실전 악성코드 개발 및 분석 방법

김경민

동의대학교

Practical Malware Development And Analysis Method

Kyung-min-Kim

Dong-Eui University

E-mail : km9311@naver.com

요 약

1986년에 최초의 악성코드인 브레인 바이러스가 발견된 이후 현재까지 다양한 종류의 악성코드가 만들어졌고 대표적으로 웜, 드로퍼, 트로젠, 백도어, 루트킷, 다운로더, 스파이웨어 등이 있다. 특히 최근에는 드라이버 형식의 악성코드가 나타남에 따라 악성코드 분석이 더욱 어려워졌다. 따라서 악성코드 분석가는 상당한 실력이 요구된다. 악성코드를 잘 분석하기 위해서는 동작 원리를 알아야 하고 이는 직접 개발을 해봐야 한다. 본 논문에서는 드로퍼, 백도어, 트로젠, 루트킷, 드라이버를 실전에 유포되는 악성코드와 유사하게 개발하고 가상 환경을 구축한 시스템에서 실행 동작을 보인다. 그리고 정적 분석과 동적 분석으로 악성코드를 빠르고 효과적으로 분석하기 위한 방법을 제안한다.

ABSTRACT

After the first malware, the brain virus, was founded in 1986, various types of malwares have been created Including worm, dropper, trojan, backdoor, rootkit and downloader. Especially in recent years, driver-type malware have made then more difficult to analyze. therefore, malware analyst require competitive skills. To analyze malware well, you need to know how it works and have to do it by yourself. So in this paper, we develop the dropper, backdoor, trojan, rootkit and driver similar to malware distributed in the real world. It shows the execution behavior on the virtual environment system We propose a method to analyze malware quickly and effectively with static analysis and dynamic analysis.

키워드

악성코드, 드라이버, 정적 분석, 동적 분석

Ⅰ. 서 론

개인 PC의 증가와 인터넷의 보급에 따라 다양한 소프트웨어가 개발됐고 동시에 악성코드의 수도 증가했다. 네트워크로 다른 컴퓨터에 악성코드를 전파하는 웜, 컴퓨터에 침투하여 사용자를 조종하는 트로젠, 시스템의 정보를 원격지의 특정한 서버에 주기적으로 보내는 스파이웨어, 공격자가 사용자의 인증 과정 등 정상적인 절차를 거치지 않고 침입하는 백도어, 악성코드를 담고 있는 드로퍼 등 다양한 악성코드가 있다. 이러한 악성코드들은 단순히 수만 늘어난 게 아니라 기술이 더 고도화됨에 따라 분석하기 매우 어려워졌다. 특히 공격자의 흔적을 지우는 악성코드인 루트킷이 드라이버 파일로 유포되면서 악성코드 분석가는 유저 메모리 뿐만 아니라 커널 메모리도 분석하기 시작했다. 이렇게 악성코드의 기술이 높아지고 개수가 많아짐에 따라 전통적인 방식인 정적 분석과 동적 분석만으로 분석이 힘들어졌다. 그래서 최근에는 악성코드 자동 분석 시스템인 쿠쿠박스나 메모리 포렌식 도구인 볼라틸리티와 같은 프로그램으로 악성코드를 분석한다.

본 논문에서는 실전에 유포되는 악성코드와 유사하게 직접 개발하고 이를 분석해본다. 분석 방법으로는 정적 분석과 동적 분석으로 전반적인 동작을 분석하고 더불어 볼라틸리티를 이용해 드라이버 파일, 의도적으로 숨긴 프로세스 등 고도화된 악성코드를 효과적으로 분석하는 방법을 제안한다.

II. 관련 연구

2.1 드로퍼

드로퍼는 파일 내에 악성코드를 숨겨 시스템에 떨어뜨리는 악성코드다.

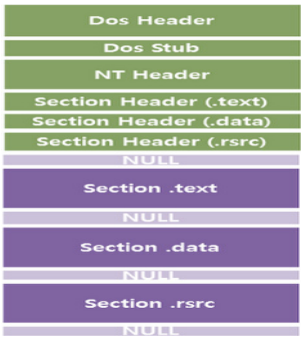

그림 1. PE 구조 [1]

윈도우의 모든 파일은 그림1와 같이 PE 구조로 되어있다. 리소스 섹션(.rsrc)에는 일반적으로 마우스 커서, 아이콘, 소리와 같은 그래픽 기반의 자원이 저장된다. 그러나 드로퍼는 리소스 섹션에 악성코드를 숨긴다. 드로퍼가 실행되면 리소스 섹션에 있는 악성코드를 꺼내 시스템에 떨어뜨린다.

2.2 백도어

백도어는 공격자가 부분 인증이나 무인증으로 컴퓨터에 접속해 쉘의 권한을 얻는 악성코드다. 쉘 권한을 얻으려는 방법으로 Bind Shell과 Reverse Shell이 있다. Bind Shell은 피해자의 서버에 포트를 열어놓고 공격자가 열린 포트를 이용해 침투하는 방법이다. 그러나 방화벽이 설치된 환경인 경우, 특정 포트 이외의 접근은 모두 차단되기 때문에 Bind Shell은 대부분의 PC 환경에서 실패한다. 이러한 문제점을 극복한 방법이 Reverse Shell이다. 이것은 공격자의 서버에 포트를 열어놓고 피해자의 서버가 공격자 서버로 접속하도록 유도하는 방법이다. Reverse Shell의 장점은 피해자의 IP를 몰라도 되고 악성코드가 실행되기만 하면 자동으로 피해자 서버에 침투한다.

2.3 루트킷

루트킷은 공격자의 흔적을 지우거나 숨기는 악성코드다. 대표적으로 스텔스 프로세스가 있는데 이는 특정 프로세스를 숨기는 기법이다. 프로세스가 할당되면 커널에서 EPROCESS 구조체가 생성된다. EPROCESS 구조체는 해당 프로세스의 정보를 저장하고, 이 중에서 Flink 필드와 Blink 필드를 이용해 프로세스끼리 더블 링크드 리스트 방식으로 가리킨다.

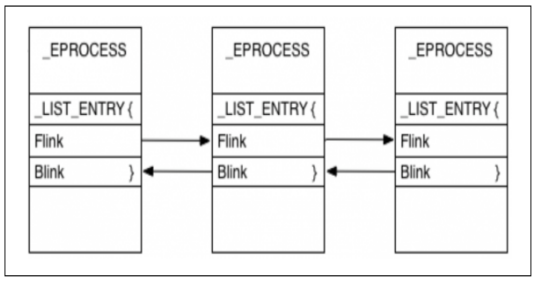

그림 2. EPROCESS의 Flink, Blink 필드 [2]

그림2에서 프로세스는 더블 링크드 리스트로 서로 가리키고 있다. 특정한 프로세스를 숨기고 싶으면 Flink와 Blink를 끊으면 된다.

2.4 볼라틸리티[3]

볼라틸리티는 메모리 분석에 사용되는 오픈소스 소프트웨어다. 악성코드가 실행된 메모리를 덤프하여 파일로 저장하고 이 파일을 볼라틸리티가 분석한다. 최근 악성코드는 자신의 흔적을 지우기 때문에 분석하기 어렵지만, 메모리에는 반드시 흔적이 남는다. 그래서 지워진 흔적을 볼라틸리티가 효과적으로 분석할 수 있다.

III. 악성코드 실행

본 장에서는 악성코드 실행 과정을 보인다. 먼저 실행 환경은 표1와 같이 구성했다.

3.1 실행 환경

표 1. 실행 환경

구분	내용	
OS	공격자	Kali Linux 64it
	피해자	Windows7 32bit
RAM	공격자	4G
	피해자	2G
IP	공격자	192.168.56.102
	피해자	192.168.56.101
Virtual Machine	virtualbox 5.122	
Malware	dropper.exe rootkit.sys spyware.exe backdoor.exe	

3.2 동작 과정

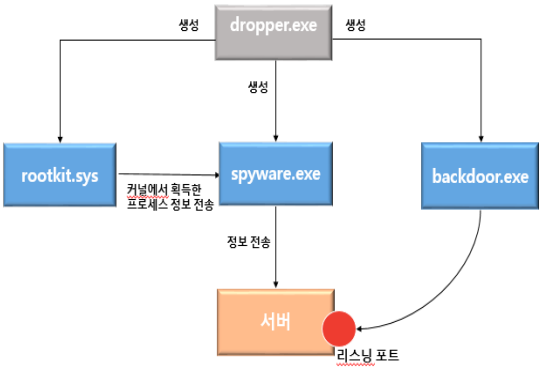

그림 3. 동작 과정

악성코드의 전반적인 동작은 그림3와 같다. dropper.exe을 실행하면 리소스 섹션에서 rootkit.sys, spyware.exe, backdoor.exe를 꺼내와 시스템 내부에 떨어뜨린다. rootkit.sys의 역할은 두 가지가 있다. 첫 번째는 프로세스를 계속 감시하다가 생성 또는 종료되면 해당 프로세스의 정보를 spyware.exe에게 보낸다. 두 번째는 spyware.exe를 스텔스 프로세스로 만든다. 그러면 사용자는 spyware.exe가 실행 중이더라도 알아채지 못한다. spyware.exe는 rootkit.sys으로부터 받은 프로세스 정보를 공격자 보내는 역할이다. backdoor.exe는 피해자의 로컬 시스템 쉘 권한을 공격자가 얻도록 한다.

3.3 dropper.exe

그림 4. dropper.exe

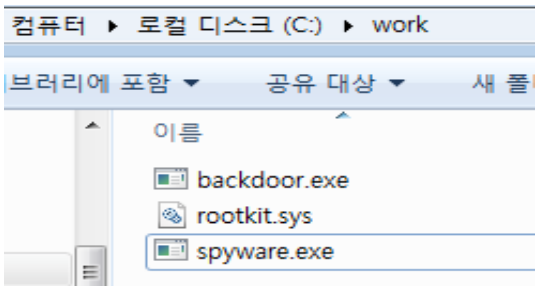

그림 5. c:\work 폴더에 파일들이 생성

그림4와 그림5에서 dropper.exe를 실행하면 c:\work 폴더에 backdoor.exe, rootkit.sys, spyware.exe가 생성된다.

3.4 rootkit.sys

rootkit.sys은 spyware.exe를 스텔스 프로세스로 만든다.

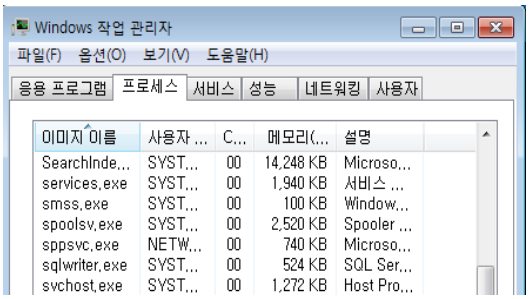

그림 6. 작업 관리자

그림 6에서 spyware.exe가 실행 중이지만 작업 관리자에서 보이지 않는다.

3.5 spyware.exe

그림 7에서 프로세스가 생성/종료 될 때마다 spyware.exe는 공격자에게 해당 프로세스의 정보를 보낸다.

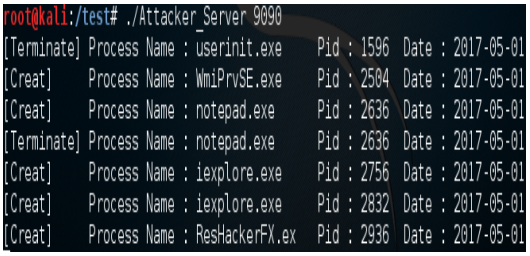

그림 7. 공격자가 프로세스 정보 획득

3.6 backdoor.exe

공격자의 리스닝 포트(4444 port)를 열어 놓고 backdoor.exe을 실행하면 피해자는 열린 포트로 공격자에게 접속을 한다. 필자는 모의 해킹 툴인 메타스플로잇을 이용하여 리스닝 포트를 열었다.

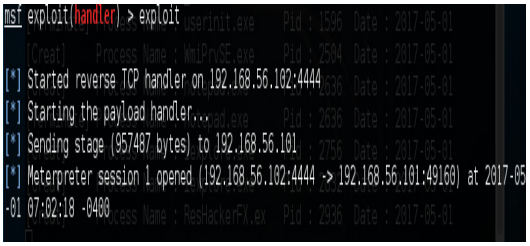

그림 8. 메타스플로잇을 이용한 리스닝 포트

그림 8에서 공격자의 리스닝 포트로 피해자가 접속해 meterpreter 쉘을 획득했다. meterpreter 쉘은 메타스플로잇이 제공하는 해킹 쉘이다.

Ⅳ. 악성코드 분석

4.1 볼라틸리티 분석

볼라틸리티의 다양한 플러그인을 이용하여 숨긴 프로세스나, 드라이버를 쉽게 분석할 수 있다.

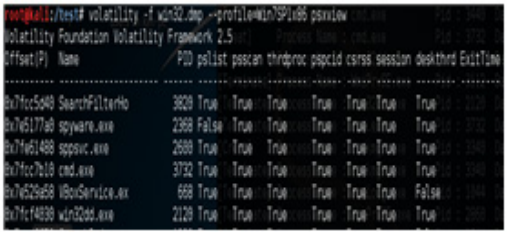

그림 9. psxview 플러그인

그림9는 psxview 플러그인으로 찾아낸 프로세스 목록이다. 유독 spyware.exe만 pslist가 False다. 이것은 프로세스는 동작 중이지만 EPROCESS 구조체의 Flink와 Blink가 끊어졌다는 뜻이다. 즉 공격자가 spyware.exe를 고의로 숨겼다.

다음은 modules 플러그인을 이용해 커널 드라이버의 목록을 살펴봤다.

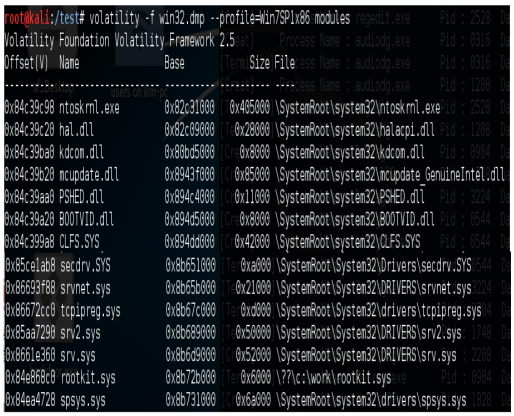

그림 10. modules 플러그인

그림10에서 대부분 파일이 \SysRoot\System32 경로로 시작하지만 유독 rootkit.sys만 \??로 시작한다. 이는 시스템이 아니라 사용자가 사용한 드라이버이다. 메모리 덤프를 뜰 때 악성코드를 제외한 다른 프로그램은 실행하지 않았다. 그러므로 악성코드가 rootkit.sys를 사용했음을 알 수 있다.

Ⅴ. 결 론

본 논문에서는 대표적인 악성코드 종류인 드로퍼, 백도어, 드라이버, 스파이웨어의 개발과 실행 동작을 보였고 볼라틸리티를 이용해 분석하기 어려운 악성코드를 효과적으로 분석하는 방법을 제안했다. 하지만 악성코드는 본 논문에서 보여준 종류 말고도 엄청나게 다양하며 지금도 수백 개의 악성코드가 새롭게 생성된다. 기업의 입장에서 매일 수백 개의 악성코드를 자세히 분석하기란 불가능한 일이다. 따라서 쿠쿠 샌드박스와 같은 악성코드 자동 분석 시스템으로 대량의 악성코드를 먼저 분석 후에 그중에서 의심스러운 악성코드를 자세히 분석하는 순서로 진행해야 한다.

참고문헌

[1] http://blog.kimtae.xyz/8
[2] https://www.alienvault.com/blogs/labs-research/malware-hiding-techniques-to-watch-for-alienvault-labs
[3] Volatility, http://www.volatilityfoundation.org
[4] 이승원 지음, 리버싱 핵심 원리, 2012