Journal of The Korea Society of Computer and Information
Vol. 18, No. 5, May 2013
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2013.18.5.069

유한 오토마타를 이용한 악성코드 실시간 분석 시스템에 관한 연구

김 효 남*, 박 재 경**, 원 유 헌*

A Study on the Malware Realtime Analysis Systems Using the Finite Automata

Hyo-Nam Kim*, Jae-Kyoung Park**, Yoo-Hun Won*

요 약

최근에 인터넷 환경에서 악성코드를 이용한 사이버 공격이 문제가 되고 있으며, 악성코드로 인한 피해가 점차 심각해지고 규모도 증가하고 있는 추세이다. 그리고 새로운 악성코드의 출현과 더불어 기존의 악성코드를 이용한 변종 역시 커다란 피해를 주고 있다. 본 논문에서는 악성코드 분석방법에서 악성코드라고 의심되어지는 파일을 보다 정확하게 판단하기 위해 악성행위에 대해서 유한 오토마타(Finite Automata) 기법을 이용한 프로파일링 기법을 도입하여 수동이 아닌 자동으로 실시간 악성코드를 분석할 수 있는 효과적인 방법을 제안하고자 한다. 파일 내부에서 사용되는 함수들을 유한 오토마타로 표현하여 상호 관계 및 연관성을 파악하여 해당 파일에 대한 악성코드 여부와 정상파일 여부를 실시간적으로 분석할 수 있는 실시간 악성코드 분석 시스템(Realtime Malware Analysis System)을 제안한다.

▸ Keywords : 악성코드, 유한 오토마타, 실시간 분석

Abstract

In the recent years, cyber attacks by malicious codes called malware has become a social problem. With the explosive appearance and increase of new malware, innumerable disasters caused by metaphoric malware using the existing malicious codes have been reported. To secure more effective detection of malicious codes, in other words, to make a more accurate judgment as to whether suspicious files are malicious or not, this study introduces the malware analysis system, which is based on a profiling technique using the Finite Automata. This new analysis system enables realtime automatic detection of malware with its optimized partial execution

•제1저자 : 김효남 •교신저자 : 김효남
•투고일 : 2013. 4. 10, 심사일 : 2013. 4. 22, 게재확정일 : 2013. 5. 14.
•이 논문은 2012학년도 홍익대학교 학술연구진흥비에 의하여 지원되었음
* 홍익대학교 컴퓨터공학과(Dept. of Computer Engineering, Hongik University)
** 한국과학기술원 사이버보안연구센타(Cyber Security Research Center, KAIST)

method. In this paper, the functions used within a file are expressed by finite automata to find their correlation, and a realtime malware analysis system enabling us to give an immediate judgment as to whether a file is contaminated by malware is suggested.

▸ Keywords : Malware, Finite Automata, Realtime Analysis

I. 서 론

2000년대 이후 대한민국 내 모든 사무실과 가정은 광케이블 통신망으로 연결되었으며, 스마트폰과 무선인터넷이 100%이상 보급되었고, 전국 어디에서나 유비쿼터스 환경에서의 디지털 사회가 구현될 정도로 인터넷이 지속적으로 발전을 거듭해 가고 있는 반면에 이를 악용하기 위한 다양한 사이버 공격으로 사회적 문제가 야기되고 있는 실정이다[1]. 최근에는 인터넷 환경에서 사회적으로 문제가 되고 있는 것이 악성코드(malware)를 이용한 사이버 공격이다. 악성코드 역시 폭발적으로 증가 추세를 보이고 있으며, 새로운 악성코드의 출현과 더불어 기존의 악성코드를 이용한 변종 역시 커다란 피해를 주고 있다[2]. 특히, 최근에 발생한 3.20 사태와 같이 특정 방송사와 금융권 전산망을 마비시키고 임직원 PC의 MBR(master Boot Record)을 망가뜨려 못쓰게 만드는 피해 유형이 발생되었다. 지난해부터 최근까지 기승을 부린 이러한 악성코드 주요 트렌드를 보면, 금융정보 탈취 위한 악성코드의 급증 및 대형 은행사이트를 사칭한 피싱의 증가, DDoS 공격을 수행하는 봇넷의 진화, 취약점 공격 툴을 이용한 악성코드 유포의 자동화, APT 형태의 복합 공격 악성코드 기승, 보안취약점을 노출되게 하는 게임 계정정보 탈취 목적의 악성코드 기승 등이다[3]. 이처럼 악성코드의 공격은 APT(Advanced Persistent Threat) 공격과 같이 목표를 정하여 공격하는 방식으로 노골적인 공격을 감행하고 있다. 또한, 사이버전과 같은 국가 간의 전쟁의 양상으로도 확대되고 있는 추세이다. 이에 대한 기술적인 대응책으로 악성코드를 차단하기 위한 기술 개발이 이루어지고 있으나 단순히 악성코드에 대한 패턴을 이용한 시스템이거나 악성코드의 배포지에 대한 URL을 차단하는 수준이다. 최근의 악성코드는 Zero-Day 공격과 같이 실시간적인 공격이 빈번하고 또한 수많은 좀비 PC를 통해 대규모적으로 공격을 감행하므로 실시간으로 악성코드를 막을 수 있는 기술적인 방안이 필수적으로 필요하다.

악성코드의 종류가 다양해진 만큼 다양한 동작과 방법으로 공격을 하고 피해를 주기 때문에 여러 단계의 필터를 통해(탐지, 분석, 검증) 악성코드를 추출하여 차단해야 한다. 인적자원의 이용을 최소화 한 능동적인 시스템으로서 실시간으로 악의적인 접속을 차단하여 악성코드로 인한 피해를 막고 자료유출을 방지하는 시스템 개발이 필요하다. 또한, 기존의 역공학(Reverse Engineering) 방식의 공개툴이나 상용화 툴을 사용하여 어셈블리나 원하는 상위 언어로 변환하여 수동적으로 악성코드를 분석하였다. 이와 같이 악성코드를 분석하기 위해 가상 환경을 사용하여 수동으로 분석하는데 시간의 비효율성 및 노동력이 많이 필요하였다.

따라서 본 논문에서는 악성코드 분석방법에서 악성코드라고 의심되어지는 파일을 보다 정확하게 판단하기 위해서 악성행위에 대한 유한 오토마타(Finite Automata) 기법을 이용해서 프로파일링 기법을 도입하고 수동이 아닌 자동으로 실시간 악성코드를 분석할 수 있는 효과적인 방법을 제안하고자 한다. 파일 내부에서 사용되는 함수들을 유한 오토마타로 표현하여 상호 관계 및 연관성을 파악하여 해당 파일에 대한 악성코드 여부와 정상 파일여부를 실시간적으로 분석할 수 있는 실시간 악성코드 분석 시스템 (RMAS : Realtime Malware Analysis System)을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해서 살펴본다. 3장에서는 새로운 악성코드 탐지를 위한 분석방법에 대해 기술하고, 4장에서는 본 논문이 제안하는 분석방법에 대한 실험 결과와 평가를 기술한다. 그리고 마지막으로 5장에서는 결론 및 향후 연구방향에 대해서 기술한다.

II. 관련 연구

1. 악성코드 종류와 분석 방법

1.1 악성코드의 종류

Malicious code 또는 malware 라고 불리는 악성코드는

악성 프로그램(malicious program)이라고도 한다. 악의적인 목적을 위해 작성된 실행 가능한 코드의 통칭으로 자기 복제 능력과 감염 대상 유무와 같이 악성코드 행위에 따라 분류된다. 악성코드의 종류는 다음과 같다[4].

- 바이러스(Virus) : 프로그램, 실행 가능한 어느 일부분 혹은 데이터에 자기 자신 혹은 변형된 자신을 복사하는 명령어들의 조합이다. 감염대상 프로그램 혹은 코드를 변형해 바이러스 코드 혹은 일부 코드를 복제해 감염시키고 다른 대상을 감염시킴으로써 확산된다.
- 웜(Worm) : 다른 프로그램의 감염 없이 자신 혹은 변형된 자신을 복사하는 명령어들의 조합이다. 네트워크 어웨어 바이러스(Network Aware virus)라고 불리기도 한다. 웜은 기억장소에 코드 형태로 존재하거나 혹은 실행파일로 존재하며 실행되면 파일이나 코드 자체를 다른 시스템으로 복사한다.
- 트로이목마(Trojan horse) : 컴퓨터의 프로그램 내에 사용자는 알 수 없도록 프로그래머가 고의로 포함시킨 자기 자신을 복사하지 않는 명령어들의 조합이다. 고의로 포함시켰다는 점에서 프로그램의 버그(bug)와 다르며, 자신을 복사하지 않는다는 점에서 바이러스나 웜과도 약간 다르다.
- 백도어(Backdoors) : 공격자가 공격 대상 시스템에 원격으로 접속해서 해당 시스템을 제어하거나 모니터링 등의 행위를 할 수 있게 네트워크 채널을 생성하는 악성코드 프로그램이다. 백도어가 실행되면 공격자가 원격에서 해당시스템에 접속해 다양한 행위를 행할 수 있게 하는 독립적인 프로그램 형태를 가지는 백도어도 있고, 나쁜 소프트웨어 개발자가 처음부터 프로그램 소스코드에 몰래 심어놓은 형태의 백도어도 있다.
- 스파이웨어(Spyware) : 스파이(spy)와 소프트웨어(software)의 합성어로 다른 사람의 컴퓨터에 잠입해 중요한 개인정보를 빼내는 소프트웨어를 말한다. 무료 공개 소프트웨어를 내려 받거나 인증서 없는 Active-X 프로그램을 실행할 때 자동적으로 함께 설치되면서 개인 및 금융정보를 유출시킨다.
- 에드웨어(Adaware) : 광고나 마케팅을 목적으로 배포하는 것으로 어떤 사람이나 조직에 관한 정보를 수집하는데 도움을 주는 프로그램이다. 컴퓨터에 직접적으로 영향을 주지 않지만, 팝업노출, 익스플로러의 시작페이지 변경 또는 고정시키는 등의 악성행위를 수행함으로써 사용자의 불편을 유발한다.

1.2 악성코드 분석 방법

악성코드를 분석하는 것은 악성코드의 공격 목표와 목적, 탈취하려는 정보, 내부 전파, 감염 시스템에서의 동작과정, 감염 경로 등의 내용의 정보를 파악할 수 있으며, 파악된 악성코드와 향후 공격하는 변종에 대처할 수 있는 기술이다. 악성코드를 분석하는 방법으로는 정적 분석(Static Analysis)와 동적 분석(Dynamic Analysis)의 두가지 방법으로 분류될 수 있다[5].

정적분석 방법은 Dis-assembler나 Decompiler를 이용하여 역공학(Reverse Engineering) 방법으로 악성코드 바이너리 파일을 실제로 실행시키지 않고 악성 바이너리에 대해서 어셈블리 명령어로 변환하여 분석하는 방법이다. 분석을 수행할 때 PE(Portable Executable) 파일 내에 API 정보를 추출함으로써 악성코드가 어떤 악성 요인을 수행하는지를 파악할 수 있으며, 포함된 텍스트 스트링을 추출함으로써 악성 행위들에 대한 정보를 추측할 수 있다[6]. 정정분석 방법은 악성코드를 직접 실행하지 않기 때문에 안전하고, 특정 실행 조건에 제한되지 않고 악성코드의 구조와 동작 진행 등에 대해서 분석이 가능한 장점을 가지고 있다. 반면, 자동화가 어렵고 노력과 시간이 많이 소모되며, 바이너리 파일이 암호화 등의 방법으로 만들어진 경우에는 분석하기 어려운 단점을 가지고 있다[6].

동적분석 방법은 악성코드를 실제로 실행시켜 악성행위 내용을 분석하는 방법이며, Emulator나 VM(Virtual Machine)에서 프로세스나 파일 및 레지스트리를 생성하는 행위와 수정하는 행위, 네트워크를 이용하는 행위, 그리고 이러한 행위를 위한 API 호출 등을 실시간으로 분석하는 방법이다. 동적분석 방법에는 API 후킹, 행위기반(Behavior Based), 유사도기반(Similarity Based) 분석 기법들이 활용된다[7,8].

- API 후킹 : Win32 API 호출을 중간에서 가로채 제어권을 얻어내서 사용자가 작성된 응용프로그램의 함수를 처리할 수 있게 해주는 방법이다.
- 행위기반 : 기존에 알려지지 않은 악성코드 및 변종코드에 적용하는 방법이며, 레지스트리가 새로 생성되거나 무결성이 파괴되는 경우, 트래픽 과부하에 대하여 모니터링하고 악성코드의 악성행위를 판단하여 탐지하는 방법이다.
- 유사도기반 : 악성코드로 의심되는 바이너리에 대한 정적 분석을 통해 특징들을 추출하여 각각의 특징들에 대한 유사도를 계산하여 악성코드를 탐지하는 방법이다.

1.3 국내·외 동향

표 1. 악성코드 탐지를 위한 국내·외 솔루션
Table 1. Domestic and International Solution for the Malware Detection

주요 장비	URL	주요 특징
TrusWatcher	www.ahnlab.com	가상 머신 기반의 신종 악성 파일 분석
Cube Defense	www.tricubelab.com	제로데이 공격에 대비와 악성코드 유포 및 경유지에 대한 접근 및 차단 통제
스나이퍼 BPS	www.wins21.co.kr	탐지와 차단시스템의 실시간 연동을 통한 신속한 악성코드 내부 확산 방지, 자체 좀비PC 치료 Agent를 통한 분석
티프론트	www.piolink.co.kr	악성코드 행위기반 분석 APT 공격 대응
TMS (Threat Management Service)	www.trendmicro.co.kr	네트워크 기반의 의심파일을 가상환경(샌드박스)에서 실행하여 악성행위 기반으로 탐지 및 치료 16억개 이상의 악성 URL 패턴을 활용한 실시간 탐지 및 치료
MPS (Malware Protection System)	www.fireeye.com	가상실행엔진(VEX)을 탑재하여 악성코트 행위 기반 분석 C&C 서버를 추적해 악성코드에 감염된 PC와 C&C 서버 간의 통신을 차단
SW 블레이드	www.checkpoint.com	공격자의 명령 유형을 식별해 차단 내부 유입 바이러스 식별 클라우드 기반의 악성코드 정보 수집
포티케이트	www.fortinet.co.kr	가상화 환경(샌드박스)에서 악성코드로 의심되는 패킷의 행위를 분석 네트워크상에서 일어나는 악성코드 비정형 활동 탐지

악성코드를 탐지하기 위해 상용화된 국내·외 제품들은 매우 뛰어난 기술력을 갖춘 보안 솔루션 개발 업체들이며, 솔루션의 성능을 향상시키기 위하여 지속적인 연구와 개발을 진행하고 있다. 그러나 현재 판매하고 있는 대부분의 장비들은 주로 URL이나 패턴 기반의 분석을 통해 악성코드를 검사하는 형태로 개발이 이루어져 왔다. 동적 콘텐츠 기반 분석이나 행위 기반 분석 기법은 공개된 가상머신을 사용하여 분석하므로 성능적인 측면에서 매우 느릴 수 밖에 없고 또한 동적 반응이 없을 경우 악성 여부를 판단하기 어려운 측면이 있다. 또한 악성코드를 감지해 악성코드를 격리하거나 치료하는 것에만 국한되어 있어 악성코드의 피해가 확산되는 것을 저지하는 역할을 수행하지 못하고 있다. 또한 자체적인 패턴 분석을 통해 패턴이나 URL을 업데이트 서버에 업데이트 하는 소극적인 시스템 형태로 구성되어 있어 근원적인 악성 코드에 대한 차단 방법을 제시하고 있지는 못한 상황이다. 표 1는 국내·외에서 판매되고 있는 악성코드 탐지 솔루션의 주요 특징들을 나타내고 있다.

III. 본 론

1. 실시간 기반의 악성코드 분석 시스템(RMAS : Realtime Malware Analysis System)

악성코드의 실시간 분석을 통한 실시간 탐지, 검증, 치료 및 2차 피해 확산을 방지하기 위해서는 네트워크로 유입되는 실시간 패킷을 분석하여 파일 기반으로 패킷을 분석하고 해당 파일에 악성코드가 감염되었을 경우 이를 실시간으로 치료 및 차단함으로써 악성코드로 인한 APT 공격을 예방하고 자료 유출을 방지할 수 있는 악성코드 분석시스템을 제안하고자 한다. 그림1은 실시간 안성코드를 분석하기 위한 RMAS의 전체적인 구조를 보여주고 있다.

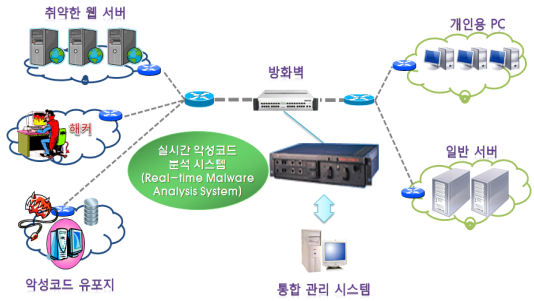

그림 1. RMAS 구조
Fig. 1. RMAS Architecture

기존 악성코드 차단의 경우 패턴이나 링크에 대한 URL만을 이용하여 분석하였으나[9-10] 본 연구에서는 실시간 분석 엔진과 실시간 검증 엔진을 통해 패턴을 자동 생성함으로써 기존 악성코드 처리방식을 획기적으로 개선할 수 있도록 하였다. 또한 알려지지 않은 악성코드의 패턴을 탐지한 후 자동 배포함으로써 다른 시스템이나 웹 사이트에도 치료 효과를 실시간으로 나타낼 수 있다.

그림 2는 실시간 정적 분석 엔진의 흐름도를 보여주고 있다. 흐름도와 같이 악성코드를 분석하기 위해서 실시간 패킷 처리 시스템으로부터 수집된 패킷들을 재조합하여 정적분석에서 사용하기 위해 실행파일을 추출한다. 추출된 실행파일은 Decompiler 툴을 이용하여 함수호출 관계를 분석할 수 있는 소스 프로그램으로 변환하게 한다. 리버싱 과정은 다음 2절에서 설명하고 있다. 리버싱 과정을 통해 만들어진 소스 프로그램은 파싱(parsing) 프로그램에서 함수 호출관계를 나타내는

오토마타로 구성하는데 사용된다. 여기서 오토마타를 구성하기 위해서 파싱(parsing) 단계를 정적 분석에서 이루어지게 하였는데, 제안하는 정적 분석 엔진의 기능은 악성코드를 리버싱한 내용만을 토대로 분석이 가능하게 하였다.

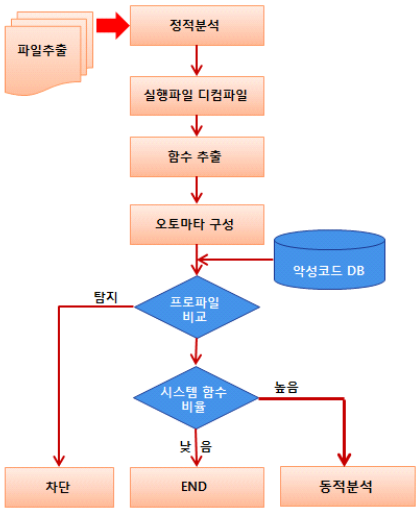

그림 2. RMAS 흐름도
Fig. 2. RMAS Flow Chart

악성코드를 탐지하기 위해 만들어진 오토마타는 기존에 등록된 악성 오토마타와 비교하여 일치할 경우 차단하게 하고 일치하지 않은 경우에는 write 행위를 하는 시스템 함수 비율을 판단하여 동적 분석 단계로 이전할지 정상처리 할지를 결정하도록 한다. 여기서 오토마타 기반의 악성코드 분석 방법을 도입한 이유는 악성 코드의 악성 행위 패턴을 프로파일링으로 등록하여 향후 변종 가능한 악성코드에 대한 프로파일을 다량으로 제작이 가능하게 할 수 있으며 또한 자동화할 수 있는 효율적인 방법으로 유사 악성코드에 대한 철저한 대비가 가능하도록 하였다.

2. 오토마타 기반의 정적 분석 엔진

리버싱 방식은 그림 3과 같이 어셈블리나 원하는 상위 언어로 변환하여 수동적으로 악성코드를 분석하였다.

그림 4는 악성코드를 탐지하기 위하여 유한 오토마타를 사용하여 정적분석하는 예를 보여주고 있다. 파일 내부에서 사용되는 함수들을 유한 오토마타로 표현하여 상호 관계 및 연관성을 파악하여 해당 파일에 대한 악성코드 여부와 정상 파일여부를 자동으로 분석하게 한다. 내부 함수를 통해서 출력된 유한오토마타 정보는 코드의 흐름을 나타내고 있고 이 정보의 경우 정상적인 경우와 비정상적인 경우를 파악할 수 있다. 기존의 정적분석 방법에서는 악성코드를 사람이 직접 분석하여 악성여부를 결정하고 악성인 경우 해당 파일을 해쉬 값으로 그 패턴을 검사하고 있다.

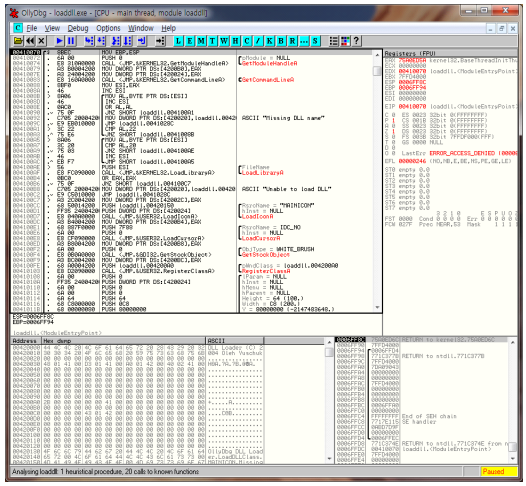

그림 3. 리버싱 툴 샘플
Fig. 3. Sample of the Reversing Tool

본 연구에서는 정적분석을 자동화하기 위해서 해쉬 값이 아닌 오토마타를 구조화하여 데이터베이스에 저장되어 있는 오토마타와 비교하여 악성인지를 판단할 수 있도록 하며 악성이 아닌 경우에도 다음 단계인 동적분석 단계에서 2차 분석이 수행하여 새로운 악성코드의 오토마타를 데이터베이스에 저장하여 정적분석 방법에 사용할 수 있게 하였다.

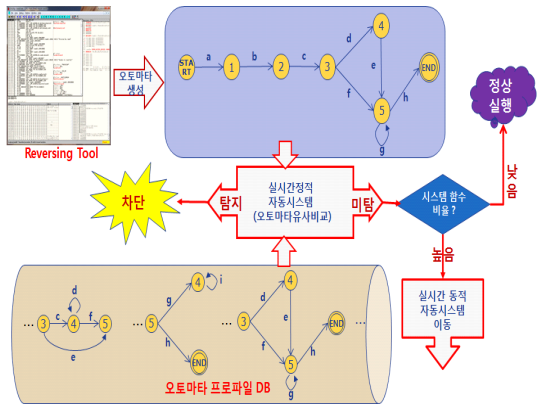

그림 4 유한오토마타 분석과정
Fig. 4. Finite Automata Analysis Process

오토마타로 표현되는 악성코드로 인식되는 패턴들은 다음과 같다.

- 무한 루프형 : 시작 시점부터 순차적인 진행이 아니라 특정 노드에서 무한 반복하는 형태의 악성코드로 시스템 내부에 실행되어 지속적인 악성행위를 하며 종료되지 않는 특징을 가지고 있다.
- 자가 복제형 : 바이러스나 웜처럼 자가 복제를 통해 시스템의 자원을 무한대로 사용하여 고갈시키는 형태로 프로세스 복제형과 파일 복제형으로 크게 나눌 수 있다.
- 시스템 데이터 유출형 : 실행된 시스템 내부의 데이터 중 주요한 파일인 패스워드 파일 및 인증서 등의 파일을 찾고 이를 외부로 유출하는 특징을 가지고 있다.
- 시스템 데이터 파괴형 : 실행된 시스템을 파괴할 목적으로 매우 민감한 함수를 통해 시스템을 자체 공격하는 형태의 특징을 가지고 있고 주로 파일 삭제나 디스크 포맷 등의 치명적인 행위를 수행한다.
- 외부 통신 의존형 : 실행 시점에는 특정한 작업을 수행하지 않으면서 외부의 통신 데이터에 따라 특정 악성행위를 하는 특징을 가지며 대개 악성코드의 유형으로 일반적인 통신 프로그램과 구별하기가 매우 어려운 특징을 가지고 있다.

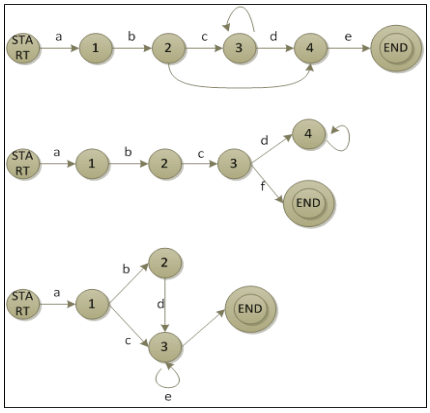

그림 5. 리버싱을 이용한 유한오토마타 샘플
Fig. 5. Finite Automata Sample using Reversing

유한 오토마타의 비정상적인 특징들을 그림5와 같이 대표적인 악성코드 유형에 대해서 보여주고 있다.

위의 특징을 가진 악성코드의 유한오토마타를 통해 특정 프로파일을 행위기반으로 생산하며 이는 기존의 알려진 패턴과는 완전히 다른 접근 방식이며 이를 통해 변형된 유사 악성코드를 실시간으로 검색 및 분석할 수 있다.

IV. 실험 및 고찰

1. 실험 환경

본 장에서는 제안하고자 하는 악성코드 분석 시스템을 위하여 실험 환경을 구축하고 시나리오에 따라 테스트를 진행하며, 테스트 결과를 바탕으로 평가를 내린다. 그림 6은 실시간 악성코드 분석 시스템을 평가하기 위한 테스트 구성을 보여주고 있다.

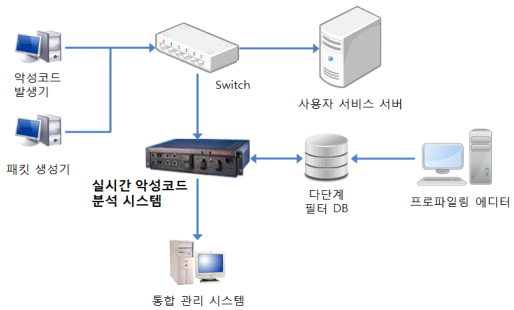

그림 6. 실시간 악성코드 분석 시스템 테스트 구성도
Fig. 6. Realtime Malware Analysis System Testing Diagram

테스트 구성도에서 악성코드를 분석하기 위해 테스트 시나리오는 다음과 같은 단계로 진행한다.

1단계 : 악성코드 발생기를 통한 악성 패킷과 패킷 생성기를 통한 일반 패킷을 동시에 발생시켜 사용자 서비스 서버에 전송한다. 전송된 패킷을 Switch단에서 Mirroring하여 악성코드 분석 시스템에 보낸다.

2단계 : 악성코드 분석 시스템에서 패킷을 분석하여 다단계 필터를 거쳐 악성코드 여부를 판단한다. 악성코드로 판단되면 다단계 필터 DB에 업데이트 한다. 악성 유무와 상관없이 분석된 정보는 통합 관리 시스템에 전달되어 각종 로그를 저장한다.

3단계 : 프로파일링 에디터는 악성코드 분석 시스템의 핵심이 되는 다단계 필터 DB(능동형 필터 DB와 동적 필터 DB)의 정보를 추가/수정/삭제 할 수 있고, 새로운 악성코드 분석 정보를 외부에서 업데이트 할 수 있어, DB에 대한 탐지율을 높이고 신뢰도를 향상시키는 역할을 한다.

2. 실험 결과

이 절에서는 실시간 악성코드 분석 시스템을 평가하기 위하여 테스트 시나리오에 따라 진행한 결과를 평가한다.

악성코드의 검증방법은 기존에 알려진 악성코드와 알려지지 않은 신규 악성코드, 정상 파일을 무작위로 보내어 시스템에 대해 탐지율, 오탐율, 신규 프로파일링 측정, 처리 단위시간 등 각 항목에 대하여 측정 및 검증한다.

(1) 탐지율 측정

악성코드 생성기에서 패킷을 보냈을 때 악성코드 패킷에 대한 탐지 확률 측정(탐지율 목표: 80%이상)

(2) 오탐율 측정

악성코드가 아닌 정상 파일 패킷을 보냈을 시에 악성코드로 판단하고 차단하는 오탐확률 측정(목표 : 10%이하)

(3) 신규 프로파일링 측정

악성코드 생성기에서 보낸 패킷이 기존에 분석된 악성코드가 아닌 새로운 악성코드로 판단되었을 경우 악성코드에 대한 프로파일링 패턴이 정확히 생성이 되고 능동형 DB에 추가가 되는지 측정

(4) 처리 단위시간 측정

정상 파일이 아닌 악성코드 패킷을 자동으로 분석하는데 까지 걸리는 시간 측정 (목표 : 200M bps 이상)

(5) 악성링크탐지 성능

링크를 통한 악성코드 유포 방법으로 악성코드 링크에 대한 탐지 성능 측정(목표 : 2G bps 이상)

표2에서는 유한 오토마타를 이용한 악성코드 분석 시스템의 검증을 위하여 테스팅 환경을 구축하여 결과를 도출한 내용이다.

표 2. 악성코드 검증을 위한 테스팅 결과
Table 2. Testing Results for the Verification of Malware

평가항목 (주요성능)	단위	국내 제품 성능	제안연구 실험결과			평가방법
			1차 실험	2차 실험	3차 실험	
악성코드링크탐지성능	bps	2G	1G	1.1G	1.2G	IXIA 테스트
악성코드분석성능	bps	200M	200M	200M	210M	IXIA 테스트
탐지확률	%	70%	69%	70%	71%	IXIA 테스트
오탐률	%	10%	3.0%	3.1%	3.4%	IXIA 테스트
프로파일링생성성능	bps	NA	1G	1G	1G	자체 데몬테스트

위 실험 결과에서는 제안하고자 하는 연구 결과와 비교하기 위하여 국내 보안장비의 성능을 비교 분석한 공인시험 성적 결과자료를 기반으로 비교하고자 한다. XIA는 테스트 장비 이름이며, Smartbit나 IXIA는 상용 장비로 네트워크 트래픽을 테스트하기 위한 시뮬레이션 환경을 제공해주는 장비이다. 네트워크 트래픽에 간헐적으로 악송코드를 섞어서 보낼 수 있는 기능을 가지고 있어서 선택한 장비이다.

테스트는 1차, 2차, 3차에 걸쳐 악성코드 분석 시스템에 대한 검증을 실시하였으며, 세 번의 테스트 실시는 기존에 알려지지 않은 악성 코드에 대한 탐지와 탐지률의 향상을 관찰하기 위한 방법이다. 결과에서처럼 악성코드 링크 탐지 성능은 1차에서 3차까지 실험에서 평균 1G 속도를 보이고 있는데 본 연구는 악성링크를 탐지하는 기술보다는 악성코드 프로그램 탐지를 위해 제안하는 내용에 초점을 맞추고 있어 국내 솔루션 목표에 는 다소 떨어진 결과를 보이고 있다. 그러나 본 연구에서 가장 중요한 항목으로 분석 성능에서 실제 악성코드 파일을 분석하는 성능이 200M 이상 나온 것은 실시간으로 파일을 분석하고 검증할 수 있다는 증거이다. 즉, 1G의 트래픽 중 파일이 포함될 확률을 10%로 보았을 때 100M 이상의 성능이 나오면 실시간 분석이 가능하므로 본 테스트의 결과는 매우 고무적이다.

탐지확률은 평균 70%로 국내 솔루션 성능과 유사하게 나오고 있으며 80% 이상 향상시킬 수 있는 추가 연구가 필요하다고 할 수 있다. 그리고 오탐률은 본 연구에서 가장 높은 성능 테스트 결과로 국내 솔루션 10%에 비해 3%는 매우 높은 성능 결과를 보여주고 있다. 이는 테스트 데이터의 편차에 따라 달라질 수 있는 요소지만 실제 시스템을 제작하는 과정에서는 이 또한 매우 중요한 평가 항목으로 취급할 필요가 있다.

따라서 테스트 결과에서처럼 성능과 탐지률에 대한 평가 요소들이 목표에 근접한 결과 값을 갖는 것은 효과적인 악성코드 분석 시스템을 수행하고 있음을 알 수 있다.

V. 결 론

최근에 3.20 사태와 같이 사회적으로 문제가 되고 있는 것이 악성코드를 이용한 사이버 공격이다. 새로운 악성코드의 출현과 더불어 기존의 악성코드를 이용한 변종 역시 커다란 피해를 주고 있다. 악성코드의 종류가 다양해진 만큼 다양한 동작과 방법으로 공격을 하고 피해를 주기 때문에 탐지에서 분석 그리고 검증 단계의 필터를 통해 악성코드를 추출하여 차단해야 한다. 그리고 인적자원의 이용을 최소화 한 자동화 시스템으로서 실시간으로 악의적인 접속을 차단하여 악성코드로 인한 피해를 막고 자료 유출을 방지하는 시스템 개발이 필요하다.

따라서 본 연구에서는 악성코드 분석방법에서 악성코드라

고 의심되어지는 파일을 보다 정확하게 판단하기 위해서 악성 행위에 대해서 유한 오토마타를 이용한 프로파일링 기법을 도입하고 또한 이를 이용하여 수동이 아닌 자동으로 실시간 악성코드를 판단할 수 있는 실시간 악성코드 분석 시스템(RMAS : Realtime Malware Analysis System)을 제시하였다.

향후 연구에서는 현재 연구된 자동화 방안들이 프로그램의 함수를 외형적으로 분석하였으나 실제로 동적 모듈에서는 해당 함수가 다르게 동작하는 경우도 간혹 발생하고 있어 분석의 정확도를 높이기위해서는 동적 검증 모듈 기능이 보다 정확하게 설계될 필요가 있다.

참고문헌

〔1〕 Symantec Corporation, "Symantec Global Internet Security Threat Report," Apr. 2010.

〔2〕 AV-TEST - The Independent IT-Security Institute, www.av-test.org

〔3〕 K. Thomas, and D.M. Nicol, "The Koobface Botnet and The Rice of Social Malware," IEEE Int. Conf. Malicious and Unwanted Software (Malware'10), pp. 63-70, Oct. 2010.

〔4〕 Boo Joong Kang, Kyoung Soo Han, Eul Gyu Im, "Malware Current Status and Detection Technology," Communications of the Korea Information Science Society, vol. 30, no. 1, pp. 44-53, Jan. 2012.

〔5〕 Intel Corporation, Intel 64 and IA-32 Architectures Software Developer's Manual Volume 2B: Instruction Set Reference, N-Z, Intel Corporation, March 2010.

〔6〕 A. Moser, C. Kruegel, and E. Kirda, "Limits of Static Analysis for Malware Detection," In ACSAC, pp. 421-430, Dec. 2007.

〔7〕 JooBeom Yun, YoungJoo Shin, "MiGuard : Detecting and Guarding against Malicious Iframe through API Hooking," IEICE Electronics Express, pp. 460-465, 2011.

〔8〕 G. Jacob. H. Debar, and E. Filiol, "Behavioral Detection of Malware: from a Survey towards an established Taxonomy," Journal in Computer Virology, vol. 4, no. 3, pp. 251-266, 2008.

〔9〕 Dwan Dong, Bavid Brumley, BitBlaze, "A New Approach to Computer Security via Binary Analysis," ICISS 2008, pp. 1-25, 2008.

〔10〕 Zhiqiang Lin, Xiangyu Zhang, "Automatic Reverse Engineering of Data Structures form Binary Execution," NDSS 2010, 2010.

저 자 소 개

김 효 남

1988 : 홍익대학교 전자계산학과 이공학사.
1990 : 홍익대학교 전자계산학과 이공학석사.
현 재 : 홍익대학교 컴퓨터공학과 박사과정.
관심분야: 사이버 보안, 게임 보안
Email : hnkim@ck.ac.kr

박 재 경

1994: 동국대학교 컴퓨터공학과 공학사.
1996: 홍익대학교 전자계산학과 이학석사.
2002: 홍익대학교 전자계산학과 이학박사
현 재: 학국과학기술원 사이버보안연구센타 연구위원
관심분야: 네트워크 보안, 사이버 보안
Email : wildcur@kaist.ac.kr

원 유 헌

1972 : 성균관대학교 수학과 이학사.
1975 : 한국과학기술원 전자계산학과 이학석사.
1985 : 고려대학교 전자계산학과 이학박사.
현 재 : 홍익대학교 컴퓨터공학과 교수
관심분야: 프로그래밍 언어론, VoIP, 네트워크 보안
Email : yhwon@hongik.ac.kr