API 및 파라미터 분석을 통한 Anti-VM 악성코드 탐지

이수영 [1], 유헌창 [2]
[1] 고려대학교 SW•AI 융합대학원 소프트웨어보안학과 석사과정
[2] 고려대학교 컴퓨터학과 교수

lahoo92@korea.ac.kr
yuhn@korea.ac.kr

Detection of Anti-VM Malware through API and Parameter Analysis

Su-Young Lee[1], Heon-Chang Yu[2]
[1]Dept. of Software Security, Korea University
[2]Dept. of Computer Science and Engineering, Korea University

요 약

악성코드에 의한 공격은 연평균 수억 건 이상 발생하고 있으며, 큰 폭으로 증가하는 악성코드 개수에 비해 악성코드 분석가가 각 악성코드를 분석하고 대응하기에는 한계가 있다. 이와 같은 한계를 극복하기 위한 방법으로 악성코드 자동 분석 시스템을 사용하는 방법이 있다. 그러나 악성코드에 Anti-VM 기술들이 포함된 경우 자동 분석 시스템에서는 제대로 탐지하기 어렵다는 단점이 있다. 본 논문에서 Anti-VM 기술에 활용되는 API 와 API 의 파라미터 정보를 분석하여 API 후킹 모듈을 기반으로 Anti-VM 탐지 우회 방법을 제안한다. Anti-VM 기술은 Virtual Box 기반의 가상 환경을 탐지하는 기술을 기준으로 연구를 진행하였고 API 및 파라미터 또한 Virtual Box 기반의 가상 환경을 탐지할 때 활용되는 API 및 파라미터를 분석하였다. 분석한 데이터를 기반으로 Anti-VM 기술을 우회하기 위한 API 및 후킹 프로그램을 제작해서 연구를 진행하였다. 연구 결과 Virtual Box 기반의 가상 환경을 탐지하는 기술 중 시간 딜레이, 마우스 이벤트 등의 행위 기반 탐지 기술은 명확한 지표를 측정하기 어려워 우회가 어려웠으나 가상 환경 특성 탐지, 파일 아티팩트 탐지, 레지스트리 및 시스템 설정 탐지 기술은 식별 및 우회에 뛰어난 성능을 보였다.

1. 서론

독일의 백신 테스트 연구소 av-test 에 의하면 2023 년에는 무려 약 12 억 개에 육박하는 악성코드가 유포된 것으로 나타났다[1]. 게다가 최근 공격자들이 Malware as a Service(MaaS), Ransomware as a Service (RaaS)와 같은 사이버 범죄 비즈니스 모델을 사용함으로써 악성코드 과거에는 악성코드 및 해킹에 대한 전문적인 지식이 필요했지만, 최근에는 전문적인 지식이 없는 사람도 공격이 공격할 수 있게 됐다[2].

수많은 악성코드를 사람이 분석하고 대응하기에는 어렵다는 한계를 극복하기 위한 방법으로 악성코드 자동 분석 시스템을 활용하는 방법이 있다. 그러나 악성코드 중 Anti-VM 기술이 포함된 경우 여러 방법을 통해 가상환경을 탐지하여 회피하도록 구현되어 있어 자동 분석 시스템이 제대로 된 결과를 도출할 수 없다.

본 논문에서는 자동 분석 시스템에서 탐지가 가능하도록 API 후킹 모듈을 기반으로 악성코드가 사용하는 Anti-VM 기술 중 API 를 이용하여 탐지하는 기술을 우회하는 방법을 제안한다. 제안하는 방법은 가상환경을 탐지하기 위해서 사용하는 API 를 분석 후 API 후킹을 통해 가상환경을 탐지하지 못하도록 조작한다. 특히 API 만을 기준으로 후킹을 할 경우 같은 API 일지라도 다른 목적으로 사용했을 가능성이 있기 때문에 Anti-VM 기술을 우회했더라도 본래의 악성코드 목적을 식별하기 어려워질 수 있다. 예를 들면 악성코드가 Windows 레지스트리 키/값을 질의하는 경우 가상환경 탐지, 시스템 정보 수집 등 목적에 따라 다양하게 해석할 수 있다. 따라서 Anti-VM 기술에서 사용하는 API 와 핵심 파라미터를 동시에 분석하는 방법을 통해 정밀도를 높이는 방법론을 제시한다.

2. 관련 연구

2.1 Frida 를 이용한 API 후킹

Frida 는 개발자, 리버스 엔지니어 및 보안 연구원을 위한 동적 분석 도구로 자바스크립트를 이용해서 어플리케이션의 메소드 및 함수, API, 라이브러리 등을 후킹해서 다양한 방식의 동적 분석이 가능하다.

2.2 Anti-VM 기술 분류

Anti-VM 은 가상 환경에서 실행되는 것을 탐지하고 그 환경에서의 탐지를 회피하기 위해 사용되는 기술 또는 메커니즘이다. Anti-VM 기술은 크게 가상 환경 특성 탐지, 파일 아티팩트 탐지, 레지스트리 및 시스템 설정 탐지, 행위 기반 탐지로 분류하였고 각 기술에 활용되는 API 를 식별하였다.[3] Anti-VM 기술 분류는 <표 1>에서 보여준다.

<표 1> Anti-VM 기술 분류 및 API 함수 목록

Anti-VM 기술 분류	API 함수
가상 환경 특성 탐지	GetUserName() GetComputerName() Process32First() Process32Next() GetAdaptersAddresses() GetTickCount() GetDiskFreeSpaceEx() 등
파일 아티팩트 탐지	CreateFile() FindWindow() FindFirstFileA() FindNextFileA() GetFileAttributes() 등
레지스트리 및 시스템 설정 탐지	RegOpenKeyExA() RegQueryValueExA() 등
행위 기반 탐지	GetCursorPos() GetMessage() PostMessageA() SetWindowsHookEx() CallNextHookEx() GetDoubleClickTime() 등

2.2 Pafish 를 이용한 Anti VM 기술 식별

Pafish(Paranoid Fish)는 악성코드 패밀리와 동일한 방식으로 가상 머신 및 악성코드 분석 환경을 탐지하기 위해 다양한 기술을 사용하는 테스트 도구이다. 해당 도구에는 Anti-VM 기술이 다수 포함되어 있어 VMware, Virtual Box, qemu 등 다양한 가상환경을 탐지할 수 있다. 그 중에서 Virtual Box 를 대상으로 가상 환경 탐지 우회 방안을 연구할 예정이다.

3. API 및 파라미터 후킹을 통한 Anti-VM 우회

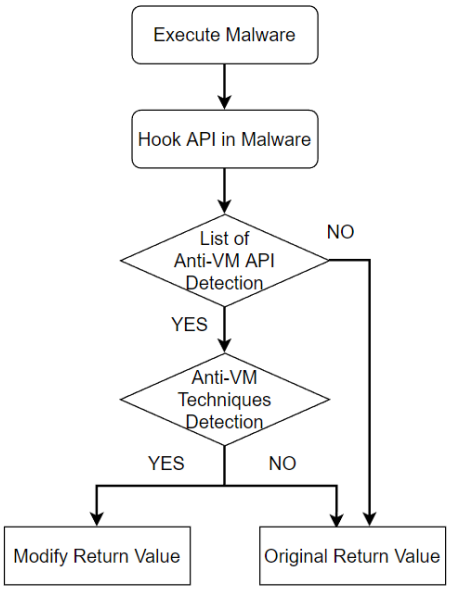
3.1 API 및 파라미터 후킹 원리

Anti-VM 기술을 우회하기 위한 방법으로는 API 후킹을 이용해서 악성코드가 가상 환경을 탐지하기 위해서 사용한 API 의 리턴 값을 조작하여 가상 환경을 탐지하지 못하도록 할 수 있다. 이때 악성코드가 사용한 API 만으로는 가상 환경을 탐지하기 위한 목적으로 사용되었는지 다른 목적으로 사용되었는지 식별하기 어렵다는 문제가 있다. 예를 들면 일부 설치 프로그램의 경우 다른 버전의 프로그램이 이미 설치되어 있는지를 확인해서 다른 버전의 프로그램을 삭제 후 설치하도록 구현되어 있다. 이때 다른 버전의 프로그램이 설치되어 있는지 여부를 RegOpenKeyEx()나 RegQueryValueEx()로 레지스트리 키, 값을 질의하여 확인할 수 있다. 이런 경우에는 해당 API 를 Anti-VM 기술로써 사용한 API 라고 볼 수 없다.

이때 API 를 후킹 해서 API 에 사용된 파라미터를 같이 검증한다면 위와 같은 오류를 줄일 수 있다. 가상 환경의 운영체제는 일반적인 운영체제와 달리 가상화 기술을 사용하여 만들어진 운영체제이기 때문에 CPU, 하드웨어와 같은 디바이스 정보부터 가상 환경을 동작시키기 위해 필요한 라이브러리, 파일 등 파일 시스템까지 다양한 부분에서 일반적인 운영체제와 상이한 부분들이 많다. 따라서 가상 환경을 탐지하기 위해서 사용하는 API 의 파라미터도 이러한 차이점을 잡기하기 위한 데이터를 저장하고 사용한다. 예를 들어 RegOpenKeyEx()나 RegQueryValueEx()로 레지스트리 키 또는 값을 질의할 때 파라미터에 "VMware", "VBox", "Virtual Box" 등 가상 환경을 의미하는 문자열이 포함된 키 혹은 값을 질의하여 가상 환경인지 확인한다.

3.2 API 및 파라미터 기반 Anti-VM 우회 방안

악성코드는 Anti-VM, Anti-Debugging, 정보 유출, 지속성(Persistence) 확보, 방어 회피(Defense evasion) 등 여러 가지 목적으로 API 함수를 사용한다. 이때 Anti-VM 에 사용되는 API 만 식별해서 리턴 값을 조작해야 하는데 Anti-VM 관련 API 를 정확하게 식별하지 않는 경우 실질적인 악성 행위가 제대로 식별되지 않는 문제가 발생한다. 악성코드에서 API 를 후킹 하는 경우 해당 API 가 가상 환경 목적으로 사용했는지 확인이 어렵다. 따라서 API 중 가상 환경 탐지에 활용되는 API 를 후킹 한 후 API 및 파라미터 분석을 통해 해당 API 가 Anti-VM 의 목적을 갖는 API 인지 식별한 뒤 리턴 값을 조작해야 한다.



(그림 1) API 및 파라미터 기반 Anti-VM 우회

3.2 API 및 파라미터 기반 Anti-VM 기술 탐지

Anti-VM 기술에 사용되는 API 마다 판단해야 할 핵심 파라미터의 위치, 데이터 타입이 모두 상이하다. 따라서 리스트업 한 Anti-VM 관련 API 들의 구조와 핵심 파라미터를 식별 및 분석해서 Anti-VM 기술 탐지의 정확도를 상승시켰다.

<표 2> API 함수 및 파라미터 분석

API 함수	위치	타입
GetUserName()	Arg[0]	LPSTR
GetComputerName()	Arg[0]	LPSTR
Process32First()	Arg[1]	LPPROCESSENTRY32
Process32Next()	Arg[1]	LPPROCESSENTRY32
GetDiskFreeSpaceEx()	Arg[2]	PULARGE_INTEGER
CreateFileA()	Arg[0]	LPCSTR
FindWindow()	Arg[0] Arg[1]	LPCSTR LPCSTR
FindFirstFileA()	Arg[0]	LPCSTR
FindNextFileA()	Arg[0]	LPCSTR
GetFileAttributes()	Arg[0]	LPCSTR
RegOpenKeyExA()	Arg[1]	LPCSTR
RegQueryValueExA()	Arg[4]	LPBYTE

4. Anti-VM 우회 기술 검증

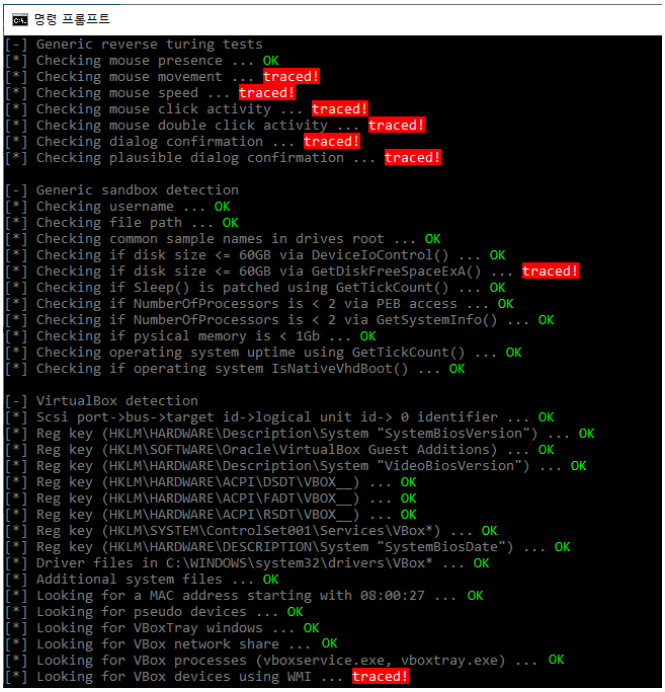
Frida 를 이용해서 API 와 파라미터 분석 결과를 기반으로 후킹을 시도하고 Anti-VM 기술 우회 여부를 검증하였다. 검증에 사용한 프로그램은 악성코드보다 많은 케이스의 Anti-VM 기술을 갖고 있는 Pafish 도구를 사용했으며 Anti-VM 기술 우회 방법 적용 전후의 결과를 기반으로 비교했다. 우회 여부를 판단할 Anti-VM 관련 API 는 가상 환경 특성 탐지, 파일 아티팩트 탐지, 레지스트리 및 시스템 설정 탐지, 행위 기반 탐지 기술에서 활용되는 API 중 Pafish 에서 사용하는 API 를 기반으로 선정하여 검증하였다. 그리고 탐지 정확도를 높일 API 의 파라미터는 Virtual Box 가상 환경을 탐지하는 파라미터 데이터를 기반으로 검증하였다.

Pafish 도구에는 qemu, VMware, Virtual Box, Wine 을 대상으로 가상 환경 특성 탐지, 파일 아티팩트 탐지, 레지스트리 및 시스템 설정 탐지 기술이 포함되어 있다. 우선 본 논문의 Anti-VM 기술 탐지 방법론을 적용하지 않고 Pafish 도구를 실행시킨 결과는 (그림 2) 에서 보여준다.

(그림 2) Anti-VM 기술 탐지 방법론 적용 전 결과

Anti-VM 기술 탐지 방법론은 적용하지 않은 상태에서는 API 를 기반으로 Virtual Box 가상 환경을 탐지하는 기술 35 개 중 23 개에서 탐지되었다. Anti-VM 에 활용된 API 는 중복으로 포함해서 약 100 개의 API 가 호출되었고 그중에서 71 개가 탐지되었다.

본 논문의 Anti-VM 기술 탐지 방법론을 적용하고 Pafish 도구를 실행시킨 결과는 (그림 3)에서 보여준다.



<그림 3> Anti-VM 기술 탐지 방법론 적용 후 결과

Anti-VM 기술 탐지 방법론은 적용한 상태에서는 API 를 기반으로 Virtual Box 가상 환경을 탐지하는 기술 35 개 중 8 개에서 탐지되었다. Anti-VM 에 활용된 API 는 중복으로 포함해서 약 100 개의 API 가 호출되었고 그중에서 27 개가 탐지되었다.

API 기반 Virtual Box 가상 환경 탐지율은 Anti-VM 기술 탐지 방법론 적용 전과 후로 65.71%에서 22.85%로 감소했고 중복으로 호출된 API 를 포함하면 71%에서 27%로 감소한 결과를 확인할 수 있었다.

5. 결론

본 논문에서는 Anti-VM 악성코드를 가상 환경 안에서 안전하게 분석하기 위해 Anti-VM 우회 기술을 구현하였다. 제안하는 기술은 Anti-VM 기술 중 API 를 기반으로 가상 환경을 탐지하는 기술을 대상으로 Anti-VM 기술을 우회하는 기술이다. 먼저 Anti-VM 기술을 가상 환경 특성 탐지, 파일 아티팩트 탐지, 레지스트리 및 시스템 설정 탐지 기술로 분류하였다. 다음으로 각 기술에서 사용되는 API 를 식별하고 후킹을 통해 Anti-VM 기술에서 사용하는 API 를 탐지하여 리턴 값을 변조함으로써 가상 환경을 인식하지 못하도록 하였다. 그 다음 우회 기법의 정확도를 높이기 위해서 단순히 API 만을 기준으로 탐지하는 방식이 아닌 API 별 가상 환경을 탐지하는 핵심 파라미터를 같이 분석하였다. 각 API 별로 핵심 파라미터 위치와 데이터 타입을 분류하였고 이는 단순 문자열을 기반의 파라미터 뿐만 아니라 포인터 형태의 파라미터 또한 포인터가 가리키는 주소의 실제 데이터를 분석하여 Anti-VM 기술 우회의 정확도를 높였다. 그 결과 실행되는 환경이 Virtual Box 기반의 가상 환경임에도 불구하고 Anti-VM 기술이 무력화되어 가상 환경을 제대로 탐지하지 못하는 결과를 도출하였다.

향후 연구로는 기존 연구에 추가로 Anti-VM 기술 중 사용자의 상호작용, 튜링 테스트와 같은 행위 기반 탐지 기술과 API 기반으로 악성코드의 실질적인 목적을 분류하여 악성코드를 유포하는 공격자들의 TTPs 까지 분석할 수 있는 악성코드 자동화 분석 시스템을 구현할 예정이다.

참고문헌

[1] "Total Amount Of Malware And PUA", 19 September 2024. [Online]. Available: https://portal.av-atlas.org/malware. [Accessed 19 September 2024].

[2] Grace Friscilla Margaretha Karo-Karo, Muhammad Sofyan Arif Harumnanda, Charles Lim, "Investigating Multiple Malware as a Service(MaaS): Analysis and Prevention Techniques", IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs), 2023, pp. 270

[3] Dong-Yeob Kim, Sang-Yong Choi, Malware Detection Technology Based on API Call Time Section Characteristics, Korea Institute of Information Security and Cryptology(KIISC), Busan Korea, 2022