http://www.jsebs.org
ISSN: 2288-3908

The Journal of Society for e-Business Studies
Vol.21, No.1, February 2016, pp.105-118
http://dx.doi.org/10.7838/jsebs.2016.21.1.105

온라인 게임 해킹대응에서 Signature 기반 탐지방법 개선에 관한 연구

A Study on Improved Detection Signature System in Hacking Response of One-Line Games

이창선(Chang Seon Lee)*, 유진호(Jinho Yoo)**

초 록

게임회사는 온라인 게임을 서비스하는 과정에서 공격자의 공격을 자주 받는다. 본 논문에서는 온라인 게임에서 해킹 모듈을 탐지하는 방식 중 하나인 Signature 탐지 방식의 한계점을 분석하고, 이러한 문제점을 보완하기 위한 Scoring Signature 탐지 방식을 제안하고자 한다. Scoring Signature 탐지 방식은 알려지지 않은 해킹 공격에 대한 수집 및 탐지를 가능토록 하여 기존의 Signature 탐지 방식보다 20배 이상의 탐지 성과로 나타났다. 이 방식을 기존에 탐지하고 있는 방식과 병행하여 적용하면 해킹 모듈 수집에 대한 번거로움을 최소화하고 미탐지로 인한 게임내의 해킹 모듈 사용도 크게 감소시킬 수 있을 것으로 판단된다.

ABSTRACT

Game companies are frequently attacked by attackers while the companies are servicing their own games. This paper analyzes the limit of the Signature detection method, which is a way of detecting hacking modules in online games, and then this paper proposes the Scoring Signature detection scheme to make up for these problems derived from the limits. The Scoring Signature detection scheme enabled us to detect unknown hacking attacks, and this new scheme turned out to have more than twenty times of success than the existing signature detection methods. If we apply this Scoring Signature detection scheme and the existing detection methods at the same time, it seems to minimize the inconvenient situations to collect hacking modules. And also it is expected to greatly reduce the amount of using hacking modules in games which had not been detected yet.

키워드 : 스코어링 시그니쳐, 온라인 게임, 해킹모듈탐지
Scoring Signature, On-line Game, Hacking Module Detection

* Lead author, Dept. of Business Administration, Sangmyung University(crattack@gmail.com)
** Corresponding Author, Dept. of Business Administration, Sangmyung University(jhyoo@smu.ac.kr)
Received: 2016-02-01, Review completed: 2016-02-18, Accepted: 2016-02-23

1. 서 론

인터넷이 확산되면서 다양한 환경에서의 게임을 실행 시키고 있다[6, 8]. 다양한 환경에서의 해킹 프로그램을 탐지하는 것은 쉬운 일이 아니다. 또한 해킹은 초기에 “개인의 호기심이나 지적욕구의 바탕 위에서 컴퓨터와 컴퓨터간의 네트워크를 탐험하는 행위”였다. 그러나 현재에는 “다른 컴퓨터 시스템을 침입하여 파괴적인 계획을 갖는 침입 행위”로 사용되고 있다[35]. 해킹 공격으로는 서비스 망을 방해/파괴하는 DDoS[19], Application에서의 취약점을 찾아 공격 코드를 생성하는 Exploit[17, 21], 불특정 다수, 특정 기업, 특정 인물 등 공격 대상을 자유자제로 공격 할 수 있는 Virus[19](Malware) 등이 존재한다. 해킹은 개인 PC와 기업에 영향을 줄 수 있고 현재 악영향을 끼치고 있다[7, 40]. 또한, 해킹 공격 대상 중 게임 해킹은 게임 서비스의 방해와 게임을 정상적으로 이용하는 유저의 게임 참여도, 업적, 성취감 등을 감소시켜 게임 이용자를 이탈하게 하는 원인 중 하나이며[11], 그로인해 게임사의 매출에 영향을 미친다[4, 16]. 온라인 게임 해킹 방식에는 DDoS, 메모리 위변조, DLL Injection을 이용한 코드/파일 삽입 등이 있다. 현재 게임사에서 사용하고 있는 온라인 해킹 탐지 방식은 Signature 방식, Heuristic 방식, 데이터마이닝 등을 이용하여, 온라인 게임 해킹을 차단, 방어하고 있다. 본 논문에서는 여러 해킹 대상 중 게임 해킹 분야에 대한 방어 방법을 논하려고 한다.

본 논문은 온라인 게임 해킹 탐지 방식 중 하나인 Signature 방식의 한계를 개선한 Scoring Signature 방식을 제안하여 온라인 게임 해킹 시도에 대한 대응책으로 논하려고 한다. Scoring Signature 방식은 Signature를 수동으로 수집하고 등록하는 방식이 아닌 사용자의 게임 환경에서 해킹 프로그램의 정보를 수집하고 수집된 정보를 Signature로 등록하도록 구현하였다.

본 논문 구성은 제 2장에서 온라인 게임 해킹 대응 방식을 설명하고 제 3장에서 Signature 기반 탐지방법의 한계에 대해 살펴본다. 제 4장에서는 본 논문에서 설계하고 구현한 Scoring Signature 방식을 서술하였으며, 제 5장에서는 Scoring Signature 구현과 실험 결과를 소개한다. 마지막으로 제 6장 결론 및 향후 연구 과제와 함께 결론을 맺는다.

2. 온라인 게임 해킹대응

2.1 DDoS

온라인 게임에서의 동시 접속자수는 매출을 산정하는 주요한 요인 중의 하나이다. 얼마나 많은 동시 접속자가 존재하느냐에 따라 게임의 성공과 실패가 가늠된다[9, 10, 15, 17, 27]. 2006년 당시 넥슨 게임의 하나인 “카트라이더”의 경우 동시 접속자수 22만 명으로 월 평균 50억 원의 수익을 얻었다[12]. 따라서, 접속자를 제한하거나 방해하는 DDoS의 경우 매출에 큰 영향을 미친다[14]. 최근 2014년 5월 7일자 뉴스에서 특정 업체의 불법 프리서버 운영자가 해당 업체를 DDoS 공격 협박을 통해 수억 원의 금품을 요구하는 사례가 기사화 되었다[4]. DDoS의 공격방식은 좀비 PC를 이용하는 방식[29]와 서비스 망의 취약점을 이용하는 방식[30]으로 분류된다. 좀비 PC의 공격 방법은 개인 PC를 이용한 공격이므로 서버에서 대응

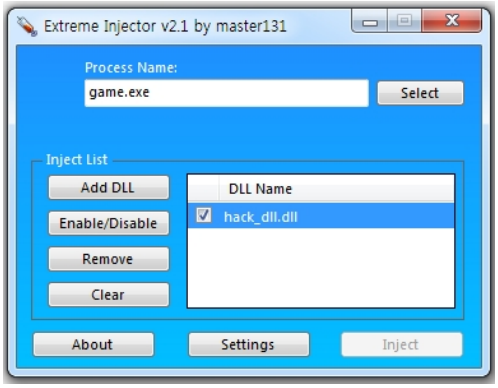
하기는 사실상 불가능하다. 따라서 서비스를 받고 있는 IDC에서의 DDoS 방어 서비스를 활용하거나 DDoS 네트워크 보안 구축, IPS/IDS 장비 구축 등을 혼합하여 대응해야 한다. 만약, 취약점을 이용한 DDoS 공격일 경우 보안 패치를 통해 공격을 차단할 수 있다.

2.2 메모리 위변조

메모리 위변조 공격은 게임에서 사용되는 획득 점수, 자원 등을 메모리상에 존재하는 값을 변조하는 방식으로 메모리 변조 프로그램을 활용하여 위변조 할 수 있다. 메모리 위변조의 한 예시로 Microsoft사에서 Windows 7 이전 제공했던 핀볼 게임을 대상으로 메모리 위변조를 테스트가 가능하다. 핀볼 게임은 정상적으로 게임 할 경우 일정 조건이 충족할 때 점수를 획득할 수 있다[13]. 하지만, 메모리 위변조 프로그램인 Cheat Engine[1]을 이용해 메모리 위변조를 진행 할 경우 높은 수치의 점수를 일정 조건이 충복하지 않아도 획득 할 수 있다. 핀볼 게임의 해킹 원리는 획득 점수를 메모리에 보관하고 있으며, 획득 점수에 대한 검증 없이 게임에 점수를 반영하고 있다. 그러므로 획득 점수를 보관하는 메모리 공간을 찾은 뒤 임의로 획득 점수 변경하면, 핀볼게임이 수정된 점수가 반영된다. 메모리 위변조 공격에 대한 대응 방안으로는 서버에서의 게임 내 사용하는 수치 자료를 감시, 비교해야 한다. 또한, 메모리 위변조 공격을 진행할 경우 클라이언트 분석인 Reverse Engineering이 필요하므로 클라이언트 분석을 방어하는 코드인 Anti-Reversing 기법[32]을 적용하여 2차 공격 피해를 차단해야 한다. 그러나 Anti-Reversing 기법을 적용하였다고 해서 Reverse Engineering을 모두 차단하는 건 아니지만 스크립트 키드[19]의 공격은 차단이 가능하므로 적용해야 한다.

2.3 DLL Injection

DLL(Dynamically Linked Library)은 실행 파일 중 하나로서, 독립적으로 실행되진 않지만 프로그램에서 호출하여 사용하는 파일 포맷이다[35]. DLL Injection은 게임 클라이언트에 공격자가 만든 해킹 프로그램을 강제로 Injection하여 해킹 프로그램을 실행하는 방식을 의미한다[25, 28].

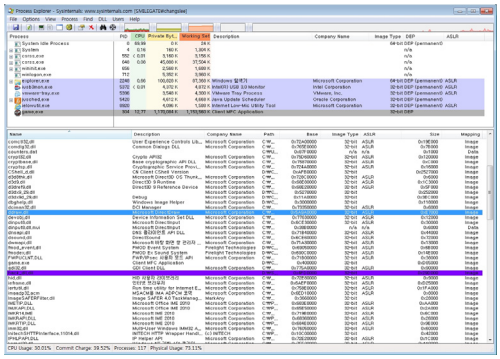


〈Figure 1〉 DLL Injector

<Figure 1>은 hack_dll.dll 파일을 Game.exe 프로그램에 Injection하게 도와주는 Injector이다. 위와 같이 공격 할 경우 <Figure 2>와 같이 Process Explorer[24]을 이용하여 프로세스 내에 사용 중인 DLL 리스트에 해킹 프로그램인 hack_dll.dll이 추가된 것을 확인 할 수 있다.

<Figure 3>은 정상적으로 게임을 실행 했을 경우 화면이다. 하지만, 공격자가 해킹 프로그램을 게임 클라이언트에 DLL Injection 할 경우

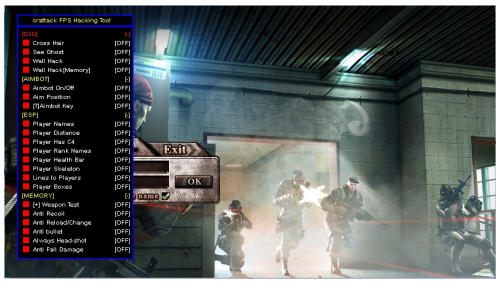
<Figure 4>처럼 해킹 프로그램에서 제공하는 해킹 메뉴가 보이고 해킹 기능이 동작하게 된다. 이에 대한 대응 방법은 여러 종류가 있지만 그 중 Signature 방식을 통한 대응책이 존재한다.



〈Figure 2〉 Hack_dll.dll Module in the Game.exe Process is Injection



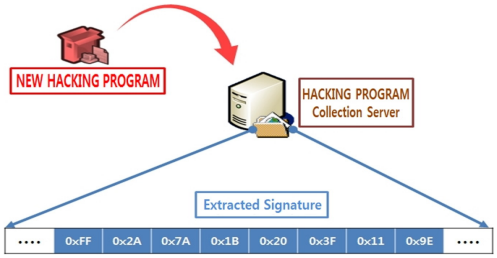
〈Figure 3〉 Progress Screen of the Normal Game



〈Figure 4〉 The Screen of the Hacking Module Injection Success

3. Signature 기반 탐지방법의 한계

Signature란 바이너리 상에서의 고유의 패턴을 찾는 것을 의미한다[5, 38]. Signature기반의 탐지 방식은 프로그램을 실행한 상태의 메모리에서 추출하거나 프로그램을 실행 하지 않은 파일 상태에서 고유한 문자열을 추출한다[3]. 수집된 파일을 분석하여 해킹 프로그램이라고 인식되면 HASH 값(MD5)을 생성하여 이후 프로세스 내에 존재하거나 파일로 존재할 경우 탐지하는 방식이다[7].

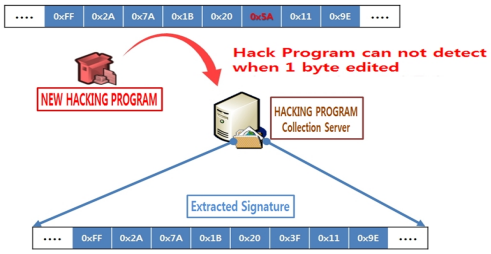


〈Figure 5〉 A New Hacking Module Signature Extraction

Signature 방식은 수집된 온라인 해킹 프로그램에 고유한 Hex 값, 문자열, HASH 값 등을 이용하여 탐지하는 방식[5, 38]으로 온라인 해킹 프로그램의 탐지를 신속하게 진행 할 수 있지만, Signature 기반의 솔루션의 단점인 다형성 기법에 대한 한계가 드러나고 있다[39].

Signature의 단점은 첫째, <Figure 5>와 같이 해킹 프로그램이 수집되어야 탐지가 가능하며, 둘째 <Figure 6>과 같이 추출한 Signature에서 1byte라도 변경되면 탐지되지 않다는 것이다. 그로 인해 우회가 용이하다[26]. 또한, 수집되지 않은 해킹 프로그램이 많아 알려지지 않은 해킹 DLL을 탐지하는 건 불가능하다.

이러한 이유로 Signature 기반의 해킹 프로그램 탐지 방식은 게임 서비스에서 온라인 해킹 프로그램이 증가하면 수집된 해킹 프로그램을 긴급하게 탐지하는 역할로 사용된다. 하지만, 지속적인 탐지 방식이 아니므로, 사용에 있어서 제한적일 수밖에 없다.



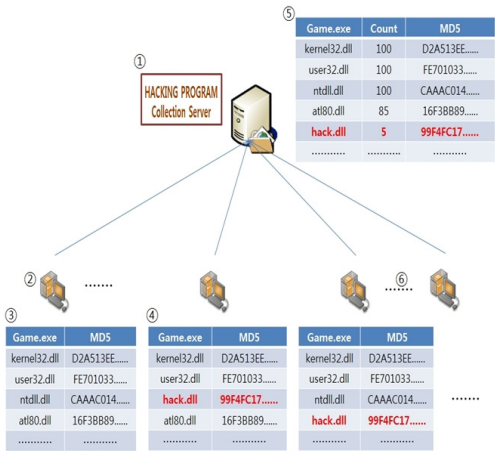
〈Figure 6〉 Bypass Techniques of New Hacking Module

따라서 본 논문에서는 Signature 탐지 방식에서 독립적인 해킹 프로그램(EXE)을 탐지하는 것이 아닌 해킹 DLL을 탐지하는 방식에 대한 개선방안을 제안하고자 한다.

4. 연구방법 및 실험설계

4.1 Scoring Signature 방식 설계 제안

기존의 Signature 탐지 방식은 이미 수집된 게임 해킹 프로그램만 탐지가 가능하기 때문에. 본 논문에서는 수집된 프로그램뿐만 아니라 수집 되지 않은 프로그램에 대해서도 탐지 가능한 Scoring Signature 방식을 제안하고자 한다. Scoring Signature는 사용자가 실행중인 게임 프로세스 내에서 동작하는 프로그램을 조사하여, 소수에서만 사용되는 프로그램(카운트가 낮은 프로그램)에 대한 수집 및 분석을 통해 수집되지 못한 해킹 프로그램을 탐지하는 방식이다. 본 논문에서 제시하고자 하는 Scoring Signature 방식의 전체 구조도는 아래와 같다.



〈Figure 7〉 The Configuration of the Scoring System Foundation Detection Method

<Figure 7>은 Scoring Signature 시스템 구성을 도식화 한 것이다. 각 구성별 역할은 다음과 같다.

① [해킹 프로그램 수집 서버] HACKING PROGRAM Collection Server는 게임 프로그램 내에 동작하는 해킹 프로그램을 수집하거나 정상 파일의 정보 수집하는 데이터 서버 역할을 한다.

② [이용자 컴퓨터] 일반 사용자 컴퓨터로 현재 게임을 진행하고 있는 컴퓨터 중 한 대이다.

③ [정상적인 사용자] 일반 사용자 컴퓨터에서 얻은 Game 프로그램에서 동작하는 모듈 정보를 수집한 내용을 도식화 한 것이며, 정상적인 환경에서 게임을 할 경우 일반적으

로 사용하는 모듈이 게임에서 사용되어야 한다. 또한 수집하는 정보는 파일의 MD5를 같이 수집하여, 파일 명을 기반으로 탐지하는 스트링 기반이 아닌 HASH 값을 이용한 탐지 방식을 채택하였으며, 향후 분석을 위해 의심 모듈을 같이 수집한다.

④ [해킹 프로그램 사용자] 해킹 DLL을 사용한 사용자의 모듈 정보 내용이다. 해당 유저는 hack.dll이라는 DLL을 사용했다는 가정을 하에 수집된 정보를 해킹 프로그램 수집 서버에 전달한다.

⑤ [해킹 프로그램 수집 서버 DB] 전달 받은 정보는 DB로 저장하고 Scoring을 통해 안정적인 모듈이 무엇이며, 불안정한 것은 무엇이고 악의적인 모듈은 무엇인지를 체크한다. 체크 방식은 각 모듈 별로 동적 분석과 정적 분석을 통해 진행하는 것이 좋지만 수집된 카운트를 기반으로 한 탐지도 가능하기 때문에 카운트 통계를 유지한다. 그러나 카운트 기반의 탐지 방식은 오류 탐지의 소지가 있으므로 2~3일의 기간 동안 모니터링을 진행하여 해당 파일의 탐지 수치 증가를 측정한 후 카운트 기반의 탐지 방식을 적용 하는 것이 좋다.

⑥ [다른 사용자] 다른 사용자의 컴퓨터도 ②-⑤를 반복한 뒤 수집된 데이터를 해킹 프로그램 수집 서버로 전달한다.

4.2 수집-분석-적용 단계별 테스트 방안

<Figure 7>과 같이 구현하기 위해 3단계로 구분하여 적용하였다. 각 단계는 다음과 같다.

4.2.1 수집: 프로세스 내의 Module HASH 수집

<Figure 7> ②번 [이용자 컴퓨터]에서 동작하는 게임의 프로세스가 Game.exe라고 가정하고, Game.exe는 게임이 시작하면서 프로세스내의 DLL 정보를 수집한다. 수집된 데이터는 주기적으로 수집하며 기존에 수집한 정보와 추가로 수집한 정보를 비교하여 변경된 내용만 ①번 서버로 전달하게 된다. ①번 서버에서는 수집된 DLL 모듈의 HASH 값을 생성하고 생성된 데이터를 ⑤번에 저장한다. 저장되는 정보의 예시는 <Table 1>과 같다.

4.2.2 분석: 수집된 정보 분석

수집된 DLL를 대상으로 정적 분석과 동적 분석을 진행한다.

4.2.2.1 정적 분석

정적분석 방식은 Virustotal[33] 사이트에서 HASH 검사 API[34]를 활용하여 평판 검사를 진행한다. 모든 파일이 Virustotal DB에 존재하지 않기 때문에 해당 내용을 정적 분석을 추가로 진행 한다. 정적분석은 binary search 알

〈Table 1〉 Information Collected of DLL Module

Filename	HASH
C:\Windows\SysWOW64\ntdll.dll	A8E40A2533C67A44EB6E878C6596F7E45A230D88
C:\Windows\syswow64\kernel32.dll	419A830FD94C6C41EFADD1CE5CD80A5E3EAC460F
C:\Windows\syswow64\KERNELBASE.dll	DE92EE7E65DCEA59B5B1F59EE9960090F88BE801
C:\Windows\SysWOW64\SYSFER.DLL	665270B866DD63AF4FBD7264A5AB2B7441E9D9E9
.....	.....

고리즘[31]을 활용하여 수집 파일의 문자열을 비교한다. 비교 문자열은 해킹 프로그램에서 많이 사용하고 있는 문자열을 기반으로 검색을 실시한다. 문자열은 다음과 같다.

〈Table 2〉 Hacking program comparison string

String Compare
xxxxxx
AIM
Head Shot
Cross Hair
See Ghost
AIM Bot
.....

<Table 2>와 같은 문자열이 바이너리에 3개 이상 있을 경우 분석 담당자에게 전달한다.

4.2.2.2 동적 분석

정적분석에서 필터링 된 해킹 프로그램 의심 파일은 2차 정적분석 과정과 동적 분석 과정을 걸치게 된다. 2차 정적분석 과정은 IDA와 같은 역어셈 프로그램을 통하여 해당 해킹 의심 파일이 정말 해킹 프로그램인지 판단한다. Anti-Reverse 기능이 적용될 경우 역어셈 프로그램으로 분석이 불가능 하므로 동적 분석으로 해킹 의심 프로그램을 분석한다. 동적 분석 과정[23]은 ollydbg와 같은 디버깅 프로그램을 활용하여 메모리에 Load된 해킹 프로그램을 분석하는 과정을 걸친다. 어떤 메모리에 접근하여 수정하는지, 또는 어떤 메모리를 참조하는지를 분석하여 해킹 프로그램 유무를 판단한다. 동적 분석을 통한 탐지 패턴 만들기는 다양한 방식들이 존재하므로 순차적으로 연구되어진 자료를 활용하여 적용한다.

4.2.3 적용: 해킹 프로그램 Signature 적용

수집된 정보 분석을 통해 얻은 자료를 기반으로 해킹 프로그램으로 확정될 경우 해킹 프로그램용 HASH Table에 추가 한다. 추가된 자료는 정기적, 비정기적으로 [이용자컴퓨터]로 전달하여 해킹 프로그램을 사용하고 있는지를 검출한다. 또는 [이용자컴퓨터]에서 취득한 Signature를 서버로 전달하여 비교하여 해킹 프로그램 사용 유무를 확인한다.

4.3 테스트 환경 구성

본 연구에서의 제시한 탐지 개선방안에 대한 시험환경 구성은 아래와 같다.

조건 1) 테스트에 활용된 PC는 5대이며, 총 10회 실시
조건 2) 게임 모듈을 구동한 뒤 임의의 해킹 모듈을 DLL Injection하여 탐지 카운트를 수집
조건 3) Game.exe를 실행 시킨 이후 5개의 Hack Module를 DLL Injection 한다.

5. 시험결과 분석

본 논문에서 제안한 방식을 기반으로 테스트를 진행하는 절차는 다음과 같다. 해킹 프로그램 수집 서버에는 <Table 3>과 같이 HASH 값이 수집된 상태이다. 따라서 <Table 3>의 HASH가 존재할 경우 해킹 프로그램을 사용한다고 판단할 수 있다.

〈Table 3〉 HASH of the Hacking Program

A0EC6B1D1C5ADBF1E0B080DC4E20F087D3B2D526

5.1 해킹 프로그램이 없는 환경

Game.exe 프로그램을 실행 시킨 이후 탐지 Scoring Signature 모듈을 실행 시킨다. 결과는 <Table 4>와 같이 얻을 수 있다.

〈Table 4〉 HASH List of Clean Environment

A8E40A2533C67A44EB6E878C6596F7E45A230D88
419A830FD94C6C41EFADD1CE5CD80A5E3EAC460F
DE92EE7E65DCEA59B5B1F59EE9960090F88BE801
...
1235207D281014EE6E42EF96EAA89607B4A261E1
D1B569E81D02A243215203078E19D69ED2D29480
CBFED52987CF42ABF107C42138C315299674B31D
6EF7F3E48E795F9BC29CAF2546B62600B37C3394
820DE8165097075C04ABAB022249DBFC99179D5D
ACDC10425361BE26443036FC3587B9E50FB81FAF

해킹 프로그램이 실행되지 않은 환경이라서 해당 HASH 목록을 서버에 보관하면 기준 데이터로 활용할 수 있다. 단, 해당 리스트는 OS와 서비스 팩, 3rd Part 프로그램에 따라서 변경 될 수 있으므로 완벽한 기준 데이터를 만들기에 무리가 있으므로 참고 항목을 사용하되, 추가로 수집된 자료는 평판 조사와 정적, 동적 분석을 통하여 기준 데이터를 만드는 작업을 진행해야 한다.

5.2 v.1 해킹 프로그램 Injection

Game.exe 프로그램에 해킹 프로그램을 Injection 하여 추가 후 탐지 유무를 확인한다. 해킹 프로그램의 HASH는 테스트하기 전에 언급한 <Table 3>을 활용하였다.

〈Table 5〉 Detection of the Hacking Program HASH

A8E40A2533C67A44EB6E878C6596F7E45A230D88
419A830FD94C6C41EFADD1CE5CD80A5E3EAC460F
DE92EE7E65DCEA59B5B1F59EE9960090F88BE801
...
1235207D281014EE6E42EF96EAA89607B4A261E1
D1B569E81D02A243215203078E19D69ED2D29480
CBFED52987CF42ABF107C42138C315299674B31D
A0EC6B1D1C5ADBF1E0B080DC4E20F087D3B2D526
6EF7F3E48E795F9BC29CAF2546B62600B37C3394
820DE8165097075C04ABAB022249DBFC99179D5D
ACDC10425361BE26443036FC3587B9E50FB81FAF

<Table 4>에서 수집된 내용과 유사하나, 해킹 프로그램에서 추출한 HASH 값을 <Table 5>에서 탐지 할 수 있었다. 따라서 해당 사용자가 해킹 프로그램을 사용하고 있음을 확인할 수 있다.

5.3 v.1 해킹 프로그램 변종-1

기존에 수집된 해킹 프로그램의 1 byte를 수정하여 새로운 HASH가 생성하여, 변종된 해킹 프로그램의 탐지 여부를 확인 하였다.

〈Table 6〉 A Variant of the Hacking Program HASH

A8E40A2533C67A44EB6E878C6596F7E45A230D88
419A830FD94C6C41EFADD1CE5CD80A5E3EAC460F
DE92EE7E65DCEA59B5B1F59EE9960090F88BE801
...
1235207D281014EE6E42EF96EAA89607B4A261E1
D1B569E81D02A243215203078E19D69ED2D29480
CBFED52987CF42ABF107C42138C315299674B31D
9CCD7620DF9014063D7A9E7E6D69B62EC5BD7A8D
6EF7F3E48E795F9BC29CAF2546B62600B37C3394
820DE8165097075C04ABAB022249DBFC99179D5D
ACDC10425361BE26443036FC3587B9E50FB81FAF

<9CCD7620DF9014063D7A9E7E6D69B62EC5BD7A8D>의 HASH가 신규로 수집된 것을 확인 할 수 있다. 따라서 해당 파일을 수집하여, 해킹 프로그램 유무를 확인하고 해킹 프로그램일 경우 Signature DB에 추가해 향후 해킹 프로그램 사용 유무를 탐지 할 수 있다.

이와 같은 방법으로 프로세스 내에서 추출한 HASH 값 중에 다수가 사용하는 정상적인 HASH 테이블을 <Table 7>과 같이 수집할 수 있었다.

〈Table 7〉 HASH Table of Normality File

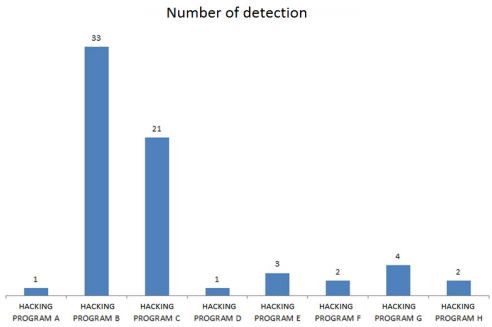
MD5	Count
D2A513EE880D71BDE7F0257F38B9D019	50
FE70103391A64039A921DBFFF9C7AB1B	50
CAAAC014C5C56A69F710B5F1B836DE22	50
16F3BB89525EE0A857923E63206409D9	35
99F4FC172A5ACE36CF00AA7038D23F2C	50
....	....

이후, 의심되는 해킹 프로그램 수집하여 분석 한 결과 <Table 8>과 해킹 프로그램임을 인지할 수 있었다.

〈Table 8〉 HASH Table of Hacking Module

Filename	Count	MD5
Hack1	3	7620AFB6A653E9C56D69BDE8F2BD5926
Hack2	1	4AA683DC35D101EE1FBADCEB8D56B3AC
Hack3	2	CFC3837ADC7C901C64B856485EB165AD
Hack4	6	69DE08BD19272AD8CB53743ACE4C24BD
Hack5	25	06F7B595326BE76C420BA1B4E9DB2262

본 논문에서 제시한 방식을 실제 게임서비스에 적용한 결과는 다음과 같다. 적용한 서비스는 A사의 B 게임으로 다양한 국가에 서비스를 하고 있는 게임이며, B 게임은 해킹 모듈을 탐지하는 Signature 기반의 탐지 방식을 사용하고 있었다. Signature 신규 패턴을 업데이트 한 이후 2시간 후 탐지 우회되는 것을 확인하였다. 아래의 <Figure 9>은 Scoring Signature를 적용하기 전의 탐지 수치이다. <Figure 9>에서는 수집된 해킹 모듈을 기반으로 탐지하기 때문에 탐지 수치가 높지 않다. 또한, 서비스 업데이트 이후 게임 사이트가 서비스를 시작할 때 해킹 모듈이 즉시 배포되지 않고 일정 시간이 진행 된 이후 해킹 모듈이 배포되기 때문에 신규 해킹 모듈을 수집하는 데에는 어느 정도 시간이 소요된다.



〈Figure 8〉 Numeric Value of the Previous Application of the Scoring System

초기에는 많은 해킹 모듈이 배포되지 않기 때문에, 수집하는데 한계가 있어 총 67건이 탐지 되었다. 하지만, Scoring Signature 방식을 적용할 경우 <Table 9>와 같이 적용 전에 비해 약 20배 이상 탐지되는 것을 알 수 있었다.

〈Table 9〉 Numerical Comparison and from Applying the Scoring System

	A	B	C	D	Hack Module Suspected	Total
before	1	33	21	1	0	56
after	1	33	21	1	1,105	1,161

해당 수치는 중복으로 탐지되는 것을 허용하였기 때문에 정확히 20배라는 수치를 얻는 것은 아니다. 그러나 기존에 탐지되었던 56건에 비해 훨씬 높은 수치의 탐지 건수 결과를 얻을 수 있었다. 또한, 1,105건이 모두 해킹 모듈인지 여부는 분석을 통해 확인을 해야 하며, 서비스를 제공하고 있는 게임에서 오류 탐지는 치명적일 수 있기 때문에 2차 분석 과정을 통하여 해킹 모듈 유무를 판단 한 뒤 Signature를 다시 생성하여 업데이트를 진행해야 한다.

해당 탐지 수치는 분석을 통해 해킹 모듈이라고 판단된 상황은 아니지만 수집되지 않은 해킹 모듈을 사용하고 있다는 의심 수치이고 이전 방법으로는 탐지 되지 않은 것을 찾아내었다는데 의미가 있다. 뿐만 아니라 초기 온라인 게임 해킹 모듈에서 해킹 모듈이 배포되기 전에 공격자의 테스트 과정을 파악 할 수 있으며, 또한 해킹 모듈을 수집할 수 있다는 장점을 확인하였다.

6. 결론 및 향후 연구 과제

본 논문에서는 온라인 게임에서 해킹 모듈을 탐지하는 방식 중 하나인 Signature 탐지 방식의 한계점을 분석하고, 이러한 문제점을 보완하기 위한 Scoring Signature 탐지 방식을 제안하였다. Scoring Signature 탐지 방식은 알려지지 않은 해킹 공격에 대한 수집 및 탐지를 가능토록 하여 기존의 Signature 탐지 방식보다 20배 이상의 탐지 성과로 나타났다. 본 논문에서 제시한 방식을 활용할 경우 서비스 오픈 후 해킹 프로그램을 처음 사용한 유저나 해킹 프로그램 개발자에 대한 공격 시도에 대한 탐지를 개선할 수 있을 것으로 판단된다. 서비스 오픈 이후 특정한 모듈이 지속적으로 카운트 될 경우 해킹 프로그램 개발자가 테스트하는 과정이라고 판단 할 수 있을 것이다. 배포되기 전 해당 해킹 프로그램을 수집하여 탐지 로직으로 분류한다면 서비스를 유지하는 과정에서 해킹 프로그램의 무분별한 사용을 차단할 수 있을 것으로 판단된다. 또한, 서비스가 운영되는 과정에서 많은 해킹 프로그램이 암묵적으로 전파되고 사용되고 있는 상황에서 유포 사이트를 방문해 해킹 프로그램을 수집하는 업무는 한계가 있다. 따라서 본 논문에서 제시한 Scoring Signature System은 해킹 프로그램 수집에 한계를 벗어날 수 있는 방법 중 하나로 판단된다. 분석에서의 리소스를 줄이기 위해서 해킹 프로그램의 MD5[37] HASH를 추출하여 VirusTotal[33]에 바이러스 조사를 통해 해당 모듈에 대한 해킹 프로그램 또는 바이러스 유무를 확인 할 수도 있을 것으로 판단된다. 그 이후, 정적 분석으로 해킹 프로그램 의심 파일의 내부 스트링을 확인하는 프로그램을 개발하고 마지막으로 동적 분석을 통해 Signature를 추출한다면 많은 리소스를 투여하지 않아도 활용이 가능할 것으로 보인다.

향후 연구에서는 DLL Injection을 통한 공격뿐만 아니라 메모리 분산 공격을 탐지하는 방식에 대한 연구 진행할 예정이다. 메모리 분산 공격의 경우 DLL Injection의 방식을 통하지 않고 프로세스 권한을 획득한 뒤 게임 메모리에 자신의 해킹 모듈의 일정 부분을 분산하여 Injection하는 기법이다. 해당 기법을 탐지하기 위해서는 메모리의 특성과 해킹 모듈의 메모리 분산 공격에 대한 특징을 분석하는 작업이 진행될 필요가 있다.

References

[1] Cheat Engine, http://www.cheatengine.org/.

[2] Chess, D. M. and White, S. R., “An undetectable computer virus,” In Proceedings of the 2000 Virus Bulletin Conference, 2000.

[3] Faloutsos, C. and Christodoulakis, S., “Description and Performance Analysis of Signature File Methods for Office Filing,” ACM TOIS, Vol. 5, No. 3, pp. 237-257, 1987.

[4] Ha, K. M., “Threatened to free server operator of DDoS attacks tear money ‘cyber gang’,” newsis, 2014.

[5] Jo, M. J. and Shin, J. S., “A Performance Enhancement Scheme for Signature-based Anti-Viruses,” Journal of the Korea Industrial Information Systems Research, Vol. 20, No. 2, pp. 65-72, 2015.

[6] Jung, J. H. and Lee, C. M, “Analysis of C2C Internet Fraud and Its Counter Measures,” The Journal of Society for e-Business Studies, Vol. 20, No. 2, pp. 141-153, May 2015.

[7] Kang, H.-K. et al, “Development of an automatic document malware analysis system,” IT Convergence and Security 2012, Vol. 215, pp. 3-11, 2013.

[8] Kang, H. M., Bang, J. H., Lee, E. H., “Choice Satisfaction of the Broadband Internet Network Services,” The Journal of Society for e-Business Studies, Vol. 16, No. 3, pp. 47-66, 2011.

[9] Kim, H. J., “Cruise in the off-season second quarter of the neowiz game,” money-today, 2008.

[10] Kim, J. S., “Freestyle, Southeast Asia in exports for three countries to globalization,” NEWSWiRE, 2008.

[11] Kim, S. M. and Kim, H. K, “A research on improving client based detection feature by using server log analysis in FPS games” Journal of The Korea Institute of Information Security and Cryptology, Vol. 25, No. 6, 2015.

[12] Lee, B. H., “High-income part paid game the key to success,” TheGames, 2007.

[13] Lee, C. S., “Hackers even unknown hacking story-let the others ride in memory” Microsoft Magazine, 2006.

[14] Lee, D. W., “FIFA 3, respite---KT CDN DDoS Attack,” ZDNet Korea, 2014.

[15] Lee, J. N., “Casual game is not a big hit?

Not that there is a national game Kart-Rider," the hankyoreh, 2005.

[16] Lim, G. G. and Lee, H. S., "An Exploratory Study on the status and classification of Cyber Money," Proceedings of the CALSEC Conference, pp. 17-28, 2005.

[17] "High-flying popular throughout the fifth anniversary of the service 'maple story'" NEXON, 2008.

[18] https://www.exploit-db.com/.

[19] Moran, D. B., "Trapping and Tracking Hackers: Collective security for survival in the Internet Age," Third Information Survivability Workshop. IEEE Computer Society Press, 2000.

[20] Nazario, J., "BlackEnergy DDoS Bot Analysis," 2007.

[21] Notorious, "hacking case 20 election that broke the IT companies," http://www.itworld.co.kr/slideshow/86870, 2014.

[22] Pascal Bouchareine, "Format String Vulnerability." https://www.win.tue.nl/~aeb/linux/hh/kalou/format.html, July 18 2000.

[23] Park, J.-W., Moon, S.-T., Son, G.-W., Kim, I.-K., Han, K.-S., Im, E.-G., and Kim, I.-G., "An Automatic Malware Classification System using String List and APIs," Journal of Security Engineering, Vol. 8, No. 5, pp. 611-626, 2011.

[24] Process Explorer, https://technet.microsoft.com/ko-kr/sysinternals/bb896653.

[25] Richter, J., "Load Your 32-bit DLL into Another Process's Address Space Using INJLIB," Microsoft Systems Journal, Vol. 9, No. 5, May. 1994.

[26] Schultz, M. G., Eskin, E., Zadok, E., and Stolfo, S. J., "Data Mining Methods for Detection of New Malicious Executables," IEEE Symposium on security and privacy, pp. 38-49, 2001.

[27] Shin, H. S., Song, Y. U., and Sung, N. H., "The Impact of Perception on the Difference Between Mobile and Stationary Internet Toward the Intention to Use Mobile Internet," The Journal of Society for e-Business Studies, Vol. 15, No. 3, pp. 99-129, 2010.

[28] Skape, Jarkko Turkulainen, "Remote Library Injection" http://www.nologin.org/Downloads/Papers/remote-library-injection.pdf, p. 14.

[29] Son, K. H., "CD Networks, DDoS cause "amplification attacks" increased" ZDNET Korea, 2015.

[30] Son, K. H., "Google Maps exploiting vulnerabilities DDoS occurs," ZDNET Korea, 2015.

[31] Tian, R., Batten, L., Islam, R., and Versteeg, S., "An automated classification system based on strings and of trojan and virus families," In Proceedings of MALWARE, 2009.

[32] Tully Joshua, "An Anti-Reverse Engineering Guide," 9 Nov 2008.

[33] Virus Total, https://www.virustotal.com/ko/#search.

[34] Virus Total API, https://www.virustotal.com/ko/documentation/public-api/.

[35] Wikipedia, http://en.wikipedia.org/wiki/Dynamic-link_library.

[36] Wikipedia, http://en.wikipedia.org/wiki/Hacking.

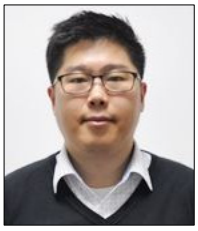
[37] Wikipedia, https://ko.wikipedia.org/wiki/MD5.

[38] Xu, J.-Y., Sung, A. H., Chavez, P., and Mukkamala, S., "Polymorphic malicious executable scanner by api sequence analysis," In Proc. of the 4th International Conference on Hybrid Intelligent Systems (HIS'04), Kitakyushu, Japan, IEEE, pp. 378-383, 2004.

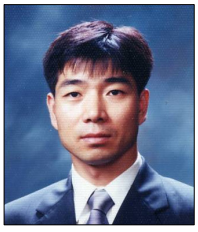
[39] Yoo, H., Yun, J.-H., and Shon, T., "White-list-based anomaly detection for industrial control system security," The Journal of Korean Institute of Communications and Information Sciences, Vol. 38B, No. 8, pp. 641-653, 2013.

[40] Yoo, J. H., "Comparison of Information Security Controls by Leadership of Top Management," The Journal of Society for e-Business Studies Vol. 19, No. 1, pp. 63-78, 2014.

저 자 소 개



이창선	(E-mail: crattack@gmail.com)
2004년	동서울대학 (전문학사)
2009년	공학사
2013년	전남대학교 (이학석사)
2015년~현재	상명대학교 정보시스템보안 (박사과정)
관심분야	온라인 게임보안, 데이터마이닝, 모바일 보안, Reverse Engineering



유진호	(E-mail: jhyoo@smu.ac.kr)
1992년	고려대학교 수학과 졸업
1994년	고려대학교 통계학과 (석사)
2010년	고려대학교 정보보호 (박사)
1993년~1999년	한국전자통신연구원 연구원
2000년~2004년	IBM KOREA 전문차장
2004년~2012년	KISA 인터넷문화진흥단장
2013년~현재	상명대학교 경영학과 교수
관심분야	정보보호, 개인정보보호, MIS, 인터넷윤리