악성코드의 Anti-VM 기법 방지를 위한 방안 연구

최석준*, 김득훈*, 곽진**
*아주대학교 컴퓨터공학과 정보보호응용및보증연구실
**아주대학교 사이버보안학과
e-mail:sjchoi.isaa@gmail.com*, dhkim.isaa@gmail.com*, security@ajou.ac.kr**

A study on the Prevention of Malware Anti-VM Technique

Suk-June Choi*, Deuk-Hun Kim*, Jin Kwak**
*ISAA Lab., Department of Computer Engineering, Ajou University
**Department of Cyber Security, Ajou University

요 약

악성코드 분석을 실제머신에서 진행하는 도중 시스템이 손상될 경우 복원에 어려움이 있다. 이에 따라 역공학 분석가들은 일반적으로 가상환경에서 분석을 진행한다. 가상환경의 경우 실제머신 호스트 운영체제와는 독립되어 있고 악성코드를 발현하여 시스템이 손상이 되더라도 이전 스냅 샷으로 복원하는 것이 용이하기 때문에 가상환경에서 악성행위 동작을 분석한다. 하지만, 최근 들어 악성코드가 점점 지능화되고 있으며 이에 따라 악성코드가 VM(Virtual Machine)환경에서는 동작하지 않는 Anti-VM 기술이 적용된 악성코드가 증가하고 있다. 따라서, 본 논문에서는 가상환경에서 Anti-VM 기술이 적용된 악성코드를 분석할 수 있는 위한 환경 연구를 진행한다. 이에 따라, 악성코드가 가상환경을 탐지하는 기법을 분석하고 각 탐지기법을 방지할 수 있는 방안을 제안한다. 이를 통해 동적분석을 하지 못하도록 하는 분석 방해·지연 기술 중 하나인 Anti-VM 기술이 적용된 악성코드의 분석 결과를 향상 시키는 것을 목표로 한다.

1. 서론

최근 신·변종 악성코드가 점점 지능화됨에 따라 악성코드는 자신을 보호하기 위해 분석 방해·지연 기술을 적용한 악성코드가 증가하고 있다. 또한 악성코드로 인한 개인정보 유출, DDoS 공격, 대규모 스팸 발송 등과 같은 침해사례가 증가하고 있어 신속한 대응이 필요하다[1]. 이러한 신·변종 악성코드에 대응하기 위해선 문제가 되는 악성코드 분석 방법론에 따른 분석이 필요하다.

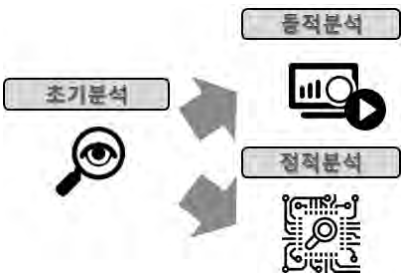
현재 악성코드는 자신을 보호하기 위해 실행 압축, Anti-Debugging, Anti-VM 등과 같은 기술이 적용되어 악성행위 패턴의 분석을 방해·지연하기 때문에 악성코드에 적용된 보호기법을 방지하는 방안 연구가 시급하다[2]. 본 논문에서는 악성코드의 동적분석을 통해 시스템이 손상 되더라도, 실제머신에 큰 영향을 미치지 않는 가상환경에서 분석을 할 수 있는 방안을 연구한다. 이를 위해선 Anti-VM 기술이 적용된 악성코드가 가상환경을 탐지할 수 없도록 가상환경의 특징들을 변경, 은닉 등을 하는 방안을 연구한다.

본 논문의 구성은 다음과 같다. 2장에서는 악성코드 분석 방법론과 악성코드에 적용된 보호기법 중 하나인 Anti-VM 기술을 설명하고, 악성코드의 가상머신 탐지기법을 분석한다. 3장에서는 악성코드가 가상머신을 탐지할 수 없도록 Anti-VM 기술 방지방안을 제안하여, Anti-VM 기술이 적용된 악성코드를 분석할 수 있는 방안을 연구한다. 마지막으로, 4장에서 결론을 맺는다.

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2014R1A2A1A11050818).

2. 관련연구

2.1 악성코드 분석 방법론

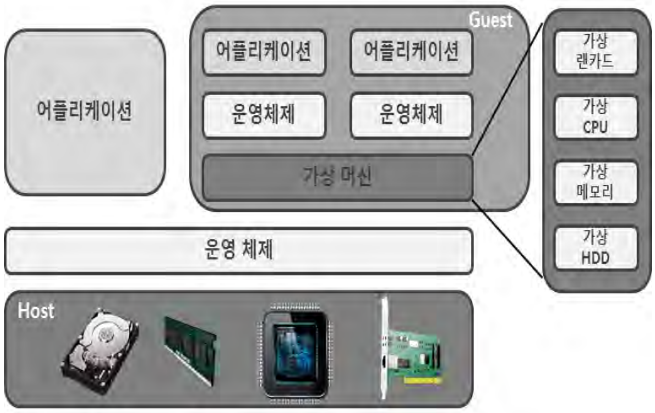


(그림 1) 악성코드 분석 절차

악성코드를 분석하기 위한 방법으로는 (그림 1)과 같이 초기분석, 동적분석, 정적분석 단계로 총 3 단계의 절차를

거쳐 분석을 진행한다[3]. 초기분석의 경우 수집한 악성코드를 초기에 분석하는 것을 말하며, 악성코드를 동작하여 분석하는 것이 아닌 외형을 보고 분석하는 단계를 말한다. 동적분석은 악성코드를 분석환경에서 동작하여 끝날 때까지 프로세스, 파일, 레지스트리 등의 변화를 감지하고 분석하는 것이며, 일반적으로 동적분석 도중 시스템이 손상되더라도 실제머신에 큰 영향을 미치지 않는 가상환경에서 분석을 진행한다. 마지막으로 정적분석은 악성코드 파일을 디컴파일하여, 사용된 함수 및 프로그램 코드 등과 같이 세부적인 분석을 하는 것을 말하며, 잘 알려진 OllyDbg, IDA pro와 같은 역분석 툴을 이용하여 분석한다.

2.2 가상환경의 구조



(그림 2) 가상머신 구조

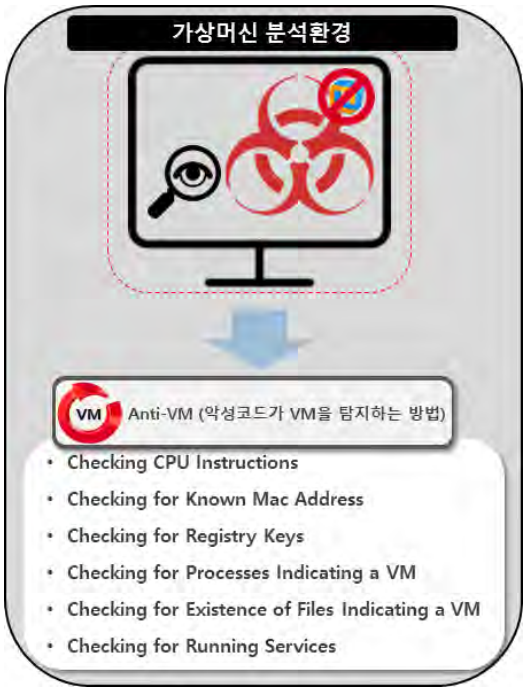
가상환경이란 물리적인 컴퓨터 환경을 소프트웨어로 구현한 것을 말한다. 이는 실제머신환경인 Host 운영체제 위에 Guest 운영체제를 추가로 설치할 수 있도록 하여, 가상의 환경을 구현하며 위의 (그림 2)와 같다[4].
가상머신 위의 설치된 운영체제는 가상으로 생성된 하드웨어 장치(가상 랜카드, 가상 CPU, 가상 메모리, 가상 HDD)를 실제 장치로 인식하며, 가상 하드웨어 장치는 실제 하드웨어 장치와 다른 이름을 가진다. 또한 이를 사용하기 위해 관련된 드라이버가 설치되기 때문에, 가상머신을 식별할 수 있는 정보가 존재하게 된다. 악성코드는 이러한 정보인 CPU Instruction, MAC 주소, 레지스트리 키, 파일, 프로세스 및 서비스 등을 통해 가상머신을 탐지하며, 본 논문에서는 동적분석을 위해 대표적으로 사용되는 가상머신인 Vmware, VitualBox에 대한 내용을 중점으로 다루고 있다.

2.3 Anti-VM 기술

악성코드 개발자들은 자신의 악성코드를 분석 하지 못하도록 보호하기 위한 방법으로 Anti-VM 기술을 적용한다. Anti-VM 기술이 적용된 악성코드는 자신이 동작하는 환경이 실제머신환경인지 프로세스, 파일시스템, 레지스트리 등을 탐지하여 판단하고, 가상환경이라 탐지될 경우, 악성코드가 동작하지 않도록 하여, 악성코드 분석 기술을 우회하는 방법 중 하나이다[5].

2.4 악성코드의 가상머신 탐지기법

Anti-VM 기술이 적용된 악성코드는 가상환경에서 악성행위를 하지 않는다. 이는 악성코드가 가상머신을 탐지하여 실제머신에서만 동작함을 의미하는데, 악성코드가 가상환경을 탐지하는 방법은 (그림 3)과 같다[6,7].



(그림 3) 악성코드의 가상머신 탐지기법

• D1. Checking CPU Instructions
가상환경에서 사용되는 특정 CPU Instruction이 존재하며, 이를 식별할 수 있는 Instruction은 <표 1>과 같다.

<표 1> 가상환경에서 사용하는 CPU Instruction

방법	설명
CPUID	- EAX Instruction을 통해 CPUID를 호출할 경우 가상화 제공업체 문자열 반환하는 특징을 통해 탐지
MMX	- 그래픽 처리를 위해 사용되는 Instruction이며, 가상환경에서는 지원되지 않는 기능을 이용한 탐지
Magic Number	- 특정 코드를 실행하여, 가상환경에서 호스트와의 통신이 특정 I/O 포트에서 이루어지는 특징을 이용한 탐지

• D2. Checking for Known MAC Address

MAC 주소의 상위 3 바이트는 제조사를 나타내는 고유번호를 사용하는데 <표 2>와 같이 가상환경에서는 특정 MAC 주소를 사용한다.

<표 2> 가상환경에서 사용하는 MAC 주소

접두사	제공업체
00 : 05 : 69	Vmware, Inc.
00 : 0C : 29	Vmware, Inc.
00 : 1C : 14	Vmware, Inc.
00 : 50 : 56	Vmware, Inc.
08 : 00 : 27	VirtualBox, Inc.

• D3. Checking for Registry Keys

레지스트리 키 값에는 가상환경에 설치되어 있는 소프트웨어 및 디바이스 드라이버에 대한 정보를 담고 있으며 <표 3>과 같이 가상환경에서는 특정 레지스트리가 존재한다.

<표 3> 가상환경에 저장된 레지스트리 키

No	레지스트리 키
1	HKLM\SOFTWARE\Vmware Inc
2	SYSTEM \CurrentControlSet\Control \VirtualDeviceDrivers
3	SYSTEM \CurrentControlSet\Control \CriticalDeviceDatabase\root#vmwvmcihostdev

• D4. Checking for Processes Indicating a VM

프로세스는 컴퓨터에서 연속적으로 실행되고 있는 컴퓨터 프로그램을 말하는데, 가상환경 특징을 가진 프로세스가 존재하며 <표 4>와 같이 Vmware, VirtualBox의 가상환경을 가리키는 프로세스들이 존재한다.

<표 4> 가상환경에서 사용하는 프로세스

프로세스	제공업체
Vmtoolsd.exe	Vmware, Inc.
Vmwareetrat.exe	Vmware, Inc.
Vmwareuser.exe	Vmware, Inc.
Vmacthlp.exe	Vmware, Inc.
vboxservice.exe	VirtualBox, Inc.
vboxtray.exe	VirtualBox, Inc.

• D5. Checking for Existence of Files Indicating a VM

가상환경에서 정상적으로 동작하기 위해 가상화 소프트웨어가 존재하며 <표 5>와 같이 가상환경을 가리키는 특정 파일들이 존재한다.

<표 5> 가상환경에 존재하는 파일

존재하는 파일	환경
Vmmouse.sys	Vmware
vm3dgl.dll	Vmware
VBoxMouse.sys	VirtualBox
VBoxGuest.sys	VirtualBox

• D6. Checking for Running Services

가상환경을 제공하기 위해서 특정 서비스가 동작하며 <표 6>과 같이 가상환경을 가리키는 특정 서비스가 존재한다.

<표 6> 가상환경에서 동작하는 서비스

No	동작하는 서비스
1	VMTools
2	Vmhgfs
3	Vmmouse
4	Vmusbmouse

3. 악성코드 Anti-VM 기술 방지방법론

본 절에서는 앞 절에서 정의한 악성코드의 가상머신 탐지 기법을 방지할 수 있는 방안을 제안하여, 악성코드가 가상환경을 탐지하지 못하도록 하는 것을 목적으로 한다. 이에 대한 방법으로는 가상환경의 특징으로 판단될 수 있는 CPU 인스트럭션, MAC 주소, 파일, 레지스트리, 서비스 등을 은닉하거나 API를 후킹하여 악성코드가 가상환경 특징정보를 검색할 수 없도록 하는 방식이 있다[8]. 이에 따라 Anti-VM 기술 방지방안은 <표 7>과 같고 각 가상머신 탐지 기법 방지 방안에 대한 설명은 아래와 같다.

• R1. Checking CPU Instructions

가상환경에서만 사용되는 특정 CPU Instruction이 존재하며, 악성코드는 특정 인스트럭션을 사용하여 반환되는 값을 보고 가상환경에서만 발생할 수 있는 특징을 탐지하여 가상머신임을 알 수 있다. 따라서 악성코드가 호출한 인스트럭션의 반환 값을 변경하여, 가상환경 탐지를 방지할 수 있다.

• R2. Checking for Known MAC Address

가상환경에서 사용되는 MAC 주소의 상위 3 바이트의 특정 MAC 주소를 사용하기 때문에, 악성코드는 이를 탐

지하여 가상머신임을 알 수 있다. 따라서 상위 3 바이트의 MAC 주소 값을 변경하여, 가상환경 탐지를 방지할 수 있다.

• R3. Checking for Registry Keys

가상환경에서는 설치되어 있는 소프트웨어 및 디바이스 드라이버에 대한 특정 레지스트리 키 값이 존재하기 때문에, 악성코드는 이를 탐지하여 가상머신임을 알 수 있다. 따라서 악성코드가 특정 레지스트리 값의 키워드를 변경하거나, 검색 자체를 무력화 하는 API 후킹을 하는 방법을 통해, 가상환경 탐지를 방지할 수 있다.

• R4. Checking for Processes Indicating a VM

프로세스는 컴퓨터에서 연속적으로 실행되고 있는 컴퓨터 프로그램을 말한다. 악성코드는 가상환경에서만 동작하는 프로세스가 존재하기 때문에, 이를 탐지하여 가상머신임을 알 수 있다. 따라서 가상머신에서 사용되는 모든 프로세스를 은닉하여, 가상환경 탐지를 방지할 수 있다.

• R5. Checking for Existence of Files Indicating a VM

가상환경에서는 시스템이 동작하기 위한 가상화 소프트웨어가 존재하는데, 악성코드는 이 파일들을 탐지하여 가상머신 임을 알 수 있다. 하지만 이러한 파일의 이름을 변경할 경우, 가상환경이 제대로 동작하지 않을 수 있기 때문에, 악성코드가 이러한 파일을 검색을 무력화 하는 API 후킹을 하는 방법을 통해, 가상환경 탐지를 방지할 수 있다.

• R6. Checking for Running Services

가상환경에서는 서비스를 제공하기 위해 특정 서비스가 존재하며, 악성코드는 이를 탐지하여 가상머신임을 알 수 있다. 따라서 프로세스를 은닉하는 방법과 동일하게 동작하고 있는 서비스를 은닉하여, 가상환경 탐지를 방지할 수 있다.

<표 7> Anti-VM 기술 방지방안

방지방안	Anti-VM 기술 방지분석					
	R1	R2	R3	R4	R5	R6
API 후킹			✓	✓	✓	✓
가상환경 특징 은닉				✓	✓	✓
가상환경 특징 변경	✓	✓	✓			

4. 결론

본 논문에서는 Anti-VM 보호기법이 적용된 악성코드를 분석하기 위해 이러한 Anti-VM 기술을 우회하는 방안을 제안하였다. 이를 위해 악성코드가 가상환경을 탐지하는 기법에 대한 분석을 하였다. 또한, Anti-VM 기술이 적용된 악성코드를 앞서 분석한 각각의 악성코드의 탐지기법을 방지하여, 악성코드가 가상환경을 탐지하지 못하도록 하는 방안을 제안하였다.

결과적으로, 신・변종 악성코드에 대응하기 위해선 악성행위 발현을 통한 동적분석이 필요하다. 악성코드에 적용된 보호기법 중 하나인 Anti-VM 기술을 우회하기 위한원시적인 방법으로 악성코드를 실제머신에서 분석하는 방안이 존재한다. 하지만 실제머신에서 동적분석을 할 경우, 분석 도중 시스템이 손상되더라도 복원이 불가능한 문제점을 가지고 있다.

따라서 본 논문에서는 악성코드에 적용된 Anti-VM 기술을 방지하는 방법으로, 가상환경을 탐지하는 기법을 분석한다. 이를 통해 실제머신에 큰 영향을 미치지 않고 복원이 가능한 장점을 가진 가상환경에서 악성행위를 분석을 할 수 있도록 하는 방안을 연구하였다.

향후 연구로는 본 연구에서 제안한 Anti-VM 기술 우회 방안을 방법론인 특정 레지스트리 값을 검색자체를 불가능 하게 하는 API 후킹, 가상환경에서만 존재하는 특정 프로세스, 서비스 등을 숨길 수 있는 은닉 기술을 구현 하여 실질적으로 보호기법이 적용된 악성코드를 가상환경에서 분석이 가능하도록 하는 연구가 진행될 것이며, 이에 대한 검증을 진행할 것이다. 이에 대한 연구가 진행될 경우 보호기법이 적용된 신・변종 악성코드 대응에 활용 될 것으로 기대한다.

참고문헌

[1] 한국인터넷진흥원, "2016년 1분기 사이버 위협 동향 보고서", 2016.05.

[2] Xu Chen, Jon Andersen, Z.Morley Mao, Michael Bailey, Jose Nazario "Towards an Understanding of Anti-virtualization and Anti-debugging Behavior in Modern Malware", International Conference on Dependable Systems&Networks: Anchorage. 2008.09.

[3] 박남열, 김용민, 노봉남 "우회기법을 이용하는 악성코드 행위기반 탐지 방법", 정보보호학회논문지, 2006.06.

[4] N. Rin "Virtual Machines Detection Enhanced," 2013.11.

[5] 임채태, 오주형, 정현철 "최신 악성코드 기술동향 및 분석 방안 연구", 정보과학회지, 2010.11.

[6] Ming-Kung Sun, Mao-Jie Lin, Michael Chang, Chi Sung Laih, Hui-Tang Lin "Malware Virtualization-Resistant Behavior Detection," IEEE 17th International Conference on Parallel and Distributed Systems, 2011.07.

[7] Yaniv Assor "Anti-VM and Anti-Sandbox Explained", 2016.05.

[8] Alberto Ortega, "Your malware shall not fool us with those anti-analysis tricks", 2012.11.