패킹된 악성코드 분류 자동화를 위한 상용 패커 동향분석

조현준, 정혜선, 이지원, 김경백

e-mail : limdugmin2@naver.com, albaneo0724@gmail.com,
maum-@hanmail.net, kyungbaekkim@jnu.ac.kr

전남대학교 정보보안협동과정

Analysis of Commercial Packers for Automation of Packed Malware Classification

Hyeon-jun Jo, Hyeseon Jeong, Ji-won Lee, Kyungbaek Kim
Chonnam National University

요 약

인터넷상의 배포되고 있는 코드의 익명성 및 정보보호를 위해 난독화를 위한 패커가 개발되고 있다. 하지만 패커의 순기능을 악용하여 악성코드 탐지 및 분석을 어렵게 하기 위해 난독화 기능을 악성코드에 적용하는 사례가 늘고 있다. 특히, 난독화된 악성코드와 일반 코드가 섞여있을 경우 탐지에 더욱더 어려움을 주고 있다. 이에 따라 최근 상용 패킹들의 동향 분석과 상용 패킹(UPX, MPRESS, ASPACK, UPACK, THEMIDA, VM Protect) 들의 특징에 대한 연구를 수행하였다. 이러한 특징을 이용하여 악성코드 분류 자동화 가능 여부를 분석하고자 한다.

I. 서론

최근, 실행 파일(Portable Executable)을 압축, 소스 코드를 볼 수 없도록 하여 코드 보호하기 위하여 패커를 사용되고 있다. 하지만 악성코드와 결합하여 패커를 사용될 경우, 파일의 크기가 줄어들고 파일 내부 코드를 난독화되어 악성코드 탐지를 위한 정적 분석이 어려워지고 있다.

패킹된 파일은 실행 압축 상태로 존재한다. 즉, 패킹된 파일을 실행하기 위해서는 파일을 메모리에 로드하고 압축해제(unpacking)를 수행해 주어야 한다. 특히, 단순 PE 파일의 사이즈 압축(Compress) 방식이 아닌 코드 보호(Protector) 방법으로 패킹된 경우, 많은 Anti-Reversing 기술이 내재되어 있어 정적 분석을 위한 원본 PE 파일을 얻기 위한 unpackig 방법은 더욱 더 어려워진다.

이에 따라, 패킹된 파일을 실행하지 않고 악성코드 탐지를 위한 정적 분석을 수행하기 위해서는, 패커들의 특징, 알고리즘, 동작원리에 대한 조사가 필요하며, 이를 기반으로 정적 분석 자동화 연구가 필요하다.

II. 상용 패커 동향 및 특징 분석

2. 1 상용 패커 동향

Kisa 2017~2018 사이버 위협 동향 보고서에 따르면 2016년에 발생한 오퍼레이션 어나니머스 팬텀(Operation Anonymous Phantom) 악성코드는 2017년 더미다(Themida) 패커로 패킹된 국내 에너지 관련 연구소에서 발견되었으며 지속적으로 변형하여 활동 중이다.[1] 2018년 발생한 채굴형 악성코드는 VM Protect로 패킹되어 유포되었다.[2] 2019 ASEC 보고서에 따르면 오퍼레이션 레드 솔트 공격에 사용된 악성코드 중 다운로더(Downloader)는 UPX 패커로 패킹되어 사용되고 있다. 2020년 5월 이후 안랩 ASEC 악성코드 탐지 통계[3] 보고서에 의하면 전체 악성코드 비중의 34.7%를 차지하는 GuLoader 는 악성코드 탐지 진단을 우회하기 위해 패킹 되어 배포되었다.[3] 진단을 우회시키기 위하여 패킹되어 있으며 대부분 구글 드라이브를 악용하여 추가 악성코드를 다운로드 한다.

2. 2 상용 패커 특징

패커는 일반적인 PE파일 구조중 특정 섹션을 압축하고 이를 해제할 때 필요한 정보를 포함

한 섹션을 추가한다. 이러한 패커의 특징에 따라, 각 상용 패커들은 자기만의 섹션 구조를 가진다. 표1에서는 각 상용 패커의 섹션구조를 정리하였다.

표 1. 상용 패커 별 섹션 구조 정리

악성코드파일	packer	section
AveMariaRAT.exe	UPX	upx0 upx1 .rsrc
Trojan/Win32.Keylogger.exe	UPACK	PE, MZ 헤더부분 결합 NULL
	ASPACK	.aspack .adata
	MPress	.MPRESS1 .MPRESS2 .rsrc
	THEMIDA	.themida
	VMProtect	.fs50 .fs51

그림 1. 각 상용 패커의 섹션 상세 구조

① UPX

UPX1 섹션 영역은 압축된(Encoding) 내용과 디코딩 루틴이 포함되어 있다. UPX1에 있는 압축된 내용(encoding)을 디코딩 하여 UPX0 섹션에 넣는다. 전체 디코딩 루틴이 끝난 후 UPX0 섹션이 원래의 코드 섹션의 역할을 한다.

UPX는 리소스 부분을 압축하지 않기 때문에 .rsrc 섹션이 그대로 존재한다. 디코딩 과정에서 기본적으로 필요한 API들을 사용하기 위하여 Import Table이 필요하다. Import Table이 .rsrc 섹션에 추가되여 존재할 수 있다. .rsrc 섹션 이 존재하지 않는 경우에는 UPX2 섹션이 만들어져서 이곳에 들어가게 된다.

② UPACK

헤더 부분을 MZ 헤더(IMAGE_DOS_HEADER)와 PE 헤더(IMAGE_NT_HEADERS)를 교모하게 겹쳐 사용함으로써 복잡성을 증가시켜 분석을 어렵게 만드는 패커이다.[4]

UPACK으로 패킹된 파일은 LZMA decompression E8/E9 decompression, Import table rebuilding, OEP jumping 나뉜다. Unpackig 과정은 LZMA 압축 해제를 하고 Call/JMP 문 복구 그 후 Import Table을 복구한 후에 OEP로 진입한다.

③ ASPACK

단순 압축(Compressor) 지원과 코드 난독화를 위해 ASProtect에서 사용되고 있다. 섹션 부분은 .aspack과 .adata 섹션이 존재하며, .aspack 섹션은 섹션의 시작 주소를 디코딩 한 루틴, Import Table이 들어 있는 영역이다.

ASPack은 언패킹하기 위해서 메모리 할당이 필요함에 따라 VirtualAlloc() 통해 메모리를 할당하여 디코딩 수행 후 VirtualFree() 할당된 메모리를 해제하는 것을 반복한다.

④ MPress

압축(Compressor)만을 지원하며 손실성을 낮추고 향상성 높이기 위해 LZMA, LZMAT 압축방법을 사용하여 패킹을 수행한다. LZMA는 압축률이 매우 높지만 멀티 코어를 활용할 수 없다. LZMAT는 언패킹시 메모리가 필요하지 않아 패킹-언패킹을 보다 신속하게 수행할 수 있다. 이 두 압축방법의 구별을 위해 "force to use"옵션의 확인 필요하다.

섹션 부분은 .MPRESS1 .MPRESS2 가 존재하며 리소스 섹션이 존재할 경우 .rsrc 섹션도 존재한다. .MPRESS1 섹션에는 인코딩 된 데이터가 존재하며, .MPRESS2 섹션에는 Import Table, 뒤에 디코딩 루틴이 존재한다.

⑤ Themida

VMware, 디버거, 프로세스 모니터(ProcMon)

등의 분석을 방지하는 기능을 갖고 있다. 이중, ProcMon은 커널 컴포넌트를 가지고 있으며, ANTI Debugging 과 ANTI 관련 기능 분석을 어렵게 만든다.

많은 기능을 포함하고 있어 패킹된 파일은 커진다. 대다수 패커와 다르게 Themida 코드는 원본 프로그램이 실행하는 전체 시간 동안 계속 동작한다.

⑥VM Protect

메모의 데이터 무결성, API 리스트 보호, PE 파일을 압축, 디버거(ebugger) 와 가상환경(VirtualizationTools) 탐지하는 옵션을 가지고 있다. 악성코드를 VM Protect로 패킹할 경우 코드 보호 영역에 마커(Marker)를 삽입하여 해당 역을 자체 제작된 가상머신에서 실행되도록 코드를 변경하는 가상화 난독화를 사용할 경우 코드 분석을 하기위한 방법을 무력화 시킬 수 있다.

생성된 .fs50 영역은 압축된 내용과 디코딩 루틴이 포함된 영역이며 .fs51 API를 사용하기 위해 Import Table을 담고 있는 영역이다

표 2. Packer 시그니처 코드(signature)

UPX0 UPX1	{55 50 58 30 00 00 00} {55 50 58 31 00 00 00}
MPRESS1 MPRESS2	{60 E8 00 00 00 00 58 05 [2] 00 00 8B 30 03 F0 2B C0 8B FE 66 AD C1 E0 0C 8B C8 50 AD 2B C8 03 F1 8B C8 57 51 49 8A 44 39 06 88 04 31 75 F6}
aspack adata	{2E 61 73 70 61 63 6B 00} {2E 61 64 61 74 61 00 00}
themida	{ B8 00 00 00 00 60 0B C0 74 58 E8 00 00 00 00 58 05 43 00 00 00 80 38 E9 75 03 61 EB 35 E8 }
upack	{ 81 3A 00 00 00 02 00 00 00 00 }
fs50 fs51	{2E 76 6D 70 30 00 00 00} {2E 76 6D 70 31 00 00 00}

2. 3 패킹된 악성코드 분류 자동화 기법

패킹된 악성코드에 대한 보다 정확한 탐지를 위해서는 언패킹 과정이 필요하다. 이때, 패킹된 악성코드가 어떤 패커를 사용했는지를 정확히 분류해야, 해당 패커에 알맞은 언패킹 과정을 수행할 수 있다. 패커 분류의 한 방법으로 패커 고유 시그니처 코드를 활용하는 분류기법을 생각할 수 있다. 앞서 살펴본 바와 같이 하나의 패커는 고유의 섹션 구조를 가지고 있으며, 이를 통한 패커 고유의 시그니처 코드를 표 2와 같이 추출할 수 있다. 추출된 시그니처 코드를 활용해 Yara와 같은 패턴 매칭 툴을 이용하면, 패킹된 악성코드 분류 자동화를 구현할 수 있다.

하지만 시그니처 코드 기반 분류 기법은 새로운 패커에 대한 탐지가 불가능하다. 이를 보완하기 위해, 보다 다양한 패커 구조분석에 기반한 머신러닝 및 인공지능 기반 분류 자동화 기법에 대한 연구도 필요할 것으로 분석된다.

III. 결론 및 향후 연구

악성코드가 패킹된 파일은 악성코드 탐지 및 분석을 어렵게 하기 위한 난독화, 암호화 등을 적용한 패커 사용 빈도 횟수가 증가되고 있다. 이에 패킹된 악성코드 탐지를 위해서는 상용 패킹들의 동향 분석과 상용 패킹 들의 특징에 대한 파악이 중요하다. 패커 들의 고유한 시그니처를 Yara의 Ruleset으로 정의하여 패킹된 악성코드들을 패커 별로 분류하는 자동화 기법을 구현하였다.

향후, 머신러닝 또는 인공지능 기반 패커 분류 기법 및 패킹된 악성코드를 언패킹 하기 위한 자동화 기법을 연구하고자 한다.

Acknowledgements

“이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017R1A2B4012559). 이 논문은 정부(과학기술정토봉신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2019-0-01343, 융합보안핵심인재양성).”

[참고문헌]

[1] 한국인터넷진흥원 “2017 사이버 위협 동향 보고서” 2분기, 2017

[2] 한국인터넷진흥원 “2018 사이버 위협 동향 보고서” 1분기, 2018

[3] ASEC “주간 악성코드 통계(20200525~20200531)” 2020

[4] 한승원, 이상진 “악성코드 포렌식을 위한 패킹 파일 탐지에 관한 연구” 정보처리학회논문지C, 제16권, 제5호, 555-562, 2009

[5] YaraRules Project “http://github.com/Yara-Rules/rules”