Opcode와 API의 군집화와 유사도 분석을 활용한 랜섬웨어 탐지모델 연구

이계혁, 황민채, 구영인, 현동엽, 유동영
홍익대학교 컴퓨터정보통신공학과
gdsmsla@g.hongik.ac.k, hminchae@g.hongik.ac.kr,
ku1718@g.hongik.ac.kr, hyunfree0106@g.hongik.ac.kr, ydy@hongik.ac.kr

A Study on the Ransomware Detection Model Using the Clustering and Similarity Analysis of Opcode and API

Gye-Hyeok Lee, Min-Chae Hwang, Young-In Ku, Dong-Yeop Hyun,
Dong-Young Yoo
Department of Software and Communications Engineering, Hongik University

요 약

최근 코로나 19 팬더믹 이후 원격근무의 확대와 더불어 랜섬웨어 팬더믹이 심화하고 있다. 현재 안티바이러스 백신 업체들이 랜섬웨어에 대응하고자 노력하고 있지만, 기존의 파일 시그니처 기반 정적분석은 패킹의 다양화, 난독화, 변종 혹은 신종 랜섬웨어의 등장 앞에 무력화될 수 있고, 실제로 랜섬웨어의 피해 규모 지속 증가가 이를 설명한다. 본 논문에서는 기계학습을 기반으로 한 단일 분석만을 이용하여 탐지모델에 적용하는 것이 아닌 정적 분석 정보(.text Section Opcode)와 동적 분석 정보(Native API)를 추출하고 유사도를 바탕으로 연관성을 찾아 결합하여 기계학습에 적용하는 탐지모델을 제안한다.

1. 서론

최근 코로나 19 팬더믹 이후 원격근무의 확대와 더불어 랜섬웨어 팬더믹이 심화하고 있다. 캐나다 사이버 보안센터에 따르면 2021년 상반기 발생한 랜섬웨어 공격은 2020년 동기 대비 151% 증가했고, 지난해 1월부터 7월까지 미국 연방수사국(FBI)이 확인한 랜섬웨어 공격 신고는 2,084건에 달했으며, 전 세계 기업은 공격받은 데이터를 복구하기 위해 약 21억 6600만 원을 지출했다[1]. 현재 안티바이러스 백신 업체들이 랜섬웨어에 대응하고자 노력하고 있지만, 기존의 파일 시그니처 기반 정적분석은 패킹의 다양화, 난독화, 변종 혹은 신종 랜섬웨어의 등장 앞에 무력화될 수 있고, 실제로 앞서 서술한 랜섬웨어의 피해 규모 지속증가가 이를 설명한다. 이러한 문제점을 해결하기 위해 기계학습을 기반으로 한 랜섬웨어 동적 분석 연구가 활발히 진행되고 있다. 관련 연구에는 이규빈 외 2명은 랜섬웨어 동적 분석을 위한 시그니처 추출 및 선정 방법에 관해 연구하였다[2]. 본 논문에서는 단일 분석만을 이용하여 탐지모델에 적용하는 것이 아닌 정적 분석에서는 .text Section Opcode와 동적 분석에서는 Native API 정보를 추출하고 연관성을 찾아 결합하여 기계학습에 적용하는 탐지모델을 제안한다. 이는 단일 분석의 한계를 극복하고 각 분석 방법의 장점만을 끌어낼 수 있는 새로운 탐지모델로 활용될 수 있다.

2. 관련 연구

랜섬웨어 및 악성코드를 탐지하는 방식에는 대표적으로 정적 분석, 동적 분석 두 가지로 나눌 수 있다. 정적 분석은 악성코드를 실행하지 않고 파일 자체가 가지고 있는 내용을 통해 분석하는 방법이다. 관련 연구에는 박정빈 외 3명은 악성코드 분류를 위한 중요 연산부호 선택 후 그 유용성에 관해 연구하였다[3]. 정적 분석의 종류에는 기초적인 해시 정보 검증부터 고유 시그니처 분석, PE 헤더 분석, 파일의 구조 파악, 문자열 확인, Opcode(Operation Code) 분석 등이 있다. 동적 분석이란 악성코드를 실행 시켜 실제로 동작하는 환경을 구성해 분석하는

방법이며, 동적 분석의 종류에는 프로세스 모니터링, 파일 접근 및 변경 행위 탐지, 네트워크 모니터링, 시스템 로그 정보 등 각종 행동 패턴을 분석한다. 관련 연구에는 D. Sgnandarra 외 3명은 랜섬웨어의 동적 분석 결과를 특징정보로 선정 후, 다양한 기계학습 알고리즘을 통해 랜섬웨어를 분류하는 'ElderRan'방식을 제안하였다[4]. 덧붙여, 동적 분석의 대표적 분석 도구로 꼽히는 Cuckoo Sandbox에서 얻을 수 있는 분석 정보들은 [표 1]과 같다.

표 1. 정적 및 동적 분석 정보[5]

정적 분석 정보	동적 분석 정보
1. PE Import	1. Process Info
2. PE VersionInfo	2. API Info
3. PE Resource	3. Log Info
4. PE Imphash	4. File Info
5. PE Sections	5. Registry Info
6. PE Timestamp	6. Network Info
	7. Buffer Info

```
push    dword ptr [edi+0DCh]
push    dword ptr [edi+0D8h]
push    dword ptr [esp+54h+Time2+4]
push    dword ptr [esp+58h+Time2]
call    sub_4174DF
push    dword ptr [edi+214h]
mov     dword ptr [esp+60h+Time2], eax
push    dword ptr [edi+210h]
mov     dword ptr [esp+64h+Time2+4], edx
push    dword ptr [esp+64h+var_18+4]
push    dword ptr [esp+68h+var_18]
call    sub_4174DF
```

그림 1. Opcode 추출

Address	Ordinal	Name
0046B244		SetFileAttributesW
0046B248		GetFileSizeEx
0046B24C		Sleep
0046B250		DeleteFileW
0046B254		GetTickCount
0046B258		SetFilePointer
0046B25C		WriteFile
0046B260		CreateThread

그림 2. Windows API 추출

3. Opcode 및 API 활용 분석기법 설계

본 논문에서는 정적 분석의 많은 정보 유형 중 바이너리 상에서 추출이 가능한 Opcode(Operation Code)를 이용한다. Opcode 의 빈도수는 소프트웨어의 특징을 반영하는 데 이용될 수 있으므로[6] 이를 변수로 사용해 유사도를 측정하고 시각화하여 랜섬웨어와 정상 파일 탐지 모델에 이용하고자 한다.

동적 분석에서는 프로그램이 실제로 동작하는 과정에서 호출하는 API 정보를 이용하여 API의 호출 흐름을 파악하고 특정 API에 대한 호출 빈도를 시각화하여 행위가 유사한 샘플들을 묶어 그룹을 형성하고 해당 그룹의 API 출현 빈도를 유사도 계산에 활용해 랜섬웨어와 정상 파일 탐지 모델에 이용하고자 한다. 앞서 제안한 Opcode 의 빈도수 [그림.1]와 API 호출 빈도수 [그림.2]를 결합하는 방법은 각각 다음과 같다. Opcode 에서 가장 빈번하게 나타나는 10개 항목의 Opcode 를 선정하고 빈도수를 계산한다. 동적 분석에서도 얻을 수 있는 API 호출 정보에서 가장 빈번하게 나타나는 n개 항목의 API를 선정하고 빈도수를 계산한 뒤, 이를 각각 라벨링 하여 하나의 데이터로 결합하고 사용하여 정상 실행 파일들과 랜섬웨어 군집 사이에 유사도를 파악한다. 두 군집의 Opcode 추출 및 API 추출 차이점을 파악하여 랜섬웨어와 정상 실행 파일 간 제안하고자 하는 탐지 모델 학습의 중요한 지표로 활용된다.

4. 기계학습 기법 분석

4.1. 랜섬웨어 특징정보 선정 및 기법 활용

각 랜섬웨어, 정상 실행 파일에서 얻을 수 있는 Opcode 와 실제 동작 시 호출하는 API의 전체 정보는 무수히 많다. 하지만 얻어낸 모든 특징 정보를 선택하게 되면 불필요하고 전혀 무관한 특징 벡터를 사용하게 되어 모델 탐지율에 악영향을 끼친다. 무관한 특징 벡터 사용에 따른 악영향의 예시로는 "FF" 혹은 "00" 과 같이 Section 을 구분하고 조정(Align)하기 위해 사용되는 값들을 유사도 계산에 사용하기 때문에 이러한 잘못된 입력 잡음(Input Noise)으로 인해 탐지 시 그 정확도가 떨어지는 결과를 초래한다[7]. 따라서 이러한 문제를 개선하기 위하여 이 논문에서는 정적 Opcode 와 동적 API 데이터에서도 유의미한 특징이 있을 것으로 추정되는 부분만 추출한다. 따라서 정적 Opcode 데이터에서 사용할 PE파일 Section은 .text Section이다. 대표적인 PE 파일의 각 Section은 [표 2]와 같다.

표 2. 대표적인 PE파일의 각 구역 및 특징

구역	특징
.text	실행 파일의 실제 Source Code
.data	정적변수, 전역변수 등 저장
.rdata	읽기 전용, 정의된 문자열 등
.bss	초기화되지 않은 전역변수 저장
.rsrc	윈도우 Application Resource Data
.idata	Import Table 저장
.edata	Export Table 저장

이 논문에서 제안하는 Opcode 특징정보 필터링 방식은 다음과 같다. 앞서 서술한 .text Section, 즉, 실행파일의 실제 Code가 저장되어 있는 부분을 추출하여 각 랜섬웨어 악성 파일과 정상 실행 파일의 .text Section Opcode 빈도수를 구하고 이를 서로 비교한다. 랜섬웨어 악성 파일의 경우 정상 실행 파일 대비 파일 크기가 작고 은닉하는 특성을 가지기 때문에 전체적인 빈도수가 정상 실행파일 보다 적은 빈도수를 나타낸다. 이에 따라 K-means Clustering을 거쳐 Opcode 빈도수를 기반으로 군집을 형성할 수 있다. 동적 API 데이터에서도 유의미한 분석 정보만을 추출하기 위해 우선 랜섬웨어 중 Gandcrab의 파일 암호화 악성 행위 동작 시 발생하는 API 호출 순서를 정립할 필요가 있다. 실제 Gandcrab 랜섬웨어의 API 호출은 [그림.3]과 같이 나타난다.

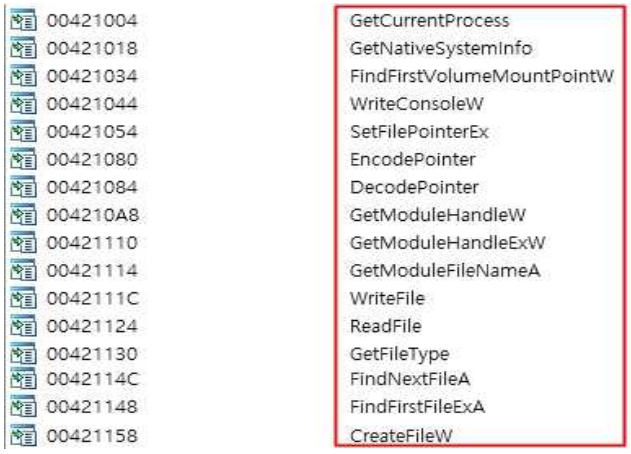


그림 3. Gandcrab 랜섬웨어의 호출 API

이 논문에서 API 특징정보 필터링 방식은 다음과 같다. [그림.3]에서 확인할 수 있는 API는 모두 커널 모드(Kernel Mode) 즉, Windows API 가 아닌 Native API 에 속한다. 이유는 랜섬웨어의 주된 악성행위인 파일의 암호화에 있다. 랜섬웨어가 동작하는 일반적인 방식은 먼저 대상 파일을 찾고 해당 파일을 열어 내용을 읽는다. 그 후 파일에 암호화 된 해시를 이용하여 데이터를 기록하고 다음 파일로 넘어가는 순서를 취한다. 따라서 제안하는 탐지모델에서는 Native API 로 범위를 구체화하여 실제 유의미한 특징 정보 필터링을 거치게 된다. 랜섬웨어 악성 파일과 정상 실행 파일의 Native API 호출 빈도수를 각각 구하고 이를 서로 비교하여 상위 빈도수를 가지는 Feature 를 선택한다. Feature 선택 후 각 Opcode, API 빈도수에 따른 특징 벡터를 활용하여 유사도를 파악하기 위해 유사도 측정 기법을 이용한다. Cosine Similarity(코사인 유사도)는 데이터 마이닝의 거리 측정 알고리즘 중 하나로 주어진 분석을 통한 두 특징 벡터의 벡터 스칼라 곱을 통해 벡터 간의 거리를 계산한다. 덧붙여, 벡터의 크기를 고려하고 싶지 않을 때 용이하게 사용된다. 두 벡터가 이루는 각이 작을수록 유사도가 높은 것이고, 각이 클수록 유사도가 작다고 생각하여 사용 결괏값으로는 0~1 사잇값이 나온다. 결괏값이 1이 나오는 경우는 두 벡터 간의 빈도수가 같으며, 유사하다고 결론 내릴 수 있다. 본 논문에서 사용되는 Cosine Similarity 계산식은 아래 수식[1]과 같다.

$$\text{수식 } 1 : CS = \cos(\theta) = \frac{A * B}{|A| \ |B|}$$

위와 같이 Cosine Similarity 계산방법을 통해서 나온 랜섬웨어의 특징정보들과 Cuckoo Sandbox에서 제공하는 악성코드 분석 보고서의 값들을 활용하여 랜섬웨어와 정상 실행 파일 간의 비교를 통해 랜섬웨어의 특징정보를 최종 선정한다[8].

4.2. 기계학습을 적용한 랜섬웨어 탐지 방법

악성코드 탐지시스템은 일반적으로 특징 추출(Feature extraction)과 분류(Classification) 또는 집화 (Clustering)의 2단계 처리 과정을 거친다[9].

본 논문에서는 특징추출 기법의 검증을 위해 K-Means Clustering을 사용한다. 분류를 위해 사용되는 비지도 학습 및 분할 접근의 대표적 기법이자 숫자 속성 데이터를 군집화하는 데 잘 알려진 학습 방법이다. 따라서 정상 실행 파일과 랜섬웨어 간 .text Section Opcode 항목 빈도수와 Native API 호출 빈도수를 각각 그룹화하고 시각화함에 따라 그룹별 유의미한 차이를 도출할 수 있다. 특징추출 후 기계학습에는 Random Forest, SVM을 이용한다. Random Forest 는 분류, 회귀 분석 등에서 사용되는 앙상블 학습 방법의 일종으로, 훈련 과정에서 구성한 다수의 결정 트리들을 조합하여 분류하거나 평균 예측값을 출력하는 방식이다. 데이터의 특성값 단위로 계층분류를 하므로, 정상 실행 파일과 랜섬웨어 간의 편차가 크고 특징적인 API 호출정보를 찾는데 용이하다. SVM은 분류와 회귀분석을 위해 사용되는 지도학습 방법의 하나며 데이터를 두 개의 클래스로 분류하여 판단할 수 있게 하는 비확률적 이진 선형 분류 모델을 생성한다. 따라서 .text Section의 Opcode와 Native API 빈도수의 군집을 바탕으로 하여 SVM 모델에 학습 시 유사도가 데이터가 0에 근접하면 랜섬웨어로, 1에 근접하면 정상

실행 파일로 분류하는데 유용하게 사용할 수 있다.

5. 실험 방법 및 탐지모델 설계

5.1 실험 방법 및 데이터 세트 구축

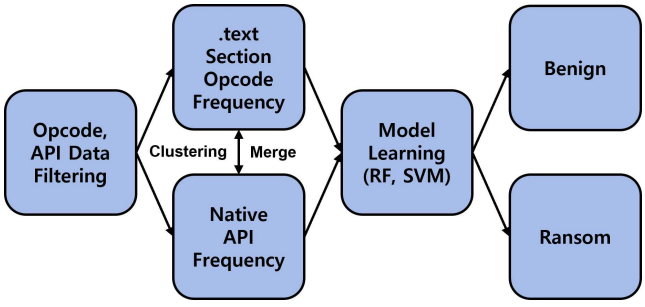
제안한 모델 [그림.4]을 사용해 모델 학습에 활용하지 않은 랜섬웨어와 정상 실행 파일을 Test Set으로 구성하고 앞서 서술한 세 개의 알고리즘을 사용해 각 랜섬웨어 탐지율을 분석하여 Voting 앙상블 기법을 적용한 뒤의 결과도 비교분석 한다. 랜섬웨어 실행 파일은 BODMAS 데이터 세트와 직접 수집한 랜섬웨어 파일들을 검증하여 총 500개의 랜섬웨어 실행 파일 데이터 세트를 구축하였으며, 정상 실행 파일도 신뢰할 수 있는 실행 파일을 실험에 이용하기 위해, 정상 실행 파일의 SHA-256 해시값을 Crawling을 통해 수집하고 VirusTotal 분석 결과와 함께 교차 검증하여 총 500개의 정상 실행 파일 데이터 세트를 구축했다. 랜섬웨어의 데이터 세트 분류는 [표.2]와 같다.

표 2. 랜섬웨어 유형 분류 정보[10]

종류	특징	확장자
Gandcrab	다양한 변종	CRAB
Wannacry	SMB 취약 전파	WNCRYT 등
Teslacrypt	AES 암호화	eoc, ezz 등
Cerber	음성 메시지	oerber, 무작위
Locky	VSC 삭제	locky, lukius 등

5.2 탐지모델 제안

그림 4. 제안하는 탐지모델



6. 결론 및 향후 연구

본 논문에서 제안하는 랜섬웨어 탐지모델은 정적 분석과 동적 분석의 결합으로 기존 단일 분석의 한계[11]를 보완하고, 그 분석 정보에서도 유의미한 특징이 있을 것으로 추정되는 .text Section Opcode와 호출하는 Native API로 구체화하여 특정 랜섬웨어[표.2]를 탐지하는데 중점을 둔다. 탐지모델 학습에 사용하는 데이터 세트도 타 연구 대비 최신의 샘플을 직접 수집하여 이용한다. 이는 향후 진행하게 될 기계학습 기반 실제 모델 구현에서 기존 단일 분석 방법 대비 탐지율 상승을 긍정적으로 기대해 볼 수 있고 각각의 알고리즘을 앙상블 기법 중 Voting 기법을 통해 여러 모델을 이용하여 데이터를 학습하고, 각 모델의 예측 평균을 구해 최적의 결과를 도출할 수 있도록 적용한다. 하지만 동적 분석기법을 우회하는 방법(Cuckoo Sandbox의 가상환경 감지)을 사용하는 랜섬웨어의 경우 본 논문에서 제안한 모델을 적용하지 못한다는 단점이 있다. 따라서 향후에는 안티 가상머신(Anti-VM) 우회에 대한 연구가 필요할 것이다. 덧붙여, 본 논문을 바탕으로 향후 랜섬웨어 탐지가 가능한 자동화된 시스템 또는 탐지 시스템 등 다양하고 활발한 연구 활동이 가능할 것으로 기대한다.

참고문헌

[1] 유혜정, "코로나發 원격업무 확대…'랜섬웨어 팬데믹'도 극성", 헤럴드경제, 2022

[2] 이규빈 외 2명, "랜섬웨어 동적 분석을 위한 시그니처 추출 및 선정 방법", 한양대학교, 2017

[3] 박정빈 외 3명, "악성코드 분류를 위한 중요 연산부호 선택 및 그 유용성에 관한 연구", 한양대학교, 2015

[4] Daniele Sgandurra 외 3명, "Automated Dynamic Analysis of Ransomware : Benefits, Limitations and use for Detection", 2016

[5] 김수정 외 5명, "Cuckoo Sandbox 와 Yara Rule 을 이용한 악성파일 감지", 한국정보과학회, 2018

[6] 원성민 외 2명, "Malware classification using statistical techniques", 이화여자대학교, 2017

[7] 김지원 외 2명, "하이브리드 분석을 통한 머신러닝 기반의 랜섬웨어 탐지 모델", 보안공학연구논문지, 2017

[8] 옥정윤, "정적 분석 정보와 동적 분석 정보를 활용한 랜섬웨어 탐지 모델 연구", 한양대학교 대학원, 2019

[9] 안태현, "멀웨어 탐지를 위한 특징 추출 설계", 을지대학교 대학원, 2020

[10] KISA, "2021 Ransomware Special Report"

[11] 최도현, "랜섬웨어 탐지를 위한 그래프 데이터베이스 설계 및 구현", 숭실대학교 대학원, 2021