한국시뮬레이션학회 논문지
Vol. 18, No. 1, pp. 63-70 (2009. 3)

윈도우 악성코드 분류 시스템에 관한 연구

서희석[1†] · 최중섭[2] · 주필환[2]

A Study on Windows Malicious Code Classification System

Hee Suk Seo · Joongsup Choi · Pillhwan Chu

ABSTRACT

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. This system provides the similarity for analyzers to save time and effort. It is used prospect system of malicious code in the future.

Key words : Malicious code, Classification methodology, Modeling

요 약

본 과제의 목표는 윈도우 환경에서 동작하는 악성코드를 분류하기 위한 방법론을 제시하고, 시험용 분류 시스템을 개발하는 데 있다. 악성코드를 크게 9개의 그룹으로 분류하고, 이를 다시 그룹의 특성이 맞는 여러 개의 클러스터로 구분하였다. 해당 클러스터에 속하는 악성코드는 최소한 클러스터의 기본 속성은 만족시킨다. 또한 악성코드가 소속되는 각각의 클러스터에서는 기준점을 기반으로 악성코드의 유사도가 계산되며, 이 유사도에 의해서 악성코드 분석가들은 기존의 악성코드와 새로운 악성코드의 유형 및 관련 정도를 파악하게 된다. 악성코드 분류 시스템은 정량적 분석과 정성적인 분석에 대한 결과를 보여주며, 챠트를 통하여 보기 쉽게 내용을 파악할 수 있다. 매일 수천 건의 악성코드가 발견되는 상황에서 악성코드 분석가들에게 기존 악성코드와의 유사도를 제공함으로써 분석의 시간과 노력을 줄여 줄 수 있다. 본 연구의 성과물은 향후 악성코드 예측 시스템의 초석으로 활용될 수 있을 것이다.

주요어 : 악성코드, 분류 방법론, 모델링, 클러스터링

1. 서 론

악성코드(malicious code)[1]란 컴퓨터에서 사용자가 원하지 않는 일을 사용자 몰래 하는 소프트웨어를 총체적으로 일컫는 것으로, 컴퓨터 바이러스, 웜, 트로이목마 프로그램 등이 모두 여기에 속한다[2,3].

2009년 1월 5일 접수, 2009년 1월 22일 채택
[1)] 한국기술교육대학교 인터넷미디어공학부
[2)] 한국정보보호진흥원
주 저 자 : 서희석
교신저자 : 서희석
E-mail; histone@kut.ac.kr

컴퓨터 악성코드는 빠르게 진화하고 있으며, 다양한 시스템 상의 취약성을 이용하여 악의적인 활동들을 수행하고 있다[4]. 최근에도 다양한 경로를 통해 악의적인 공격자에 의한 침투는 계속 되고 있는 상황으로 악성코드 분석에 대한 연구는 지속적이며 체계적으로 이루어져야 한다.

우리나라는 세계적인 IT 인프라 구축이라는 명성에 걸맞지 않게 많은 분야에서 보안상 취약점을 갖고 있는 상황이다. 최근 중국 해커에 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 이루어져야하는 과제이다[5].

악성코드를 크게 9개의 그룹으로 분류하고, 이를 다시 그룹의 특성이 맞는 여러 개의 클러스터로 구분하였다. 해당 클러스터에 속하는 악성코드는 최소한 클러스터의 기본 속성은 만족시킨다. 또한 악성코드가 소속되는 각각의 클러스터에서는 기준점을 기반으로 악성코드의 유사도가 계산되며, 이 유사도에 의해서 악성코드 분석가들은 기존의 악성코드와 새로운 악성코드의 유형 및 관련 정도를 파악하게 된다. 악성코드 분류 시스템은 정량적 분석과 정성적인 분석에 대한 결과를 보여주며, 챠트를 통하여 보기 쉽게 내용을 파악할 수 있다.

현재 국내에서는 악성코드 분류 시스템에 대한 연구는 거의 진행되고 있지 않으며, 대표적인 안티바이러스 업체인 안철수 연구소[6]와 하우리[7] 등에서 자체 시스템을 개발하여 사용하고 있는 실정이다. 해외의 사례로는 소포스, 트래드 마이크로[8], 카스퍼스키[9] 등에서 연구 중이다[10-13].

2. 악성코드 그룹

본 연구진은 악성코드 분류방법론을 위하여 악성코드의 특징에 따라서 총 9개의 그룹을 선정하였다.

2.1 TROJAN(트로이 목마)

- 그리스의 병사들이 목마에서 나와 트로이를 멸망 시킨 것에서 유래
- 사용자가 눈치 채지 못하게 프로그램을 실행
- 합법적인 접근으로 시스템에 침입 허락되지 않은 정보를 획득하는 것을 말함
- 전자메일, 사용자의 파일 공유을 통해 감염
- 바탕화면의 임의의 아이콘 생성, 삭제 등 시스템 변경 파괴, 심한 경우 시스템을 마비시킴

2.2 DOWNLODER(다운로더)

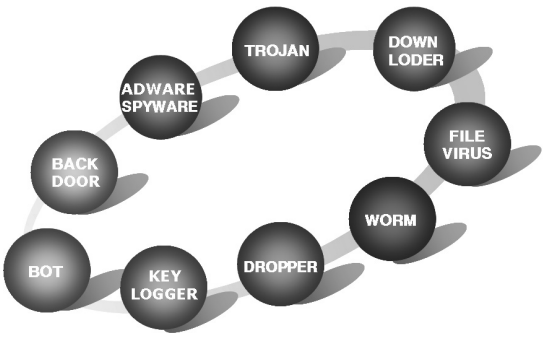

그림 1. 악성코드 그룹의 분류

- 인터넷을 통한 다운로드 하는 프로그램, 악성코드를 다운 함
- 전자메일 첨부 파일 또는 ACTIVE - X를 통하거나 설치 프로그램을 통해 실행
- 시스템마비, 트래픽 증가 등 문제점이 발생

2.3 FILE VIRUS(파일 바이러스)

- 파일에 감염되는 바이러스를 말함
- 실행파일과 자료파일로 구분 할 수 있음
- 전자메일 또는 파일 공유, USB, 엑셀 워드의 매크로, 악의적인 인터넷 웹페이지를 통해 감염
- 시스템 성능저하, 프로그램 작동 불능, 시스템 파괴

2.4 WORM(웜)

- 프로그램 실행 코드 자체로 번식을 할 수 있음
- 사용자의 합법적인 권한을 사용해 시스템의 방어 체제에 침투하여 접근이 허락되지 않는 정보를 획득
- 전자메일, 파일 공유, 스스로 복제를 통해 감염
- 부트영역에 침입하거나 메모리에 상주, 파일에 침입하여 감염
- 시스템 성능에 영향을 미침

2.5 DROPPER(드롭퍼)

- 사용자가 인지하지 못하는 순간에 바이러스 혹은 트로이 목마 프로그램을 컴퓨터에 설치하는 프로그램을 말함
- 프로그램 실행이 특정 위치에 악성코드를 드롭
- 정상프로그램으로 위장 또는 다른 정상프로그램을 인젝션해 실행 시킴

2.6 KEYLOGGER(키로거)

- 키보드의 키 입력 감지를 하여 기록하는 악성 프로그램 사용자의 정보를 해커에게 전송
- 금융정보(신용카드 비밀번호, 비밀번호), 신용 정보(주민등록번호, 전화번호) 등 사용자 정보 획득이 주 목적임
- 다운로더, 드롭퍼 등 악성코드를 통한 설치 또는 인터넷 웹페이지, 전자메일을 통해 감염

2.7 BOT(봇)

- 사용자나 다른 프로그램 도는 사람의 행동을 흉내, 대리자의 역할 수행
- 전자메일 또는 웹페이지, 악성코드를 통해 감염

- 감염시 해커의 명령에 따라 특정 웹사이트를 공격(DDOS) 또는 정보를 유출하는 역할을 함

2.8 BACKDOOR(백도어)

- 비인가된 접근을 허용하게 하는 프로그램
- 시스템의 취약점 또는 악성코드 등 을 통해 감염
- 특정 포트를 오픈하고 해커가 언제나 들어 올 수 있도록 항상 백그라운드로 실행 되어 있음
- 자신의 존재를 해커에 알리거나 해커의 존재를 확인하게 위해 특정 패킷(PING)등 의 신호를 보내기도 함

2.9 ADWARE/SPYWARE(애드웨어/스파이웨어)

- 사용자의 동이 없이 특정 사이트에 접속하게 만들거나 정보를 유출하는 악성코드를 말함
- ACTIVE-X 또는 특정 웹사이트 접속 전자 메일, 악성 프로그램을 통해 감염
- 웹브라우져의 변화, 주기적으로 특정 사이트 접속, 웹페이지에 홍보 등의 악성 목적을 하는 베너 출력, 사용자 정보 유출 등의 역할을 함

3. 그룹 내 클러스터

본 논문은 그룹 내에 클러스터 개념을 적용하여, 그룹을 다시 세분화 하였다. 이것은 크게 두 가지의 의미를 갖게 되는데, 첫 번째로 최소한의 속성을 정의할 수 있다. 부연 설명을 하면 다음과 같다. Trojan 그룹의 악성코드라고 하더라도 어떤 악성코드는 파일을 사용하고, 어떤 악성코드는 파일과 관련된 내용을 사용하지 않을 수 있다. 또한 어떤 악성코드는 파일, 레지스트리의 내용을 변경하는 경우가 있고, 어떤 악성코드는 프로세스, 네트워크의 내용을 변경하거나 사용할 수 있다. 이런 다양한 형태의 악성코드가 존재할 수 있는데, 클러스터는 이러한 악성코드가 어느 정도의 최소 속성을 만족하는지 알 수 있게 된다. 아래 표 1의 3번 클러스터에 포함되는 악성코드는 파일과 관련된 작업들을 수행하고, 레지스트리의 내용을 건드리기는 악성코드이다. 또한 8번의 경우는 파일, 프로세스, 레지스트리, 네트워크의 모든 내용을 사용하거나 변경시키는 악성코드라는 것을 보장할 수 있다. 두 번째의 장점은 유사도 계산을 위한 계산 및 검색의 속도를 향상시키기 위해서 필요한 개념이다. 본 분류방법론에 따라 분류 시스템을 개발하였는데, 어떤 악성코드가 파일만을 변경하는 악성코드라면 파일만을 건드리는 악성코드만을 비교하여 유사도를 계산해 본다면 계산의 속도가 많이 향상될 수 있다. 같은 Trojan 그룹이라고 하더라도 굳이 네트워크만을 사용하는 악성코드는 검색할 필요가 없기 때문이다.

표 1. Trojan 클러스터 현황

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 2. Backdoor 클러스터 현황

No	PACKET 전송
1	0
2	1

표 3. Keylogger 클러스터 현황

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

본 연구진은 9개의 그룹에 대하여 표 1-3과 같이 클러스터를 구분하였다. 표 1-3은 대표적인 클러스터 현황을 나타낸 것이다.

4. 유사도 계산 방법

4.1 정량적 분석

<유사도 계산 방법>

- 표 4와 같이 [MALWARE CODE], [FILE], [REGISTRY], [NETWORK], [PROCESS]의 속성들이 모든 그룹의 악성코드를 채점하는데 사용되며, 각 그룹의 특성마다 속성의 비중이 달라진다.
- 각각의 특징적인 속성이 반영되어 클러스터의 특징을 살린다.

① 각 항목은 100점 만점으로 점수를 부여한다.

표 4. Trojan의 배점 기준

테이블 이름	속성		점수
MALWARE CODE	TYPE		10
	합계		10
FILE	CREATE DIR		10
	MODIFY DIR		
	DELETE DIR		
	CREATE NAME		10
	MODIFY NAME		
	DELETE NAME		
	합계		20
REGISTRY	CREATE KEY		10
	MODIFY KEY		
	DELETE KEY		
	CREATE NAME		10
	MODIFY NAME		
	DELETE NAME		
	합계		20
NETWORK	네트워크 미사용	USE	20
	네트워크 사용	PORT	10
		OPEN_URL	10
	합계		20
PROCESS	EXECUTE NAME		5
	CALL_API		25
	DLL_INJECTION_NAME		
	합계		30
총 합계			100

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

③ 각 테이블별 채점 기준을 사용한다.

- FILE

○ 비교하는 두 악성코드가 생성하는 파일이 같은 것이 있을 때에는

(같은 것의 개수) * (배정된 점수 / Bigger 파일생성갯수)

로 점수를 부여한다.

※ 여기서 “Bigger 파일생성갯수 ”라는 것은 두 악성코드 중 파일 생성 개수가 많은 것을 의미한다.

표 5. Worm의 배점 기준

테이블 이름	속성	점수
MALWARE CODE	TYPE	10
	합계	10
FILE	CREATE DIR	10
	MODIFY DIR	
	DELETE DIR	
	CREATE NAME	10
	MODIFY NAME	
	DELETE NAME	
	합계	20
REGISTRY	CREATE KEY	10
	MODIFY KEY	
	DELETE KEY	
	CREATE NAME	10
	MODIFY NAME	
	DELETE NAME	
	합계	20
NETWORK	USE	20
	PORT	10
	OPEN_URL	10
	합계	10
PROCESS	EXECUTE NAME	10
	CALL_API	20
	DLL_INJECTION_NAME	
	합계	30
총 합계		100

표 6. 비교표

Sample1	Sample2
arp_.exe	arp1.exe

○ 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

위 두 파일은 파일이름이 완벽하게 똑같지는 않으나 4

글자 중 3글자, 즉 75%가 유사하기 때문에 Trojan의 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.
- ○ 나머지 레지스트리, 네트워크, 프로세스도 위와 비슷한 방법으로 정량적인 점수를 계산하게 된다.

4.2 정성적 분석

악성코드는 분석은 여러 방법이 존재하고 있다. 현재 백신 업체(안철수, 하우리 등)에서는 각 분석 방법이 상이하며 기준도 상이한 실정이다. 더욱이 중요한 것은 파일의 변화나 레지스트리에 변화, 프로세스의 변화만을 가지고 악성코드를 판단한다는 것이다. 그리고 악성코드를 분석하여 백신을 만들게 되는 분석가들 사이에도 악성코드 하나에 대한 동작이나 위험정도, 치료 가능 정도 등 눈에 보이지 않는 객관적인 지표에서 판단하는 기준이 매우 다르다는 것이다. 따라서 그 판단하는 객관적인 기준의 필요성이 대두된다. 이에 악성코드에 대한 정성적(위험정도, 치료정도 등)인 것에 수치적인 판단이 필요 하다.

NVD(National Vulnerability Database)는 취약점을 분석, 수집, 저장하고 그 정보를 제공하는 미국의 정부기관이다. NVD에서는 기존에 저장하고 있는 데이터를 토대로 CVSS(Common Vulnerability Scoring System)을 제공하고 있다. CVSS는 취약 정도에 따른 정도를 수치적으로 표현 할 수 있도록 도와주고 있는데 그 판단은 객관적인 지표로 표현 할 수 있는 정도가 아닌 주관적인 판단에 따른 정도를 수치적으로 표현하고 있다. 본 연구에서는 CVSS 의 내용을 참조하여 악성코드에 대한 정성적인 계산을 할 수 있도록 하였다[14-15].

- 결과 출력

항목에서 분석가가 주관적인 방법으로 계산을 하게 되면 시스템의 피해 정도, 기본 점수 및 종합 점수를 출력하게 된다. 결과는 아래와 같다.

- ○ Exploitability 악성코드 접근 범위 점수
- ○ Impact 피해 정도, base 악성코드 정성적 점수
- ○ TemporalScore 악성코드에 대한 치료법에 대한 점수
- ○ EnvironmentalScore 총체적인 점수

5. 시스템 구조

악성코드 분류 방법을 위해 점수를 채점하는 시스템은 위와 같은 구조로 작동한다. 시스템은 악성코드의 정보를 입력받은 INPUT 모듈과 해당 정보로 점수를 계산하는 SCORE 모듈, 그리고 최종적으로 분석결과를 출력하는 OUTPUT 모듈로 이루어진다.

INPUT 모듈은 Web페이지를 통해 이루어진다. INPUT 모듈은 JSP 로 구성된 Web 페이지에 분석 악성코드 정보를 입력받는 역할을 한다. 사용자가 한층 다루기 쉽게 하기 위해 Input 페이지는 Web 페이지로 구성되어 있고 악성코드 분석에 필요한 악성코드기본정보, 파일수정정보,

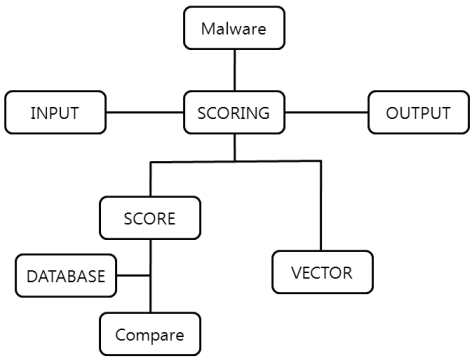

그림 2. 시스템 구조도

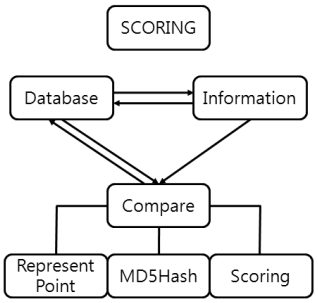

그림 3. SCORING 모듈

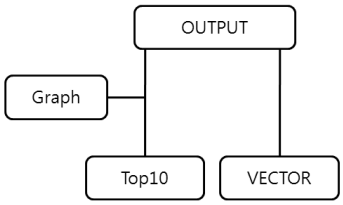

그림 4. OUTPUT 모듈

레지스트리 수정정보, 네트워크 사용, 프로세스 변화 정보들을 입력받는다.

SCORE 모듈은 정량적인 점수를 계산하는 악성코드 비교, 정성적인 벡터 계산으로 나뉘는데 악성코드 비교의 경우 Database에서 비교대상 악성코드를 참조해야 하지만 정성적인 분석인 경우에는 자기 자신의 악성 코드 위험도를 나타내는 지표이기 때문에 자신 외에 다른 정보를 필요로 하지 않는다.

SCORE 모듈은 Database에 있는 악성코드 정보를 이용해 점수 계산을 하고 그 결과를 다시 Database 에 기재하는 역할을 한다.

정성적인 점수의 경우 information은 악성코드의 정보를 Database에서 얻어오며, 점수 부여시 필요한 Compare가 만점점수를 Database에서 얻어와 점수 계산을 하게 된다. 악성코드 비교는 MD5Hash 비교 메소드와 기준점 비교 메소드, 그리고 점수 부여 메소드를 통해 이루어지게 된다.

MD5Hash가 같은 악성코드의 경우 일치하는 악성코드이므로 만점을 가져가게 된다. 악성코드 비교는 전체 정도를 다 비교하는 것이 아닌 기준점을 이용하여 악성코드 비교의 성능을 높인다. 점수부여 메소드는 두 악성코드의 정보들을 비교하여 두 악성코드 간의 유사도를 계산한다.

정성적인 점수를 나타내는 방법은 자신의 정보 외에 다른 정보는 필요로 하지 않는다. 설문조사과 비슷한 느낌을 받을 수 있는 정성적 분석 입력페이지를 통해 얻은 자신의 정보로 정성적인 점수를 계산한다.

OUTPUT 계산된 점수를 출력한다. 출력페이지에서는 분석된 악성코드와 가장 가까운 10개의 악성코드를 출력해주며, 세부 내용을 그래프로 표현한다. 그래프는 Top10 악성코드에 대한 정보를 파일/레지스트리/네트워크/프로세스 분야로 나누어 세세하게 출력해줌으로써 사용자의 비주얼적 독립적인 점수인 정성적 벡터는 다른 악성코드와 비교가 아닌 자기 자신의 위협도 점수를 출력한다.

INPUT 모듈에서 입력받은 정보가 Serial_Compare 클래스로 전달되고 Serial_Compare 클래스에서 Information 객체에 입력받은 정보를 저장한다. 이 information 객체를 Scoring 클래스에서 받음으로써 분석 정보를 각 토큰으로 나누어 점수를 부여한다. 부여된 점수는 다시 Serial_Compare 로 전달되어 각 기준점들과의 비교 등을 통해 유사도 Top10 악성코드를 정리하여 OUTPUT 으로 전달하게 된다.

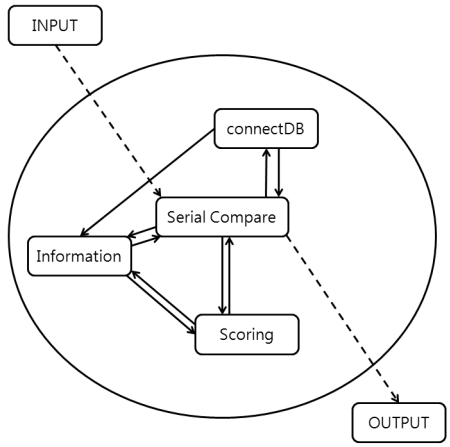

그림 5. 클래스 데이터 흐름도

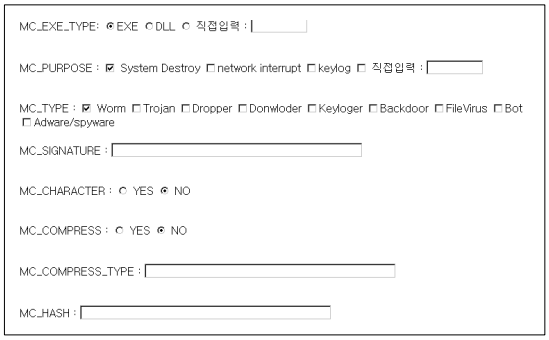

그림 6. 입력 페이지

6. 테스트 시스템

각 사용자들의 분석 할 악성 코드를 입력 페이지를 통해 등록해야 한다. 입력 페이지는 총 6개의 페이지로 나뉘어져 있다. 첫 번째 페이지는 악성코드의 정보를 입력하는 페이지, 두 번째 페이지는 악성코드의 파일 정보를 입력하는 페이지, 세 번째 페이지는 악성코드의 레지스트리 정보를 입력하는 페이지, 네 번째 페이지는 악성코드의 네트워크 정보를 입력하는 페이지, 다섯 번째 페이지

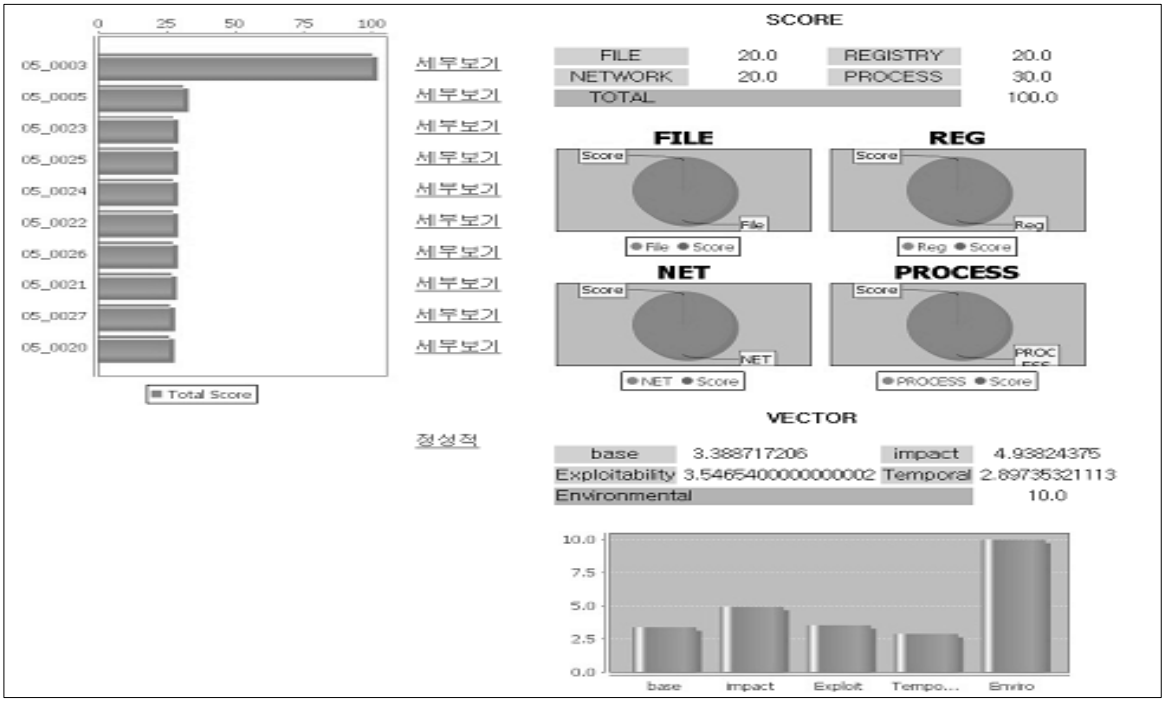

그림 7. 출력 페이지

는 악성 코드의 프로세스 정보를 입력하는 페이지, 여섯 번째 페이지는 악성코드의 프로세스 정보를 입력하는 페이지, 일곱 번째 페이지는 분석가의 정보를 입력하는 페이지로 나눌 수가 있다. 입력 시 문자열 입력에는 ‘;’ , ‘,’ 토큰으로 분리해 입력해야 한다. 기본적인 디폴트 값을 넣어 두었기 때문에 모든 사항을 입력해야만 동작하는 것은 아니다. 하지만 분석의 정확도를 높이기 위해서는 분석가가 분석한 최대한의 많은 내용을 입력하는 것이 바람직하다.

각 입력 페이지에서 받은 정보를 통합하여 챠트로 보여주게 된다. DB의 내용이 자꾸 쌓임에 따라서 분석 시간이 오래 걸려 페이지 전체 표시 시간이 오래 걸린다.

출력 페이지에는 데이터베이스에 저장되어 있는 악성코드의 정보를 분석해서 출력하게 되는데 출력페이지 오른쪽 챠트는 유사도가 가장 가까운 악성코드를 10개 보여주는 그림이다.

그리고 가운데 세부사항으로 각 악성코드가 분석하려는 악성코드와 얼마나 가까운지를 보여준다.

왼쪽 챠트들은 그 악성코드가 어떻게 해 점수를 획득하는 지를 각 파일 네트워크 레지스트리 프로세스 별로 보여 준다.

7. 결론 및 향후 연구계획

본 연구에서는 악성코드 분류 방법론을 연구하였고, 테스트 시스템을 구축하였다. 악성코드 분류를 위하여 9개의 큰 대 그룹을 선정하였으며, 각각의 그룹에 대하여 여러 개의 클러스터로 내용을 분류하였다. 각각의 클러스터를 대상으로 악성코드의 유사도를 측정하게 되는데, 악성코드의 개수가 늘어남에 따라 계산 시간이 증가하게 될 것이다. 이를 개선하기 위하여 기준점이라는 개념을 도입하여 기준점에 악성코드가 포함되도록 구축하였다. 분석의 결과는 정량적 분석과 정성적 분석으로 나누어진다. 정량적 분석은 분석가가 분석한 데이터에 의존하여 수치적으로 계산되는 값이며, 정성적인 분석은 벡터값이다.

분석가가 발견된 악성코드에 대해서 파일, 네트워크, 레지스트리, 프로세스에 대하여 값을 넣고 분석을 하게 되면 해당 클러스터에서 가장 유사도가 높은 10개의 목록을 보게 된다. 각각의 값을 선택하면 계산된 결과를 챠트로 볼 수 있게 된다. 본 연구의 다음 계획으로는 클러스터에 저장되는 DB의 양이 방대해 짐에 따라 계산의 속도 문제가 발생될 것으로 예상된다. 따라서 클러스터 내의 악성코드 분석에 있어서 이를 효율적으로 계산하기 위한

방법이 필요하다. 현재는 기준점의 도입으로 어느 정도의 해결은 되었다고 판단되나 하루에 200 여개의 악성코드가 DB로 쌓이는 경우에는 다른 해결책을 강구해야할 것으로 판단된다. 따라서 이의 해결 방안이 추후 과제라고 할 수 있다. 본 연구를 추후 악성코드 예측 시스템을 구축하기 위한 기초 시스템으로 활용될 수 있다.

참 고 문 헌

1. 안철수 연구소, 2008년 4월 악성코드 동향 분석 보고서, ASEC 리포트.
2. 최준호, 곽효승, 공현장, 김판구, 이병권, 오은숙, "악성코드 분류 및 명명법에 관한 연구", 정보과학회지, 제 20권 제 11호, 2002년 11월.
3. 염용진, 배병철, "악성 프로그램의 진화", 정보통신연구진흥원 주간 기술동향 1244호, 2006. 5.
4. 장영준, 차민석, 정진성, 조시행, "악성 코드 동향과 그 미래 전망", 정보보호학회지 제 18권 제 3호, 2008. 6.
5. 월간 정보보호뉴스 (Vol. 128 No. 5 2008년 5월).
6. http://www.ahnlab.com/
7. http://www.hauri.co.kr/
8. http://www.trendmicro.co.kr/
9. http://www.kasperskylab.co.kr
10. Nancy R.Mead et. al., "Survivable Network Analysis Method", CMU/SEI-2000-TR-013, Sep. 2000.
11. Robert J. Ellison, David A. Fisher, Richard C. Linger, Howard F. Lipson, Thomas A. Longstaff, Nancy R. Mead "Survivability: Protecting Your Critical Systems," IEEE Internet Computing, November December, Vol. 3, pp. 55-63, 1999.
12. F. Cohen, "Simulating Cyber Attacks, Defenses, and Consequences," Computer & Security, Vol. 18, pp. 479-518, 1999.
13. M. Bishop, "Vulnerabilities Analysis," Proceedings of the Recent Advances in Intrusion Detection, pp. 125-136, September, 1999.
14. NVD: National Vulnerability Database - http://nvd.nist.gov
15. CVE: Common Vulnerabilities and Exposures - http://cve.mitre.org
16. CVSS: Common Vulnerability Scoring System - http://www.first.org/cvss

서 희 석 (histone@kut.ac.kr)

2000 성균관대학교 산업공학과(공학사)
2002 성균관대학교 전기전자및컴퓨터공학과(공학석사)
2004～2005 (주)정보감리평가원 선임연구원
2005 성균관대학교 전기전자및컴퓨터공학과(공학박사)
2005～현재 한국기술교육대학교 인터넷미디어공학부 정보보호전공 교수

관심분야 : 네트워크보안, 보안 시뮬레이션, USN

최 중 섭 (jschoi@kisa.or.kr)

1993 인천대학교 전자계산학과 졸업
1995 숭실대학교 대학원 컴퓨터학과 석사
2000 숭실대학교 대학원 컴퓨터학과 박사
1986～1994 대우통신(주) 연구원
1995～1996 한국전산원 연구원
2000～현재 KISA 인터넷침해사고대응지원센터 해킹대응팀 팀장

관심분야 : 인터넷침해사고대응, 정보보호

주 필 환 (chu@kisa.or.kr)

2003 조선대학교 정보통신공학과 졸업
2005 전남대학교 대학원 정보보호 석사
2005～2006 대우정보시스템 사원
2006～현재 KISA 인터넷침해사고대응지원센터 해킹대응팀 연구원

관심분야 : 인터넷침해사고대응, 정보보호