특집 04

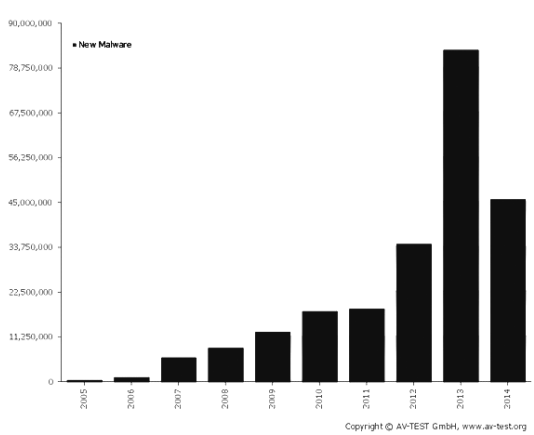
Dependable 시스템을 위한 악성코드 자동 분석 시스템 기술 동향 및 발전 방향

윤종희 (영남대학교)

목 차 »

1. 서 론

최근 해킹 기법과 컴퓨터 네트워크의 급속한 발전 등으로 인해 (그림 1)과 같이 다양한 악성코드 및 그 변종들에 의한 위협이 많아지고, 그 수 또한 기하급수적으로 증가하고 있다[1]. 이에 따라, 빠르고 정확하게 악성코드를 식별하여 차단하고 전파되는 것을 방지하는 것이 매우 중요하고, 이를 위해서는 효과적이고 빠른 악성코드 자동 분석 기법에 대한 필요성이 점점 더 많아지고 있다. 왜냐하면, 발견되는 악성코드 수의 폭증으로 인해 악성코드 분석전문가를 통한 수동분석으로는 수집된 악성코드를 모두 분석하는 것은 거의 불가능하기 때문이다. 그래서 본 논문에서는 현재 연구개발 중이거나 상품화가 완료된 악성코드 자동 분석시스템에 대한 비교 분석 및 연구동향을 알아보도록 하겠다.



(그림 1) 최근 10년간 악성코드 발생 빈도

2. 악성코드 종류 및 분석 기법 소개

이번 장에서는 간단하게 악성코드의 종류와 이러한 악성코드를 분석하는 기법에 대해서 소개하도록 하겠다. 악성코드의 종류는 전파 방법, 악성행위 등에 의해서 분류된다.

2.1 악성코드의 종류

악성코드는 악성행위와 증식, 전파 방법 등에

따라 아래와 같이 구분 할 수 있다.

2.1.1 웜(Worm)

인터넷과 같은 네트워크 환경을 기반으로 자신을 복제해서 전파할 수 있는 프로그램을 뜻한다[2]. 대부분 특별하게 컴퓨터에 영향을 주거나 피해를 주지 않고, 자기증식에만 집중해서 위험하지 않을 경우도 있지만, 기하급수적인 자기증식을 통해서 컴퓨터 시스템이나 네트워크 장비에 과부하를 걸어 시스템의 성능을 현격하게 떨어트리거나 DDOS(Distributed Denial of Service)와 같은 공격의 수단으로 사용될 수 있다.

2.1.2 바이러스(Virus)

다른 프로그램의 파일 그 자체나 메모리영역, 부트영역 또는 심지어 운영체제 자체에 자신의 프로그램을 추가하는 특징을 가지며, 이렇게 파일과 파일을 통해서 전파되어 다른 시스템을 감염시켜 나간다[3]. 대부분의 바이러스는 단독으로 실행되지 못하고, 실제 동작하는 프로그램에 자신의 코드를 적절하게 추가해 그 프로그램이 실행될 때 자신도 실행되어 복제, 시스템 파괴 등의 원하는 행위를 수행하게 된다.

2.1.3 루트킷(Rootkit)

커널 모듈, 디바이스 드라이버, 시스템 라이브러리 등에 침투 동작하는 바이러스 프로그램들을 뜻하며[4], 유저 모드 레벨보다 아래에서 동작하기 때문에 자신의 파일, 프로세서 자체를 숨기고 네트워크 연결등도 보이지 않게 해 사용자가 감염여부를 알 수 없는 것이 특징이다.

2.1.4 트로이목마(Trojan Horse)

보통 유용한 유틸리티와 같은 프로그램으로 자신을 포장하여 전혀 악성행위와 관련 없이 보이는 프로그램처럼 자신을 속이지만, 설치되고 난 이후에 다른 악성 프로그램들을 생성하거나 다운로드해서 시스템의 정보를 유출하거나 파괴하는 프로그램을 말한다[5].

2.1.5 스파이웨어(Spyware)

컴퓨터 시스템에 몰래 침투하여 중요한 비밀정보나 개인정보 등을 공격자에게 유출하는 프로그램을 말한다. 보통 신용카드 정보, 인터넷 계정정보, 주민등록번호, 은행계좌와 같은 금융정보 등을 노린다.

2.1.6 봇(Bot)

인터넷 상의 특정 마스터 프로그램으로부터 원격으로 조정 받는 프로그램을 말하며 일반적으로 이러한 봇과 마스터들이 네트워크상에서 연결되어 있는 것을 봇넷(Botnet)이라한다. 이러한 봇은 마스터 프로그램(보통 C&C 서버라 불림)에 의해서 시스템의 정보 등을 유출하거나 대량 스팸메일 발송, DDoS 공격 등에서 많이 활용되지 때문에 특히 위험하다고 볼 수 있다[6].

2.1.7 드로퍼(Dropper)

사용자가 모르는 사이에 바이러스, 스파이웨어, 봇 등을 컴퓨터 시스템에 설치하는 프로그램이다. 인터넷을 통해서 특정 악성 프로그램을 다운로드 할 수도 있고 자신의 프로그램 속에 미리 인코딩 해 놓은 악성 프로그램을 설치 할 수도 있어 악성코드 전파를 위한 용도로 많이 사용되는 기법이다.

2.2 악성코드 분석 기법

악성코드 분석 기법은 크게 프로그램을 실행하면서 분석하는 동적 분석 기법과 프로그램 자체

를 분석하는 정적 분석 기법으로 나눌 수 있다.

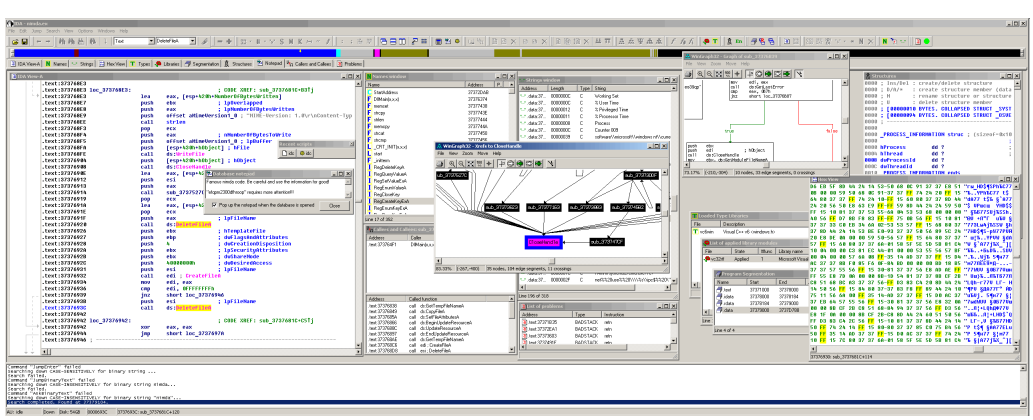
2.2.1 정적 분석

프로그램을 실행하지 않고 분석하는 것을 의미하며, 보통 소스코드가 주어진 경우와 바이너리 파일만 주어진 경우를 구분해 분석 기법을 적용할 수 있다. 대부분의 악성코드는 바이너리, 즉 실행 파일만이 존재하기 때문에 악성코드 분석에서의 정적 분석은 대체로 바이너리 파일 분석을 의미 한다. 안타깝게도 대부분의 기존 정적 분석 기법은 소스 코드를 분석하는 툴들을 연구개발한 상태이지만, 바이너리에 대한 툴은 그렇게 다양하지 않다. 가장 대표적으로는 디어셈블러를 통해 바이너리 파일을 어셈블리로 변환하여 분석하고 다시 이 어셈블리를 디컴파일해서 소스코드로 변환 한 후 분석하는 방법이다. 대부분의 악성코드들이 IA32 명령어 셋을 가진 인텔 CPU와 윈도우 시스템 상에서 동작하는 프로그램으로 작성되어 있는데, IA32 명령어 셋 자체가 가변 길이를 가지고 있기 때문에 디어셈블을 시도할 때 실패하거나 전혀 다른 코드로 나타날 수 있고, 이러한 문제점을 악성코드 제작자도 미리 알고 더욱 더 디어셈블이 불가능하도록 코드를 작성하는 경우도 있다. 또한, 어셈블리에서 소스코드로 변환하는 디컴파일의 경우에도 많은 어려움이 있어, 대체로 완벽한 소스코드를 복원하는 것은 불가능하다. 대표적은 도구로는 IDA pro[7] 등이 있다.

2.2.2 동적 분석

정적분석과 다르게 동적 분석은 프로그램을 실제로 실행시켜서 그 행위를 관찰하여 특징을 추출하여 악성 여부를 판단하는 방법이다. 동적분석은 어떤 실행 정보들에 대해서 관심을 가지는가에 따라 그 종류를 나눌 수 있다.

가장 대표적인 방법은 프로그램에서 호출되는 함수를 트래킹하는 방법이다. 즉, 함수 호출이 있을 때마다 이 정보를 기록하고 나중에 이런 정보들을 모아서 분석하는 기법으로 단순히 호출되는 함수만을 트래킹 할 수 있지만, 함수 호출에 사용되는 인자(Arguments)와 리턴 값까지도 정보를 모으는 경우도 많다. 이렇게 함수 호출에 관심을 가지는 이유는 함수가 프로그램에서 특정 기능 하나 하나를 모아놓은 것으로 볼 수 있고, 또한 라이브러리 함수의 경우 이미 그 역할이 다 공개되어 있어서 프로그램의 의도 파악이 용의하기 때문이다. 이런 함수 호출 정보를 뽑아내는 주요



(그림 2) IDA Pro

한 방법은 함수 hooking 기법, DLL injection 기법 등이 있다. 이런 기법들을 분석가가 직접 코드를 작성하여 동적 분석을 시도 할 수 있지만, 대부분 많은 기술적인 요소와 시간이 요구 된다. 그래서 실제로는 자동 동적 분석 시스템을 활용하는 경우가 많다. 다음 장에서는 현재 사용되거나 연구개방중인 자동 분석 시스템에 대해서 알아보기로 하겠다.

3. 자동 동적 악성코드 분석 연구 동향

현재 많은 자동 동적 악성 코드 분석 시스템들이 개발되어 공개되어 있다. 이들 중 대부분이 가상머신을 기반으로 동작하고 있다. 왜냐하면, 악성코드를 동적으로 분석하기 위해서는 실행을 시켜야 하는데 일반 시스템에서 실행할 경우 분석 시스템 자체가 오염될 가능성이 높기 때문이고, 또한 가상 시스템을 사용할 경우 메모리 분석, 네트워크 분석 등에 더욱 더 용이하기 때문이다. 본 장에서는 가장 대표적은 악성코드 분석 시스템 3가지에 대해서 간단하게 소개하도록 하겠다.

3.1 CW Sandbox[8]

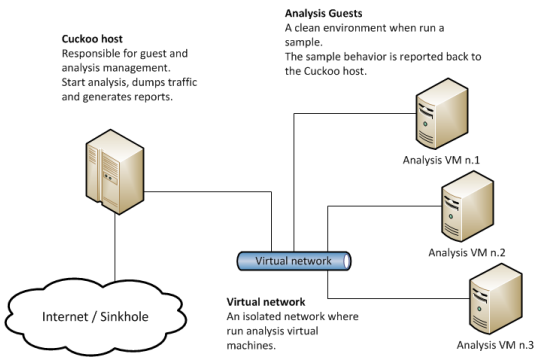
가상 시스템상의 윈도우 환경이나 실제 컴퓨터 환경에서 악성코드 샘플을 실행하며, 기본적으로 API 후킹(hooking)기법과 DLL injection 기법을 사용해서 네트워크 통신, 파일 및 레지스트리 접근, API 함수 호출 정보와 같은 악성코드의 행위를 모니터링 한다. API 후킹은 샘플이 메모리에 로딩 될 때 이루어지며, 메모리에 로딩 된 모든 라이브러리들의 exported API 함수의 리스트를 보고 적절한 후킹을 시도하게 된다. 그 후 분석용 프로그램을 DLL injection 기법으로 적용하여, 앞서 말한 정보들을 모니터링하고 이외에 system call 정보등도 모니터링하여 악성행위를 분석한다. 최종적으로 분석 보고서가 만들어지고 이 보고서를 기반으로 악성코드 분석가가 악성 여부를 판단 할 수 있게 된다. 현재는 ThreatAnalyzer라는 이름으로 상용화되어 판매되고 있다[9].

3.2 Anubis

UC Santa Barbara의 연구팀에서 개발한 TTAnalyze[10]에 기반 한 악성코드 분석 시스템으로, Qemu상의 Windows XP에서 샘플 파일을 실행하고, 윈도우 API 함수 호출 및 시스템 서비스 호출을 모니터링한다. 이때 함수들에게 전달되는 인자값 들도 모두 기록한다. CPU 에뮬레이션 기법을 사용하기 때문에 가상머신 기반 기법에 비해서 모든 CPU 명령어들을 추적/분석할 수 있는 장점을 가지고 있지만, 그 대신 느리다는 단점도 가지고 있다. 동적 data flow analysis를 지원해서 오염된(tainted) 입력값이나 메모리주소로부터 발생되는 결과값들도 오염된 것으로 처리해 악성행위 분석을 용의하게 한다. 현재 웹페이지를 통해서 직접 윈도우 실행파일을 업로드해서 분석 결과를 확인할 수 있는 환경을 제공하고 있다.

3.3 Cuckoo Sandbox

Cuckoo sandbox[11]는 현재 가장 활발하게 개발되고 있는 오픈소스 프로젝트 형태의 악성코드 자동분석 시스템이다. (그림 3)은 Cuckoo sandbox의 구성도를 보여준다. 서버(host)와 가상머신상의 클라이언트(guests)들로 구성되어 있고, 가상 네트워크 인터페이스를 통해서 서로 연결된 구조이다. 사용자가 악성코드로 의심되는 샘플을 서버에 전달하면, 서버가 현재 사용가능한 가상머신상의 클라이언트를 선택해 샘플을 전송하고, 가



(그림 3) Cuckoo Sandbox 구성도

상머신 상에서 실행시켜서 다양한 동적 행위 정보들을 추출해서 사용자에서 보여주는 방식으로 구성되어 있다. 오픈소스 정책으로 개발되고 있기 때문에 다양한 변형이 가능하고 그 분석 기능 발전도 다른 툴들에 비해서 빠른 편이라 주목할 만하다.

4. 결 론

본 글에서는 현재 기하급수적으로 증가하고 있는 악성코드들의 행위를 분석하는 방법과 그것을 자동으로 하는 시스템에 대해서 소개하였다. 물론 소개된 것은 전체 분석 시스템 중에 극히 일부이지만, 최근 10여 년 동안 이 부분에 대한 많은 연구와 개발이 진행되고 있음을 알 수 있다. 특히, 공개로 개발되고 있는 Cuckoo Sandbox의 경우 다른 상용 프로그램들에 비해서 쉽게 사용할 수 있고, 그 기능도 변경할 수 있어 아주 유용하다고 볼 수 있겠다. 물론 이렇게 공개적인 시스템은 악성 코드 제작자들도 손쉽게 사용할 수 있어, 자신들의 악성코드를 테스트하고, 분석 시스템을 회피하는 기능을 추가하여 무력화 될 수 있는 가능성이 높지만, 악성코드 분석가 또는 악성코드 분석 시스템 개발자에게는 더 없이 좋은 자료라 볼 수 있겠다. 마지막으로 이런 악성코드 분석 시스템들의 결과 리포트를 활용하는 다양한 연구들도 추가로 진행 될 수 있을 것이다. 즉, 자동 분석 시스템에서 뽑아낸 각 악성 코드들의 특징들을 가지고 악성 코드 분류(classification)을 하거나, 악성 코드 제작자(또는 제작 그룹)을 특정 짓는 작업등의 많은 응용에 사용될 수 있을 것이다. 앞으로 악성 코드의 수가 더욱 더 많아지고 지능화되어 갈 것이 분명하기 때문에 이에 대응하는 자동 악성코드 분석시스템에 대한 연구 개발도 더욱 더 활발해 질 것으로 기대된다.

참 고 문 헌

[1] AV-TEST, http://www.av-test.org/en/statistics/malware/
[2] Barwise, Mike. "What is an internet worm?". BBC. Retrieved 9 September 2010.
[3] Aycock, John (2006). Computer Viruses and Malware. Springer. p. 14. ISBN 978-0-387-30236-2.
[4] "Rootkits, Part 1 of 3: The Growing Threat". McAfee. 2006-04-17.
[5] "What is the difference between viruses, worms, and Trojans?". Symantec Corporation. Retrieved 2009-01-10.
[6] Ramneek, Puri (2003-08-08). "Bots &; Botnet: An Overview". SANS Institute. Retrieved 12 November 2013
[7] IDA, https://www.hex-rays.com/products/ida/
[8] Willems, C., Holz, T., and Freiling, F., Toward Automated Dynamic Malware Analysis Using CWSandbox. IEEE Security and Privacy 5, 2 (Mar. 2007), 32-39.
[9] http://www.threattracksecurity.com/enterprise-security/sandbox-software.aspx
[10] Christopher Kruegel and Engin Kirda and Ulrich Bayer, "TTAnalyze: A Tool for Analyzing Malware," in *Proceedings of the 15th European Institute for Computer Antivirus*

Research (EICAR 2006) Annual Conference, 2006.
[11] cuckoo, http://www.cuckoosandbox.org

저 자 약 력

윤 종 희

이메일 : youn@yu.ac.kr

- 2003년 경북대학교 전자전기공학부(학사)
- 2011년 서울대학교 전기컴퓨터공학부(박사)
- 2011년~2012년 강릉원주대학교 컴퓨터공학과 강의전담교수
- 2012년~2013년 한국전자통신연구원 부설연구소 연구원
- 2013년~현재 영남대학교 컴퓨터공학과 조교수
- 관심분야 : 사이버 보안, 컴파일러, 소프트웨어 최적화, 임베디드 시스템