Themida API Wrapping 자동 역난독화 시스템

김민호,* 조해현,* 이정현*

*숭실대학교

Automated Deobfuscation System for Themida API Wrapping

Minho Kim,* Haehyun Cho,* Jeong Hyun Yi*

*Soongsil University

요 약

최근 유행하는 악성코드 중 난독화 기법이 차지하는 비율은 약 31%으로 많은 비중을 차지하고 있다. 실제 유포된 악성코드 중에는 상용 난독화 도구 중 하나인 Themida에 의해 보호되는 사례가 발견되고 있다. Themida의 대표적인 난독화 기법은 API Wrapping 기법으로 2019년 Themida의 버전 업데이트가 되어 난독화 기법의 동작 방식이 크게 변경되었다. 이로 인해 기존의 역난독화 기법들을 적용할 수 없는 문제가 발생했다. 본 논문은 최신 버전의 Themida API Wrapping 기법이 적용된 악성코드를 분석하고 자동으로 역난독화하여 분석에 필요한 정보를 제공할 수 있는 자동 역난독화 시스템을 제안한다.

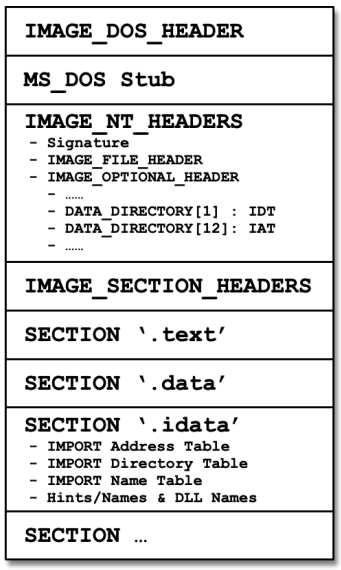
I. 서론

난독화 기법이란 프로그램 실행 결과는 동일하지만 내부 코드의 동작을 분석이 어렵게 변형하는 기법으로 프로그램을 보호하기 위한 목적을 가지고 있다. 이러한 난독화 기법이 적용되는 경우는 전체 악성코드의 약 31%를 차지하고 있다 [1]. 실제로 난독화 된 악성코드 중에는 Themida에 의해 보호되는 악성코드가 존재한다 [2, 3]. Hatching Triage [4]에 보고된 Themida로 난독화 된 악성코드 140개를 대상으로 직접 분석한 결과, 약 10%의 악성코드에서 Themida의 대표적인 난독화 기법인 API Wrapping 기법이 적용된 것을 확인했다.

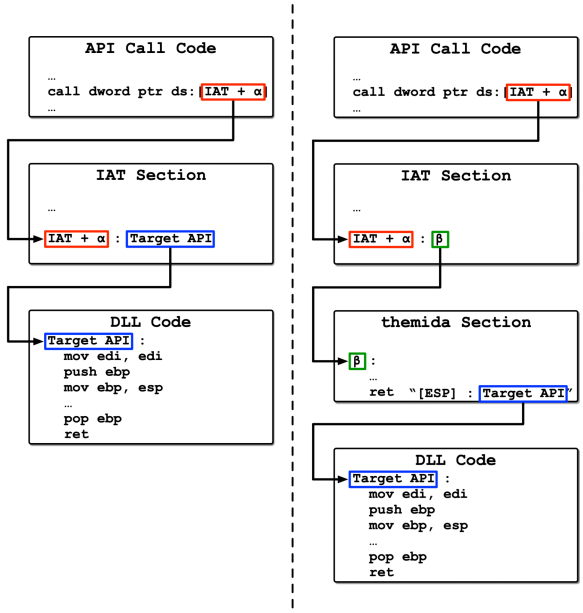
기존 Themida 난독화 기법에 대한 분석 및 역난독화 연구는 활발히 진행되었으나, Themida의 버전 업데이트로 난독화 기법의 동작 방식이 크게 변경되었다. 이에 따라 기존의 역난독화 방법은 최신 버전의 Themida에서 적용 불가능한 문제가 있다 [5, 6, 7]. 또한 최신 버전의 Themida를 대상으로 API Wrapping 연구가 진행되었으나 자동화된 역난독화 방법을 제안하지 못했다 [8].

본 논문의 Ⅱ장에서는 PE(Portable Executable) 파일 포맷 중 IMPORT 섹션 내용을 설명하고, 최신 버전의 Themida API Wrapping 동작 원리에 대한 분석을 설명한다. Ⅲ장에서는 Themida API Wrapping 기법을 자동으로 역난독화하기 위해 제안한 시스템을 설명한다. Ⅳ장에서 제안 시스템의 성능은 역난독화 된 결과를 가지고 검증한다. 마지막으로 Ⅴ장에서는 본 논문의 결론 및 추후 연구에 대해 기술한다.

이 논문은 2017년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2017-0-00168, 사이버 위협 대응을 위한 Deep Malware 자동 분석 기술 개발)



[그림 1] PE Format.



[그림 2] Call chain of Normal API Call and Obfuscated API Call.

II. 배경지식

2.1 윈도우 PE 파일 포맷

윈도우 PE 파일은 윈도우 운영체제에서 실행 가능한 프로그램으로 [그림 1]과 같은 구조로 구성되어 있다.

PE 파일 포맷에서 IMPORT 섹션과 관련된 항목은 DATA_DIRECTORY와 idata 섹션이다. DATA_DIRECTORY의 1, 12번째 인덱스 값은 각각 IDT(IMPORT Directory Table), IAT(IMPORT Address Table)의 오프셋 값으로 테이블의 시작 위치를 의미한다 [9]. idata 섹션은 바이너리가 API를 호출 시 필요한 심볼을 가지고 있다. 다른 DLL의 API를 호출할 때 호출하려는 API의 정보를 가져올 때 사용한다. 섹션은 DLL과 API 이름이 저장된 Hint/Names 테이블과 이를 가리키는 INT(IMPORT Name Table), IAT, IDT로 구성되어 있다. 바인딩 과정에서 위 구성 요소를 참조하여 가져온 API 주소를 IAT에 기록하고, API 호출 시 IAT에 기록된 주소로 분기한다.

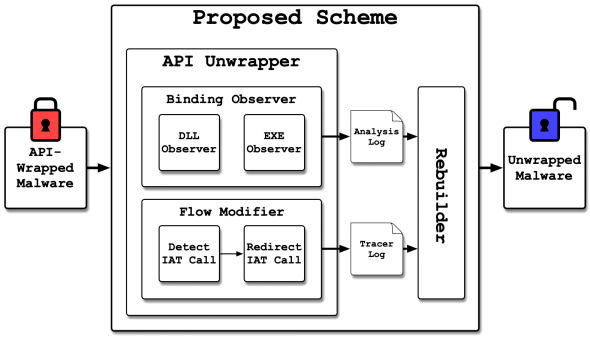
2.2 Themida API Wrapping 기법 분석

Themida는 디버거, 가상머신 탐지와 같은 분석 방지 기법과 String Encryption, API Wrapping과 같은 난독화 기법을 제공하여 프로그램의 분석 방지 및 보호 목적으로 사용하는 상용 프로텍터 프로그램이다 [10]. 본 논문에서는 Themida 3.0.7.0 버전을 기준으로 API Wrapping 기법을 분석했다.

Themida API Wrapping 기법에 의한 난독화 적용 여부에 따른 동작 방식은 [그림 2]와 같다. 난독화가 되지 않은 경우는 IAT에 API 주소가 바인딩 과정에서 기록된다. 그리고 API 호출 시 IAT를 참조하여 기록된 API 주소로 분기한다. 반면에 난독화가 된 경우는 IAT에 API와 1대1로 대응하는 Themida 코드 블록의 주소가 기록된다. 이후, API 호출 시 IAT를 참조하여 Themida 코드 블록으로 분기한다. Themida 코드 블록은 더미 코드와 API 주소를 계산하는 코드가 실행된다. Themida 코드 블록 실행이 종료되면 계산된 API 주소로 분기하는 방식으로 동작한다.

III. 시스템 설계 및 구현

본 논문에서 제안하는 Themida API Wrapping 역난독화 방법은 Intel PIN [11]을 활용하여 구현했다.



[그림 3] Proposed Scheme.

pFile	Data	Description	Value
0000804C	0001B870	Hint/Name RVA	02D3 GetStartupInfoW
00008050	0001B852	Hint/Name RVA	0571 SetUnhandledExceptionFilter
00008054	0001B836	Hint/Name RVA	05B1 UnhandledExceptionFilter
00008058	0001B820	Hint/Name RVA	0366 InitializeSListHead
0000805C	00000000	End of Imports	KERNEL32.dll
00008098	0001B450	Hint/Name RVA	002E __vcrt_GetModuleFileNameW
0000809C	0001B436	Hint/Name RVA	0035 _except_handler4_common
000080A0	0001B42C	Hint/Name RVA	0048 memset
000080A4	0001B40E	Hint/Name RVA	001D __current_exception_context
000080A8	0001B3F8	Hint/Name RVA	001C __current_exception
000080AC	0001B46C	Hint/Name RVA	002F __vcrt_GetModuleHandleW
000080B0	0001B486	Hint/Name RVA	0031 __vcrt_LoadLibraryExW
000080B4	0001B3D8	Hint/Name RVA	0025 __std_type_info_destroy_list
000080B8	00000000	End of Imports	VCRUNTIME140D.dll
000080E8	0001B66A	Hint/Name RVA	0545 strcat_s
000080EC	0001B676	Hint/Name RVA	008E __stdio_common_vsprintf_s
000080F0	0001B64E	Hint/Name RVA	0073 __p__commode
000080F4	0001B6A4	Hint/Name RVA	0197 _initialize_onexit_table

[그림 4] Normal PE File IAT List.

3.1 시스템 구조

[그림 3]은 API Wrapping 역난독화 시스템의 구조도이다. 본 논문의 제안 시스템은 크게 PIN을 활용하여 동적으로 바이너리를 분석하는 API Unwrapper 모듈과 API Unwrapper 모듈에서 생성한 로그를 바탕으로 IMPORT 섹션을 재구성하는 Rebuilder 모듈로 구성된다.

3.2 API Unwrapper

API Unwrapper 모듈은 Intel PIN을 이용하여 바이너리를 동적으로 분석하는 모듈로, 크게 Binding Observer 모듈과 Flow Modifier 모듈로 이루어져 있다.

3.2.1 Binding Observer

Binding Observer 모듈은 바이너리를 실행할 때 바이너리와 DLL이 메모리에 로드되는 것을 관찰한다. 바이너리에 Themida 섹션이 존재하면 Themida 섹션 정보를 기록한다. 그리고 DLL의 API 목록 정보와 다른 DLL로 포워딩되는 API 목록을 기록한다.

3.2.2 Flow Modifier

Flow Modifier 모듈은 IAT에 기록된 순서대로 모든 Themida 코드 블록이 실행되도록 실행 흐름을 변경한다. 실행 중인 바이너리의 IAT를 참조하여 API 호출 루틴을 탐지하면 실행 흐름을 IAT에 기록된 Themida 코드 블록의 주소로 변경하고 분기한다. Themida 코드 블록은 종료되기 직전에 해당 Themida 코드 블록에 대응되는 API 주소가 메모리에 기록되는 특징을 가지고 있다. 이러한 특징을 이용하여 메모리에 기록된 값이 API 주소값이면 기록한다. 그리고 API 주소가 기록된 메모리의 값을 다음 Themida 코드 블록의 주소로 변경한다. 이렇게 하면 실제 API 호출 대신 다음 Themida 코드 블록이 호출된다. 이렇게 차례대로 모든 Themida 코드 블록을 실행시키면 어떤 API가 난독화 되었는지 파악할 수 있다.

3.3 Rebuilder

Rebuilder 모듈은 API Unwrapper 모듈에서 생성한 로그를 분석하고 재구성하는 역할을 한다. API Unwrapper 모듈에서 기록한 로그는 실제 포워딩되어 실행되는 API 정보가 기록되어 있다. 이 로그를 이용하여 IMPORT 섹션을 구성하면 원본 실행 파일의 IMPORT 섹션과 차이점이 존재한다. IMPORT 섹션은 포워딩 전 API를 기준으로 기록하기 때문에 API Unwrapper 모듈에서 생성한 로그를 수정한다. 이를 통해, 원본 실행 파일의 IMPORT 섹션과 동일한 IMPORT 섹션을 재구성한다.

IV. 실험 결과

Themida API Wrapping 기법이 적용된 프로그램을 대상으로 IMPORT 섹션을 재구성하여 제안 시스템에 대한 성능을 검증한다.

[그림 4]는 PEview를 사용하여 원본 프로그램의 Themida API Wrapping 기법 적용 전 IMPORT 섹션을 분석한 모습이다. [그림 5]는 [그림 4]의 프로그램을 API Wrapping 기법 적용 후 제안 시스템을 사용하여 역난독화한 결

```
kernel32.dll ¦ GetStartupInfoW ¦ 774a1550
kernel32.dll ¦ SetUnhandledExceptionFilter ¦ 774a1720
kernel32.dll ¦ UnhandledExceptionFilter ¦ 774b4f20
kernel32.dll ¦ InitializeSListHead ¦ 777985c0
vcruntime140d.dll ¦ __vcrt_GetModuleFileNameW ¦ 7217ae40
vcruntime140d.dll ¦ _except_handler4_common ¦ 72174c70
vcruntime140d.dll ¦ memset ¦ 721743d0
vcruntime140d.dll ¦ __current_exception_context ¦ 72176b90
vcruntime140d.dll ¦ __current_exception ¦ 72176b80
vcruntime140d.dll ¦ __vcrt_GetModuleHandleW ¦ 7217ae60
vcruntime140d.dll ¦ __vcrt_LoadLibraryExW ¦ 7217ae70
vcruntime140d.dll ¦ __std_type_info_destroy_list ¦ 72179df0
ucrtbased.dll ¦ strcat_s ¦ 720cfc30
ucrtbased.dll ¦ __stdio_common_vsprintf_s ¦ 720c6da0
ucrtbased.dll ¦ __p__commode ¦ 720aac10
ucrtbased.dll ¦ _initialize_onexit_table ¦ 72097420
```

[그림 5] Rebuilded API List.

과이다. [그림 4]와 비교했을 때 IMPORT 섹션 재구성 기능이 수행되어 난독화 된 프로그램이 사용하는 API 정보를 확인할 수 있다.

V. 결론

악성코드를 보호하기 위한 기법 중 난독화 기법이 차지하는 비중은 매우 크며, 난독화 된 악성코드는 계속해서 보고되고 있다. 그중, 상용 난독화 도구인 Themida에 의해 보호되는 악성코드의 사례가 존재한다. Themida API Wrapping 기법은 기존 연구를 통해 다양한 방식으로 역난독화 기법이 제시되었으나 Themida의 버전 업데이트로 최신 버전에 적용 불가능한 문제가 발생했다. 이에 본 논문에서는 최신 버전의 Themida API Wrapping 기법을 분석했고 IMPORT 섹션에 존재하는 모든 API에 대응하여 자동으로 역난독화하는 시스템을 제시했다.

본 논문의 제안 시스템을 통해 악성코드에 적용된 Themida API Wrapping 기능이 무효화 된 것을 확인할 수 있다. 본 논문의 제안 시스템을 사용하면 난독화 된 프로그램의 IMPORT 섹션에 존재하는 모든 API를 대상으로 역난독화하여 IMPORT 섹션을 복구 가능하다는 장점이 있다. 그러나 모든 Themida 코드 블록을 실행함으로써 역난독화가 완료되기까지 실행 시간이 증가한다는 단점을 가지고 있다.

추후 연구를 통해 다른 난독화 도구에 대해 분석하고 다양한 난독화 기법을 자동화하여 역난독화 할 수 있도록 연구를 진행할 예정이다.

[참고문헌]

[1] FireEye Mandiant, “M-Trends 2020”, https://content.fireeye.com/m-trends-kr/rpt-m-trends-2020-kr/, Accessed May 18, 2021.

[2] ESET, “Ousaban: Private photo collection hidden in a CABinet”, https://www.welivesecurity.com/2021/05/05/ousaban-private-photo-collection-hidden-cabinet/, Accessed May 19, 2021.

[3] ESET, “Lazarus supply-chain attack in South Korea”, https://www.welivesecurity.com/2020/11/16/lazarus-supply-chain-attack-south-korea/, Accessed May 19 2021.

[4] H. Triage, “Hatching Triage”, https://tria.ge/, Accessed April 14, 2021.

[5] J. H. Suk, J. Lee, H. Jin, I. S. Kim and D. H. Lee, "UnThemida: Commercial obfuscation technique analysis with a fully obfuscated program," Software: Practice and Experience, 48(12), pp. 2331-2349 2018.

[6] Y. Kang, M. C. Park and D. H. Lee, "Implementation of the Automated De-Obfuscation Tool to Restore Working Executable," Journal of the Korea Institute of Information Security & Cryptology, 27(4), pp. 785-802 2017.

[7] J. Lee, J. Han, M. Lee, J. Choi, H. Baek and S. Lee, "A study on API wrapping in Themida and unpacking technique," Journal of the Korea Institute of Information Security & Cryptology, 27(1), pp. 67-77 2017.

[8] J. Lee, B. Lee and S. Cho, "A Study on the Analysis Method to API Wrapping that Difficult to Normalize in the Latest Version of Themida," Journal of the Korea Institute of Information Security & Cryptology, 29(6), pp. 1375-1382 2019.

[9] Microsoft, “PE Format”, https://docs.microsoft.com/en-us/windows/win32/debug/pe-format#the-idata-section/, Accessed May 22, 2021).

[10] Oreans Technology, “Themida”, https://www.oreans.com/Themida.php/, Accessed May 22, 2021.

[11] Intel Corporation, “Pin - A Dynamic Binary Instrumentation Tool”, https://software.intel.com/content/www/us/en/develop/articles/pin-a-dynamic-binary-instrumentation-tool.html/, Accessed May 24, 2021.