# PIN 과 머신러닝을 이용한 효과적인 악성코드 판별 기법

김창현, 김태성, 박용수
한양대학교, 한양대학교, 한양대학교

changhyun4@naver.com, wowboya@hanyang.ac.kr, yongsu@hanyang.ac.kr

# An Effective Malware Detecting Technique using PIN and Machine Learning

Kim Chang Hyun, Kim Tae Sung, Yongsu Park
Hanyang Univ., Hanyang Univ., Hanyang Univ.

요 약

본 연구는 악성코드 피해를 최소화하기 위해 머신러닝을 이용해 악성코드 판별을 자동화하는 것을 목표로 한다. 이를 위해 정상 프로그램과 악성코드 샘플을 각각 500 개씩 수집하고 Intel PIN DBI (Dynamic Binary Instrumentation)를 이용하여 Feature(API 사용여부, Instruction Frequency, API 실패율)들을 추출하였다. 각 Feature 들에 해당하는 데이터를 추출하고 Random Forest 알고리즘을 이용하여 인공지능을 학습시켰다. Feature 를 다르게 설정하여 총 세 번의 실험을 수행하였는데, 결과적으로 위 3 가지의 Feature 를 모두 사용했을 경우 정확도가 가장 높았다.

## Ⅰ. 서 론

네트워크 통신을 하다보면 악성코드가 유무선 통신을 타고 들어올 수 있다. 이러한 악성코드들은 수신자의 통신기기에 들어와 여러 문제(정보갈취, 손상 등)를 낳는다. 이에 대해서 우리는 주로 백신을 통해 들어온 파일이 악성코드인지 아닌지에 대해 판단한다. 그러나, 실행파일의 경우에는 정적분석만으로는 알 수 없는 경우가 많다. 따라서, 우리는 이에 대해 동적으로 분석할 수 있어야 한다. 백신이 주로 사용하는 악성코드 탐지 방법은 시그니쳐 기반 탐지방식과 휴리스틱 기반 탐지방식이 있다[1]. 시그니쳐 기반 탐지방식은 악성코드의 hash 를 저장해 놓고, 파일과 비교해 악성코드를 탐지하는 기법이다. 정확도가 매우 높고, 속도가 매우 빠른 장점이 있지만, 파일의 내용이 조금만 달라져도 hash 값은 바뀌기 때문에 탐지할 수 없다는 단점이 있다. 휴리스틱 기반 탐지방식은 악성코드에서 자주 발견되는 패턴을 미리 정의해 놓고, 의심스러운 행동을 하는 것이 발견될 경우 악성코드로 판단하는 것이다. 이 방법은 오진 가능성이 크고 속도가 느리다는 단점이 있다. 위에서 언급한 두 가지 악성코드 탐지 기법 중에서 자동화할 수 있는 것은 휴리스틱 기반 탐지 방식이다. 머신러닝을 통해 인공지능이 스스로 악성코드의 특징을 찾아 내고, 실제 분류까지 할 수 있도록 만들어 내는 것이 본 논문의 목표이다. 악성코드인지 아닌지 분류하는 것은 binary classification 문제로 볼 수 있는데, 이를 해결하는 전통적인 머신러닝 알고리즘에는 SVM, Random forest 등이 있다.[2] 이러한 알고리즘들을 사용하기 위해서는 우선 Feature 들을 설정해야 한다. 그리고 나서 동적분석 또는 정적분석을 통해 각 Feature 에 해당하는 데이터를 구해야 하고 마지막으로 추출한 데이터를 전처리하여 csv 파일로 만들어 머신러닝 알고리즘의 입력으로 사용될 수 있도록 가공해야 한다.

PIN[3]을 이용해 얻은 데이터를 전처리하여 Random forest 알고리즘으로 인공지능을 학습시켰다. 이 때, API 사용여부 + Instruction Frequency + API 실패비율을 모두 학습에 사용한 경우에 가장 높은 정확도(86.2%)를 보여주었다.

## Ⅱ. 본론

우리는 Intel 에서 제공하는 PIN DBI(Dynamic Binary Instrumentation)를 이용하였다[3]. PIN 은 실행파일 분석기로서 Windows / Linux 상의 x86, 64 바이너리를 지원한다. PIN 의 강력한 점은 대부분의 안티디버깅 기술을 우회할 수 있다는 것과, C++ 기반으로 되어 있어서 추출하고 싶은 데이터만 선택해서 빠르게 추출할 수 있다는 것이다. VMware 와 같은 가상환경에서 분석을 시도하면 가상환경을 인식해서 실제 환경과는 다른 행동을 보이는 Anti VM 기술을 사용하는 악성코드들도 분석할 수 있다.

학습에 사용한 Feature 는 먼저 API 사용여부이다. 특정 API 사용여부는 윈도우에서 동작하는 악성코드들은 어떤 행위를 할 때 API 함수를 호출한다. 따라서 모든 API 의 사용 여부를 조사하면 악성코드가 어떤 의심스러운 행동(파일 생성/수정/삭제, 소켓 연결, 레지스트리 변경 등)을 하는지 알아낼 수 있다. 그러나, API 의 종류는 매우 많기 때문에 모든 API 를 Attribute 로 사용할 수는 없다. 따라서 업로드 된 샘플 프로그램을 모두 분석하여, 악성코드만 사용하는 API 와, 정상 파일만 사용하는 API, 자주 사용하는 API 200 개를 추출하였고, 여기에 악성코드에서 자주 사용하는 API 100 여개를 추가했다. 이렇게 하여 총 600 개의 API 를 Attribute 로 사용하였다[1]. 각 API 들의 사용여부를 One-hot encoding 하여 0 또는 1 로 표현하였다.

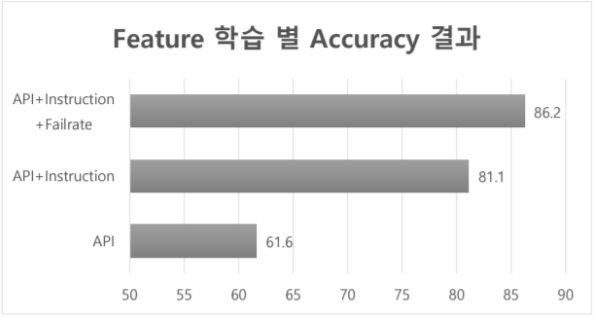
다음 Feature 는 Instruction Frequency 이다. Instruction Frequency 는 Instruction 의 사용 빈도를 나타낸 것인데, Instruction 의 종류는 매우 많으므로 자주 사용되는 instruction 만을 20 개 정도 선택하여 Attribute 로 사용했다.

마지막으로 사용한 Feature 는 API 실패비율이다. API 실패비율은 모든 API 호출 중 실패한 호출의 비율을 나타낸 것이다. 우리 예상으로는 악성코드는 모든 경우의 수를 이용해 악성 행위를 하기 때문에 API 의 실패 비율이 정상 파일보다 높을 것이라고 생각했다. 실제로 악성코드의 API 호출 실패 비율이 약간 높은 것을 확인할 수 있었다.

## Ⅲ. 결론

지금까지 위에서 설명한 Feature 를 실제로 데이터로 추출하기 위해 PIN Tool 을 수정하였다. PIN Tool 이란 어떤 정보를 추출할지 프로그래밍 하여, 이를 PIN 을 이용하여 분석대상 프로세스를 PIN 위에서 실행시킬 때 사용하는 dll 을 의미한다. PIN 의 가장 큰 장점은 이 PINtool 을 사용자가 커스터마이징 할 수 있다는 점이다.

PIN 을 이용해 얻은 데이터를 전처리하여 Random forest 알고리즘[2]으로 인공지능을 학습시켰다. 우리는 총 세 번의 실험(1. API 사용여부, 2. API 사용여부 + Instruction Frequency, 3. API 사용여부 + Instruction Frequency + API 실패비율)을 하였다. 실험 결과는 다음 그래프와 같이 나왔다.




학습 결과를 보면, 위의 세 개의 Feature 를 모두 학습한 경우에 가장 높은 정확도(86.2%)를 보여주었다.

ACKNOWLEDGMENT

This work was funded by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT), grant number 2020R1F1A1048443.

## 참 고 문 헌

[1] Hauri, “악성코드의 악성 행위와 탐지 기법” [Online]. Available: https://hauri.co.kr/security/issue_view.html?intSeq=93&page=21&article_num=92, accessed on Jul. 29, 2020.

[2] Breiman, “Random Forests”, Machine Learning, 45(1), 5-32, 2001.

[3] Intel, “Pin - A Dynamic Binary Instrumentation Tool” [Online]. Available: https://software.intel.com/content/www/us/en/develop/articles/pin-a-dynamic-binary-instrumentation-tool.html, accessed on Jul. 29, 2020.