PE 기반 악성코드 자동 분석 결과 관리를 위한 DB 설계†

강홍구*, 오주형*, 임채태*, 정현철*
*한국인터넷진흥원 융합보호R&D팀
e-mail: redball@kisa.or.kr, jhoh@kisa.or.kr, chtim@kisa.or.kr, hcjung@kisa.or.kr

A DB Design for Management of Malware Automated Analysis based on PE

Hong-Koo Kang*, Joo-Hyung Oh*, Chae-Tae Im*, Hyun-Cheol Jung*
*Team of Convergence Security R&D, KISA

요 약

오늘날 인터넷 기술의 발전과 대중화와 함께 다양한 악성코드가 빠르게 제작, 유포되고 있다. 최근 빠르게 증가하는 악성코드를 신속하게 대응하기 위해 자동화된 분석 기법에 대한 연구가 활발히 진행되고 있다. 자동화된 악성코드 분석 결과로 생성되는 데이터는 안티바이러스 업체나 관련 기관 등에서 알려지지 않은 악성코드에 대응할 수 있는 시그니처를 생성하는데 활용된다. 따라서 저장되는 악성코드 분석 결과는 악성코드 사이의 행위와 특성 관계가 고려되어 저장되어야 한다. 즉, 자동화된 악성코드 분석 결과를 효율적으로 저장할 수 있는 DB 설계가 필요하다. 본 논문에서는 악성코드의 대부분을 차지하는 PE를 대상으로 자동화된 악성코드 분석 결과를 효율적으로 저장할 수 있는 DB 설계를 제안하고자 한다.

1. 서론

오늘날 인터넷 기술이 발전하고 인터넷 사용이 대중화되면서 악의적인 목적을 갖는 다양한 악성코드가 빠르게 제작, 유포되고 있다. 최근에는 악성코드가 금전적인 이익을 얻는 수단으로 활용되고 악성코드를 자동으로 제작하는 프로그램까지 등장하면서 악성코드는 하루가 다르게 급속히 증가하고 있다[1,2].

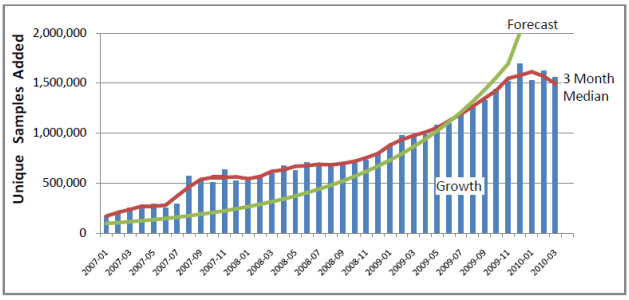
그러나 악성코드로 보고되는 샘플 개수는 크게 증가되는 반면에 분석자가 수동으로 분석할 수 있는 악성코드 개수는 제한적이기 때문에 사전에 샘플을 입수하더라도 사전 대응은 매우 어렵다. 이에 국내외 안티바이러스 업체나 관련 기관에서는 급증하는 악성코드의 신속한 대응을 위해 자동화된 악성코드 분석 기술을 활발히 연구하고 있다[3,4].

자동화된 악성코드 분석 결과로 생성되는 데이터는 안티바이러스 업체나 관련 기관 등에서 알려지지 않은 악성코드에 대응할 수 있는 시그니처를 생성하는데 활용된다. 따라서 저장되는 악성코드 분석 결과는 악성코드 사이의 행위와 특성 관계가 고려되어 저장되어야 한다. 즉, 자동화된 악성코드 분석 결과를 효율적으로 저장할 수 있는 DB 설계가 필요하다. 본 논문에서는 악성코드의 대부분을 차지하는 PE(Portable Executable)[5]를 대상으로 악성코드 자동 분석 결과를 효율적으로 저장할 수 있는 DB 설계를 제안하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대해 기술한다. 그리고 3장에서 정적 분석 결과 관리 DB 설계를 설명하고 4장에서 동적 분석 결과 관리 DB 설계를 설명한다. 마지막으로 5장에서 결론에 대해 기술한다.

2. 관련연구

국내외에서 악성코드 변종이 빠르게 증가하고 동시에 악성코드가 서로 융복합되고 지능적이 되면서 안티바이러스 업체에서 악성코드를 개별 분석하는 것은 현실적으로 어렵게 되었다. (그림 1)은 신규 악성코드 샘플 수집 통계를 보여준다.



(그림 1) 신규 악성코드 샘플 수집 통계

(그림 1)과 같이 신규 악성코드는 최근 몇 년간 빠르게 증가되었으며 2010년에는 매달 150만개 이상의 신규 악성코드가 발생하고 있는 것으로 나타났다[6]. 그리고 최근

† "본 연구는 지식경제부 및 한국산업기술평가관리원의 IT산업원천기술개발사업의 일환으로 수행하였음. [10035427, 지능형 악성코드 자동 분석 및 경유/유포지 탐지 기술 개발]"

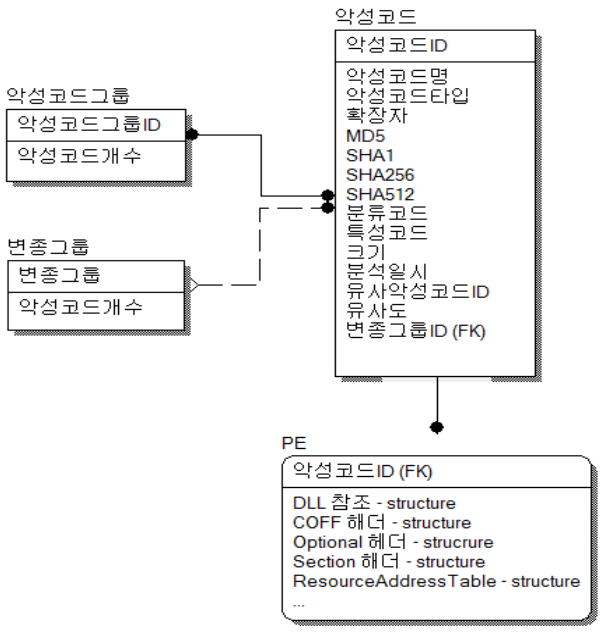
발견되는 많은 수의 악성코드는 기존 악성코드의 변종인 경우가 많고 다양한 악성코드들의 집합체 형태를 가지는 악성코드도 많이 증가하고 있는 추세이다. 이러한 상황에서 악성코드를 자동적으로 분석할 수 있는 기술 연구가 활발히 진행되고 있다[4].

2008년 오스트리아의 Vienna 대학교, 프랑스의 Eurocom사 등으로 구성된 ISEC lab에서 자동화된 행위 분석 시스템을 개발하여 공개하였으나, Conficker, Storm 등과 같은 커널 레벨 실행 은닉 기술을 사용하는 악성코드를 분석하지 못하는 문제점이 있다. 그리고 미국 조지아 공대, 퍼듀 대학교 등에서 자동화된 악성코드 정적 분석 및 행위 분석 기술에 대한 연구를 수행하고 있다. 국내에서는 KISA가 행위 기반 봇넷 자동 분석 기술을 개발하였고 현재 정적/동적 분석이 결합된 악성코드 자동 분석 기술을 개발 중에 있다.

본 논문에서는 현재 가장 많이 사용되는 운영체제인 Windoes를 고려하여 PE 기반 악성코드를 대상으로 한다. PE는 Win32에서 사용하고 있는 파일 구조로 Win32 실행 파일이 적재되는 가상주소, Import 함수 목록, Export 함수 목록, 데이터, 코드 등의 정보를 관리하기 위해 파일의 첫 부분에 여러 가지 구조체 묶음으로 되어 있다. PE 파일 포맷은 COFF(Common Object File Format)라는 포맷을 계승한 파일 포맷으로서 COFF의 확장이다[5].

3. 악성코드 정적 분석 결과 관리 DB 설계

악성코드 정적 분석은 악성코드를 실행하지 않고 코드 자체에 대한 분석을 의미한다. 일반적으로 악성코드의 소스를 얻기는 불가능하기 때문에 바이너리를 디버거(Debugger)나 디스어셈블리(Disassembler)를 이용하여 분석한다[3]. (그림 2)는 악성코드 정적 분석 결과를 저장을 위한 ERD를 보여준다.



(그림 2) 정적 분석 결과 저장 ERD

(그림 2)와 같이 악성코드 정적 분석 결과를 저장하는 테이블은 악성코드 테이블, 악성코드그룹 테이블, 변종그룹 테이블, PE 테이블로 구성된다. <표 1>은 악성코드 테이블 정의서를 보여준다.

<표 1> 악성코드 테이블 정의서

No	속성명	데이터타입	길이	Key
1	악성코드ID	VARCHAR2	10	PK
2	악성코드명	VARCHAR2	20	
3	악성코드타입	NUMBER	1	
4	확장자	CHAR	3	
5	MD5	CHAR	32	
6	SHA1	CHAR	40	
7	SHA256	CHAR	64	
8	SHA512	CHAR	128	
9	분류코드	NUMBER	1	
10	특성코드	RAW	7	
11	크기	NUMBER	10	
12	유사악성코드ID	VARCHAR2	10	
13	유사도	NUMBER	3	
14	분석일시	DATE		
15	변종그룹ID	VARCHAR2	10	FK

<표 1>과 같이 악성코드 테이블은 악성코드 기본 정보, 분류 정보, 변종 정보를 가지고 있다. 악성코드 기본 정보는 PE 파일에 대한 정보로 악성코드명, 악성코드타입, 확장자, 해쉬값(MD5, SHA1, SHA256, SHA512), 크기, 분석일시가 해당된다. 악성코드 분류 정보는 동적 분석 결과로 생성되며 분류코드 속성과 특성코드 속성으로 표현된다. 먼저 분류코드 속성은 자기 복제 여부와 감염 대상 여부에 따라 0(Virus), 1(Worm), 2(Trojan)로 표현하고 특성코드 속성은 악성코드 특성을 7개 비트로 표현하였다. (그림 3)은 7개 악성코드 특성을 보여준다. 단, 자동 분석 대상과 정책에 따라 특성 개수는 변경될 수 있다.

1	2	3	4	5	6	7
다운로더	드롭퍼	스파이웨어	키로거	커뮤니케이터	백도어	루트킷

(그림 3) 악성코드 특성

(그림 3)과 같이 악성코드의 특성은 하나가 아닌 복합적으로 발생할 수 있다. 예를 들어, 악성코드 특성이 다운로더, 키로거이면 1001000 비트 스트링이 저장된다. 마지막으로 악성코드 변종 정보는 코드 및 행위 유사도에 따라 정의되며 유사악성코드ID, 유사도, 변종그룹ID 속성이 해당된다. <표 2>는 변종그룹 테이블 정의서를 보여준다.

<표 2> 변종그룹 테이블 정의서

No	속성명	데이터타입	길이	Key
1	변종그룹ID	VARCHAR2	10	PK
2	악성코드개수	NUMBER	10	

<표 2>와 같이 변종그룹 테이블은 유사성을 갖는 악성코드들의 그룹 정보를 가지고 있다. 변종그룹ID 속성은 악성코드 테이블의 외래키로 설정되어 특정 악성코드가 어떤 변종그룹에 포함되는지 알 수 있고, 악성코드개수 속성으로 변종그룹에 몇 개의 악성코드가 존재하는지 알 수 있다. <표 3>은 악성코드그룹 테이블 정의서를 보여준다.

<표 3> 악성코드그룹 테이블 정의서

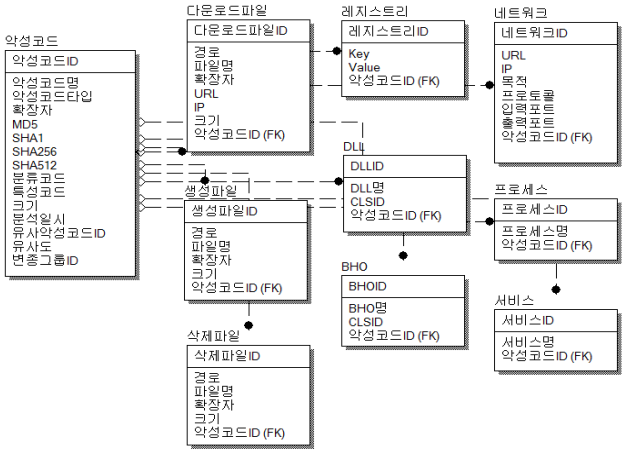
No	속성명	데이터타입	길이	Key
1	악성코드그룹ID	VARCHAR2	10	PK
2	악성코드개수	NUMBER	10	

<표 3>과 같이 악성코드그룹 테이블은 복합 형태의 악성코드들의 그룹 정보를 가지고 있다. 악성코드 그룹은 여러 악성코드들로 구성되고 하나의 악성코드는 여러 악성코드 그룹에 포함될 수 있기 때문에 악성코드 테이블과 악성코드그룹 테이블은 N:M관계를 갖는다. 그리고 악성코드개수 속성으로 악성코드 그룹에 몇 개의 악성코드가 존재하는지 알 수 있다.

본 논문에서 대상으로 하는 악성코드는 PE 형태를 가지고 있기 때문에 PE 테이블을 포함하고 있다. PE 테이블은 PE 파일이 가지고 있는 속성 데이터를 포함하고 있다. 하지만 PE 파일의 속성 데이터는 종류가 매우 다양하고 대게 구조체로 이루어져 있기 때문에 하나의 테이블로는 표현하기 어렵고 악성코드 대응 시그니처 생성에 필요한 PE 파일의 속성 데이터를 정의할 기준이 없기 때문에 본 논문에서는 모든 PE 속성 데이터를 제공하는 것으로 가정한다.

4. 악성코드 동적 분석 결과 관리 DB 설계

악성코드 동적 분석은 악성코드를 실제로 실행하여 악성코드의 행위에 대한 분석을 수행한다[3]. (그림 4)는 악성코드의 동적 분석 결과 관리를 저장하기 위한 ERD를 보여준다.



(그림 4) 동적 분석 결과 저장 ERD

(그림 4)와 같이 악성코드 동적 분석 결과를 저장하는 테이블은 다운로드파일 테이블, 생성파일 테이블, 삭제파일 테이블, 레지스트리 테이블, DLL 테이블, BHO 테이블, 네트워크 테이블, 프로세스 테이블, 서비스 테이블로 구성된다. <표 4>, <표 5>, <표 6>은 각각 다운로드파일, 생성파일, 삭제파일 테이블 정의서를 보여준다.

<표 4> 다운로드파일 테이블 정의서

No	속성명	데이터타입	길이	Key
1	다운로드파일ID	VARCHAR2	10	PK
2	경로	VARCHAR2	100	
3	파일명	VARCHAR2	20	
4	확장자	CHAR	3	
5	URL	VARCHAR2	30	
6	IP	CHAR	15	
7	크기	NUMBER	10	
8	악성코드ID	VARCHAR2	10	FK

<표 5> 생성파일 테이블 정의서

No	속성명	데이터타입	길이	Key
1	생성파일ID	VARCHAR2	10	PK
2	경로	VARCHAR2	100	
3	파일명	VARCHAR2	20	
4	확장자	CHAR	3	
5	크기	NUMBER	10	
6	악성코드ID	VARCHAR2	10	FK

<표 6> 삭제파일 테이블 정의서

No	속성명	데이터타입	길이	Key
1	삭제파일ID	VARCHAR2	10	PK
2	경로	VARCHAR2	100	
3	파일명	VARCHAR2	10	
4	확장자	CHAR	3	
5	크기	NUMBER	10	
6	악성코드ID	VARCHAR2	10	FK

<표 4>, <표 5>, <표 6>과 같이 악성코드로 인해 다운로드, 생성, 삭제되는 파일에 대한 정보를 가지고 있다. 대부분의 악성코드는 악성 행위를 위해 하나 이상의 악성파일을 다운로드하거나 생성하므로 다운로드, 생성, 삭제 파일 행위 내용을 저장한다. <표 7>은 레지스트리 테이블 정의서를 보여준다.

<표 7> 레지스트리 테이블 정의서

No	속성명	데이터타입	길이	Key
1	레지스트리ID	VARCHAR2	10	PK
2	Key	VARCHAR2	100	
3	Value	VARCHAR2	20	
4	악성코드ID	VARCHAR2	10	FK

<표 7>과 같이 레지스트리 테이블은 악성코드로 인해 변경된 레지스트리 정보를 가지고 있다. 대부분 악성코드

가 실행되면 시스템 시작시 악성코드가 자동 실행 또는 로드되도록 하거나 백신 및 방화벽에 의한 탐지 및 회피를 위해 레지스트리 값이 변경되므로 레지스트리 변경 행위 내용을 저장한다. <표 8>과 <표 9>는 순서대로 DLL과 BHO 테이블 정의서를 보여준다.

<표 8> DLL 테이블 정의서

No	속성명	데이터타입	길이	Key
1	DDLID	VARCHAR2	10	PK
2	DDL명	VARCHAR2	10	
3	CLSID	VARCHAR2	30	
4	악성코드ID	VARCHAR2	10	FK

<표 9> BHO 테이블 정의서

No	속성명	데이터타입	길이	Key
1	BHOID	VARCHAR2	10	PK
2	BHO명	VARCHAR2	10	
3	CLSID	VARCHAR2	30	
4	악성코드ID	VARCHAR2	10	FK

<표 8>, <표 9>와 같이 악성코드로 인해 호출되는 DLL, BHO 정보를 가지고 있다. 특정 악성코드는 실행에 필요한 DLL이나 BHO를 호출하므로 DLL, BHO 호출 행위 내용을 저장한다. 악성코드 <표 10>은 네트워크 테이블 정의서를 보여준다.

<표 10> 네트워크 테이블 정의서

No	속성명	데이터타입	길이	Key
1	네트워크ID	VARCHAR2	10	PK
2	URL	VARCHAR2	100	
3	IP	CHAR	15	
4	목적	NUMBER	1	
5	프로토콜	NUMBER	1	
6	입력포트	NUMBER	4	
7	출력포트	NUMBER	4	
8	악성코드ID	VARCHAR2	10	FK

<표 10>과 같이 네트워크 테이블은 악성코드로 인해 수행된 네트워크 정보를 가지고 있다. 최근 대부분의 악성코드는 사용자 시스템을 감염시킨 후에 네트워크를 통해 공격자에 연결하는 기능을 가지므로 네트워크 행위 내용을 저장한다. <표 11>과 <표 12>는 순서대로 프로세스와 서비스 테이블 정의서를 보여준다.

<표 11> 프로세스 테이블 정의서

No	속성명	데이터타입	길이	Key
1	프로세스ID	VARCHAR2	10	PK
2	프로세스명	VARCHAR2	20	
3	악성코드ID	VARCHAR2	10	FK

<표 12> 서비스 테이블 정의서

No	속성명	데이터타입	길이	Key
1	서비스ID	VARCHAR2	10	PK
2	서비스명	VARCHAR2	20	
3	악성코드ID	VARCHAR2	10	FK

<표 11>, <표 12>와 같이 악성코드로 인해 생성되는 프로세스, 서비스 정보를 가지고 있다. 악성코드도 실행되는 프로그램 중 하나이기 때문에 실행시 하나의 프로세스로 나타나고 특정 서비스를 호출할 수 있다. 따라서 프로세스, 서비스 생성 행위 내용을 저장한다. 그밖에 시스템 변수 변경, 브라우저 변경, 메모리 접근 등 악성코드 자동 분석으로 얻어지는 행위 정보가 존재한다.

5. 결론

본 논문에서는 PE 기반의 악성코드 자동 분석 결과를 관리할 수 있는 DB 설계에 대해 제안하였다. 악성코드 자동 분석 결과는 정적 분석과 동적 분석을 통해 생성되며 이들 데이터를 각각 저장할 수 있는 테이블을 설계하였다. DB에 저장된 데이터는 안티바이러스 업체나 기관에서 악성코드 대응 시그니처를 생성할 수 있는 악성코드 DNA 요소로서 활용될 수 있을 것으로 기대된다.

본 연구에서는 악성코드 자동 분석 결과 중 기본적인 데이터에 국한된 DB 설계를 보이고 있다. 따라서 보다 세부적인 분석 결과와 실제 안티바이러스 업계나 기관에서 요구하는 속성들에 대한 조사가 반영된 세부 DB 설계가 필요하다.

참고문헌

[1] 서희석, 최중섭, 주필환 “윈도우 악성코드 분류 방법론의 설계,” 정보보안학회논문지, 2009, pp.83-92.

[2] 최준호, 곽효승, 공현장, 김판구, 이병권, 오은숙, “악성코드 분류 및 명명법에 관한 연구,” 정보과학회지, 제20권, 제11호, 2002, pp.24-29.

[3] 인터넷침해사고대응지원센터(KISC), “관리자를 위한 악성프로그램(Malware) 분석방법,” 2004.

[4] 오주형, 임채태, 정현철, “커널 콜백 매커니즘을 이용한 Win32 실행파일 타입의 악성코드 자동 분석,” 한국인터넷정보학회, 2010, pp443-446.

[5] Microsoft Corporation, “Microsoft Portable Executable and Common Object File Format Specification,” 2008.

[6] Trend Micro, “Trend Micro Endpoint Comparative Report Performed by AV-Test.org,” 2010.