情報保護學會論文誌
第19卷 第2號, 2009. 4

윈도우 악성코드 분류 방법론의 설계

서 희 석,[1†‡] 최 중 섭,[2] 주 필 환[2]
[1]한국기술교육대학교, [2]한국정보보호진흥원

Design of Classification Methodology of Malicious Code in Windows Environment

Hee-Suk Seo,[1†‡] Joong-Sup Choi,[2] Pill-Hwan Chu[2]
[1]korea University of Technology and Education,
[2]Korea Information Security Agency

요 약

인터넷 기술의 발전과 더불어 다양한 악성코드들이 제작되고 있다. 본 연구에서는 윈도우 환경에서 동작하는 악성코드를 분류하기 위한 방법론을 제시하고 시험용 분류 시스템을 소개한다. 악성코드는 매일 수천 건씩 발생하고 있으며, 이를 체계적으로 분류하여 발견된 바이러스가 기존의 악성코드와 어느 정도 유사한지에 대한 판단기준을 설정할 필요가 있다. 변종인 경우에는 이전 악성코드와의 유사성이 어느 정도인지에 대한 유사도 제시가 필요할 것이다. 이러한 분석은 악성코드 분석가들의 업무 노드를 줄여줄 수 있을 뿐만 아니라, 악성코드 분석가들의 성향에 따라 다르게 분석될 수 있는 오류를 줄여 줄 수 있다. 본 연구에서는 악성코드를 크게 9개의 그룹으로 분류하고, 이를 다시 그룹의 특성이 맞는 여러 개의 클러스터로 구분하였다. 악성코드가 소속되는 각각의 클러스터에서는 기준점을 기반으로 악성코드의 유사도가 계산되며, 이 유사도에 의해서 악성코드 분석가들은 기존의 악성코드와 새로운 악성코드의 유형 및 관련 정도를 파악하게 된다.

ABSTRACT

As the innovative internet technologies and multimedia are being rapidly developed, malicious codes are a remarkable new growth part and supplied by various channel. This project presents a classification methodology for malicious codes in Windows OS (Operating System) environment, develops a test classification system. Thousands of malicious codes are brought in every day. In a result, classification system is needed to analyzers for supporting information which newly brought malicious codes are a new species or a variety. This system provides the similarity for analyzers to judge how much a new species or a variety is different to the known malicious code. It provides to save time and effort, to less a faulty analysis. This research includes the design of classification system and test system. We classify the malicious codes to 9 groups and then 9 groups divide the clusters according to the each property.

Keywords: malicious code, classification method, similarity, virus, clustering

I. 서 론

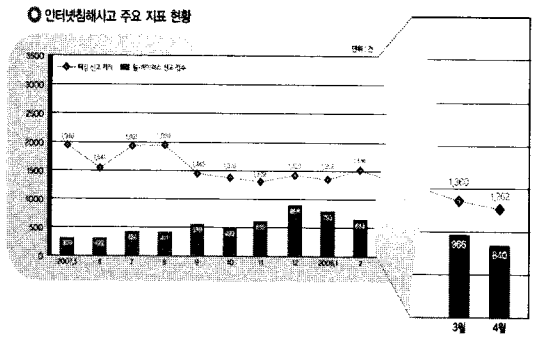
악성코드(malicious code)란 컴퓨터에서 사용자가 원하지 않는 일을 사용자 몰래 하는 소프트웨어를 총체적으로 일컫는 것으로, 컴퓨터 바이러스, 웜, 트로이목마 프로그램 등이 모두 여기에 속한다. 컴퓨터 악성코드는 빠르게 진화하고 있으며, 다양한 시스템 상의 취약성을 이용하여 악의적인 활동들을 수행하고 있다. 최근에도 다양한 경로를 통해 악의적인 공격자에 의한 침투는 계속 되고 있는 상황으로 악성코드 분

접수일(2008년 12월 29일), 수정일(2009년 2월 3일), 게재확정일(2009년 2월 10일)
† 주저자, histone@kut.ac.kr
‡ 교신저자, histone@kut.ac.kr

석에 대한 연구는 지속적이며 체계적으로 이루어져야 한다[1,2].

우리나라는 세계적인 IT 인프라 구축이라는 명성에 걸맞지 않게 많은 분야에서 보안상 취약점을 갖고 있는 상황이다. 최근 중국해커에 의한 (주)옥션 공격, 유럽입자물리연구소(CERN)의 사이트 해킹 사고 등 다양한 분야에서 공격이 이루어지고 있음을 감안할 때 악성코드를 체계적으로 분류하기 위한 방법론의 구축은 시급이 이루어져야할 것이다.

[그림 1]은 2007년 5월에서부터 2008년 4월까지의 악성코드 신고 접수를 정리한 그림이다. 3월에 비해 4월의 신고건수가 약 10% 정도 줄어들었다고 하더라도 2007년의 신고 건수에 비하여 2-3배 정도로 신고건수가 많아진 것을 알 수 있으며, 신고자체가 이루어지지 않은 것을 감안한다면 악성코드로 인한 피해 건수를 가히 막대하다고 할 수 있다[3,4].



[그림 1] 최근 악성코드 신고 현황

[표 1]에서 보이는 바와 같이 주요 문제의 악성코드는 트로이목마 류가 대부분임을 알 수 있다. 그 외에 웜과 드롭퍼가 뒤를 잇고 있다.

악성코드는 매일 수천 건씩 발생하고 있으며, 이를 체계적으로 분류하여 발견된 바이러스가 기존의 악성코드와 어느 정도 유사한지에 대한 판단기준을 설정할 필요가 있다[5]. 변종인 경우에는 이전 악성코드와의 유사성이 어느 정도인지에 대한 유사도 제시가 필요할 것이다. 이러한 분석은 악성코드 분석가들의 업무 노드를 줄여줄 수 있을 뿐만 아니라, 악성코드 분석가들의 성향에 따라 다르게 분석될 수 있는 오류를 줄여줄 수 있다[6,7].

본 논문에서는 악성코드를 크게 9개의 그룹으로 분류하고, 이를 다시 그룹의 특성이 맞는 여러 개의 클러스터로 구분하였다. 악성코드가 소속되는 각각의 클러스터에서는 기준점을 기반으로 악성코드의 유사도가 계산되며, 이 유사도에 의해서 악성코드 분석가들은 기존의 악성코드와 새로운 악성코드의 유형 및 관련 정도를 파악하게 된다.

II. 악성코드 분류

2.1 악성코드 그룹

악성코드는 악성 또는 악용 가능한 소프트웨어의 집합으로, 바이러스, 웜, 스파이웨어, 악성 애드웨어 등 사용자와 컴퓨터에게 잠재적으로 위험이 되는 모든 소프트웨어를 총칭하는 말이며 사전적의미로 멀웨어(malware)는 'malicious software(악의적인 소프트웨어)'의 약자로, 사용자의 의사와 이익에 반해 시스템을 파괴하거나 정보를 유출하는 등 악의적 활동을 수행하도록 의도적으로 제작된 소프트웨어를 말한다[8-10]. 국내에서는 '악성 코드'로 번역되며, 자기 복제와 파일 감염이 특징인 바이러스를 포함하는 더 넓은 개념이라고 할 수 있다. 본 논문에서는 악성코드를 그 특성에 따라 총 9개 분류[11-18]하였으며 각각의 특징은 아래와 같다.

2.1.1 트로이 목마(Trojan)

유용한 프로그램으로 가장하여 사용자가 그 프로그램을 실행하도록 속인다. 사용자가 의심하지 않고 그 프로그램을 실행하게 되면 실제 기대했던 기능을 수행한다. 실제 목적은 사용자의 합법적인 권한을 사용해 시스템의 방어체제에 침해하여 접근이 허락되지 않는

[표 1] 2008년 최근 3개월간 유형별 신종(변종) 악성코드 발견 현황

	웜	트로이	드롭퍼	스크립트	파일	매크로	부트	부트/파일	유해가능	비윈도우	합계
02월	43	281	21	3	3	0	0	0	5	0	356
03월	29	675	48	6	9	1	0	0	46	0	81
04월	32	573	64	3	9	0	0	0	19	0	700

정보를 획득하는 것이다. 전자 메일 메시지와 함께 전송되는 전자 메일 첨부 파일을 통해 주로 확산된다.

2.1.2 다운로더(DownLoader)

인터넷을 통해 파일을 다운로드하는 프로그램이다. 다운로더는 최근 프로그램에서 지원하는 자체 업데이트 기능으로 볼 수 있다. 하지만, 악성코드나 스파이웨어로 분류되는 다운로더는 대체로 설치된 제품과 독립적으로 실행되며 사용자에게 동의나 통보 없이 실행파일을 다운로드하여 시스템에 설치한다. 주로 스팸메일 또는 사이트 방문 시 설치되는 Active-X 컨트롤을 통하여 설치하는 것이 일반적인 설치 경로이다. 최근에는 악성코드 자체에는 다운 기능만 하고 받은 악성코드가 시스템에 영향을 미친다.

2.1.3 파일바이러스(file virus)

감염되는 파일은 파일 바이러스의 종류에 따라 실행 파일과 자료 파일의 두 가지로 나눌 수 있다. 실행파일에는 COM 파일, EXE 파일 등 단독 실행 파일, 오버 레이 파일(overlay file), 주변기기 구동 프로그램(device driver) 등이 있으며, 자료 파일은 MS 오피스(워드, 엑셀 등)에서 사용되는 매크로(macro)를 포함하는 파일이 대부분이지만 최근에는 HTML, VBS(Visual Basic Script)같은 텍스트 형태의 파일도 컴퓨터 바이러스 감염대상이 되어 문제가 되고 있다. 바이러스에 감염되면, 컴퓨터 기동시간이 평소보다 오래 걸리거나, 기동 자체가 되지 않거나, 프로그램이 실행되지 않는다. 그리고 프로그램을 실행시키는 시간이 평소보다 오래 걸리거나, 파일목록을 확인하는 명령을 하였을 때 목록이 화면에 나타나는 시간이 오래 걸린다. 또 화면에 이상한 글자가 나타나거나, 프로그램의 크기가 달라져 있거나, 프로그램의 작성일자 또는 파일의 이름이 바뀌는 등의 증세를 나타낸다. 이처럼 시스템에 영향을 주는 행동을 한다.

2.1.4 웜(WORM)

웜 프로그램은 실행코드 자체로 번식하는 유형을 말하며 주로 pc 상에서 실행된다. 감염대상을 갖고 있지 않다는 면에서 바이러스와 다소 차이가 있지만, 나쁜 의도로 만들어진 프로그램이라는 것을 일반인들에게 쉽게 인식시키기 위해서 보통 웜 바이러스라고 통용된다. 실제 목적은 사용자의 합법적인 권한을 사용해 시스템의 방어체제에 침투하여 접근이 허락되지 않는 정보를 획득하는 것이다. 웜은 자기 스스로의 증식을 목적으로 하는데, 보통은 파일 스스로 그런 기능을 가지고 있거나 윈도우의 운영체제인 경우 시스템 등에 자기 자신을 감염시킨다.

2.1.5 드로퍼(Dropper)

드로퍼는 자신을 복제하는 기능은 없지만 컴퓨터 바이러스를 전파시킬 수 있는 위험이 있다. 바이러스나 웜, 스파이웨어 등의 악성 프로그램을 내부에 포함하여 실행 시 내부 프로그램을 특정 위치에 생성하고 실행한다. 정상 프로그램으로 위장하여 실행을 유도하거나, 다른 악성코드 및 스파이웨어에 의해 실행된다.

2.1.6 키로거(Keylogger)

키보드 상의 키 입력을 감지하여 은밀히 기록하는 악성 프로그램이다. 인터넷 뱅킹 또는 사이버 증권 거래 시 키보드로 입력하는 아이디, 각종의 비밀번호, 신용카드 번호, 주민 등록번호, 웹사이트 접속 ID 와 비밀번호 등의 타이핑된 기록을 암호화되기 이전 상태에서 가로채어 외부의 해커에게 전송한다. 백그라운드로 실행되는 프로그램으로 키보드상의 키 입력을 모두 기록한다. 일단 키 입력이 기록되면 나중에 유출할 수 있도록 컴퓨터내부에 은닉되거나 공격자에게 그대로 전송된다. 공격자는 암호나 유용한 정보를 찾아내고자 이를 세심히 추적하고, 그것들을 사용해 시스템을 위험에 빠뜨리거나 다른 공격수단으로 사용한다.

2.1.7 봇(Bot)

인터넷상에서 가장 보편적으로 존재하는 봇들은 스파이더, 크로울러로 불리는 프로그램들로서 웹사이트들에 주기적으로 방문하여 검색엔진의 색인을 위한 콘텐츠를 모아오는 일을 한다. 하지만 요즘엔 이를 악용하여 인터넷을 돌아다니며 웹 페이지의 정보를 수집해 놓다가 해커의 명령에 따라 정보를 유출하거나 특정 사이트를 공격하는 악성 프로그램으로 정의되며 이 악성 봇에 IRC봇 중 한 종류이다. 해커는 공격대상 pc에 window 취약점, Email, 인터넷 웹페이지를 통해 악의적원 에이전트를 실행시켜며 이를 통해 정보획득, 특정 사이트 공격(DDOS)에 이용한다.

2.1.8 백도어(Backdoor)

백도어란 시스템에 비 인가된 접근을 가능하게 하는 프로그램을 나타낸다. 백도어는 시스템 설계자나 관리자에 의해 고의로 남겨진 시스템의 보안 취약점으로 응용 프로그램이나 운영체제에 삽입된 프로그램 코드이다. 즉 백도어는 시스템 접근에 대한 사용자 인증 등 정상적인 절차를 거치지 않고 응용 프로그램 또는 시스템에 접근할 수 있도록 한다. 특정 포트를 항상 오픈하고 해커의 들어 올 수 있도록 도와주며 항상 백그라운드로 실행되며 명령 서버에 자신에 존재한다는 특정 패킷(ping 등)을 보내기도 한다.

2.1.9 스파이웨어(spyware)/애드웨어(adware)

스파이웨어(spyware)는 스파이(spy)와 소프트웨어의 합성어로, 본래는 어떤 사람이나 조직에 관한 정보를 수집하는 데 도움을 주는 기술을 뜻한다. 그러나 최근에는 다른 사람의 컴퓨터에 몰래 숨어들어가 있다가 중요한 개인정보를 빼가는 프로그램을 지칭한다. 대개 인터넷이나 PC통신에서 무료로 공개되는 소프트웨어를 다운로드 받을 때 함께 설치된다. 애드웨어란 제한 없이 무료로 사용되는 프리웨어(freeware), 일정한 금액으로 제품을 사야 하는 셰어웨어(shareware) 같이 "광고를 보면 사용할 수 있는 프로그램"을 뜻한다. 그리고 광고를 보여주는 프로그램이라는 의미로도 사용된다. 하지만 요즘엔 강제적인 광고를 사용자 pc에 띄워서 사용자의 불편과 프라이버시를 야기하는 악성 프로그램을 뜻한다.

2.2 그룹별 클러스터

본 연구에서는 악성코드를 총 9개의 그룹으로 나누었다. 각 그룹에 포함되어 있는 악성코드는 그 특징에 따라 클러스터로 구분이 되는데 이렇게 클러스터를 만든 이유는 최소한의 기준을 맞추기 위함이다. 예를 들어 다운로더인 경우 다양한 악성코드가 존재하는데, 이 중 몇몇의 악성코드는 네트워크와 파일의 내용을 변경하고, 몇몇 악성코드는 레지스트리와 파일의 내용을 변경한다고 가정할 때 두 클러스터 간의 구분을 용이하게 하기 위하여 그룹을 클러스터로 세분화 하였다. 즉 〔표 2〕에서 1번 클러스터에 속해있는 악성코드는 트로이목마 이면서 파일의 내용만 바꾸는 악성코드이다. 또한 〔표 2〕의 6번 클러스터에 속해있는 악성코드는 트로이목마 이면서 파일, 프로세스, 레지스트리의 내용을 변경하는 악성코드이다. 다운로더의 경우 다운받은 악성코드가 어떤 그룹에 속하느냐에 따라 해당 그룹의 클러스터를 적용 시킨다. 애드웨어와 스파이웨어는 주소만 가지고 접속하기 때문에 클러스터를 나누지 않는다.

※F:file, P:process, R:registry, :network

〔표 2〕 트로이목마 클러스터

No	F_ CREATE	P_CREATE _OTHER	R_ MODIFY _YN	N_ USE
1	1	0	0	0
2	0	0	1	0
3	1	0	1	0
4	0	1	1	0
5	0	0	1	1
6	1	1	1	0
7	0	1	1	1
8	1	1	1	1

〔표 3〕 다운로더 클러스터

No	TYPE
1	TROJAN
2	DROPPER
3	FILE VIRUS
4	KEYLOGGER
5	BOT
6	WORM
7	BACKDOOR
8	ADWARE/SPYWARE

〔표 4〕 백도어 클러스터

No	PACKET 전송
1	0
2	1

〔표 5〕키로거 클러스터

No	F_CREATE	N_USE
1	1	0
2	0	1
3	1	1

〔표 6〕 파일 바이러스 클러스터

No	F_CREATE	P_HOOKING_API_YN	R_MODIFY_YN
1	1	0	0
2	1	1	0
3	1	0	1
4	1	1	1
5	0	1	0
6	0	0	1
7	0	1	1
8	1	1	1

〔표 7〕 드로퍼 클러스터

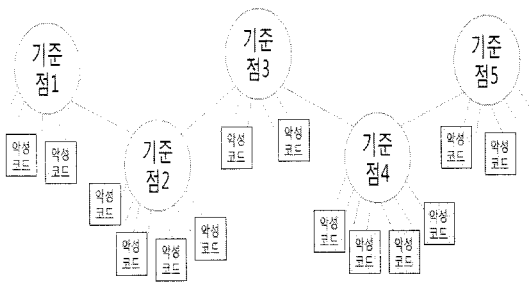
No	P_DLL_INJECTION	R_MODIFY_YN	N_USE
1	1	0	0
2	1	1	0
3	1	0	1
4	1	1	1
5	0	1	0
6	0	0	1
7	0	1	1
8	1	1	1

〔표 8〕 웜 클러스터

No	F_CREATE	P_CREATE_OTHER	R_MODIFY_YN
1	1	0	0
2	1	0	1
3	1	0	0
4	1	1	1
5	0	0	1
6	0	1	1
7	0	0	1
8	1	1	1

2.3 기준점

각 클러스터에는 여러 개의 기준점(representative point)을 설정하였다. 기준점을 설정한 이유는 검색의 효율을 높이기 위해서 이다. 각 클러스터에는 수천 수만개의 악성코드가 존재할 수 있다. 이러한 상황에서 새로운 악성코드를 분석하여 기존의 악성코드와 유사도를 비교하는 것은 검색의 효율이 떨어질 수 있으므로, 기준점을 두어 기준점과 비교를 하여 가장 유사도가 높은 기준점 내의 악성코드들만을 대상으로 유사도를 비교해 보게 된다. 모든 악성코드는 이러한 기준점에 매달리게 된다.



〔그림 2〕 기준점

III. 악성코드 데이터베이스 설계

악성코드 분류 시스템을 개발하기 위해 구성한 DB의 내용은 아래와 같이 구성되어 있다. 분석가가 수행할 수 있는 기본적인 분석사항만을 포함하여 구성하였으며 여러 개의 내용이 있는 경우는 기본적으로 토큰(세미콜론 사용";")을 사용하여 입력할 수 있도록 구성하였다.

〔표 9-14〕는 악성코드를 분류하기 위하여 구성된 DB의 정보를 나타낸다.

〔표 9〕 레지스트리 관련 DB

R_MODIFY_YN	- 0(사용하지 않음) - 1(사용함)	레지스트리를 변경하는가
R_MODIFY_LOCATION	- text	어떤 레지스트리 값을 변경했는가?
R_CREATE_VALUE	- text	어떤 레지스트리를 생성하였는가? (DB 입력시시 (키:내용) (키:내용) 방법 사용)
R_DELETE_VALUE	- text	어떤 레지스트리 값을 삭제했는가? (DB 입력시시 (키:내용) (키:내용) 방법 사용)
R_MODIFY_VALUE	- text	어떤 값을 변경했는가? (DB 입력시시 (키:내용) (키:내용) 방법 사용)

〔표 10〕 네트워크 관련 DB

N_USE	- 0(사용하지 않음) - 1(사용함)	네트워크 연결이 발생하는가?
N_INPORT	- 포트번호	정보를 어떤 포트로 수신하는가?
N_OUTPORT	- 포트번호	정보를 어떤 포트로 전송하는가?
N_PROTOCOL	- TCP - UDP - ARP - SMTP - other	사용하는 프로토콜은 무엇인가?
N_DOWNLOAD	- 0(사용하지 않음) - 1(사용함)	네트워크를 사용해 다운로드하는가?
N_OPEN_URL	- text	악성코드가 요청 모든 URL/IP는 어떻게 되는가?
N_CONNECT_URL	- text	악성코드가 접속 성공한 모든 URL/IP는 어떻게 되는가?
N_UP_URL	- text	업로드 하는 URL/IP는 어떻게 되는가?
N_DOWN_URL	- text	다운로드하는 URL/IP
N_BPS	- number	네트워크 트래픽양은 어떻게 되는가?
N_PPS	- number	네트워크 패킷양은 어떻게 되는가?
N_IRC_SERVER	- text	접속하는 서버 주소은 무엇인가?
N_IRC_CHANNEL	- text	접속하는 IRC 채널은 무엇인가?
N_IRC_NICKNAME	- text	명령자 이름은 무엇인가?
N_IRC_USERNAME	- text	사용자 이름은 무엇인가?
N_IRC_USERMODE	- text	사용자 형태는 어떻게 되는가?
N_IRC_PASSWORD	- text	사용자 패스워드는 어떻게 되는가?

〔표 11〕 프로세스 관련 DB

P_EXECUTE_NAME	- text	실행된 프로그램명 (';' 으로 구별)
P_CREATE_OTHER	- 0(사용하지 않음) - 1(사용함)	자신 외에 다른 프로세스 실행 여부
P_COMPRESS	- 0(사용하지 않음) - 1(사용함)	실행압축이 되어있는가? 되어있다면 어떤 실행압축인가?
P_MODIFY_INJ	- 0(사용하지 않음) - 1(사용함)	특정 DLL이 프로세스에 인젝션 여부
P_CALL_API	- text	불러진 API 목록
P_HOOKING_API	- text	후킹된 API 모든 목록
P_HOOKING_API_YN	- 0(사용하지 않음) - 1(사용함)	API후킹 사용 여부
P_MEMORY_USAGE	- number	프로세스의 메모리 사용량
P_AUTORUN	- 0(사용하지 않음) - 1(사용함)	프로세스의 자동실행 여부
P_AUTORUN_LOCATION	- reg - start - service - other	프로세스 자동실행 등록 위치
P_DLL_INJECTION	- 0(사용하지 않음) - 1(사용함)	DLL 인젝션을 사용 여부
P_DLL_INJECTION_NAME	- text	DLL 인젝션시 사용되는 모든 이름

〔표 12〕 파일 관련 DB

F_SIZE	- file size	악성코드의 용량
F_CREATE	- 0(사용하지 않음) - 1(사용함)	악성코드가 파일을 생성 여부
F_CREATE_TYPE	- exe - dll - txt - html - tmp - com - sys - inf - js - other - bat - vbs - ini - log	파일의 종류
F_CREATE_DIR	- text	생성 위치
F_CREATE_NAME	- text	생성한 파일의 이름
F_DELETE_NAME	- text	삭제된 파일명
F_CREATE_NUM	- number	-악성코드가 생성하는 파일의 수
F_MODIFY_NAME	- Host - iexplorer.exe - scvhost.exe - loader.exe - explorer.exe - other	어떤 파일을 변경했는가?
F_MODIFY_NUM	- number	-몇 개의 파일이나 변경되는가?

〔표 13〕 악성코드 정보 DB

MC_EXE_TYPE	- dll - application - other	실행파일인가 dll 파일인가?
MC_PURPOSE	-network interrupt - keylog -system destroy - remote control -other	악성코드의 목적은 무엇인가? (복수 선택 가능)
MC_TYPE	- worm - trojan - dropper - downloader - password stealer - backdoor - file virus - bot - rootkit - other	악성코드의 종류는 무엇인가? (복수 선택 가능)
MC_SIGNATURE	- text	악성코드가 갖는 특정문자열 혹은 특징은 무엇인가?
MC_CHARACTER	- 0 (잠복형) - 1 (실시간)	악성코드의 성격은 무엇인가?
MC_COMPRESS	- 0 (사용하지 않음) - 1 (사용)	실행압축이 되었는가?
MC_COMPRESS_TYPE	- text	실행압축 종류은 무엇인가?
MC_POINTER	- text	악성코드가 속해있는 기준점인가?
MC_MD5_HASH	- text	악성코드 hash값은 무엇인가?

〔표 14〕 분석 관리 DB

M_CHECK_DATE	- YY-MM-DD	분석한 날짜는 언제 인가?
M_CHECK_Analyzer	- text	분석한 사람 이름은 무엇인가?
M_CHECK_NAME	- text	악성코드 이름은 무엇인가?
M_CHECK_NUMBER	- number	악성코드 일련번호는 어떻게 되는가?

IV. 악성코드 분류 시스템

4.1 점수 계산 방법

새롭게 분석된 악성코드와 기존의 악성코드와의 비교를 위하여 각각의 그룹별로 점수를 계산하는 방법을 사용하였다.

본 연구에서는 9개의 그룹에 대하여 각각의 정량적인 배점기준을 정하고 이에 따라서 점수를 부여하였다. 〔표 15〕는 트로이목마의 점수를 계산하기 위한 배점표이다. 악성코드의 유사도를 계산하는 순서는 아래와 같은 순서로 점수를 부여하게 된다.

〔표 15〕 트로이목마 배점표

테이블 이름	속성		점수	배점 설명
MALWARE CODE	TYPE		10	같은 그룹일 경우 최소한의 점수
	합계		10	
FILE	CREATE DIR		10	CREATE, MODIFY, DELETE를 모두 참조하여 점수를 준다
	MODIFY DIR			
	DELETE DIR			
	CREATE NAME		10	CREATE, MODIFY, DELETE를 모두 참조하여 점수를 준다
	MODIFY NAME			
	DELETE NAME			
	합계		20	
REGISTRY	CREATE KEY		10	CREATE, MODIFY, DELETE를 모두 참조하여 점수를 준다
	MODIFY KEY			
	DELETE KEY			
	CREATE NAME		10	CREATE, MODIFY, DELETE를 모두 참조하여 점수를 준다
	MODIFY NAME			
	DELETE NAME			
	합계		20	
NETWORK	네트워크 미사용	USE	20	NETWORK를 사용하지 않은 경우
	네트워크 사용	PORT	10	NETWORD를 사용할 경우 PORT 와 OPEN_URL에 10점씩 부여
		OPEN_URL	10	
	합계		20	
PROCESS	EXECUTE NAME		5	프로세스상에 같은 이름일 경우 의심
	CALL_API		25	API 조합으로 악성코드 의심
	DLL_INJECTION_NAME			
	합계		30	
총 합계			100	

① 각 항목은 100점 만점으로 점수를 부여한다.

② MD5 Hash 값과 악성코드의 Signature 를 우선적으로 비교한다. 이것이 같으면 거의 흡사한 악성코드이며 100% 의 유사도를 보여준다.

③ 각 테이블별 채점 방법은 다음과 같다.

- FILE

○ 비교하는 두 악성코드가 생성하는 파일이 같은 것이 있을 때에는
(같은 것의 개수) * (배정된 점수 / Bigger 파일생성갯수) 로 점수를 부여한다.

* 여기서 "Bigger 파일생성갯수 " 라는 말은 두 악성코드 중 파일 생성 개수가 많은 것을 의미한다.

○ 문자열로 비교하기 때문에 F_CREATE_NATE 내에서 70%이상 유사한 것은 개별 점수의 50% 의 점수를 준다.

○ 파일 뿐만 아니라 DIR 도 같은 방법으로 점수를 준다.

ex) Trojan 그룹의 두 악성코드가 생성하는 파일이 다음과 같을 때

Sample1	Sample2
arp_.exe	arp1.exe

위 두 파일은 파일이름이 완벽하게 똑같지는 않으나 4글자 중 3글자, 즉 75% 가 유사하기 때문에 트로이목마의 FILE CREATE_NAME 점수의 50% 인 5점을 획득하게 된다.

○ 악성코드의 파일 수정이 부분 집합이 되는 경우, 즉 A는 5개의 파일을 생성하였고 B는 3개의 파일만을 생성하였지만 B가 생성한 파일 3개가 A의 그것에 완전 포함될 경우, 완전 같지만 파일생성 개수만 차이가 있는 경우가 있을 수 있다. 이 개수의 차이가 Bigger 악성코드(여기에선 A 악성코드) 의 개수의 절반보다 차이가 작을 때에만 개별 점수의 90% 의 점수를 준다.

○ 위 사항은 A는 10개의 파일을 생성하는 반면 B가 1개만의 파일을 생성할 때 1/10 확률로 우연히 포함될 수 있다. 하지만 이러한 경우엔 높은 점수를 주지 않기 위한 사항이다.

ex) 트로이목마 그룹의 두 악성코드가 생성하는 파일이 다음과 같을 때

Sample1	Sample2
winavxx.exe	winavxx.exe
hadjajr.ini	hadjajr.ini
vtr.dll	printer.exe
printer.exe	
~DF6C0F.tmp	

위 두 악성코드에서 볼 수 있듯이 Sample1 은 5개, Sample2 는 3개의 파일을 생성하였다. 하지만 Sample2 의 생성파일은 Sample1 의 생성파일에 모두 포함된다.

또한 파일 생성 개수의 차이가 2로, Sample1 의 파일생성 개수의 절반(2.5)보다 작기 때문에 Sample2 악성코드는 Trojan FILE CREATE_NAME 의 90% 점수인 9점을 받게 된다.

- REGISTRY

○ 레지스트리도 FILE 의 배점 방식과 비슷하다.

○ 비교하려는 두 악성코드가 접근하는 레지스트리 같은 것이 있을 때에는 (같은 것의 개수) * (배정된 점수 / Bigger 레지스트리접근갯수) 로 점수를 부여한다.

※ 여기서 Bigger 라는 것은 두 악성코드 중에 레지스트리에 접근하는 개수가 많은 것을 의미한다.

- NETWORK

○ Network를 사용하지 않은 경우와 사용하는 경우로 나뉘어 질 수 있다.

○ USE 항목을 통해 Network 를 사용하지 않는 경우엔 USE 항목의 점수를 부여하고 그렇지 않은 경우엔 PORT와 OPEN_URL을 통해 점수를 부여하게 된다.

- PROCESS

○ API 는 중복을 제외한 전체 API 중에 얼마나 많은 API 가 같은가를 가지고 점수를 부여한다. 예를들어 A에게 25개의 API 목록이 있고, B에게 20개의 API 목록이 있다면 해당 점수표의 API 점수를 25개의 API 에 할당하고 25개중에 몇 개가 같은 지에 비례하여 점수가 부여된다.

○ 만약 API 에 20점이 부여되어 있는 클러스터에서, A가 40개의 API 를 호출하고 B가 30개의 API를 호출할 때 B의 API 와 A의 API가 15개 일치 할 경우 (20점 / 40개) * 100 * 15개 로 계산하여 7.5 점을 부여하게 된다.

V. 결 론

본 연구진은 악성코드를 크게 9개의 그룹으로 분류하고 각 그룹에 대해서 여러 개의 클러스터를 구성하였다. 이렇게 나누어진 그룹에 대한 유사도 계산 알고리즘이 정해지고, 정해진 배점 방식에 따라 기존의 악성코드와 새로운 악성코드의 유사도를 계산할 수 있다. 각각의 클러스터에 포함되는 악성코드의 개수가 증가함에 따라 각 클러스터 내에서 비교해야하는 악성코드의 개수가 증가하게 되는데 이러한 부분에서 성능을 개선하기 위하여 각 클러스터 내에 기준점 되는 악성코드를 선정하였다. 이 기준점을 우선 비교하여 유사도 검색에 있어서 성능 향상을 꾀할 수 있다. 이러한 장점을 사용하여 악성코드 분석가들이 보다 쉽게 기존의 악성코드와의 유사도를 식별함으로써 분석의 효율을 높일 수 있다.

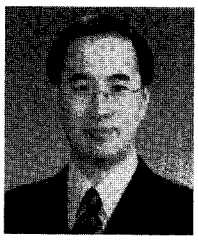
향후 계획으로는 설계된 악성코드 분석 시스템을 바탕으로 방대한 양의 악성코드 DB를 구축하고, 유사도 검색을 위한 보다 유연한 인터페이스를 개발하고자 한다.

참 고 문 헌

[1] 정진성, "최근의 보안 패러다임 변화와 2004년 악

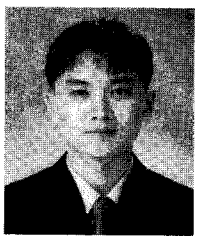
성코드 동향," 정보과학회지, 23(1), pp. 8-14, 2005년 1월.
[2] 박희환, 박대우, "Windows 시스템 파일에 기생하는 악성코드의 치료 방법 연구," 한국컴퓨터정보학회 학술대회논문집, 14(2), pp. 255-262, 2006년 12월.
[3] 신화수, "4월 인터넷 침해사고 민원접수, 처리현황 및 분석," 월간 정보보호뉴스, Vol. 128, No. 5, pp. 12-13, 2008년 5월.
[4] 안철수 연구소, "악성코드 동향 분석 보고서," ASEC 리포트, pp. 2-13, 2008년 4월.
[5] http://www.trendmicro.co.kr/
[6] http://www.ahnlab.com/
[7] http://www.hauri.co.kr/
[8] 최준호, 곽효승, 공현장, 김판구, 이병권, 오은숙, "악성코드 분류 및 명명법에 관한 연구," 정보과학회지, 20(11), pp. 24-29, 2002년 11월.
[9] 염용진, 배병철, "악성 프로그램의 진화," 정보통신연구진흥원, 1244호, pp. 36-42, 2006년 5월.
[10] 장영준, 차민석, 정진성, 조시행, "악성 코드 동향과 그 미래 전망," 정보보호학회지, 18(3), pp. 1-16, 2008년 6월.
[11] NVD: National Vulnerability Database - http://nvd.nist.gov
[12] CVE: Common Vulnerabilities and Exposures - http://cve.mitre.org
[13] CVSS: Common Vulnerability Scoring System - http://www.first.org/cvss
[14] http://www.kasperskylab.co.kr/
[15] http://www.wikipedia.org/
[16] http://ko.wikipedia.org/
[17] http://cafe.naver.com/malzero.cafe/
[18] http://cafe.naver.com/securityplus.cafe/

〈著者紹介〉



서 희 석 (Hee Suk Seo) 정회원
2000년 2월: 성균관대학교 산업공학과 (공학사)
2002년 2월: 성균관대학교 전기전자및컴퓨터공학과 (공학석사)
2005년 2월: 성균관대학교 전기전자및컴퓨터공학과 (공학박사)
2005년 3월~현재: 한국기술교육대학교 인터넷미디어공학부 정보보호전공 조교수
〈관심분야〉 악성코드 분석, 네트워크보안, 보안 시뮬레이션, USN

최 중 섭 (Joong Sup Choi) 정회원
1993년 2월: 인천대학교 전자계산학과 졸업
1995년 8월: 숭실대학교 대학원 컴퓨터학과 석사
2000년 8월: 숭실대학교 대학원 컴퓨터학과 박사
1986년 1월~1994년 3월: 대우통신(주) 연구원
1995년 6월~1996년 2월: 한국전산원 연구원
2000년 7월~현재: KISA 인터넷침해사고대응지원센터 해킹대응팀 팀장
〈관심분야〉 인터넷침해사고대응, 정보보호



주 필 환 (Pill Hwan Chu)) 정회원
2003년 2월: 조선대학교 정보통신공학과 졸업
2005년 2월: 전남대학교 대학원 정보보호 석사
2005년 1월~2006년 5월: 대우정보시스템 사원
2006년 6월~현재: KISA 인터넷침해사고대응지원센터 해킹대응팀 연구원
〈관심분야〉 인터넷침해사고대응, 정보보호