KIPS Trans. Comp. and Comm. Sys.
Vol.11, No.10 pp.363~372
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)

https://doi.org/10.3745/KTCCS.2022.11.10.363

A Study on the Cerber-Type Ransomware Detection Model Using Opcode and API Frequency and Correlation Coefficient

Gye-Hyeok Lee† · Min-Chae Hwang† · Dong-Yeop Hyun† · Young-In Ku† · Dong-Young Yoo††

ABSTRACT

Since the recent COVID-19 Pandemic, the ransomware fandom has intensified along with the expansion of remote work. Currently, anti-virus vaccine companies are trying to respond to ransomware, but traditional file signature-based static analysis can be neutralized in the face of diversification, obfuscation, variants, or the emergence of new ransomware. Various studies are being conducted for such ransomware detection, and detection studies using signature-based static analysis and behavior-based dynamic analysis can be seen as the main research type at present. In this paper, the frequency of ".text Section" Opcode and the Native API used in practice was extracted, and the association between feature information selected using K-means Clustering algorithm, Cosine Similarity, and Pearson correlation coefficient was analyzed. In addition, Through experiments to classify and detect worms among other malware types and Cerber-type ransomware, it was verified that the selected feature information was specialized in detecting specific ransomware (Cerber). As a result of combining the finally selected feature information through the above verification and applying it to machine learning and performing hyper parameter optimization, the detection rate was up to 93.3%.

Keywords : Ransomware, Cerber, Opcode, API, Malware, Machine-Learning, Detection

Opcode와 API의 빈도수와 상관계수를 활용한 Cerber형 랜섬웨어 탐지모델에 관한 연구

이 계 혁† · 황 민 채† · 현 동 엽† · 구 영 인† · 유 동 영††

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 정적 분석은 패킹의 다양화, 난독화, 변종 혹은 신종 랜섬웨어의 등장 앞에 무력화될 수 있다. 이러한 랜섬웨어 탐지를 위한 다양한 연구가 진행되고 있으며, 시그니처 기반 정적 분석의 탐지 방법과 행위기반의 동적 분석을 이용한 탐지 연구가 현재 주된 연구유형이라고 볼 수 있다. 본 논문에서는 단일 분석만을 이용하여 탐지모델에 적용하는 것이 아닌 ".text Section" Opcode와 실제 사용하는 Native API의 빈도수를 추출하고 K-means Clustering 알고리즘, 코사인 유사도, 피어슨 상관계수를 이용하여 선정한 특징정보들 사이의 연관성을 분석하였다. 또한, 타 악성코드 유형 중 웜과 Cerber형 랜섬웨어를 분류, 탐지하는 실험을 통해, 선정한 특징정보가 특정 랜섬웨어(Cerber)를 탐지하는 데 특화된 정보임을 검증하였다. 위와 같은 검증을 통해 최종 선정된 특징정보들을 결합하여 기계학습에 적용하여, 최적화 이후 정확도 93.3% 등의 탐지율을 나타내었다.

키워드 : 랜섬웨어, 케르베르, Opcode, API, 악성코드, 기계학습, 탐지

1. 서 론

신종 코로나19 팬더믹의 기승으로 정보 통신 분야를 비롯해 어느 분야를 막론하고 많은 지각변동이 일어났다. 비대면 학습 플랫폼의 증가, 원격근무 확대, 원격진료 확대에 따른 디지털 헬스케어의 확산 등 전반적인 디지털 전환 가속화의 결과를 낳았다. 하지만 이러한 환경 변화의 결과와 더불어 랜섬웨어 공격 또한 매년 꾸준히 증가하고 있다. 그 결과로 랜섬웨어 공격에 대한 관심 또한 꾸준히 증가하고 있다. 전 세계 기준 2022년 1분기 랜섬웨어 관련 시간 흐름에 따른 관심도는 평균 64.7%로 2021년 4분기 평균 17.9%에 비해 약 3.6배 증가하였다. 수많은 랜섬웨어 공격 중에서도 기업을 대상으로 파일 암호화와 데이터 탈취를 동시에 수행하는 랜섬웨어 공격이 증가하고 있다. 이는 랜섬웨어 공격의 주요 목표 중 하나인 금전 탈취에 있다. 개인에 대한 랜섬웨어 공격

※ 이 논문은 홍익대학교의 '지역특화형 스마트시티 전문대학원 구축 사업'의 지원을 받아 수행된 결과임.
※ 이 논문은 2021년 한국정보처리학회 ACK 2021의 우수논문으로 "Opcode와 API의 군집화와 유사도 분석을 활용한 랜섬웨어 탐지모델 연구"의 제목으로 발표된 논문을 확장한 것임.
† 비 회 원 : 홍익대학교 소프트웨어융합과 학사과정
†† 종신회원 : 홍익대학교 소프트웨어융합과 부교수
Manuscript Received : July 29, 2022
Accepted : August 16, 2022
* Corresponding Author : Dong-Young Yoo(ydy@hongik.ac.kr)

으로 얻는 부당수익이 크지 않다는 공격그룹들의 판단하에 기업을 표적으로 한 랜섬웨어 공격들이 무분별하게 발생하고 있다. 2022년 1월, 공격자는 원격 명령 실행 취약점(CVE-2021-22986)을 악용한 프랑스 법무부 컴퓨터 시스템에 랜섬웨어 공격을 가해 데이터를 탈취했다. 2022년 2월에는 공격자가 영국의 식료품 생산기업 KP Snacks를 랜섬웨어 공격의 표적으로 삼고 IT 시스템을 마비시켜 식료품 배송 지연, 공급부족 문제 등을 야기시켰다. 덧붙여, 랜섬웨어 공격의 증가는 숫자로도 확연히 나타난다. 컴퓨터 보안 전문 기업인 이스트시큐리티(ESTsecurity)에서 탐지한 2022년 1분기 랜섬웨어 수는 약 18만 건으로, 각 59,688건, 54,997건, 63,047건을 기록했으며, 지난 2021년 4분기 대비 약 14,500건 증가했다[1].

이러한 사태에 발맞추어 2022년 랜섬웨어의 피해를 조금이라도 줄이고자 미국, 영국, 호주의 공동 사이버보안 권고문이 발표되었고, 각종 안티바이러스 백신 업체들이 랜섬웨어의 악의적인 행위가 발생하기 전에 탐지 및 대응하고자 노력하고 있지만, 안티바이러스 백신의 경우 상당수는 시그니처(Signature) 기반의 패턴탐지가 주된 동작 방식이며, 이에 따른 한계로 신종, 변종 등 기존 데이터베이스(DB)에 식별되지 않는 랜섬웨어 탐지에는 적합하지 않다[2]. 앞에 연구에서는 시그니처(Signature) 기반 패턴탐지의 랜섬웨어 탐지모델이 아닌 행위기반 정보결합을 사용한 Cerber 랜섬웨어 탐지모델을 제안하였다[3].

이에 본 논문에서는 Cerber 랜섬웨어 실행 파일의 “.text Section” Opcode 항목을 빈도수 기반으로 선정하고, 실제 Cerber 랜섬웨어가 악의적인 동작을 행할 때 사용하는 Native API 항목을 선정하고 추출하여 사용한다. 앞서 서술한 두 정보를 각 군집화하고 유사도 검증을 통해 유의미한 특징정보로 사용할 수 있음을 검증하고 이를 하나의 데이터 세트로 결합하여 기계학습의 특징 정보(Feature)로 이용하는 Cerber 랜섬웨어 탐지모델을 제안하고 검증한다.

2. 관련 연구

2.1 관련 연구 분석

Deepti Vidyarthi[4]의 연구에서는 랜섬웨어의 식별을 위해 랜섬웨어의 전체적인 작동방식을 연구하고 랜섬웨어 파일의 고유한 속성에서 얻을 수 있는 정보를 이용한 분류 기법을 제안한다. 해당 연구에서 사용한 방법은 PE 파일의 정적분석을 통해 PE Header에 정의된 속성 값을 매개변수로 사용했다. 다만, Deepti Vidyarthi이 제안한 논문의 경우, PE Header 단일 분석만 사용하였기에 앞서 서술한 시그니처(Signature) 기반의 패턴탐지 한계점을 극복하진 못했다.

İlker Kara[5]의 연구에서는 특정 랜섬웨어 Cerber계열의 동작 방식과 참조하는 레지스트리 파일, 경로 탐색 및 C&C 서버와 통신하는 과정에서의 네트워킹 행위 분석을 통해 특정 랜섬웨어를 분석했다. 다만 해당 논문에서는 단일 랜섬웨어 파일에 대한 행위 분석만 진행했을 뿐 기계학습을 이용한 직접 분류에 대한 연구는 진행되지 않았다.

Philip O'Kane[6]의 연구에서는 지지기반 기계학습을 이용한 악성 실행 파일과 정상 실행 파일 간의 분류를 진행했다. 해당 연구에서는 악성 실행 파일과 정상 실행 파일의 연산부호(Opcode)를 민감도(Opcode Sensitivity) 기반으로 나타내어 분류하는 데 사용했다. 다만, O’Kane 등이 제안한 방법의 경우 Opcode를 추출하는 과정에서 동적 분석을 이용했지만, 본 논문에서는 Opcode 추출 시 정적분석을 기반으로 한다. 덧붙여, O'Kane이 제안한 Opcode 동적 분석의 경우, 실제 프로그램이 동작하는 과정에서 접근하지 않는 Opcode의 누락, 반복문 수행 등 Opcode 항목의 빈도수를 도출하는 과정에서 정보가 왜곡되거나 편향될 수 있는 문제점을 지닌다. 본 논문에서는 각 실행 파일 별 Opcode를 추출하고 정적분석의 특징정보로만 사용하며, 정보 왜곡을 방지하기 위해 “.text Section”의 Opcode로 범위를 특정한 뒤 탐지모델에 이용한다.

M. Zhang[7]의 연구에서는 시그니처(Signature) 기반의 패턴 탐지 방식이 악성코드 바이트코드(bytecode)의 변형으로 쉽게 우회될 수 있음을 입증하였고, 이를 보완하고자 실제 악성코드가 동작 시 호출하는 API에 대한 의존성 그래프를 추출하고 이에 가중치를 부여하여 악성코드와 정상 파일을 분류하는 연구를 수행하였다.

Mahbub Hasan[8]의 연구에서는 기계학습을 이용한 악성 프로그램과 정상 프로그램의 분류를 진행했는데, 위 연구에서 이용한 정보는 다음과 같다. 실제 호출하는 함수의 길이 기반 빈도(FLF)와 프로그램 내부에서 인쇄 가능한 문자열 정보(PSI)를 결합하여 교차검증을 통한 분류 모델을 구현했다. 본 논문에서 제안 및 검증하고자 하는 랜섬웨어 탐지모델은 먼저 정적 분석과 동적 분석의 결합으로 기존 단일분석의 한계를 보완하고, 정적 분석정보에는 실행 파일의 실제 Code가 동작하는 범위로 특정한 뒤 각 정상 실행 파일과 특정 랜섬웨어 실행 파일의 Opcode 항목별 빈도수를 특징정보로 사용하며, 동적 분석정보에서는 실제 호출하는 Native API의 항목별 빈도수로 구체화한다. 그 후, 각 분석정보를 기반으로 클러스터링을 통한 각 파일 유형별 군집을 형성하고 유사도 검증을 진행한다. 마지막에는 두 분석정보를 결합한 하나의 데이터 세트를 도출하여 랜섬웨어 탐지모델 기계학습에 사용함에 따라 Cerber 랜섬웨어의 행위기반 탐지에 중점을 둔다.

2.2 Cerber형 랜섬웨어와 Locker형 랜섬웨어 비교 분석

본 연구에서 데이터 세트로 선정하고 검증한 Cerber 랜섬웨어와 비슷한 시기에 널리 유포되어 많은 피해를 끼쳤던 Locky 랜섬웨어를 분석했다. 분석 항목은 랜섬웨어 감염 사실 통보방식, 주요 유포경로, 파일 내 RSA 공개키 저장 여부, VSC(Volume Shadow Copy) 삭제 여부, UAC(사용자 계정 컨트롤) 우회 여부를 비교 분석하였다. 먼저 감염 사실 통보

Table 1. Comparison of Cerber and Locky Ransomware

Feature	Cerber Family	Locky Family
Infection Notification	Text-To-Speech,Background	Create Ransom Notes, Background
Main Distribution Path	Malvertising, Vulnerability Exploit, etc	E-Mail Attachments, etc
RSA Public key Whether to Save	O	X
Whether to Delete VSC	O	O
Whether to Bypass UAC	O	X

의 경우 Cerber 랜섬웨어는 Locky 랜섬웨어와 달리 Text-to-Speech, 즉 VBS 스크립트 파일을 이용하여 음성으로 감염 사실을 전파한다. 두 번째는 주요 유포경로이다. Locky 랜섬웨어의 경우 대부분의 유포 방식은 이메일 첨부파일을 이용한 VBS 스크립트 파일 실행이다. Cerber 랜섬웨어의 유포경로 또한 이메일의 첨부파일 기능을 이용하지만 주된 유포경로는 웹 사이트 상 노출되는 온라인 광고 서비스를 이용하여 악성코드를 유포하는 형태의 멀버타이징(Malvertising)[9]과 시스템 취약점 공격(Exploit)으로 이루어지는 경우가 대다수이다. 세 번째로 Cerber 랜섬웨어는 RSA 공개키를 PE파일 리소스(Resources) 영역 안에 저장하고 암호화 대상, C&C서버 정보 등도 함께 저장한다. 이는 Locky 랜섬웨어과 비교 시 가장 큰 차이점으로 꼽을 수 있다. 네 번째는 VSC(Volume Shadow Copy) 삭제 여부이다. 두 랜섬웨어 모두 VSC를 삭제하는 과정을 거치며 이는 윈도우 시스템 복구를 막는다. 다섯 번째는 UAC(사용자 계정 컨트롤)의 우회 여부이다. Cerber 랜섬웨어는 Locky 랜섬웨어와 달리 파일 암호화 전 먼저 레지스트리 값을 확인하여 피해자 PC 운영체제에 UAC 기능이 존재하는지 확인한다. 그 후, UAC 기능의 존재가 파악되면 UAC 우회가 가능한 PE 파일을 window 시스템 폴더 경로에서 찾아 임시 폴더에 복사한 후, 프로그램 코드 진입 시 Cerber 랜섬웨어를 실행시키는 코드를 삽입하여 UAC 기능을 우회한다. 또한, 동작 중인 프로세스에 특정 API를 호출하는 코드를 삽입하여 모든 프로세스에 대한 접근 권한을 획득한다[10]. Table 1은 위 서술한 분석 결과를 표로 작성하였다.

3. Opcode 항목별 빈도수 분석

3.1 Opcode 항목 선정

Opcode(Operation Code)는 컴퓨터가 수행할 명령어를 나타내는 부호를 말한다. Opcode 빈도수는 소프트웨어의 특징을 반영하는 데 이용될 수 있으므로 본 논문에서는 이를 독립변수로 사용해 코사인 유사도(Cosine Similarity)를 측정하고 이를 클러스터링 기법을 통해 시각화하여 본 논문에서 제안하는 Cerber 랜섬웨어 탐지모델에 사용한다[2]. 분석에 앞서 Opcode를 추출하기 위해서는 어셈블리어로 변환하는 Disassemble 과정이 필요하다. 따라서 실행 파일의 바이너리 정보를 확인할 수 있도록 디스 어셈블러(Disassembler)인 Objdump를 사용하였다. 먼저, Objdump로 실행 파일을 Disassmble 하고 "-d-j .text" 옵션을 통해 파일의 실제 동작하는 코드(Code)구역인 ".text Section"으로 특정하여 Opcode를 추출하였다. 이 정보를 이용하여 정상 실행 파일과 Cerber 랜섬웨어 실행 파일의 항목별 Opcode 빈도수를 도출하였으며 Opcode 분석 결과 각 실행 파일의 Opcode 항목별 비율은 Fig. 1과 같이 구성되어 있다.

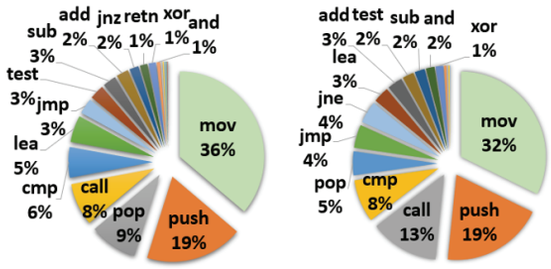

Fig. 1. Most Frequent 13 Opcode for Benign and Cerber Ransomware

분석 결과 정상 실행 파일과 Cerber 랜섬웨어 실행 파일의 Opcode 상위 5개 항목(mov, push, pop, call, cmp)이 같다는 결과를 도출하여 위 5개 항목과 그 외 5개의 항목(jmp, lea, sub, add, test)을 더하여 특징정보(Feature)로 선정하였다. 최종 선정한 상위 10개 opcode 항목의 빈도수를 계산하여 도출한 랜섬웨어 실행 파일과 정상 실행 파일 간의 비교 결과는 다음 Fig. 2와 같다.

Fig. 2에서 확인할 수 있는 정보는 다음과 같다. Cerber 랜섬웨어 실행 파일의 경우, 정상 실행 파일 대비 파일의 크기가 작고 악의적인 행위를 수행하기 때문에 은닉하는 특성을 가진다. 결과적으로 전체적인 Cerber 랜섬웨어 실행 파일의 Opcode 빈도수는 정상 실행 파일의 Opcode 빈도수보다 낮은 결과를 보인다.

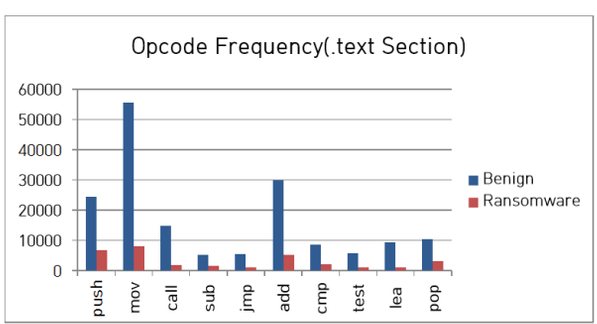

Fig. 2. Differences in Opcode Frequency Between Benign and Cerber Ransomware

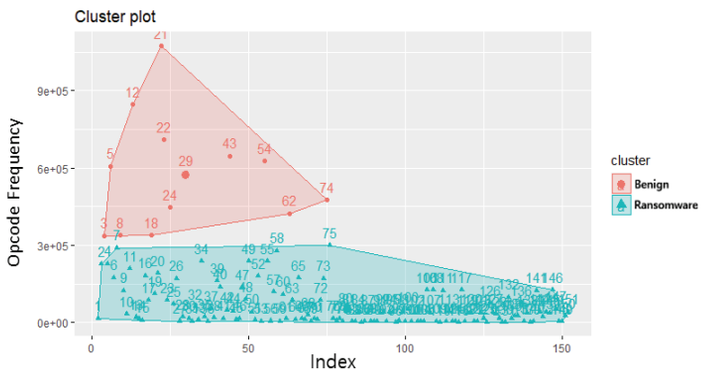

Fig. 3. Result of K-means Clustering

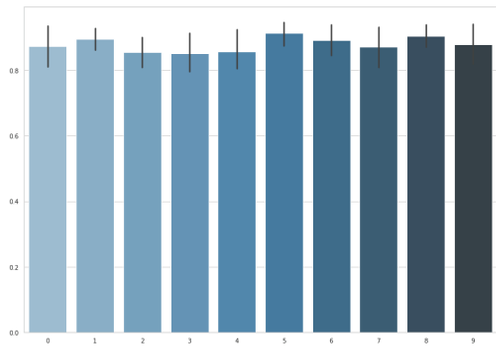

Fig. 4. Cosine Similarity of Cerber Ransomware Opcode

3.2 Opcode 항목 클러스터링 및 군집화

Fig. 3은 앞서 선택한 총 10개의 Opcode 항목 빈도수에 대한 K-means Clustering 알고리즘을 이용해 나타낸 그래프이다. K값의 경우, K=2로 설정하여 두 군집으로 나타내었다. 클러스터링을 통한 군집화 결과, Opcode 항목의 빈도수가 전반적으로 낮은 랜섬웨어 실행 파일의 특성을 이용하여 정상 실행 파일과 랜섬웨어 실행 파일을 분류하기 위한 기준으로 사용할 수 있다.

3.3 Opcode 항목 코사인 유사도 측정

Fig. 4에서 확인할 수 있는 정보는 다음과 같다. 앞서 3.1에서 선정한 총 10개의 Opcode 항목에 대해 각 Cerber 랜섬웨어 실행 파일별 코사인 유사도를 측정한 결과, 각 Cerber 랜섬웨어 실행 파일의 Opcode 빈도수 기반 벡터 간 유사도는 0.8에서 0.9 사이의 값을 나타낸다. 이는 앞서 선정한 총 10개의 Opcode 항목으로 실행 파일의 유형을 탐지할 수 있는 독립변수로 사용될 수 있다는 정당성을 가질 수 있다.

4. 호출하는 API 항목별 빈도수 분석

4.1 Native API 항목 선정

의심스러운 악성 파일이 존재할 경우, 현재 사용하는 컴퓨터 환경에서 파일을 실행시키는 행위는 매우 위험하다. 따라서, 게스트(Guest) PC-가상환경(Virtual Machine)을 이용한 격리된 환경에서 파일을 실행하여 분석하고 파일에 대한 정보를 획득하는 Sandboxing 기술을 적용한다. 본 논문에서는 Cuckoo Sandbox를 이용하여 각 실행 파일을 분석한다. 분석이 완료되면 .json 형식의 결과보고서가 생성되는데 해당 파일을 이용하여 랜섬웨어가 실제로 동작하는 과정에서 호출하는 API를 후킹(Hooking) 하여 실제 동작 시 호출하는 API의 추출이 가능하다. 그중 Native API만을 추출하여 호출 빈도수를 비교한 결과, 상위 10개로 Native API 항목을 선정했다. Native API로 범위를 좁힌 이유는 관리자 권한에서 수행되는 System Call의 일종으로 관리자 권한을 획득하여 공격하는 다양한 종류의 악성코드들을 탐지하는 데에 사용되기 때문이다[11]. 따라서 Cerber 랜섬웨어와 정상파일 사이에 유의미한 차이를 나타내는 상위 Native API를 선정하기 위해 Cerber 랜섬웨어 실행 파일과 정상 실행 파일 간의 출현 빈도수를 도출한 다음, 빈도수를 비교하여 큰 차이를 보이는 총 10개의 Native API들을 선정했다. 선정된 Native API 항목은 Table 2와 같다.

Fig. 5에서 확연한 차이를 나타내는 FindFirstFile, SearchPathW API의 경우[12], 암호화 대상 파일 및 디렉터리(Directory)의 경로를 아스키코드(ASCII)로 전달하기 때문에 일부 백신에서 파일이 변조되는 것을 감지하여 해당 프로세스를 종료하는 방식으로 랜섬웨어 대응 시 적용하거나, 지정된 경로에서 파일을 검색하여 암호화 대상 파일을 탐색하

Table 2. Top Native APIs Run by Cerber Ransomware

Native API	Feature
CryptEncrypt	Encrypts Data. The Algorithm Used to Encrypt the Data
CreateThread	Creates a Thread to Execute within the Virtual Address Space
FindFirstFile	Searches a Directory for a File or Subdirectory with a Name
FindResourceExW	Determines the Location of the Resource
FindResourceEx	Determines the Location of the Resource
GetFileAttributes	Retrieves File System Attributes for a Specified File
SearchPathW	Searches for a Specified File in a Specified Path
SetFilePointer	Moves the File Pointer of the Specified File
SetFilePointerEx	Moves the File Pointer of the Specified File
SetFileAttributes	Sets the Attributes for a File or Directory

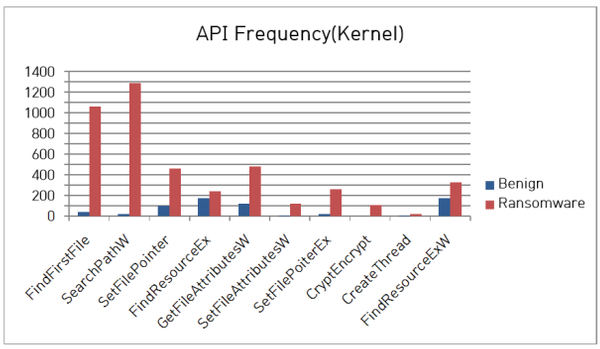

Fig. 5. Differences in Native API Frequency Between Benign and Cerber Ransomware

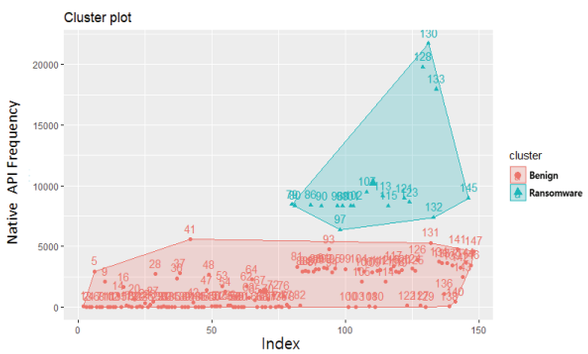

Fig. 6. Result of K-means Clustering

는 데 사용하므로, 앞서 선정한 총 10개의 Native API 항목 빈도수의 경우 정상 실행 파일에 비해 Cerber 랜섬웨어 실행 파일의 Native API 실제 호출 빈도수가 정상 실행 파일과 비교 시 확연히 높다는 것을 확인할 수 있다.

4.2 Native API 항목 클러스터링 및 군집화

Opcode Clustering과 마찬가지로 Fig. 6에서는 앞서 선택한 총 10개의 Native API 항목 빈도수에 대한 K-means Clustering 알고리즘을 이용해 나타낸 그래프이다. K값의 경우, K=2로 설정하여 두 군집으로 나타내었다. 클러스터링을 통한 군집화 결과, API 총 빈도수가 전반적으로 높다는 랜섬웨어 실행 파일의 특성을 이용하여 정상 실행 파일과 랜섬웨어 실행 파일을 분류하기 위한 기준으로 사용할 수 있다[13].

4.3 Native API 항목 코사인 유사도 측정

Fig. 7에서 확인할 수 있는 정보는 다음과 같다. 앞서 4.1에서 선정한 총 10개의 Native API 항목에 대해 각 Cerber 랜섬웨어 실행 파일별 코사인 유사도를 측정한 결과, 각 Cerber 랜섬웨어 실행 파일의 Native API 빈도수 기반 벡터 간 유사도는 평균 0.9 이상의 값을 나타낸다. 이는 앞서 선정한 총 10개의 Native API 항목 또한 실행 파일의 유형을 탐지할 수 있는 독립변수로 사용될 수 있다는 정당성을 가질 수 있다.

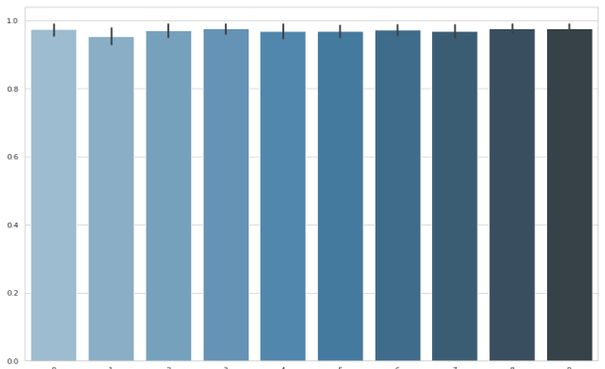

Fig. 7. Cosine Similarity of Cerber Ransomware Native API

5. Opcode와 API 분석정보의 결합

5.1 Opcode와 API 상관관계 파악

피어슨 상관계수(Pearson Correlation Coefficient)를 사용하여 Opcode 특징정보(Feature)와 API 특징정보 사이의 상관관계를 파악한다. 피어슨 상관계수의 정의는 아래의 Equation (1)과 같다.

$$\rho_{xy} = \frac{cov(X, Y)}{\sigma_x \sigma_y} \tag{1}$$

피어슨 상관계수는 두 변수 간 선형적인 상관관계를 모수적으로 나타낸 수치이다. 상관계수의 값을 -1과 1 사이의 값으로 평균화시키기 위해 두 변수의 공분산을 각 변수의 표준편차로 나눈다. 공분산이 한 변수의 값이 커질 때 다른 변수의 값도 커지는 경향을 가지면 양수 값, 한 변수의 값이 커질 때 다른 값이 작아지는 경향을 가지면 음수 값을 가진다. 표본의 상관계수는 r이라고 부르고 크기 n인 표본의 경우 아래의 Equation (2)를 이용하여 도출할 수 있다.

$$\gamma_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{2}$$

5.2 Opcode와 API 분석정보의 결합

Fig. 8에서 확인할 수 있는 정보는 다음과 같다. 본 연구에서 선정한 각 10개의 Opcode와 API의 연관성을 입증하기 위해 피어슨 상관계수(Pearson correlation coefficient)를 사용하여 시각화했다. 두 변수의 상관관계를 보여주는 우측 상단과 좌측 하단의 자색 부분이 -0.3 < r < -0.1의 약한 음(-)의 상관관계를 가짐에 따라 반비례의 특성을 가지는 것을 확인할 수 있다. 피어슨 상관계수를 이용한 연관성 검증을 통해 얻은 결과로 앞서 선정한 두 특징정보를 하나의 데이터 세트로 결합하는 방법은 각 특징정보의 항목을 서로 더하여 결

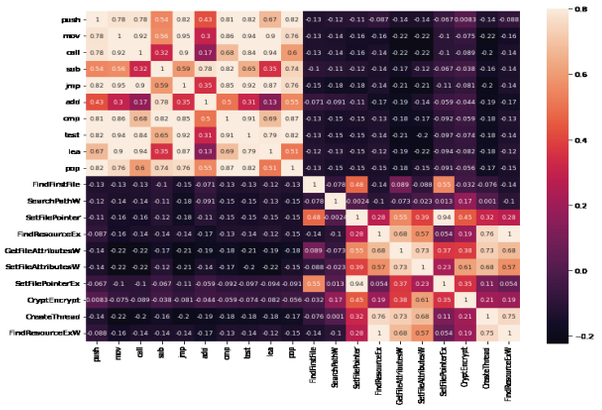

Fig. 8. Heatmap Between Opcode Feature and API Feature

합하는 방법을 선정하였다. 즉, 앞서 선정한 Opcode 항목 10개와 Native API 항목 10개를 더하여 총 20개의 특징정보를 가지는 하나의 데이터 세트를 도출하여 탐지모델의 기계학습에 사용한다. 이는 두 분석정보가 서로 가지는 뚜렷한 반비례의 특성을 하나의 데이터 세트로 결합하여 기계학습에 사용했을 때 상호 보완할 수 있는 요소로 사용된다.

6. Cerber형 랜섬웨어와 Worm의 분석 및 결과

6.1 Cerber 랜섬웨어와 Worm의 분석 필요성 및 분석 결과

앞서 3.1, 4.1절에서 선정한 각 10개의 특징정보와 5.2절에서 결합한 총 20개의 특징정보가 다양한 유형의 악성코드에 해당하는 것이 아닌 본 논문에서 사용하는 특정 랜섬웨어(Cerber)를 탐지하기 위한 특화된 특징정보임을 입증하기 위해 타 악성코드 유형 중 웜과의 탐지 결과를 도출하였다. 먼저 본 검증에 앞서 사용될 Cerber형 랜섬웨어와 Worm의 특징과 차이점을 비교하고 검증하기 위해 항목별로 비교 분석하였다. 분석 항목은 주요 특징, 존재 형태, 전파 경로, 대표적인 계열 순이다. 첫 번째로 주요 특징의 경우, Cerber형 랜섬웨어는 감염 PC의 파일을 암호화하여 금전 탈취를 목적으로 하고, 웜은 자가 복제를 통해 네트워크 장악을 통한 시스템 자원 고갈, 네트워크 트래픽 과부하, 감염 PC에 백도어 설치 등의 악성 행위를 수행한다. 두 번째로 존재 형태이다. Cerber형 랜섬웨어는 암호화된 파일 형태로 존재하는 반면 웜은 자가 복제된 독자적인 형태로 존재한다. 세 번째는 전파 경로이다. Cerber형 랜섬웨어는 주로 시스템 취약점 공격(Exploit), 첨부파일 등으로 전파되고 웜은 첫 감염 후 네트워크를 통해서 스스로 증식하고 전파한다. 마지막 비교 항목은 각 대표적인 계열이다. Cerber형 랜섬웨어의 다른 유형에는 Cerber 1, 2, 3, 4.0.1, 5.0.1, 6.0.1, 최종 CRBR의 진화 과정을 거친다. 또한, Cerber의 후속 랜섬웨어로 불리는 Magniber가 있다. 웜의 대표적인 계열에는 Moris Worm과 Blaster Worm 등이 있다. Table 3은 위 서술한 비교 분석한 내용을 표로 작성하였다.

Table 3. Comparison of Cerber Ransomware and Worm

Feature	Cerber	Worm
Main Feature	Encrypting Files and Stealing Money	System Resource Exhaustion and Backdoor Installation
Form of Existence	File Encryption Form	Self-replicating Form
Main Distribution Path	Vulnerability Exploit, Attached file, etc	Self-propagating Over the Network
Representative Family	Cerber3, 4, 5, 6 CRBR, Magniber	Moris Worm, Blaster Worm

Table 4. RandomForest Evaluation

N_estimator Max_depth	Accuracy	Precision	Recall	F1-Score
20/5	97.4%	100%	93.3%	96.5%
100/100	94.8%	93.3%	93.3%	93.3%
96/2	97.4%	93.7%	100%	96.7%

Table 5. SVM Evaluation

Cost	Accuracy	Precision	Recall	F1-Score
C = 1	89.7%	100%	73.3%	92.85%
C = 5	89.7%	86.6%	86.6%	86.6%
C = 100	94.8%	100%	86.6%	92.85%
C = 139	94.8%	93.3%	93.3%	93.3%

위와 같은 특징을 가진 웜 실행 파일 120개에서 특징정보(Feature)를 추출하고 학습용, 검증용 각 8:2 비율로 나누어 실험 데이터로 사용하였다. 탐지모델 학습에 사용한 특징정보는 앞서 선정한 총 20개의 특징정보를 동일하게 사용하였으며, 각 RandomForest, SVM 알고리즘을 이용한 탐지모델의 검증 결과는 Table 4, Table 5와 같다.

6.2 실험 결과

Table 4, Table 5에서 확인 가능한 정보는 다음과 같다. RandomForest 알고리즘을 사용한 탐지모델에서는 N_estimators/depth 값이 96/2일 때, SVM 알고리즘을 사용한 탐지모델에서는 Cost 값이 138일 때 각 97.4%, 94.8%로 나타났다. 위 평가 지표는 본 논문에서 선정한 특징정보(Feature)가 악성코드 유형 중 특정 랜섬웨어(Cerber)에 국한되며 동시에 특정 랜섬웨어 탐지에도 효과적임을 입증할 수 있다.

7. 제안 모델 프레임워크 흐름도 및 실험 결과

7.1 프레임워크 흐름도

Fig. 9는 제안하는 탐지모델에 대한 실험 데이터 수집부터

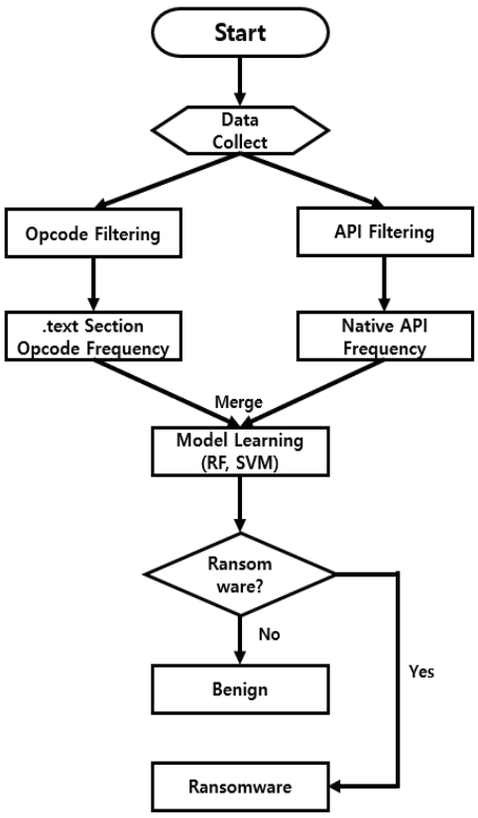

Fig. 9. Framework Flowchart

Cerber 랜섬웨어 탐지까지 진행되는 프레임워크 흐름도이다. 첫 번째는 초기 데이터 수집 후 Opcode와 API 데이터에서 각 ".text Section"부분과 실제 사용했던 Native API 부분으로 필터링을 거쳐 데이터 전처리를 수행한다. 그리고 두 번째는 앞서 검증을 통해 선정한 Opcode 항목, Native API 항목 각 10개, 총 20개의 특징정보를 추출한다. 세 번째로 추출한 각 10개의 특징정보 항목을 서로 병합하여 20개의 항목을 가진 하나의 데이터 세트로 만든 후 탐지모델의 기계학습에 이용한다. 마지막으로 탐지모델을 통한 정상 실행 파일과 Cerber 랜섬웨어 실행 파일의 분류를 통해 본 논문에서 제안하는 탐지모델을 검증하고 평가하는 과정을 거친다.

7.2 실험 환경

본 연구의 실험 환경은 Table 6과 같다. 먼저 Host OS로 사용하는 Ubuntu 20.04 LTS 버전을 구축한다. 또한, Guest PC(가상환경)의 구축을 위해 VirtualBox 6.1.32 버전과 Windows 7 Ultimate K SP1 버전을 Guest OS로 사용한다. 또한, 분석 및 정보의 추출과 분석 데이터의 전처리를 위해 Cuckoo SandBox 2.0.7, binutils 2.34 버전을 사용한다. 그 후, 앞서 분석한 각 분석정보의 클러스터링 및 군집화를 위해 Python 2.7.18 버전을 사용한다. 마지막으로 본 논문에서 제안한 탐지모델의 학습에는 Python 3.7, Google Colaboratory를 사용하였다.

Table 6. Analysis Environment

Classification	Name	Version
Host OS	Ubuntu	20.04 LTS
Ubuntu	VirtualBox	6.1.32
	Cuckoo SandBox	2.0.7
	Objdump	binutils 2.34
	Windows 7 (guest)	Ultimate K SP1
Windows	Python	2.7.18
Machine Learning	Google Colaboratory	-
	Python	3.7

7.3 실험 결과

위에서 진행한 분석정보의 검증으로 앞서 선정한 Opcode 항목과 Native API 항목의 빈도수가 랜섬웨어 실행 파일과 정상 실행 파일을 분류할 수 있는 유의미한 특징정보임을 입증했다. 본 논문에서 제안하는 탐지모델의 검증을 위해 scikit-learn 라이브러리의 RandomForest와 SVM 알고리즘을 사용하였고, RandomForest의 경우, 파라미터인 결정트리의 개수를 의미하는 n_estimators, 트리의 깊이를 의미하는 depth, SVM에서는 이상치 허용을 조정하는 C(cost)를 모델 최적화에 사용하였으며 모델의 정확도와 성능 등을 평가하기 위해 교차행렬을 사용하여 아래의 Equation (3), (4), (5), (6)을 기준으로 Precision, Accuracy, Recall, F1-score를 기계학습 모델의 평가 지표로 사용하였다.

$$Accuarcy = \frac{TP+TN}{TP+TN+FP+FN}*100 \quad (3)$$

$$\mathrm{Precision} = \frac{TP}{TP+FP}*100 \quad (4)$$

$$Recall = \frac{TP}{TP+FN}*100 \quad (5)$$

$$F1-score = 2*\frac{\mathrm{Precision}*Recall}{\mathrm{Precision}+Recall}*100 \quad (6)$$

제안한 탐지모델에서의 Accuracy 값은 Cerber 랜섬웨어 실행 파일과 정상 실행 파일을 각각 정확하게 분류한 비율을 의미하고, Precision 값은 탐지한 Cerber 랜섬웨어 중 실제 Cerber인 개수에 대한 비율, Recall 값은 Cerber 랜섬웨어 중 설계모델에서 실제로 탐지한 랜섬웨어 개수에 대한 비율, F1-score 값은 Precision과 Recall의 조화평균으로 제안한 탐지모델을 전반적으로 평가하고 검증하기 위한 수치로 사용된다. 정상 실행 파일 100개와 특정 랜섬웨어 계열(Cerber) 73개, 총 173개의 실행 파일 분석정보에서 앞서 선정한 특징정보(Feature)를 추출하여 탐지모델 학습 및 검증에 사용하기 위해 데이터 세트를 학습용, 검증용 각 8:2 비율로 나누어 실험 데이터로 사용하였다. 각 RandomForest, SVM 알고리즘을 이용한 탐지모델의 검증 결과는 Table 7, Table 8과 같다.

Table 7. RandomForest Evaluation

N_estimator Max_depth	Accuracy	Precision	Recall	F1-Score
20/5	93.54%	100%	88.8%	94.1%
100/20	96.77%	100%	94.11%	96.9%
100/100	96.77%	100%	94.11%	96.9%

Table 8. SVM Evaluation

Cost	Accuracy	Precision	Recall	F1-Score
C = 1	87.0%	100%	77.7%	87.5%
C = 3	87.0%	92.8%	81.2%	86.7%
C = 8	87.0%	92.8%	81.2%	86.7%
C = 100	83.8%	92.8%	76.4%	83.8%

Table 7, Table 8에서 확인 가능한 정보는 다음과 같다. RandomForest 알고리즘을 사용하여 적용한 탐지모델에서는 N_estimators/depth 값이 100/20일 때, SVM 알고리즘을 사용한 탐지모델에서는 Cost 값이 각 3, 8일 때 각 96.7%, 87.0%로 나타났다.

7.4 탐지모델 파라미터 최적화

초기 모델 학습 후 최적화 과정을 통해 탐지모델의 성능을 향상하고 정상 실행 파일과 랜섬웨어 실행 파일에 대한 탐지 정확도를 높이기 위해 하이퍼 파라미터를 튜닝(Tuning)하였다. 본 논문에서는 파라미터 튜닝을 위해 자동화 라이브러리 중 하나인 Optuna를 사용하였다. 기존의 튜닝 라이브러리인 GridSearchCV 등은 파라미터 값을 사용자가 직접 임의로 설정해야 하는 단점이 있지만, Optuna의 경우 파라미터 값에 대한 범위를 설정하고, 그 범위 내에서 자동탐색을 통해 최적의 파라미터 값을 도출할 수 있다. RandomForest의 경우, 기존 N_estimators값과 Max_depth값의 범위를 각각 1~100, 1~10로 설정하고 과적합 방지를 위해서 min_sample_split(내부 노드 분할에 필요한 최소 표본 데이터 수)와 min_sample_leaf(단말 노드에 필요한 최소한의 표본 데이터 수) 파라미터 값에 대한 범위를 2~10, 1~5로 설정하였다. SVM의 경우 기존 cost 값의 범위를 1e-5~1e5로 설정하였고, Gamma(경계값 곡률 조정) 값의 범위를 1e-5~1e5로 설정하여 각 100번의 시도 과정을 거쳤다. 아래와 같이 시도 별로 최적의 파라미터를 탐색하는 과정과 그 과정에서 얻은 최적의 파라미터를 제안하는 탐지모델의 분류 알고리즘에 적용하고 분류 알고리즘에 적용하고 검증한 선도와 탐지모델의 평가 지표이다.

Fig. 10은 RandomForest의 하이퍼 파라미터인 각 N_estimators, Max_depth, Min_sample_split, Min_sample_leaf값의 변화에 따른 최적의 파라미터 값을 찾고 이를 탐색하는 과정을 시각화한 그림이다. Table 9는 최적화 과정을 거쳐 나온 최적의 값이며, Table 10은 파라미터 값을 적용한 뒤, RandomForest 알고리즘을 사용하여 학습하고 검증한 탐지모델의 평가결과이다.

Fig. 11은 SVM의 하이퍼 파라미터인 Cost값과 Gamma값에 대한 변화에 따른 최적의 파라미터 값을 찾고 이를 탐색

Table 9. RandomForest Best Parameter

N_estimator	Max_depth	Min_sample_split	Min_sample_leaf
42	5	6	1

Table 10. RandomForest Optimization Result

Accuracy	Precision	Recall	F1-Score
97.1%	100%	92.3%	96.0%

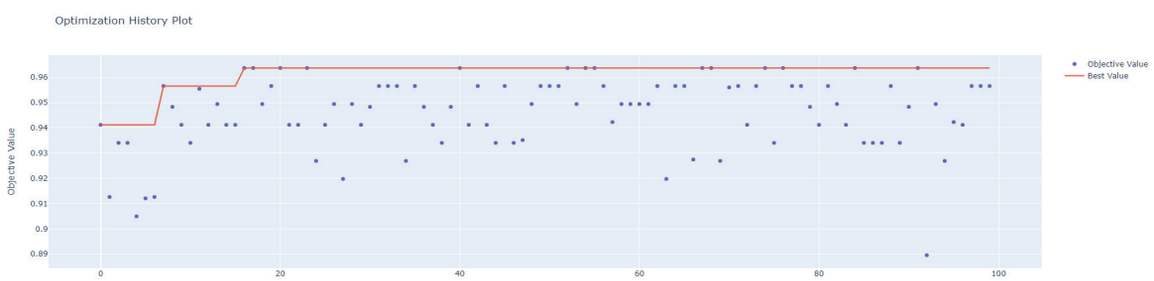

Fig. 10. RandomForest Optimization Process

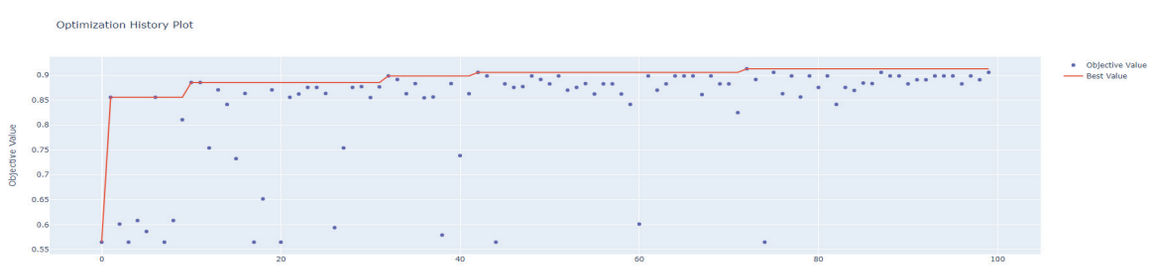

Fig. 11. SVM Optimization Process

Table 11. SVM Best Parameter

Cost	Gamma
7296	0.001

Table 12. SVM Optimization Result

Accuracy	Precision	Recall	F1-Score
91.4%	100%	76.9%	86.9%

하는 과정을 시각화한 그림이다. Table 11은 최적화 과정을 거쳐 나온 최적의 값이며, Table 12는 최적의 파라미터 값을 적용한 뒤, SVM 알고리즘을 사용하여 학습하고 검증한 탐지 모델의 평가결과이다.

8. 결론 및 향후 연구

본 논문에서는 정적 분석과 동적 분석을 결합한 특징정보를 바탕으로 기계학습 알고리즘을 이용한 Cerber 랜섬웨어 분류 및 탐지모델을 제안하고 검증하였다. ".text Section" Opcode 와 실제 프로그램 동작 시 호출하는 Native API로 구체화하여 선정한 Opcode와 API 항목의 빈도수에서 파일 유형별 군집화와 유사도 분석을 통해 Cerber 랜섬웨어를 특정할 수 있는 특징정보를 선정하였다. 또한, Cerber 랜섬웨어 실행 파일과 웜 악성코드 실행 파일 간의 탐지 실험을 수행하여 선정한 특징정보가 Cerber 랜섬웨어의 탐지에 특화된 정보임을 검증하였다. 위와 같은 분석과 검증을 거쳐 선정된 특징정보를 결합한 데이터 세트를 기계학습 하여 서론에서 서술한 시그니처 기반의 패턴탐지 한계를 보완한 Cerber 랜섬웨어의 행위기반 탐지모델을 구현하였다. 이 연구에서 선택하여 검증한 랜섬웨어 계열은 케르베르(Cerber) 계열이며, 앞서 선정한 Opcode 빈도수 항목과 Native API 빈도수 항목을 결합한 총 20개의 특징정보로 해당 계열 랜섬웨어 실행 파일과 정상 실행 파일 간의 분류탐지실험을 진행하였고, 최적화 이후 두 분류 알고리즘에서 정확도 97.1%, 91.4%의 탐지율을 보여 성능을 입증하였다. 그러나 동적 분석에서 선택한 Sandboxing 기술의 경우 Anti-VM 기법이 적용된 랜섬웨어 실행 파일의 분석이 불가능한 한계를 지닌다. Anti-VM 기능을 가진 랜섬웨어는 가상환경의 특징(CPU ID, IO Port, 레지스트리 값, 서비스 실행 여부 등)을 인식하여, 실행을 중단하거나 일정 시간 지연을 통해 분석을 회피하고 있어, 향후에는 가상환경 정보를 수집하는 Anti-VM형 랜섬웨어에 대한 특징정보 분석과 데이터 학습을 통해 개선할 계획이다.

References

[1] K. M. Kim, J. S. Kim, and Y. J Lee, "Ransomware trends & Statistics, First Quarter for 2022," KISA, pp.2, 2022. https://seed.kisa.or.kr/kisa/Board/130/detailView.do

[2] G. H. Lee, M. C. Hwang, Y. I. Ku, D. Y. Hyun, and D. Y. Yoo, "A study on the ransomware detection model using the clustering and similarity analysis of opcode and API," *Proceedings of the Annual Spring Conference of Korea Information Processing Society Conference (KIPS)*, Vol.29, pp.179-180, 2022.

[3] G. H. Lee, M. C. Hwang, Y. I. Ku, D. Y. Hyun, and D. Y. Yoo, "A study on the ransomware detection model using the clustering and similarity analysis of opcode and API", *Proceedings of the Annual Spring Conference of Korea Information Processing Society Conference (KIPS)*, Vol.29, pp.182, 2022.

[4] D. Vidyarthi, C. Kumar, S. Rakshit, and S. Chansarkar, "Static malware analysis to identify ransomware properties," *IJCSI International Journal of Computer Science Issues*, Vol.16, Iss.3, pp.1-8, 2019.

[5] İ. Kara and M. Aydos, "Static and dynamic analysis of third generation cerber ransomware," *International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism*, 2018.

[6] P. O'Kane, S. Sezer, K. McLaughlin, and E. G. Im, "SVM training phase reduction using dataset feature filtering for malware detection," *Journal of IEEE Transactions on Information Forensics and Security*, Vol.8, No.3, pp.500-509, 2013.

[7] M. Zhang, Y. Duan, H. Yin, and Z. Zhao, "Seman-tics-aware android malware classification using weighted contextual API dependency graphs," *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp.1105-1116, 2014.

[8] M. Hasan and M. Rahman, "RansHunt: A support vector machines based ransomware analysis framework with integrated feature set," *2017 20th International Conference of Computer and Information Technology (ICCIT)*, Dec. 2017.

[9] K. W. Moon and J. H. Lee "Analysis of the latest ransomware features," Sangmyung University, 2018.

[10] Ahnlab ASEC, "Cerber ransomware demanding money by voice," 2016.

[11] O. C. Kwon, S. J. Bae, J. I. Cho, and J. S. Moon, "Malicious codes re-grouping methods using fuzzy clustering based on native API frequency," *Journal of The Korea Institute of Information Security and Cryptology*, Vol.18, No.6, pp.115-127, 2008.

[12] INFOSEC, Windows functions in malware analysis part1, May 26, 2015. https://resources.infosecinstitute.com/topic/windows-functions-in-malware-analysis-cheat-sheet-part-1/.

[13] J. W. Kim, "A study on Machine Learning-based Ransomware Detection Model using Hybrid Analysis," Konkuk University for Master's Degree in Korea, 2017.

이 계 혁

https://orcid.org/0000-0002-5444-1907
e-mail : gdsmsla@naver.com
2017년 ~ 현 재 홍익대학교
소프트웨어융합과 학사과정
관심분야 : 정보보호, 융합보안 등

황 민 채

https://orcid.org/0000-0002-3596-1165
e-mail : hminchae@gmail.com
2021년 ~ 현 재 홍익대학교
소프트웨어융합과 학사과정
관심분야 : 정보보호, 융합보안 등

현 동 엽

https://orcid.org/0000-0002-6063-3630
e-mail : hyunfree0106@naver.com
2017년 ~ 현 재 홍익대학교
소프트웨어융합과 학사과정
관심분야 : 정보보호, 융합보안 등

구 영 인

https://orcid.org/0000-0002-0990-946X
e-mail : ku1718@naver.com
2017년 ~ 현 재 홍익대학교
소프트웨어융합과 학사과정
관심분야 : 정보보호, 융합보안 등

유 동 영

https://orcid.org/0000-0002-8231-5203
e-mail : ydy@hongik.ac.kr
1997년 숭실대학교 컴퓨터학부(학사)
2000년 숭실대학교 컴퓨터학과(석사)
2011년 고려대학교 컴퓨터학과(박사)
2014년 ~ 2018년 숭실대학교
정보과학대학원 겸임교수
2019년 ~ 2020년 한국인터넷진흥원 수석연구원
2021년 ~ 현 재 홍익대학교 소프트웨어융합과 부교수
관심분야 : 정보보호, 융합보안, IoT, 블록체인 등