한국컴퓨터정보학회 동계학술대회 논문집 제25권 제1호 (2017. 1)

IAT 후킹 탐지 방안에 대한 연구

임하빈[O], 오인수[*], 이경률[**], 임강빈[*]
[O]고려대학교 정보보호대학원
[*]순천향대학교 정보보호학과
[**]순천향대학교 보안안전융합기술사업화센터
e-mail: habin103@korea.ac.kr[O], catalyst32@sch.ac.kr[*], carpedm@sch.ac.kr[**], yim@sch.ac.kr[*]

Countermeasure for Detecting IAT Hooking

Habin Yim[O], Insu Oh[*], Kyungroul Lee[***], Kangbin Yim[*]
[O]Center for Information Security Technologies (CIST), Korea University
[*]Dept. of Information Security Engineering, Soonchunhyang University
[**]R&BD Center for Security and Safety Industries, Soonchunhyang University

● 요 약 ●

악성코드는 매년 그 수가 증가하고 있으며, 악성코드의 공격기법이 지능적이고 복합적으로 진화함에 따라 이에 대한 분석과 대응이 요구된다. 하지만 일부 악성코드는 감염여부를 숨기기 위하여 분석에 대한 회피방법으로 루트킷을 통하여 방어자에 의한 악성코드의 코드 분석을 우회함으로써 은폐된 상태로 악의적인 공격을 수행한다. 따라서 본 논문에서는 유저레벨에서 IAT (Import Address Table)의 정보를 후킹하여 악성 행위를 수행하는 루트킷을 탐지하는 대응방안을 제안한다.

키워드: 역공학(reverse-engineering), 악성코드(malware), 루트킷(rootkit), 후킹(hooking)

I. Introduction

초기의 공격자들은 자신의 기술을 과시하거나 호기심으로 악성코드를 제작하였다. 하지만 현재 인터넷 보급이 활발해짐에 따라 악성코드를 활용하여 금전적인 이득을 취하는 목적의 공격이 증가하고 있으며, 그림 1과 같이 AV-TEST 기관의 조사[1]에 따르면 2016년을 기준으로 390,000개 이상의 새로운 악성코드가 지속적으로 발견되어 그 수가 매해 증가하는 추세를 보이고 있다. 기술이 발전함에 따라 고도화된 형태로 변화하는 악성코드는 방어자로부터의 분석을 회피하기 위한 방법으로 IAT의 정보를 후킹하는 루트킷을 활용한 공격을 시도한다. 루트킷은 일반적인 악성코드보다 그 수가 적지만 점점 증가하는 추세[2]이며, 해당 기술로 IAT의 정보가 변경되어 악성코드 탐지에 대한 우회가 가능하다.

본 논문에서는 IAT 후킹 시나리오를 통하여 공격기법을 검증하고, 이에 대한 대응방안으로 메모리 영역의 검증을 통한 IAT 후킹 탐지방안을 제안한다.

II. The Proposed Scheme

1. Attack Scenario of IAT Hooking

프로그램은 실행 이후에 별도로 로드한 라이브러리 파일을 제외하고 대부분 IAT를 사용하여 라이브러리를 임포트한다. 공격자는 악성코드를 통하여 프로세스 내부 IAT의 주소값을 변경시키는 방법으로 대상 프로그램을 후킹하고 분석을 회피하는 것이 가능하다.

본 논문에서는 루트킷 샘플을 구현하여 타겟 프로그램에서 사용하는 함수를 후킹하여 그 결과를 분석한 후, 제안하는 탐지 프로그램을 통하여 타켓 프로그램의 후킹여부를 탐지한다. 타겟 프로그램은 그림 2와 같이 Test.exe 파일명을 가지고 프로세스 ID를 출력하는 프로그램이며, 출력하는 프로세스 ID는 프로세스의 정보를 관리하거나 중지하는 등의 목적을 위하여 운영체제에서 활용하는 정보이다.

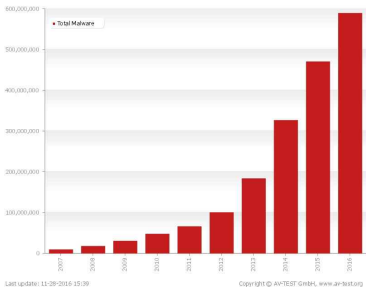

Fig. 1. Malware Trend in Recent 10 Years

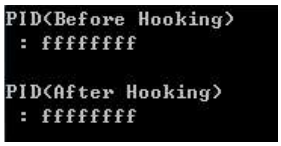

Fig. 2. Execution Result of Target Program

루트킷 샘플은 대상 대상프로그램의 프로세스 ID를 kernel32.dll의 GetCurrentProcess() 함수[3]를 후킹한 후, 이를 변경함으로써 방어 자로부터 자신을 보호하는 것이 가능하다. 이를 위하여 대상 프로세스 이름을 입력하여 후킹을 수행하면, 임의의 프로세스 ID를 반환하며, 그림 3과 같이 프로세스 ID가 0xFFFFFFFF에서 0x87654321로 변경된 것을 확인할 수 있다.

Fig. 82. Execution Result of Rootkit Sample Program

Fig. 83. IAT Hooking Result

2. Countermeasure for detecting IAT Hooking

루트킷 샘플은 타겟 프로그램의 IAT 내 후킹된 함수의 주소를 자신의 코드가 준비된 주소로 변경한다. 즉, 이러한 후킹 과정에서는 원래의 주소가 루트킷에 의하여 삽입된 주소로 변경되기 때문에, 프로세스 내부의 메모리 영역이 달라지는 특징이 발생한다. 따라서 본 논문에서는 이러한 차이를 점검하여 IAT 내 후킹된 함수를 탐지하는 방안을 제안한다. 이를 위하여 대상 프로세스에 임포트된 모든 DLL을 추출한 후, 추출된 DLL에 포함된 함수 중 해당 DLL의 주소 영역을 벗어나는 함수를 검사하여 후킹여부를 판단한다. 그림 5에 탐지 결과를 나타내었으며, 후킹된 함수인 GetCurrentProcess의 주소가 해당 DLL의 영역을 벗어났기 때문에 후킹된 것이 탐지되었다. 따라서 루트킷 혹은 악의적인 목적으로 프로세스 내 IAT에 포함된 함수를 후킹한 경우에는 해당 DLL의 주소 영역을 벗어나게 되므로 IAT가 후킹된 프로세스 및 함수를 효과적으로 탐지하는 것이 가능하다.

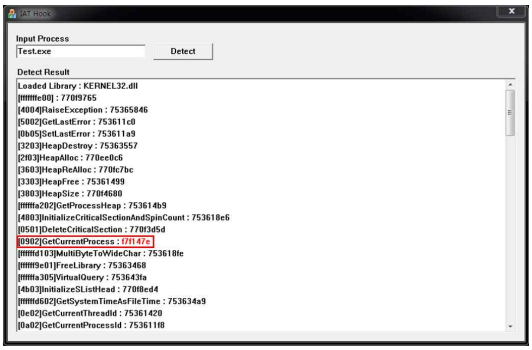

Fig. 84. Experiment Result of IAT Detection

III. Conclusions

오늘날 악성코드는 다양한 기술을 반영하여 진화하고 있으며, 악성코드 자체를 보안 프로그램에서 탐지하는 것을 회피하는 기법을 적용하는 형태로 진화되고 있다. 따라서 본 논문에서는 루트킷에 감염된 프로그램에 대한 후킹 여부를 탐지하는 방법으로 메모리 영역의 차이를 비교한 방안을 제시하였으며, 공격 시나리오를 제시하고 이를 검증함으로써 효과적으로 IAT 후킹을 탐지하는 것을 확인하였다.

Acknowledgment

이 논문은 2015년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임
(No. NRF-2015R1D1A1A01057300).

References

[1] AV-TEST, http://www.av-test.org
[2] Microsoft, “Malware Protection Center: Infection Reported”, http://www.microsoft.com
[3] Microsoft “GetCurrentProcess function”, https://msdn.microsoft.com/ko-kr/library/windows/desktop/ms683179(v=vs.85).aspx