산업융합연구 제20권 제5호 pp. 61-68, 2022
ISSN 2635-8875 / e-ISSN 2672-0124
DOI : https://doi.org/10.22678/JIC.2022.20.5.061

지능적 탐지 모델을 위한 악의적인 코드의 특징 정보 추출 및 분류

황윤철
한남대학교 탈메이지 교양융합대학 조교수

Extraction and classification of characteristic information of malicious code for an intelligent detection model

Yoon-Cheol Hwang
Assistant Professor, Department of Talmage Liberal Arts·Convergence College, Hannam University

요 약 최근에는 발전하는 정보통신 기술을 이용하여 악의적인 코드들이 제작되고 있고 이를 기존 탐지 시스템으로는 탐지하는게 역부족인 실정이다. 이러한 지능적이고 악의적인 코드를 정확하고 효율성 있게 탐지하고 대응하기 위해서는 지능적 탐지 모델이 필요하다. 그리고, 탐지 성능을 최대로 높이기 위해서는 악의적인 코드의 주요 특징 정보 집합으로 훈련하는 것이 중요하다. 본 논문에서는 지능적 탐지 모델을 설계하고 모델 훈련에 필요한 데이터를 변환, 차원축소, 특징 선택 단계를 거쳐 주요 특징 정보 집합으로 생성하는 기법을 제안하였다. 그리고 이를 기반으로 악의적인 코드별로 주요 특징 정보를 분류하였다. 또한, 분류된 특징 정보들을 기반으로 변형되거나 새로 등장하는 악의적인 코드를 분석하고 탐지하는데 사용할 수 있는 공통 특징 정보를 도출하였다. 제안된 탐지 모델은 제한된 수의 특성 정보로 학습하여 악의적인 코드를 탐지하기에 탐지 시간과 대응이 빨리 이루어져 피해를 크게 줄일 수 있다. 그리고, 성능 평가 결과 값은 학습 알고리즘에 따라 약간 차이가 나지만 악의적인 코드 대부분을 탐지할 수 있음을 평가로 알 수 있었다.

키워드 : 탐지 모델, 악의적인 코드, 특징 정보, 추출, 분류

Abstract In recent years, malicious codes are being produced using the developing information and communication technology, and it is insufficient to detect them with the existing detection system. In order to accurately and efficiently detect and respond to such intelligent malicious code, an intelligent detection model is required, and in order to maximize detection performance, it is important to train with the main characteristic information set of the malicious code. In this paper, we proposed a technique for designing an intelligent detection model and generating the data required for model training as a set of key feature information through transformation, dimensionality reduction, and feature selection steps. And based on this, the main characteristic information was classified by malicious code. In addition, based on the classified characteristic information, we derived common characteristic information that can be used to analyze and detect modified or newly emerging malicious codes. Since the proposed detection model detects malicious codes by learning with a limited number of characteristic information, the detection time and response are fast, so damage can be greatly reduced and Although the performance evaluation result value is slightly different depending on the learning algorithm, it was found through evaluation that most malicious codes can be detected.

Key Words : Detection model, Malicious code, Characteristic information, Extraction, Classification

This work was supported by 2021 Hannam University Research Fund.

*Corresponding Author : Yoon-Cheol Hwang(dolpin2010@gmail.com)

Received March 11, 2022 Revised April 22, 2022
Accepted May 20, 2022 Published May 28, 2022

1. 서론

현재 우리사회는 4차산업혁명이 절정기를 향해가고 있는 시기로서 급속하게 발전하는 정보통신 기술들로 인해 인간의 삶 전반은 편리하고 윤택해지고 있는 반면 이런 기술들을 악용하여 타인의 삶을 파괴하는 부작용 역시 급증하고 있는 현실이다. 이런 부작용들은 대부분 악의적인 코드를 사용하여 발생되고 있으며 매년 개인이나 사회를 대상으로 발생되는 침해 사고가 급증하고 있고 오픈 소스와 자동화 도구를 이용하여 기존의 악의적인 코드를 변형하거나 새로운 악의적인 코드가 쉽고 지능적으로 제작되어 나타나고 있다[1]. 이러한 악의적인 행위를 탐지하기 위해 사용되는 침입 탐지 시스템 대부분은 알려진 시그니처 기반의 탐지 기법을 사용하기 때문에 기존 코드를 변형하거나 새롭게 만들어진 악의적인 코드를 탐지하는데 많이 역부족이다. 따라서 이러한 지능적인 악의적인 코드를 정확하고 효율성 있게 탐지하고 대응하기 위한 지능적 탐지 모델이 시급히 필요하고 탐지 성능이 우수한 모델을 만들기 위해서는 악의적인 코드의 특징 정보를 정확하게 파악하고 분류한 다음 이 특징 정보를 데이터로 악의적인 코드를 탐지하는 모델을 훈련하는 것이 가장 중요하다[2].

따라서 본 논문에서는 지능화 되어가고 있는 악의적인 코드를 탐지하기 위한 지능적 탐지 모델과 제안한 탐지 모델이 학습하는데 필요한 악의적인 코드들의 특징 정보를 추출 방법 제안하고 악의적인 코드별 특징 정보를 분류한다. 악의적인 코드들의 특징 정보는 기존에 알려져 있는 다양한 악의적인 코드들에서 각각의 악의적인 코드 유형별 특징 정보를 학습하기 이전 단계인 전처리기로 추출한다. 제안한 탐지 모델은 악의적인 코드 데이터 집합과 정상적인 코드 데이터 집합을 사용하여 해당 파일의 내용을 Hexdump 유틸리티를 이용하여 특성 추출해 적합한 16진법 코드로 변환하여 n-gram(4-gram)의 특징 정보 집합으로 생성한다[3]. 그런 다음 생성된 특징 정보 집합을 주성분 분석(PCA: Principal component analysis)를 이용하여 상호 연관되어있는 특징 정보를 하나의 특징 정보 처리하여 특징 정보 집합을 축소한다[4]. 축소된 특징 정보 집합을 TF-IDF(Term Frequency -Inverse Document Frequency)를 사용하여 최적의 정확도와 학습 시간을 산출하는 사용 빈도가 높은 특징 정보를 선택한다. 선택된 특징들을 이용해 SVM (Support Vector Machine)과 신경망(Neural Network)을 사용하여 학습기에서 학습을 진행한다. 그리고 높은 정확도를 산출하는 분류기를 생성하기 위하여 매개 변수의 임계값을 조정해가면서 반복적으로 학습을 진행한 후 검증 과정을 거쳐 가장 탐지 성능이 우수한 지능적 탐지 모델의 분류기를 생성한다[5]. 제안한 특징 정보를 추출하는 기법은 악의적인 코드의 유형별로 중요도가 높은 특징 정보를 선별하여 분류함으로써 악성 코드를 신속하고 정확하게 탐지할 수 있고 변종과 새로운 악의적인 코드의 탐지와 대응에 활용하여 악의적인 코드에 의한 피해를 최소화하는데 도움을 줄 수 있다.

논문의 구성은 다음과 같다. 2장에서는 관련 연구로 악의적인 코드와 특징 정보에 대해 살펴보고 3장에서는 지능적 탐지 모델과 악의적인 코드에서 탐지을 효율성을 고려한 특징 정보 추출 방법을 제안하고 추출 과정을 살펴본다. 4장에서는 추출된 다양한 특징 정보들을 악의적인 코드를 식별하고 탐지하는데 활용할 수 있는 주요 특징 정보를 분류하여 기술한 다음 5장에서 연구의 내용과 향후 연구 과제를 언급하면서 결론을 맺는다.

2. 관련연구

2.1 악의적인 코드

악의적인 코드(Malware)는 정보 유출과 금전적 이익 등 악의적인 목적으로 작성되어 컴퓨터 사용자의 승인 없이 컴퓨터에 침투하거나 설치되어 악의적인 행위를 수행하는 프로그램을 말하며 아이디나 암호와 같은 개인정보를 유출하거나 주요 기관의 서비스를 중지시키는 행위 등과 같이 다양한 종류의 악의적인 행위를 한다. 악의적인 코드들이 하는 행위를 바탕으로 백도어, 트로이 목마, 바이러스, 웜, 디도스, 스파이웨어, 애드웨어, 랜섬웨어 등으로 분류된다[6,7].

2021년 상반기 수집·탐지된 악의적인 코드를 [8]에서 유형별로 분류한 내용을 분석해보면, 2021년 상반기에는 73.8%로 컴퓨터 이름, 사용자 이름, 볼륨 정보, 시스템 설치시간, 설치 프로그램 목록, 서비스 목록 실행중인 프로세스 목록 등과 같은 기기정보와 브라우저, 메일 클라이언트, FTP 등의 계정정보 관련 파일(웹데이터, 로그인데이터, 개인설정파일 등)과 같은 계정 정보를 탈취하는 정보유출형 악의적인 코드가 가장 활발하게 많이 유포되었고, 그 다음으로 디도스가 7.1%로 많았고,

랜섬웨어는 5.6%으로 디도스 다음으로 많이 나타났다.

2.2 악의적인 코드 특징 정보 유형

한국인터넷진흥원 특징 정보 분석 시스템에서는 악의적인 코드가 가지는 정보의 유형을 메타데이터(Metadata), 정적 정보(Static Info), 동적 정보(Dynamic Info), 네트워크(Network), ATT&CK Matrix, 기타 정보(ETC)와 같이 6개 카테고리로 분류하고 그 하위에 세부 특징 정보를 72가지로 분류하여 사용하고 있으며 다양한 인공지능 모델 개발이나 자동 분석에 이 특징 정보를 활용하고 있다[9]. 6개 카테고리의 내용을 살펴보면 기본적인 파일정보와 PE정보는 메타데이터로 분류한다. 개발 경로 및 문자열 등 코드 내에서 확인 가능한 정보는 정적 정보로 분류되고 레지스트리, 프로세스 등 악의적인 코드 실행 시 동작하는 주요 행위 정보는 동적 정보로 분류한다. 그리고 악의적인 코드 실행 시 접속 시도 및 파일, 메모리내 포함된 URL/IP는 네트워크 정보로 분류한다. 악의적인 코드를 전략, 전술 별(TTPs) 행위를 기술단위 별로 추출한 정보는 ATT&CK Matrix 정보로 분류되며 악의적인 코드 함수 단위 등의 정보와 악의적인 문서에 대한 정보는 기타 정보로 분류한다.

2.3 악의적인 코드 분석 방법

악의적인 코드를 분석하는 방법은 분석가에 따라 다양하게 정의되고 있으며 일반적으로 많이 사용되고 있는 방법은 정적 분석과 동적 분석이다[10].

정적 분석은 악의적인 코드를 실행하지 않고 그 코드가 갖고 있는 내용들을 통해 악의적인 여부를 진단하는 방법이다. 분석이 비교적 쉽고 빠르며, 별도의 지식 없이 모든 파일들이 가지고 있는 해시나, 사용되는 API, 그리고 문자열 등을 이용하여 분석 여부나 이후 분석 방향에 대하여 결정된다. 해시 값의 경우 VirusTotal과 같이 많은 백신들의 엔진을 통해서 비교하여 결과를 출력할 수가 있지만 악성코드가 자가 변조를 할 수 있는 경우에는 해시 값을 완전히 신뢰할 수 없다. 그리고 EXE, DLL, SYS 등의 경우 PE 구조를 가지고 있어서 PE 구조에서 악의적인 행위와 관련된 정보들을 파악할 수가 있다. 또한, 사용되는 API의 목록을 가지는 IAT(Import Address Table)을 이용할 경우에는 어떠한 DLL을 필요로 하며, 해당 DLL에서 어떠한 함수를 사용하는지 확인할 수가 있고 socket과 같은 API를 사용하는 경우에는 추가적인 분석이 필요할 경우도 있다. 문자열에는 새로운 프로세스나 파일을 생성하거나 제거, 외부 통신을 위한 IP 주소, 지속성을 위해 레지스트리에 남기는 키 값 등 악의적인 코드의 기능과 직접적으로 관련된 문자를 포함하고 있을 수 있다. 현대의 악의적인 코드들은 패킹이나 난독화와 같은 방법을 사용하여 탐지 정보를 확인할 수 없도록 하는 경우가 대부분으로 각 방법에 맞게 언패킹을 진행하거나 적절한 방법으로 난독화를 풀어야 한다[11]. 정적 분석에 사용할 수 있는 도구에는 HxD, HashTab, VirusTotal, BinText, Dependency Walker, Resource Hacker 등이 있다.

동적 분석은 의심되는 파일을 실행하여 나타나는 변화를 모니터링하고 어떠한 기능을 수행하는지 확인하여 악의적인 코드를 판별하는 방법이다. 의심되는 파일이 실제 악성 행위를 할 수 있으므로 보통 가상 환경에서 독립된 네트워크와 같이 필요한 설정 들을 구축한 뒤, 동적 분석 도구들을 통해 어떠한 동작을 하는지 확인하는 동적 분석을 수행한다[12]. 동적 분석은 의심되는 파일을 실행되었을 때 생성되는 프로세스를 가장 먼저 살펴보아야 하고 어떠한 이름의 프로세스가 생성되는지 확인하며, 하위 프로세스로 생성되는 것이 있다면 이에 대해서도 추가적인 분석도 진행되어야 한다. 그리고 악의적인 코드가 프로세스를 진행하면서 어떠한 파일에 변화가 생기는지 확인하는 것도 필요하고 파일과 관련하여 레지스트리의 변화와 어떠한 곳에서 어떠한 네트워킹이 일어나는지 파악해야 한다. 동적 분석을 하는데 많이 사용되고 있는 도구에는 Process Explorer, ProcessMoniter, WireShark, TCP View, RegShot, REGA, NTFS Log Tracker 등이 있다[13].

3. 지능적 침입 탐지 모델 및 특징 정보 추출

3.1 지능적 침입 탐지 모델

지능적 침입 탐지 모델은 Fig. 1과 같이 입력과 출력, 학습부로 이루어지며 입력에는 훈련 데이터인 악의적인 코드와 정상 코드로 구성된다. 학습부는 전처리기와 학습기로 구성되고 전처리기는 특징 추출부와 특징 선택부로 이루어진다. 그리고 학습기는 지능적 알고리즘들로 구성된다[14]. 출력는 학습을 마친 후 생성된 악

의적인 코드 분류기로 이루어진다. 그리고 학습을 마치고 생성된 분류기는 테스트 데이터를 이용해 성능을 검증하여 탐지 성능을 높였다. 지능적 탐지 모델의 동작 과정은 아래와 같이 진행된다.

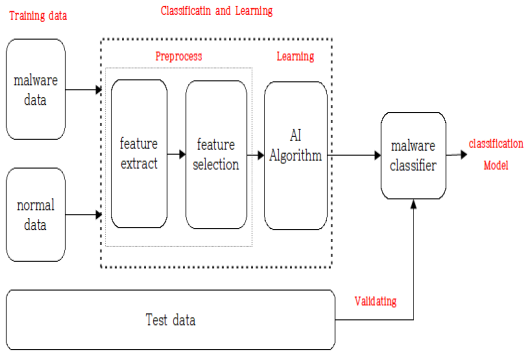

Fig. 1. Intelligent Detection Model

단계1: 훈련 데이터가 분류 및 학습 과정의 전처리기에 입력된다.

단계2: 전처리기에서 입력된 데이터를 Hexdump 유틸리티를 이용하여 특성 추출해 적합한 16진법 코드로 변환하여 n-gram(4-gram)의 특징 정보 집합을 생성한다.

단계3: 생성된 특징 정보 집합을 주성분 분석를 이용하여 상호 연관되어있는 특성 정보를 하나의 특징 정보 처리하여 특징 정보 집합을 축소한다.

단계4: 축소된 특징 정보 집합을 TF-IDF를 사용하여 최적의 정확도와 학습시간을 산출하는 사용 빈도가 높은 특징 정보를 선택한다.

단계5: 선택된 특징들을 이용해 SVM과 신경망을 사용하여 학습기에서 학습을 진행. 높은 정확도를 산출하는 분류기를 생성하기 위하여 매개 변수의 임계값을 조정해가면서 반복적으로 학습을 진행한다.

단계6: 학습이 종료되면 악의적인 코드를 최적으로 분류하는 분류기를 생성한다.

단계7: 생성된 분류기는 테스트 데이터를 사용해 분류기의 정확도를 검증한다.

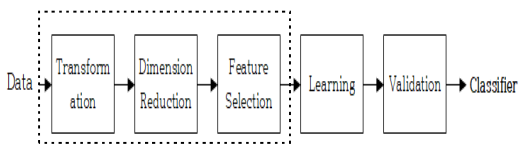

Fig. 2. Classifier Creation Process

3.2 악의적인 코드 특징 정보 추출

제안하는 특징 추출 기법은 분류기를 생성하는 전체 과정인 Fig. 2에서 학습과 검증 과정을 제외한 변환, 차원 축소, 특징 선택 3단계로 진행된다.

제안 기법은 특징 정보 데이터의 차원을 줄이기 위해 악의적인 코드의 특징 추출과 선택을 결합하여 사용하며 특징 추출과 선택이 완료되면 학습과 검증 과정을 거쳐 최종 분류기를 생성한다. 이 작업에 사용된 데이터 집합은 NSL-KDD[15]와 VX Heavend[16]에서 수집된 코드를 기반으로 진행하였고 악의적인 코드들을 분류하기 위한 특징 추출 과정은 다음과 같다.

단계1: 변환 단계 : 먼저 실행 파일을 재가공하고 Hexdump 유틸리티를 이용하여 16진수 코드로 변환하여 이들 파일의 특징을 추출한 다음 4-gram 특징을 생성한다. 생성된 데이터 집합에는 수 천개의 n-gram 특징이 포함되어 있고 이런 특징 모두가 악의적인 코드의 특징 정보를 추출하는데 필요하지 않다. 따라서 특징 선택 기법은 분류기의 원본 데이터 집합보다 특징 정보의 개수가 최소가 되는 축소된 특징 정보 데이터 집합을 생성해야 한다. 본 논문에서는 여러 연구를 기반으로 결과를 가장 좋게 산출하는 4-gram 방식을 사용한다[17].

단계2: 차원 축소 단계 : 앞 단계에서 생성된 4-gram으로 이루어진 특징 데이터 집합을 주성분 분석을 사용해 축소된 특징 정보 데이터를 생성한다[18]. 먼저 생성된 특징 데이터 집합을 1차원 벡터로 변환한 다음 공분산 행렬에 대한 고유 벡터를 구하고, 고유값 크기 순으로 나열하여 저차원의 특징 벡터로 변환하는 과정을 거쳐 축소된 특징 정보 데이터 집합을 생성한다. 이에 대한 과정은 파이썬을 이용해 데이터 정규화(Mean Centering), 공분산 행렬(Covariance Matrix) 구하기, 공분산 행렬의 고유값 분해(Eigendecom position), 고유값(eigen value) 크기순으로 정렬, 고유값 벡터(eigen vector)와 전체 집합 X를 곱하여 추출 변수 Z 구하기 순으로 진행된다. PCA를 통하여 구하는 주성분을 Z라고 했을 때 Z는 다음 식(1)과 같이 표현된다.

$$\begin{aligned} Z_1 &= a_1^T X = a_{11}X_1 + a_{12}X_2 + \cdots + a_{1p}X_p \\ Z_2 &= a_2^T X = a_{21}X_1 + a_{22}X_2 + \cdots + a_{2p}X_p \\ &\vdots \qquad\qquad \vdots \\ Z_p &= a_p^T X = a_{p1}X_1 + a_{p2}X_2 + \cdots + a_{pp}X_p \end{aligned} \quad (1)$$

식(1)에서 X는 축소하고자 하는 전체 데이터 집합이고 ai는 기저 또는 계수이고 Zi는 각 기저로 사영된 변환 후 변수(주성분)을 의미한다. 즉 a값만 구하면 Zi값을 구할 수 있다. 결국 a값이 공분산 행렬을 고유값 분해를 수행하여 얻은 고유 벡터가 된다.

단계3: 특징 선택 단계 : 악의적인 코드들을 분류 가장 비중이 많이 차지하는 특성을 선택하기 위해 TF-IDF를 사용한다. TF-IDF는 단어의 빈도(Term Frequency)와 역 문서 빈도(Inverse Document Frequency)를 토대로, 특정 문서 내에 어떤 단어가 얼마나 중요한지를 나타내는 통계적 수치로 파이썬의 tf•idf 함수를 Fig. 3과 같이 표현하여 선택하였고 악의적인 코드군의 주요 행동도 반영하여 가장 비중을 많이 차지하는 특징을 선택하도록 하였다.

```
TF-IDF
tf idf(t,d,D) = tf(t,d) • idf(t,D)
# d는 단어가 포함된 해당문서, D는 전체문서
# tf(t,d) =  단어 t가 포함된 문서의 수 / 전체 문서의 수
# idf(t,D) = 전체 문서의 수 / 단어 t가 포함된 문서의 수
```

Fig. 3. TF-IDF fuction of Python

4. 악의적인 코드 특징 정보 분류 및 평가

4.1 특징 정보 추출 방법

제안한 기법을 사용하여 추출된 특징 정보를 백도어, 스파이웨어, 애드웨어, 바이러스,트로이 목마, 웜과 같이 악의적인 코드를 분류하고 분류된 유형별로 행해지는 주요 동작도 반영하여 가장 비중을 많이 차지하는 특징 정보들만 선정하였다.

1) 백도어(Backdoor): 백도어는 운영체제나 프로그램에서 정상적인 인증 과정을 거치지 않고 바로 접근할 수 있도록 만들어진 프로그램이다. 시스템의 취약점 또는 악의적인 코드 등을 통해 감염되고 특정 포트를 오픈하고 해커가 언제나 들어 올 수 있도록 항상 백그라운드로 실행 되는 특징을 가지고 있다. 주로 파일이나 프로세스를 숨기거나 네트워크 연결을 숨기고 원격으로 명령 실행하며 관리자 권한을 획득하는 행동을 한다. 따라서 백도어를 탐지하기 위한 주요 특징 정보는 Log 정보, 해당 시스템의 호스트 명, 시스템 사용 OS, 네트워크 연결 정보(접근 시기, 접근 기간, IP 주소, Mac 주소), 생성된 서비스 포트, 프로세스 정보(프로세서 생성 경로, 생성된 프로세스 이름), 파일 생성 경로, 파일 생성 이름, 레지스트리 키, 레지스트리 키 값을 들 수 있다.

2) 스파이웨어(Spyware): 스파이웨어는 사용자의 동이 없이 설치되어 해당 시스템의 정보를 주기적으로 특정한 서버에 보내는 악의적인 프로그램이다. 특정 웹사이트 접속하거나 전자 메일과 악성 프로그램을 통해 감염된다. 웹브라우저의 환경을 변화시키거나 사용자 정보를 유출하는 행위를 한다. 주요 활동으로는 컴퓨터 내에 저장된 개인정보를 수집하고 전송하며 사용자의 행동을 감시하고 파일을 삭제하거나 변조하며 실시간 서버와 연결되어 악의적인 코드를 설치한다. 따라서 스파이웨어를 탐지하기 위한 주요 특징 정보는 네트워크 연결 정보(접근 시기, IP 주소,open URL), 레지스트리 키, 레지스트리 키 값, 생성 레지스트리, 실행 경로, 실행 파일명, 프로세스(프로세서 생성 경로, 생성된 프로세스 이름), 시스템 콜 정보를 들 수 있다.

3) 애드웨어(Adware): 애드웨어는 사용자의 동이 없이 특정 사이트에 접속하게 만드는 악의적인 프로그램이다. 특정 웹사이트를 접속하거나 전자 메일과 악성 프로그램을 통해 감염된다. 임의적으로 특정 사이트 접속하거나 웹 페이지에 홍보 등의 악의적인 목적을 하는 베너를 출력하는 행동을 한다. 따라서 애드웨어를 탐지하기 위한 주요 특징 정보는 네트워크 연결 정보(접근 시기, IP 주소,open URL), 레지스트리 키, 레지스트리 키 값, 생성 레지스트리, 실행 경로, 실행 파일명, 프로세스(프로세서 생성 경로, 생성된 프로세스 이름)를 들 수 있다.

4) 바이러스(Virus): 바이러스는 사용자 컴퓨터내에서 사용자 몰래 프로그램이나 실행 가능한 부분을 변형해 자신 또는 자신의 변형을 복제하는 악의적인 프로그램이다. 스스로를 복제하고 파일들을 감염시키는 행위를 하지만 다른 컴퓨터로 스스로 전파되지는 않는다. 따라서 바이러스를 탐지하기 위한 주요 특징 정보는 문자열 정보, 메모리 정보, 실행 스크립트 파일명, 레지스트리 키, 레지스트리 키 값, 실행 경로, 생성 레지스트리, 파일(파일 생성 경로, 파일 생성 이름), 프로세스(프로세서 생성 경로, 생성된 프로세스 이름)를 들 수 있다.

5) 트로이 목마(Trojan horse): 트로이 목마는 사용자가 눈치 채지 못하게 정상적인 프로그램으로 위장하여 시스템에서 설치되어 사용자의 컴퓨터를 조종할 수

있는 프로그램이다. 트로이 목마는 주로 시스템의 충돌을 유발하거나 파일을 삭제하고 수정하며 데이터를 오염시키고 디스크를 포맷하거나 네트워크를 이용하여 악의적인 코드를 전파하고 사용자의 민감한 정보 에 접근하여 유출하는 행동을 한다. 따라서 트로이목마를 탐지하기 위한 주요 특징 정보는 포트 정보, 레지스트리 키, 레지스트리 키 값, 생성 레지스트리, 실행 파일명, 실행 경로, 프로세스(프로세서 생성 경로, 생성된 프로세스 이름), 파일(파일 생성 경로, 파일 생성 이름), open URL를 들 수 있다.

6) 웜(Worm): 웜은 인터넷이나 네트워크를 통해서 타 시스템에 스스로 전파되고 윈도우나 응용 프로그램의 취약점과 이메일이나 공유 폴더를 통해 감염된다. 주로 시스템의 부트 영역에 침입하거나 메모리에 상주하면서 시스템과 네트워크의 성능을 저하시키고 호스트의 통제를 획득하여 통제를 유지하거나 네트워크를 이용해 다른 호스트로 전파된다. 파일과 윈도우의 레지스트리, 프로세스를 변경하며 네트워크로 접근하여 특권을 변경하고 비정상 질의를 수행하며 중요 API 호출하는 행동을 한다. 따라서 웜을 탐지하기 위한 주요 특징 정보는 네트워크 연결 정보(호스트 명, IP주소, 프로토콜), 파일(파일 생성 경로, 파일 생성 이름), 레지스트리 정보(레지스트리 키 값, 생성 레지스트리), 사용 포트 정보, open URL, 접근권한,API Call을 들 수 있다. 악의적인 코드들의 특징을 정리하면 Table 1과 같다.

Table 1. Characteristic information for malicious code

Malicious code type	Feature Information
Backdoor	log information, hostname, OS name, access time, access period, IP address, Mac address, service port, processor creation path, created process name, file creation path, created file name, registry key, registry key value
Spyware	access time, IP address, open URL, registry key, registry key value, created registry, execution path, execution file name, processor creation path, created process name, API Call
Adware	access time, IP address, open URL, registry key, registry key value, created registry, execution path, execution file name, processor creation path, created process name
Virus	string information, execution script file name, registry key, registry key value, file creation path, created file name, processor creation path, created process name
Trojan	port information, registry key, registry key value, created registry, execution path, execution file name, processor creation path, created process name, file creation path, created file name
Worm	host name, IP address, protocol, file creation path, created file name, registry key value, created registry, port information, open URL, permission, API Call

그리고, 악의적인 코드들의 변종이나 새로운 악의적인 코드들은 기존의 악의적인 코드들과 유사한 특징 정보들을 가지고 있기 때문에 기존의 악의적인 코드들에 사용되는 주요 공통 특징 정보를 도출하는 것도 중요한 의미가 있다. 악의적인 코드들에 사용되는 주요 공통 특징 정보들로는 악의적인 코드을 식별할 수 있는 문자열, 악의적인 코드들이 사용하는 명령어들 가운데 있는 opcode 리스트의 해싱값, 악의적인 코드 파일이 다른 악의적인 파일에 의해 사용되는 API 명과 DLL, 악의적인 코드가 실행되는 과정에서 파일에 대해 읽기, 쓰기, 삭제 등이 수행된 모든 파일의 경로, 악의적인 코드가 실행되면서 사용되는 프로토콜, DNS, URL, IP 주소, 포트 번호, 원격지 주소와 같은 네트워크 정보, 악의적인 코드가 접근하고 수정하는 레지스트리의 키와 키 값, 레지스트리 경로, 악의적인 코드가 생성해서 사용되는 프로세스 이름과 프로세스 생성 경로를 들 수 있다. 공통 특징 정보를 정리하면 Table 2와 같고 악의적인 코드의 주요 특징 정보를 활용한다면 변형되거나 새로운 악의적인 코드를 비교적 효과적으로 탐지할 수 있다.

Table 2. Overview of the common feature information for malware

Feature	Description
string	string information
instruction	hash value of opcode list
API	API list, DLL list
file	created file name, file create path
registry	registry key, registry key value registry path
network	protocol, DNS, IP address, URL Port number, remote address
process	process name, process create path

4.2 탐지 성능 평가

분류된 악의적인 코드의 특징 정보를 사용하여 파이썬과 싸이킷 런(Scikit-learn)을 활용하여 성능 평가를 진행하였다. 제안 모델을 학습하는 머신러닝 알고리즘으로 신경망(NN:Neural Network)[19]와 서포트 벡터 머신(SVM:Support Vector Machine)[20]를 사용하여 성능 지표로정확도와 오탐율의 지표인 정상적인 코드를 악성코드로 탐지하는 FPR(False Positive rate)과 FNR(False Negative Rate)로 평가하였다. 정

확도와 FPR, FNR는 아래 식(2)를 통해 구할 수 있다.

$$\begin{aligned} \text{Accuracy} &= (TP+TN) / (TP+TN+FN+FP) \\ \text{FPR} &= FP / (FP+FN) \\ \text{FNR} &= FN / (FN+TP) \end{aligned} \quad (2)$$

식(2)에서 TP(True Positive)는 악의적인 코드를 정상적으로 악의적인 코드로 판별한 수이고 TN(True Negative)는 악의적인 코드가 아닌 코드를 정상적으로 악의적인 코드가 아니라고 판별한 수를 의미한다. 또한 FN(False Negative)sms 악의적인 코드를 정상적인 코드로 판별한 수이고 FP(False Positive)는 정상적인 코드를 악의적인 코드로 판별한 수를 말한다. 평가는 데이터 집합 중 80%는 훈련데이터로 20%는 테스트 데이터로 사용해서 평가하였고 결과는 Table 3과 같다.

Table 3. Performance evaluation result

Evaluation matrix	NN	SVM
Accuracy	96.5	96.7
FPR	0.04	0
FNR	0	0.1

Table 3에서 보는 것과 같이 두 학습 알고리즘에 대하여 탐지 정확도는 거의 비슷하게 나오고 오탐율의 지표인 FPR는 SVM이 좋게 나왔고 FNR는 NN이 좋게 나왔다. 탐지측면에서 평가 결과를 정리해보면 악의적인 코드를 탐지하는 학습 알고리즘은 FPR보다는 FNR이 적은 NN이 약간 좋다고 평가할 수 있다.

5. 결론

현대사회는 모든 일상적인 업무가 인터넷을 통하여 진행되고 있는데 인터넷은 많은 취약점이 존재하며 이를 이용한 지속적인 악의적인 행위가 발생되고 있다. 이런 악의적인 행위는 대부분 악의적인 코드에 의해 이루어지며 사용자 개입없이 네트워크를 통해 빠르게 확산되고 있고 손쉽게 인터넷을 통해 구입할 수 있는 구성 키트를 사용하여 다양한 변종과 새로운 악의적인 코드가 생성되고 있다. 또한, 현재 악의적인 코드들은 기존 탐지 방법을 회피하기 위해 다양한 지능적 기법을 사용하여 제작되기 때문에 새로운 악의적인 코드를 탐지하는 방법의 개발이 필요하다.

따라서 본 논문에서는 기존의 악의적인 코드뿐만 아니라 변종과 새로운 악의적인 코드를 효율적으로 탐지하기 위해 기존에 발생되었던 악의적인 코드의 내용과 구조를 기반으로 주요 특징을 변환, 차원축소, 특징 선택 단계를 수행하여 추출하는 방법을 제안하고 추출된 정보를 악의적인 코드별로 분류하였다. 그리고 분류된 특징 정보들을 기반으로 변형되거나 새롭게 등장하는 악의적인 코드를 분석하고 탐지하는데 사용할 수 있는 공통 특징 정보를 도출하였다. 그런 다음, 이런 정보를 사용해 학습하여 분류기를 생성하는 탐지 모델도 설계하였다. 설계된 모델을 이용하면 제한된 수의 특징 정보를 사용하여 탐지할 수 있기 때문에 기존의 탐지 시스템보다 좀 더 탐지 시간을 단축할 수 있고 그만큼 대응도 신속하게 이루어지 때문에 악의적인 코드에 의한 피해를 최소화 시킬 수 있다. 향후에는 악의적인 코드별 탐지율을 좀 더 높이기 위한 효과적인 특징 정보를 추출하기 위한 방법과 최적의 학습 방법에 대한 연구가 진행된 다음 최적의 분류기를 탑재한 지능적인 탐지 시스템을 구현하는 연구가 진행되어야 한다.

REFERENCES

[1] AVTEST. (2021). https://www.av-test.org/en/statistics/malware/.

[2] Symantec, Symantec internet security threat report. (2018). ISTR-23-2018

[3] Chionis, I., Nikolopoulos, S. D. & Polenakis I. (2013). *A Survey on Algorithmic Techniques for Malware Detection.* Proc. 2nd Int'l Symposium on Computing in Informatics and Mathematics (ISCIM'13). 29-34.

[4] Xu, X. & Wang, X. (2005). An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines. *Advanced Data Mining and Applications, Springer,* 696-703.

[5] M. Egele, T. S. Scholte, E. Kirda & C. Kruegel. (2012). A survey on automated dynamic malware-analysis techniques and tools. *ACM Computing Surveys(CSUR). 44(2),* 1-42.

[6] H. J. Gwon, S. W. Kim & E. G. Lim. (2012). An Malware Classification System using Multi N-gram. *Journal of Security Engineering, 9(6),* 531-542.

[7] H. S. Seo. J. S. Choi & P. H. Chu. (2009). Design of Classification Methodology of Malicious Code

in Windows Environment. *Journal of the Korea Institute of Information Security & Cryptology 19(2).* 83-92. DOI: 10.13089/JKIISC.2009.19.2.83

[8] Cyber Threat Trend Report. (2021). KISA. https://www.krcert.or.kr/data/reportView.do?bulletin_writing_sequence=36189

[9] Cyber Threat Trend Report. (2021). KISA. https://www.krcert.or.kr/data/reportView.do?bulletin_writing_sequence=36076

[10] M. Sikorski & A. Honig. (2012). *Practical Malware Analysis: the hands-on guide to dissecting malicious software.* No Starch Press.

[11] Chionis, I. Nikolopoulos, S. D. & Polenakis I. (2013). *A Survey on Algorithmic Techniques for Malware Detection.* Proc. 2nd Int'l Symposium on Computing in Informatics and Mathematics (ISCIM'13). 29-34.

[12] W. K. Lee, M. J. Lee & D. S. Seo. (2020). Application of Machine Learning Techniques for the Classification of Source Code Vulnerability. *Journal of The Korea institute of imformation security & cryptology, 30(4),* 735-743.

[13] Sihwail, R., Omar, K. & Ariffin, K. Z. (2018). A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis. *Int. J. Adv. Sci. Eng. Inf. Technol, 8(4-2),* 1662-1671. DOI: 10.18517/ijaseit.8.4-2.6827

[14] K. Rieck, T. Holz, C. Willems, P. Dussel & P. Laskov. (2008). *Leaming and classification of malware behavior.* in Proceedings of the 5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, 108-125.

[15] NSL-KDD dataset [online] available: http://nsl.cs.unb.ca/nsl-kdd/

[16] VX Heaven. (2018). http://83.133.184.251/virensimulation.org

[17] Wang, W., Zhang, X. & Gombault, S. (2009). Constructing Attribute Weights from Computer Audit Data for Effective Intrusion Detection. *Journal of Systems and Software, 82,* 1974-1981.

[18] S. Cateni, et al. (2012). Variable Selection and Feature Extraction through Artificial Intelligence Techniques, Multivariate Analysis in Management. *Engineering and the Science, chapter 6,* 103-118.

[19] Makandar, A. & Patrot, A. (2015). *Malware analysis and classification using artificial neural network.* IEEE. In 2015 International conference on trends in automation, communications and computing technology (I-TACT-15). 1-6.

[20] P. Manandhar. (2014). *A Practical Approach to Anomaly-based Intrusion Detection System by Outlier Mining in Network Traffic.* Masdar Institute of Science and Technology.

황 윤 철(Yoon-Cheol Hwang) [정회원]

· 2008년 2월 : 충북대학교 전자계산학과(이학박사)
· 2019년 3월~2021년 2월 : 가천대학교 소프트웨어 중심대학 사업단 소프트웨어교육센터 초빙교수
· 2021년 3월~현재 : 한남대학교 탈메이지 교양융합대학 조교수
· 관심분야 : 네트워크 및 웹 보안, IDS, ITS, Fusion IT Technology(AI)
· E-Mail : dolpin2010@gmail.com