Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.3, Jun. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.3.555

머신러닝 기반 악성 URL 탐지 기법*

한 채 림,[1†] 윤 수 현,[1] 한 명 진,[1] 이 일 구[2‡]
[1,2]성신여자대학교 (학생, 교수)

Machine Learning-Based Malicious URL Detection Technique*

Chae-rim Han,[1†] Su-hyun Yun,[1] Myeong-jin Han,[1] Il-Gu Lee[2‡]
[1,2]Sungshin Women's University (Undergraduate student, Professor)

요 약

최근 사이버 공격은 지능적이고 고도화된 악성코드를 활용한 해킹 기법을 활용하여 재택근무 및 원격의료, 자동산업설비를 공격하고 있어서 피해 규모가 커지고 있다. 안티바이러스와 같은 전통적인 정보보호체계는 시그니처 패턴 기반의 알려진 악성 URL을 탐지하는 방식이어서 알려지지 않은 악성 URL을 탐지할 수 없다. 그리고 종래의 정적 분석 기반의 악성 URL 분석 방식은 동적 로드와 암호화 공격에 취약하다. 본 연구에서는 악성 URL 데이터를 동적으로 학습하여 효율적으로 악성 URL 탐지하는 기법을 제안한다. 제안한 탐지 기법에서는 머신러닝 기반의 특징 선택 알고리즘을 사용해 악성 코드를 분류했고, 가중 유클리드 거리(Weighted Euclidean Distance, WED)를 활용하여 사전처리를 진행한 후 난독화 요소를 제거하여 정확도를 개선한다. 실험 결과에 따르면 본 연구에서 제안한 머신러닝 기반 악성 URL 탐지 기법은 종래의 방법 대비 2.82% 향상된 89.17%의 정확도를 보인다.

ABSTRACT

Recently, cyberattacks are using hacking techniques utilizing intelligent and advanced malicious codes for non-face-to-face environments such as telecommuting, telemedicine, and automatic industrial facilities, and the damage is increasing. Traditional information protection systems, such as anti-virus, are a method of detecting known malicious URLs based on signature patterns, so unknown malicious URLs cannot be detected. In addition, the conventional static analysis-based malicious URL detection method is vulnerable to dynamic loading and cryptographic attacks. This study proposes a technique for efficiently detecting malicious URLs by dynamically learning malicious URL data. In the proposed detection technique, malicious codes are classified using machine learning-based feature selection algorithms, and the accuracy is improved by removing obfuscation elements after preprocessing using Weighted Euclidean Distance(WED). According to the experimental results, the proposed machine learning-based malicious URL detection technique shows an accuracy of 89.17%, which is improved by 2.82% compared to the conventional method.

Keywords: Machine Learning, Malicious URL, Malware, Security, Static Detection

I. 서 론

정보화 시대가 도래하고 인터넷 환경이 발전하면서 지능화된 악성코드가 인터넷을 통해 급속히 유포되고 있다. 악성코드는 악의적인 목적을 위하여 작성된 코드이다. 네트워크 트래픽 발생, 시스템 성능 저

Received(02. 28. 2022), Modified(04. 08. 2022), Accepted(04. 10. 2022)
* 이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. 2020R1F1A1061107)과 2022년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원(P0008703, 2022년 산업혁신인재성장지원사업)을 받아 수행된 연구임.
† 주저자, 20200969@sungshin.ac.kr
‡ 교신저자, iglee@sungshin.ac.kr(Corresponding author)

하, 개인 정보 유출 등 다양한 증상을 유발시킬 수 있는 악성코드가 점차 지능화됨에 따라 피해 규모도 커지고 있다. 그림 1은 2017년부터 2020년까지의 랜섬웨어 공격 수와 피해 추정액을 나타낸다[1-5].

이러한 악성코드를 사용자의 PC에 감염시킬 수 있는 사이트를 악성코드 은닉사이트 또는 악성 URL(Uniform Resource Locator)이라고 한다. 악성 URL을 통한 공격은 사용자가 웹 사이트에 방문하거나, 이메일을 확인할 때 발생한다. 이때 소프트웨어 버그를 악용하여 악성코드를 실행하고, 데이터를 읽어와 공격자의 의도대로 사용자의 브라우저를 조종한다. 이와 같은 지능적인 공격에서 악성 URL은 악성코드를 배포하기 위한 중개자 역할을 한다.

그림 2는 악성코드 자체나 악성코드 유포 URL이 은닉된 국내 사이트의 탐지율을 나타낸 그래프이다[6-9]. 악성코드를 직접 유포하는 유포지(Distribution Site)와 유포지로 연결하는 악성 스

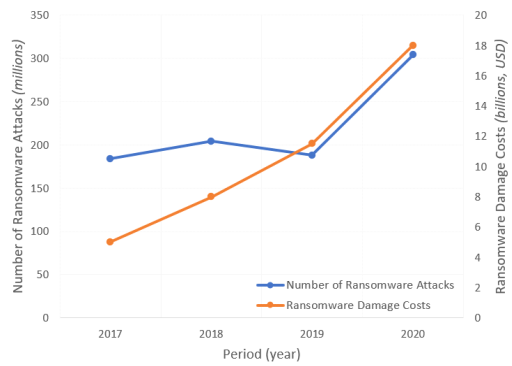

Fig. 1. Number of Ransomware Attacks and Ransomware Damage Costs

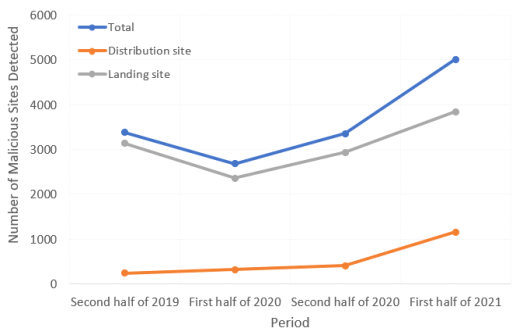

Fig. 2. Statistics of Malicious Code Hidden Site Detection

크립트가 삽입된 경유지(Landing Site)가 증가하는 추세를 보이며 악성코드 은닉 사이트는 매년 높은 탐지 건수를 보이고 있다. 한국인터넷진흥원에서 발간한 2021년 상반기 악성코드 은닉사이트 탐지 동향 보고서에 따르면 악성코드 경유지 사이트는 2020년 하반기 대비 31% 증가하였고 악성코드 유포지 사이트는 181% 증가하였다[9]. 신규 악성 URL과 그 전파 속도가 급증하는 추세이며, 인터넷에 연결된 사물인터넷이 악성 URL로 잘못 접속하게 하는 등 다양한 공격이 이루어지고 있기에 효율적인 악성 URL 탐지에 관한 연구가 필요하다.

최근 악성코드 은닉 여부를 탐지하기 위한 정적 분석 기법에 관한 연구가 활발하게 이루어지고 있지만, 순차 알고리즘에 의존하고 분산 컴퓨팅을 지원하지 않아서 막대한 런타임 오버헤드가 발생한다. 그리고 순수 메모리 기반 알고리즘은 제한된 메모리 환경에서 비효율적이다. 이와 같이 정적 분석 기법은 모델의 탐지율이 낮고, 탐지 대상이 난독화된 경우에는 탐지하는데 오래 걸린다. 그러므로 본 논문에서는 악성 URL을 효율적으로 탐지하기 위한 머신러닝 기반의 탐지 기법을 제안하였다. 제안하는 기법은 특징 선택 알고리즘을 사용해 악성 코드를 효과적으로 분류하고, 가중 유클리드 거리(Weighted Euclidean Distance, WED)를 활용하여 사전처리한 후 난독화 요소를 제거하는 3단계의 과정을 거쳐 탈독화를 진행한다.

본 연구의 기여점은 다음과 같다.

- 공유 데이터를 Native API 시퀀스(Native Application Programming Interface Sequence) 길이에 따라 소형 데이터로 축소함으로써 동기화 오버헤드를 줄이고 높은 병렬 처리를 달성한다.
- 특징벡터 평균치에 따라 난독화 요소를 효율적으로 제거함으로써 난독화 지연 문제를 해결한다.
- 실험 결과에 따르면 제안한 방법은 종래의 방법 대비 2.82% 향상된 89.17%의 높은 정확도를 달성할 수 있다.

본 논문 구성은 다음과 같다. 2장에서는 관련 연구를 비교 분석하고, 3장에서는 머신러닝 기반의 악성 URL 분석 엔진 기법을 제안한다. 4장에서는 평가 환경과 지표를 설명하고, 5장에서 평가 지표를 통해 제안한 프레임워크를 평가한다. 그리고 6장에

서 결론을 맺는다.

II. 관련연구

악성 URL 탐지와 관련된 대다수의 연구들이 주로 정적 분석을 수행하여 악성코드의 특징을 파악한다. 정적 분석은 프로그램을 실행하지 않고 악성코드를 식별하는 것을 의미한다. 본 장에서는 기존 악성 URL 분석 방식인 블랙리스트와 머신러닝 모델을 이용하는 방식에 대한 연구를 분석하고 한계점을 제시한다. 표 1은 블랙리스트 기반 악성 URL 분석 엔진에 대한 선행 연구를 분석한 내용이다.

PhishNet과 AutoBLG에서는 알고 있는 정보를 악성 목록에 추가하는 블랙리스트 기반 방식으로 악성 URL을 차단한다[12]. PhishNet은 피싱을 방지하기 위한 시스템이다. 다양한 악성 URL로부터 특징을 추출하여 블랙리스트의 항목과 비슷한 요소끼리 분류하고, 사용자가 접속한 URL에서 악성 URL 특징이 보이면 즉시 차단한다. 그리고 AutoBLG는 분석할 URL을 줄이면서 웹페이지 검색 공간을 확대하는 탐지 방법이다. 허니팟을 사용한 광범위한 분석을 통해 URL 확장, 필터링 및 확인 작업을 거친다.

그러나 블랙리스트 기반 탐지 기법은 미리 목록화한 정보만을 차단하기 때문에 난독화된 악성 URL을 탐지하기 어렵다. 그리고 신·변종 악성 URL에 대한 공격 대응이 어렵다. 블랙리스트는 수동으로 업데이트해야 하므로 어떤 탐지 기준으로 어떻게 탐지하느냐에 따라 정확성이 달라지는 문제가 있다. 또한, 특정 기간을 한정 지어 리스트를 생성할 경우 그 기간 외에 탐지율이 열화된다.

블랙리스트 방식의 문제를 해결하기 위해 등장한 머신러닝 모델은 악성 URL의 특징 정보를 추출해 학습한 후 입력된 URL을 정상 또는 악성으로 예측하여 분류한다. 머신러닝 기반의 악성 URL 탐지 모델 중 BigSpa와 HomDroid는 다중 머신러닝 모델 기반 방식으로 악성 URL을 차단한다. 이러한 머신

Table 1. Previous Research on Blacklist Detection

Previous Research	Feature	Limitation
PhishNet [10]	• Extract the characteristics of malicious URLs and classify them into elements similar to those of blacklist items • If see that feature in the URL, block it immediately	• Unable to respond to attacks on new and variant malicious URLs • New rules must be manually updated when the blacklist grows larger • Decreased accuracy
AutoBLG [11]	• Expand web page search spaces while reducing the URL to be analyzed • Using honeypot	• Decreased detection rate • Low classification accuracy • Unfavor to obfuscation

Table 2. Previous Research on Machine Learning Model

Previous Research	Feature	Limitation
BigSpa [13]	• Solving the static analysis problem through an inter procedure static analysis engine • Using data parallel algorithm	• Limited memory size • Distributed computing impossible
HomDroid [14]	• Detect covert malware by analyzing the homogeneity of the function call graph • An automated proto system	• Decreased accuracy • Overlook the call relationship between methods • Vulnerable to dynamic loading and encryption

러닝 모델 기반의 탐지 기법을 분석한 내용은 표 2와 같다.

BigSpa는 데이터 병렬 알고리즘을 사용하여 악성 실행 파일을 정적으로 탐지하는 머신러닝 모델이다. BigSpa는 각 절차마다 정적 분석 엔진을 통해 호출 그래프를 도출한다. 이때, 호출 그래프란 악성 코드를 그룹화하여 정적 분석을 수행한 함수를 의미한다.

HomDroid는 정적 기반 함수 호출 그래프를 추출하고, 그래프 간 동질성을 분석하여 은닉된 악성코드를 탐지한다. HomDroid 연구는 5개의 모델을 조합하여 악성 코드의 특징 벡터를 추출한다. 악성 데이터 샘플에서의 탐지 정확도는 92.2%이나, 단일 모델의 탐지 정확도 90.7.%와 큰 차이가 존재하지 않는다. 이처럼 선행 연구에서는 악성 URL를 탐지하고 예측하기 위한 다양한 머신러닝 알고리즘을 제안하고 있으나, 다중 머신러닝 알고리즘의 효율성을 최적화하지 못하고 있다.

머신러닝 기반 탐지 기법은 정적 분석을 수행하여 호출 그래프를 추출한다. 이때, 함수 호출 그래프는 상황 및 흐름에 민감하므로 통화 그래프로 인한 탐지 정확도가 열화된다[14]. 또한, 정적 분석 기법은 병렬화가 어렵기에 분산 컴퓨팅이 불가능하고 제한된 메모리로 인해 동적 로드 및 암호화에 취약하다.

III. 머신러닝 기반의 악성 URL 탐지 기법

본 장에서는 제안하는 머신러닝 기반의 악성 URL 탐지 기법의 메커니즘과 프레임워크를 설명한다. 그림 3은 악성 URL 탐지 프레임워크 구조도이다. 프레임워크는 크게 네 개의 기능적 모듈로 구성되며 각 모듈이 단계적으로 동작한다. 1단계 계층 검출(Layer Detection)은 수집 단계로, 악성 URL 유사도 기반 분류를 통해 난독화 흔적을 찾는다. 2단계 사전처리(Preprocessing)는 탐지 단계로 PE(Portable Executable) 헤더, 동적, 정적 정보를 바탕으로 감지한 패턴을 통해 구문 오류를 검사하고, 악성 여부를 판단한다. 3단계 난독화 제거(Remove Obfuscation)는 base64를 디코딩하고, cmdlet을 재정의하며, 정규식을 사용함으로써 난독화 요소들을 제거한다. 4단계 코드 실행(Script Execution)은 실행 단계로 결과를 기반으로 스크립트를 작동시킨다.

먼저, Layer Detection 단계에서 난독화 흔적

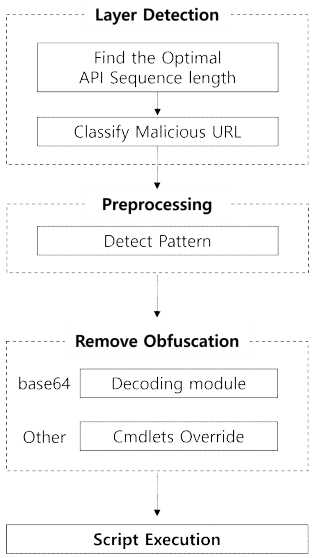

Fig. 3. Structure of Malicious URL Detection Framework

을 수집한다. 효율적인 난독화 요소 수집을 위해서는 최적의 Native API 시퀀스를 통해 악성코드를 분류해야 한다. 분석 플랫폼을 사용하여 실험 샘플에 대한 자동 분석을 수행하였고, 각 샘플의 Native API 호출 시퀀스를 기록하였다. Native API 호출 시퀀스는 분류 알고리즘으로 직접 처리할 수 없으므로 이를 특징 벡터로 변환하는 N-grams를 적용하였다. 각 짧은 시퀀스 특징의 가중치는 Term Frequency - Inverse Document Frequency (TF-IDF) 값으로 표현한다. 최적의 Native API 시퀀스 길이를 찾기 위해 특징 선택 알고리즘을 사용하였고, 특징 부분 집합을 시퀀스와 결합하여 최종적으로 14개의 특징으로 분류하였다[15].

1단계를 통해 양성과 악성 데이터에서 중복되는 Native API 시퀀스 특징의 가중치를 줄이고, 악성 데이터에서만 발견되는 특징 가중치 탐지율을 높인다. 그리고 공유 데이터를 Native API 시퀀스 길이에 따라 소형 데이터로 축소함으로써 정적 분석의 문제인 동기화 오버헤드를 개선한다.

Preprocessing에서는 PE 헤더 특성 분석 기술로 사전 처리를 진행한다. PE 파일에 난독화가 이루어지는 시점에 생성되는 헤더 파일의 특징 값으로 정상 파일과 악성 파일을 구분한다.

선정된 10개의 특징벡터의 각 악성과 정상 파일별 특징벡터 평균치를 계산하고, 가중 유클리드 거리를

통해 악성 파일 유무를 판단한다. $d(a,b)$를 a와 b 사이의 거리, CV_F, CV_L을 각각 임의의 프로그램 F와 L에 대한 특징벡터, Υ를 가시화를 위한 상수, W_i를 특징벡터 원소의 개수 i에 대한 파라미터 가중치 값, CVF,i, CVL,i을 각각 i에 해당하는 임의의 프로그램 F와 L의 특징벡터 값이라고 할 때, 특징벡터 평균치의 가중 유클리드 거리는 수식(1)과 같이 나타낸다[16].

$$d(CV_F, CV_L) = \gamma \cdot \frac{\sqrt{\sum_{i=1}^{n} W_i \cdot (CV_{F,i} - CV_{L,i})^2}}{\sum_{i=1}^{n} W_i} \quad (1)$$

위 수식을 이용하여 임의의 프로그램 F와 악성·정상 프로그램의 특징벡터 평균치 사이의 거리를 계산하여, 더 가까운 값을 탐지한다. 위 과정으로 특징벡터 평균치에 따라 난독화 요소를 효율적으로 제거함으로써 난독화 지연 문제를 해결하고, 높은 정확도로 악성 파일을 탐지할 수 있다.

Remove Obfuscation에서는 난독화를 제거한다. 난독화 제거는 은닉된 난독화 기술과 악성 URL 및 도메인 흔적을 알아내기 위한 단계이다. 난독화 제거를 통해 공격 코드 및 공격자가 탐지를 회피하기 위해 사용한 전략을 파악할 수 있다[17].

"-enc", "-Enc", "-EncodedCommand"와 같은 옵션을 사용하여 base64로 인코딩된 난독화의 경우 인코딩한 스크립트를 디코딩 모듈을 통해 디코딩함으로써 일반 스크립트를 가져온다[18]. 이후 문자 중간의 공백이나 문자 분할 등 난독화된 문자열에서 보이는 공통 패턴인 난독화 기호를 찾아 제거한다. base64로 난독화된 경우에는 난독화 요소를 제거하여 악성 URL을 확인한다.

base64로 인코딩되어 있지 않은 경우에는 cmdlet을 재정의한다. Invoke-Expression cmdlet은 문자열을 명령어로 실행시키고 식 또는 명령의 결과를 반환한다. 이를 사용하여 인코딩된 스크립트를 실행하고, 런타임에 스크립트의 디코딩된 텍스트를 가져온다[19]. 이후 난독화 기호를 찾아 제거한다.

이와 같은 과정을 통해 base64와 Invoke Obfuscation 난독화 툴로 생성할 수 있는 난독화를 해제할 수 있다. 이를 바탕으로 추가적인 파일을 가져오는 악성 스크립트와 실행 가능한 악성파일인 페이로드를 다운로드한 후 실행하는 파일 기반 악성 프로그램을 확인할 수 있다. 이 때, 프로그램에 의해 연결되는 악성 URL 및 IP 주소 정보를 확인할 수 있지만, Invoke-Expression에 의존하지 않는 난독화 유형이 포함된 경우에는 완전한 난독화가 어렵다. 하지만 이러한 특수한 경우를 제외하면 구문을 이해할 수 있는 경우가 일반적이다[20].

Layer Detection 단계와 Preprocessing 단계를 거쳐 난독화를 해제하면, Script Execution에서 코드를 실행한다.

IV. 평가 환경 및 방법

본 장에서는 제안한 기법의 성능을 평가하고, 분석하기 위한 평가 환경과 방법을 설명한다.

4.1 평가 환경

Windows XP 운영 체제 및 Program Files 디렉토리에서 946개의 양성 소프트웨어 샘플을 수집하였다. 본 양성 데이터는 PE 형식이며 그래픽, 멀티미디어, 사무용 등 다양한 유형의 소프트웨어로 구성되어있다. 그리고 VXHeavens에서 수집한 1414개의 악성 데이터를 실험에 활용했다[21]. 1414개의 악성 데이터에 대해 1298개의 데이터를 악성으로 판단하였으며, 946개의 정상 데이터 중 932개의 데이터를 정상으로 판단하였다. 이는 Preprocessing 단계에서 추출한 특징이 91.8%의 성능을 보임을 알 수 있으며, 이 단계에서 추출한 특징을 활용한 데이터 분류 정확도를 분석한 결과는 표 3과 같다.

학습한 모델은 파이썬(Python)의 사이킷런(Scikit learn) 라이브러리의 로지스틱회귀

Table 3. Significance of Entire Data and the Data Affected by Feature Extracted from Preprocessing

	Number of Data	Data Determin-ded to be Malicious	Data Determin-ded to be Normal
Malicious Data	1414	1298 (91.8%)	116 (8.2%)
Normal Data	946	14 (1.48%)	932 (98.52%)

(LogisticRegression) 클래스로 정확도를 평가하였다. 평가 기준은 오차 행렬(Confusion Matrix)를 활용해 제안한 기법이 예측한 값과 실제 데이터 값을 비교하여 평가를 진행하였다.

4.2 평가 방법

제안하는 악성 URL 탐지 프레임워크는 속도와 정확도로 평가할 수 있다. 이때, 정확도는 다음 세 가지의 평가지표로 측정 가능하다.

첫 번째는 정밀도이다. 정밀도는 측정값과 예측값 사이의 유사성에 대한 척도로서, 여러 번 측정 후 얼마나 일관성 있는 결과가 나타나는지를 설명하는 측정의 재현성을 나타낸다.

두 번째는 재현율이다. 재현율은 전체 항목 중 실제 올바르게 검색된 항목들의 비율이다. 정밀도와 재현율은 서로 보완적인 지표(Trade-off)이기 때문에 극단적인 수치 조작이 가능하다. 단적인 예로, 둘 중 하나만 점수가 좋고 다른 하나가 나쁜 분류는 성능이 좋지 않은 분류로 간주할 수 있으므로 정밀도와 재현율을 결합한 F1 점수를 활용해 평가한다.

세 번째는 AUC 점수이다. AUC 점수는 ROC에 기반한 이진 분류의 성능 측정에서 중요하게 사용되는 지표이다. ROC 곡선은 특이성과 재현율의 상관관계를 시각화한 곡선이며 분류의 성능 지표로 사용되는 것은 AUC 값으로 결정한다. AUC 점수는 ROC 곡선의 밑면적을 구한 것으로, 1에 가까울수록 좋은 수치이다.

악성 URL 탐지 프레임워크를 통해 정밀도, 재현율, AUC 점수를 측정할 수 있으며, 효율성을 정량적으로 입증할 수 있다.

V. 평가 결과 및 분석

본 장에서는 속도, 정밀도, 재현율, AUC 점수의 평가지표를 이용하여 제안한 악성 URL 탐지 기법과 최신이면서 가장 정확도가 높았던 HomDroid 모델의 성능을 동일한 환경에서 시뮬레이션하여 비교·분석한다.

5.1 속도

속도에 대한 평가는 평가 환경에서 수집한 샘플 데이터의 분류 처리 속도와 런타임 오버헤드로 측정한다.

Table 4. Comparison of Positive and Malicious Data Sorting Rates of the Proposed Malicious URL Detection Technique and HomDroid

	Proposed Technique	HomDroid
Data Sorting Rate	31.81MB/s	26.2MB/s

표 4는 제안한 악성 URL 탐지 기법과 HomDroid 모델의 분류 처리 속도를 비교한 표이다. 1초당 4.1의 평가 환경에서 수집한 양성 및 악성 프로그램을 분류하는 속도를 측정하였을 때, 제안한 악성 URL 탐지 기법은 31.81MB/s, HomDroid는 26.2MB/s로 측정되었다.

그림 4는 제안한 악성 URL 탐지 기법과 HomDroid 모델의 런타임 오버헤드를 비교한 누적분포함수이다. 이때, 런타임 오버헤드란 프로그램이 실행되고 있을 때 추가적으로 요구되는 시간을 의미한다. 처리해야 할 프로세스가 확장됨에 따라 오버헤드도 증가하지만, 처리량을 향상하기 위해서는 이러한 오버헤드를 최소화해야 한다. 그래프에서 x축은 초 단위의 런타임 오버헤드 시간, y축은 누적분포라고 설정하였다. 제안한 기법과 HomDroid에서 제시한 각각의 4단계의 악성코드 탐지 런타임을 기준으로 설정하였을 때, 제안한 악성 URL 탐지 기법의 평균 런타임은 11.02초, HomDroid는 13.38초로

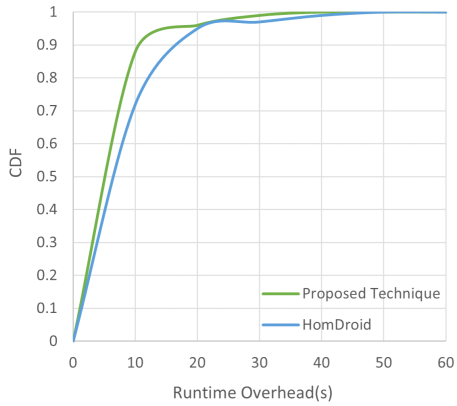

Fig. 4. The Cumulative Distribution Function (CDF) of runtime overhead of the Proposed Malicious URL Detection Technique and HomDroid

측정되었다.

5.2 정밀도와 재현율

분모 (TP+FP)는 예측을 양성으로 한 모든 데이터, 분자 FP는 예측과 실제 값이 양성으로 일치한 데이터의 비율이라고 할 때, 정밀도(Accuracy, A)를 수식(2)로 나타낸다.

$$A = \frac{TP}{TP+FP} \tag{2}$$

분모 (TP+FN)은 실제 값이 양성인 모든 데이터, 분자 TP는 예측과 실제 값이 양성으로 일치한 데이터의 비율이라고 할 때, 재현율(Recall rate, R)을 수식(3)으로 나타낸다.

$$R = \frac{TP}{TP+FN} \tag{3}$$

A를 정밀도, R을 재현율이라고 할 때, 정밀도와 재현율의 조화평균 값인 F1 점수를 수식(4)로 나타낸다.

$$F_1 = \frac{2AR}{A+R} \tag{4}$$

F1 점수는 정밀도와 재현율이 편향되지 않은 수치를 나타낼 때 상대적으로 높은 값을 가진다.

그림 5는 정밀도와 재현율의 임계값에 따른 F1 점수값 변화를 시각화한 그래프이다. 이때, 결정 임계값(Threshold)를 변수로 설정해 배열 X의 값이 결정 임계값보다 같거나 작으면 0으로, 크면 1로 반환하도록 설정하였다. Binarizer 클래스를 적용해 결정 임계값을 변화시키며 평가를 진행하였다.

아래의 그래프에서 x축을 결정 임계값, y축을 정밀도와 재현율, z축을 F1 점수로 설정하였을 때, 정밀도와 재현율은 서로 상호보완적인 관계가 성립한다. 정밀도는 임계값이 증가할수록 점점 증가하는 정비례 그래프를 보였으나, 재현율은 점차 감소하는 반비례 그래프를 보였다. F1 점수는 정밀도와 재현율의 오차가 작을수록 증가하는 양상을 보였다. 이를 통해 본 논문에서 제안하는 악성 URL 탐지 프레임워크의 정밀도와 재현율, F1 점수의 최적 결정 임계

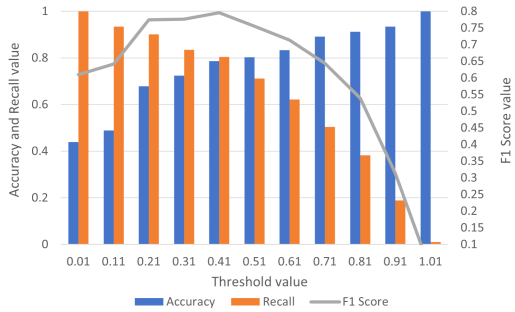

Fig. 5. Correlation Between Accuracy, Recall and F1 Score According to the Decision Threshold for the Proposed Malicious URL Detection Framework

값은 약 0.4임을 도출하였다.

5.3 AUC 점수

그림 6은 제안한 악성 URL 탐지 기법과 HomDroid 모델의 ROC 곡선을 시각화한 그래프이다. x축은 음성을 양성으로 예측한 비율(FPR)로 설정하고, y축을 재현율(TPR)로 설정하였을 때, ROC 곡선은 두 모델 모두 log x에 가까운 추세가 없는 지수 모형이 도출된다. 특히, FPR이 작을 때 재현율의 증가폭은 가파르게 상승한다. 아래의 그래프에서 점선으로 표시된 직선은 곡선의 최젓값을 나타내는데, ROC 곡선이 이 직선에서 멀어질수록 좋은 성능을 나타낸다.

ROC 곡선에 기반한 AUC 값은 1에 수렴할수록

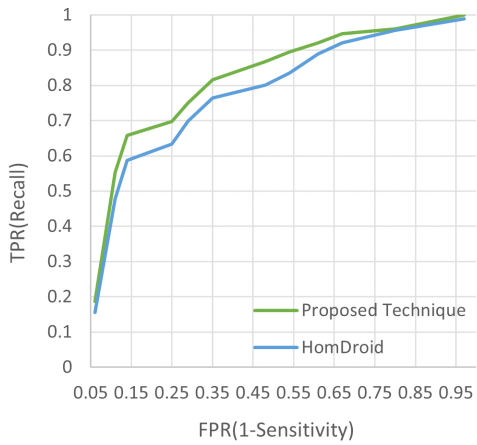

Fig. 6. ROC Curve of the Proposed Malicious URL Detection Technique and HomDroid

좋은 성능 수치를 가지는데, HomDroid 모델의 AUC 값은 0.8635로, 제안한 기법의 AUC 값은 0.8917로 높은 정확도를 보인다.

VI. 결 론

악성 URL을 활용한 공격이 점점 지능화되고, 그 피해도 급증하는 추세임에 따라 악성 URL을 효율적으로 탐지하고 차단하는 연구가 필요하다.

본 논문에서는 머신러닝 기반 악성 URL 탐지 프레임워크를 제안하였다. 기존 연구에서 제안하고 있는 블랙리스트, 다중 머신러닝 기반의 악성 URL 분석 엔진은 난독화가 어렵고, 신·변종 악성 URL 및 암호화 공격에 대처하지 못한다는 한계가 있다. 그러나 본 논문에서 제안한 악성 URL 탐지 기법은 계층 탐지, 사전처리, 난독화 요소를 해제한다. 신·변종 악성 URL 및 암호화 공격에 대한 분류 정확도를 향상하기 위해 Native API 시퀀스 가중치를 TF-IDF 값으로 표현하여 1차 분류를 진행하고, 각 특징벡터의 평균치를 계산하여 가중 유클리드 거리를 통해 2차 분류를 진행한다. 제안 기법은 종래의 방법에 비해 2.82% 향상된 89.17%의 탐지율을 보이는 것을 확인하였다.

향후 제안한 4단계 프레임워크를 기반으로 탐지의 정확도 및 속도를 더 향상시킬 수 있는 방안을 탐구할 것이며, 나아가 신규 gLTD 관련 악성 URL 탐지 기법에 대한 후속 연구를 진행할 예정이다.

References

[1] Nisar, K., Iqbal, M.W., Sarwar, M.I., Ahmad, S., and Arfan, M.T., "Ransomware Dissemination and Mitigation Techniques-A Review." The 7th International Conference on Next Generation Computing 2021, pp. 173-177, Dec. 2021.

[2] N. Caporusso, S. Chea and R. Abukhaled, "A game-theoretical model of ransomware." International Conference on Applied Human Factors and Ergonomics, pp. 69-78, July 2018.

[3] C. Bansal, P. Deligiannis, C. Maddila and N. Rao, "Studying ransomware attacks using web search logs." Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1517-1520, July 2020.

[4] Ömer Aslan Aslan and Refik Samet, "A Comprehensive Review on Malware Detection Approaches," IEEE Access, vol. 8, pp. 6249-6271, Jan. 2020.

[5] David S. Wall, "The Transnational Cybercrime Extortion Landscape and the Pandemic: changes in ransomware offender tactics, attack scalability and the organisation of offending." European Law Enforcement Research Bulletin SCE 5, pp. 45-60, Aug. 2021.

[6] Korea Internet & Security Agency, "Report on the detection trend of malicious code hidden sites in the second half of 2019", https://www.krcert.or.kr/data/reportView.do?bulletin_writing_sequence=35248, accessed Jan.7,2022, 2020

[7] Korea Internet & Security Agency, "Report on the detection trend of malicious code hidden sites in the first half of 2020", https://www.krcert.or.kr/data/reportView.do?bulletin_writing_sequence=35537, accessed Jan. 7, 2022, 2020

[8] Korea Internet & Security Agency, "Report on the detection trend of malicious code hidden sites in the second half of 2020", https://www.krcert.or.kr/data/reportView.do?bulletin_writing_sequence=35872, accessed Jan. 7, 2022, 2021

[9] Korea Internet & Security Agency, "Report on the detection trend of malicious code hidden sites in the first half of 2021", https://krcert.or.kr/data/reportView.do?bulletin_writing_sequence=36189, accessed Jan.7,2022, 2021

[10] P. Prakash, M. Kumar, R. R. Kompella and M. Gupta, "PhishNet: Predictive Blacklisting to Detect Phishing Attacks," 2010 Proceedings IEEE INFOCOM, pp. 1-5, Mar. 2010.
[11] B. Sun, M. Akiyama, T. Yagi, M. Hatada and T. Mori, "AutoBLG: Automatic URL blacklist generator using search space expansion and filters," 2015 IEEE Symposium on Computers and Communication (ISCC), pp. 625-631, Jul. 2015.
[12] Chang, H.Y., Kim, M.J., Kim, D.J., Lee, J.Y., Kim, H.K., and Cho, S.J., "An implementation of system for detecting and filtering malicious URLs." Journal of KIISE: Computing Practices and Letters, 16(4), pp. 405-414, Apr. 2010.
[13] Z. Zuo, R. Gu, X. Jiang, Z. Wang, Y. Huang, L. Wang and X. Li, "BigSpa: An Efficient Interprocedural Static Analysis Engine in the Cloud," 2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 771-780, May 2019.
[14] Y. Wu, D. Zou, W. Yang, X. Li and H. Jin, "HomDroid: detecting Android covert malware by social-network homophily analysis." Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, pp. 216-229, Jul. 2021.
[15] Jinrong Bai, Qibin Shi, "Malware Detection Method based on Dynamic Variable Length API Sequence," 2019 12th International Symposium on Computational Intelligence and Design (ISCID), pp. 285-288, Dec. 2019.
[16] Choi, Y.S., Kim, I.K., Oh, J.T., and Ryu, J.C., "PE Header Characteristics Analysis Technique for Malware Detection." Convergence Security Journal, 8(2), pp. 63-70, Jun. 2008.
[17] D. Ugarte, D. Maiorca, F. Cara and G. Giacinto, "PowerDrive: Accurate de-obfuscation and analysis of powershell malware." International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 240-259, Jun. 2019.
[18] Song, J., Kim, J., Choi, S., Kim, J., and Kim, I., "Evaluations of AI based malicious PowerShell detection with feature optimizations." ETRI Journal, 43(3), pp. 549-560, Apr. 2021.
[19] Carolyn J. Holz, "Investigating representations of obfuscated malicious PowerShell", Ph.D. Thesis, Massachusetts Institute of Technology, 2019.
[20] Giuseppe Mario Malandrone, Giovanni Virdis, Giorgio Giacinto and Davide Maiorca, "PowerDecode: a PowerShell Script Decoder Dedicated to Malware Analysis." Proceedings of the Italian Conference on Cybersecurity 2021, pp. 219-232, Apr. 2021.
[21] Pengtao Zhang, Ying Tan, "Class-wise information gain," 2013 IEEE Third International Conference on Information Science and Technology (ICIST), pp. 972-978, Mar. 2013.

〈저 자 소 개〉

한 채 림 (Chae-rim Han) 학생회원
2020년 3월~현재: 성신여자대학교 융합보안공학과 학사
〈관심분야〉 악성코드, 융합보안, 정보보호

윤 수 현 (Su-hyun Yun) 학생회원
2020년 3월~현재: 성신여자대학교 융합보안공학과 학사
〈관심분야〉 융합보안, 정보보호, 모의해킹

한 명 진 (Myeong-jin Han) 학생회원
2021년 3월~현재: 성신여자대학교 융합보안공학과 학사
〈관심분야〉 정보보호, 모의해킹

이 일 구 (Il-Gu Lee) 정회원
2003년 2월: 서강대학교 전자공학과 졸업
2005년 2월: KAIST 정보통신대학원 석사
2016년 2월: KAIST 전산학부 박사
2005년 2월~2017년 2월: 한국전자통신연구원 5G기가통신시스템연구본부 선임연구원
2017년 3월~현재: 성신여자대학교 미래융합기술공학과/융합보안공학과 조교수
〈관심분야〉 융합보안, 미래융합기술, 정보통신