Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.31, NO.6, Dec. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.6.1105

머신러닝을 활용한 행위 및 스크립트 유사도 기반 크립토재킹 탐지 프레임워크*

임 은 지,[1†] 이 은 영,[2] 이 일 구[3‡]
[1]연세대학교 (대학원생), [2,3]성신여자대학교 (대학원생, 교수)

Behavior and Script Similarity-Based Cryptojacking Detection Framework Using Machine Learning*

EunJi Lim,[1†] EunYoung Lee,[2] IlGu Lee[3‡]
[1]Yonsei University (Graduate student),
[2,3]Sungshin Women's University (Graduate student, Professor)

요 약

최근 급상승한 암호 화폐의 인기로 인해 암호 화폐 채굴 악성코드인 크립토재킹 위협이 증가하고 있다. 특히 웹 기반 크립토재킹은 피해자가 웹 사이트에 접속만 하여도 피해자의 PC 자원을 사용해 암호 화폐를 채굴할 수 있으며 간단하게 채굴 스크립트만 추가하면 되기 때문에 공격이 쉽고 성능 열화와 고장의 원인이 된다. 크립토재킹은 피해자가 피해 상황을 인지하기 어렵기 때문에 크립토재킹을 효율적으로 탐지하고 차단할 수 있는 연구가 필요하다. 본 연구에서는 크립토재킹의 대표적인 감염 증상과 스크립트를 지표로 활용하여 효과적으로 크립토재킹을 탐지하는 프레임워크를 제안하고 평가한다. 제안한 크립토재킹 탐지 프레임워크에서 행위 기반 동적 분석 기법으로 컴퓨터 성능 지표를 학습한 K-Nearest Neighbors(KNN) 모델을 활용했고, 스크립트 유사도 기반 정적 분석 기법은 악성 스크립트 단어 빈도수를 학습한 K-means 모델을 크립토재킹 탐지에 활용했다. 실험 결과에 따르면 KNN 모델은 99.6%의 정확도를 보였고, K-means 모델은 정상 군집의 실루엣 계수가 0.61인 것을 확인하였다.

ABSTRACT

Due to the recent surge in popularity of cryptocurrency, the threat of cryptojacking, a malicious code for mining cryptocurrencies, is increasing. In particular, web-based cryptojacking is easy to attack because the victim can mine cryptocurrencies using the victim's PC resources just by accessing the website and simply adding mining scripts. The cryptojacking attack causes poor performance and malfunction. It can also cause hardware failure due to overheating and aging caused by mining. Cryptojacking is difficult for victims to recognize the damage, so research is needed to efficiently detect and block cryptojacking. In this work, we take representative distinct symptoms of cryptojacking as an indicator and propose a new architecture. We utilized the K-Nearst Neighbors(KNN) model, which trained computer performance indicators as behavior-based dynamic analysis techniques. In addition, a K-means model, which trained the frequency of malicious script words for script similarity-based static analysis techniques, was utilized. The KNN model had 99.6% accuracy, and the K-means model had a silhouette coefficient of 0.61 for normal clusters.

Keywords: Malware Detection, Machine Learning, Dynamic Analysis, Static Analysis, Cyber Security

Received(10. 08. 2021), Modified(11. 24. 2021), Accepted(11. 24. 2021)

* 이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No. 2020R1F1A1061107)과 2021년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의지원(P0008703, 2021년 산업혁신인재성장지원사업)을 받아 수행된 연구임.

† 주저자, traveler5260@yonsei.ac.kr
‡ 교신저자, iglee@sungshin.ac.kr(Corresponding author)

I. 제안 배경

2009년 온라인상에서 제3자 없이 신뢰할 수 있는 거래를 보장하는 전자 화폐 시스템인 비트코인이 등장하였다. 비트코인은 노드들의 컴퓨팅 자원을 사용해 시스템의 신뢰도를 보장하는 Proof of Work(PoW)방식으로 운영된다. 구성원들은 자신의 컴퓨팅 파워를 소모하여 암호 화폐를 채굴한다. 하지만 채굴 경쟁으로 인해 점점 높은 하드웨어 성능을 필요로 하게 되었다.

코인하이브(Coinhive)는 브라우저 마이닝 기반으로 모네로를 채굴할 수 있는 Application Programming Interface(API)를 제공한다. 코인하이브의 목표는 기부, 광고 수익 없이 웹 사이트를 운영할 자금을 확보하는 방법을 제안하는 것이다. 하지만 코인하이브는 방문자가 많은 웹 사이트에 채굴 스크립트를 삽입하거나, 방문자의 동의 없이 방문자의 PC 자원을 사용하여 공격자의 암호 화폐를 채굴하는 크립토재킹에 악용되었다. 크립토재킹은 랜섬웨어처럼 피해자에게 금전을 요구하거나 피해자의 PC에 있는 파일을 암호화하는 등 가시적인 피해가 없어 피해자가 피해 사실을 빠르게 알아차릴 수 없다.

2019년 3월, 대표적으로 브라우저 기반 크립토재킹으로 악용되었던 코인하이브의 폐쇄 이후, 크립토재킹 피해가 현저히 감소하였다. 하지만 2020년의 크립토재킹 발생 건수는 81,901,858건으로 전년 대비 28% 증가한 것으로 보아 공격자들이 지속적으로 새로운 방법을 고안해 크립토재킹 공격을 진행하고 있음을 알 수 있다[1].

크립토재킹 피해를 줄이기 위해서는 효율적으로 크립토재킹을 탐지하고 차단할 방법이 필요하다. 본 논문의 2장에서는 크립토재킹 탐지와 관련된 연구를 정적 탐지 방법과 동적 탐지 방법으로 나누어 설명한다. 3장에서는 크립토재킹을 분석하고, 4장에서는 정적 탐지와 동적 탐지 단계를 결합하여 크립토재킹을 효과적으로 탐지할 수 있는 프레임워크를 제안한다. 5장은 제안한 프레임워크를 평가한다.

II. 관련 연구

2.1 정적 탐지 방법

정적 탐지는 코드, Uniform Resource Locator(URL) 등 악성코드를 실행시키지 않아도 알 수 있는 정보를 사용하여 공격을 식별한다.

AntiMiner[7], MinerBlock[8], No Coin[9]에서는 이미 알려진 정보를 차단 목록에 추가하는 블랙리스트 기반 방식으로 크립토재킹을 차단한다.

AntiMiner와 No Coin은 크립토재킹을 방지할 수 있는 브라우저 확장 프로그램이다. 이미 알려진 악성 도메인을 블랙리스트에 추가하여 사용자가 해당 도메인에 방문하면 접속을 차단한다.

MinerBlock은 브라우저 기반 암호 화폐 채굴을 차단하는 브라우저 확장프로그램이다. 크립토재킹에 사용된 리퀘스트, 스크립트 목록을 수집하여 사용자가 접속한 사이트에서 해당 리퀘스트, 스크립트를 사용하면 접속을 차단한다.

하지만 블랙리스트 기반 차단 방식은 이미 알려진 정보만을 차단할 수 있으므로 난독화 등 약간의 우회 기법만 적용해도 탐지할 수 없으며, 알려지지 않은 공격에 유연하게 대처할 수 없다. 또한, 크립토재킹 차단율은 수집된 데이터의 양과 비례하기 때문에 메모리 저장 공간과 컴퓨팅 파워가 제한된 개인 PC에서 사용하기에는 한계가 있다.

MineSweeper[10]는 브라우저 기반 크립토재킹 탐지 시스템이다. 첫 번째로 웹 어셈블리 모듈을 정적 분석하여 해시 함수를 사용하는지 확인하고, 두 번째로 CPU 캐시 이벤트를 모니터링해서 메모리 접근 패턴을 파악하여 크립토재킹을 탐지한다.

Muhammad Saad외 2인은 자바스크립트 코드를 클러스터링하여 크립토재킹을 탐지하였다[11]. 자바스크립트 코드의 순환 복잡도, 순환 복잡도 밀도, 유지 보수 스코어, 코드 줄 수를 퍼지 C-means 클러스터링 알고리즘을 사용해 클러스터링 하였고, 96.4%의 정확도를 보였다.

하지만 여전히 코드 난독화에 취약하고, 두 방법 모두 탐지율을 높이기 위해서 리소스 사용량, 네트워크 정보 등 동적 탐지 방법을 함께 사용해야 했다.

2.2 동적 탐지 방법

동적 탐지는 공격 상황에서 발생하는 크립토재킹의 행위 특징을 분석하여 악성 행위를 탐지한다.

Mining Hunter[2]는 크립토재킹을 식별할 수 있는 웹 크롤링 프레임워크이다. 브라우저 기반 마이닝은 웹 소켓 트래픽이 Java Script Object Notation(JSON)으로 이루어져 있어 수집이 용이하다. 크롬 개발자 도구 프로토콜을 사용하여 웹 소

캣 트래픽과 기타 지표에서 URL, 특정 문자열, 특정 해시값 등을 분석하여 크립토재킹을 확인한다. Mining Hunter는 난독화하여도 크립토재킹을 탐지할 수 있어 블랙리스트 기반 차단 방식보다 효과적이다.

Tanana외 1인이 제안한 크립토재킹 탐지 알고리즘[3]은 동적 분석 결과를 토대로 CPU 사용량, 램 사용량 등의 크립토재킹 특징을 이용한 알고리즘을 제안하였다. 최종적으로 브라우저 기반 크립토재킹 40개 중 32개를 탐지하였다.

SEISMIC[4]은 크립토재킹을 탐지하는 인라인 스크립트 모니터이다. 인라인 스크립트를 모니터링하여 마이닝 작업에 필수적인 암호화를 탐지한다. 이때, 웹 어셈블리의 opcode를 Support Vector Machine(SVM)를 사용해 인라인 스크립트의 중요도를 판단한다.

CMTracker[5]는 해시 기반 프로파일러와 스택 구조 기반 프로파일러를 통해 크립토재킹을 탐지한다. 해시 기반 프로파일러는 스크립트 내부에 해시 계산에 사용되는 스크립트가 있는지 확인한 뒤, 해시에 사용된 웹사이트 누적 시간을 계산하여 해시 실행 시간이 10%이상일 때, 크립토재킹으로 판단한다.

CoinPolice[6]는 크립토재킹 동적 분석 데이터를 분석하여 크립토재킹 탐지 방법을 연구하였다. CPU 사용량, Hyper Performance Counters(HPC), 자바스크립트와 웹 어셈블리 실행시간을 데이터로 사용하고, 심층 신경망 분류기를 구현하여 최종적으로 97.87%의 크립토재킹 탐지율을 확인했다.

하지만 동적 탐지 방법은 크립토재킹에 피해를 입어 증상이 나타난 뒤에야 차단할 수 있다. 또한, 다양한 동적 탐지 요소 간의 유기적인 관계가 드러나지 않는 모델이 대다수이기 때문에 요소 간 관계를 파악하여 효과적으로 탐지하는 방법이 필요하다.

III. 크립토재킹 분석

3.1 크립토재킹 원리

웹 기반 크립토재킹[12]은 흔히 'Drive-By Mining'이나 'Drive-By 크립토재킹'으로 불린다. 특정 웹사이트에 접속한 경우, 웹 사이트 내의 스크립트를 통해 암호화폐를 채굴할 수 있게 된다. 비트코인의 경우 막대한 처리 능력을 요구하기 때문에, 주로 모네로를 활용하게 된다. 그리고 모네로는 익명성이 상대적으로 우수하므로 해커들이 추적을 피하기 용이하여 크립토재킹 공격의 주요 대상이 되고 있다. 악성 자바스크립트[13]를 통해 악성 자바 파일 또는 웹 어셈블리 파일을 로드하게 되며, 채굴 이후 공격자의 지갑 주소에 가상화폐가 전송된다. 공격자의 입장에서는 단순한 웹 접속만으로 채굴이 가능하므로 간편한 채굴 방식으로 이용된다. 홈페이지에 스크립트를 추가하기만 하면 언제든지 방문자들의 PC를 해킹하여 가상화폐 채굴기를 심을 수 있다.

3.2 코인 하이브

코인하이브[14]란, 웹사이트에 설치하도록 고안된 스크립트에 의존하는 가상화폐 채굴 서비스를 말한다. 이 기술은 이용자들에게 방해가 되는 광고를 실행하지 않고 수익을 올리기 위해 개발되었다. 하지만, 사이버 범죄자들은 이 기술을 악용하기 시작했고, 자바스크립트를 통해 웹사이트에 사용자 모르게 혹은 허가 없이 접근하면서 심각한 멀웨어 중 하나로 자리 잡게 되었다. Texthelp에서 만든 Browsealoud[15]라는 합법적인 플러그인에 코인하이브를 연결하거나, 유튜브 광고[16]에 크립토재킹 코드를 숨기는 등 여러 가지의 시도가 있었다. 그러나, 1년에 모네로[17]의 가치가 85% 넘게 추락하면서, 하드 포크 및 알고리즘 업데이트를 통해 코인하이브 사이트를 중단을 선언하게 되었다. 이로 인해, 백신 제품과 광고 차단 확장 프로그램 등에서 코인하이브의 도메인의 사용이 금지되었다. 여기서 주목해야 할 점은, 코인하이브가 수많은 범죄의 수단으로 활용되었지만, 재정적인 문제로 폐쇄가 된 것이지 불법적인 활동 때문은 아니라는 것이다. 결과적으로, 사이트의 폐쇄가 크립토재킹 범죄의 끝은 아님을 의미한다.

3.3 최신 크립토재킹

코인하이브[18]는 현재 폐쇄되어, 클라이언트와 서버 간에 데이터를 주고받는 필수 웹 소켓(WebSocket)이 서버에 연결되지 못하는 상황이다. 2017년 9월 크립토재킹이 급격히 늘어난 이후, 최근까지도 공격 빈도와 방법이 다양해지고 있으며, Cryptoloot, JSEcoin, Deepminer 등 여전히 다수의 크립토재킹 사이트들이 존재한다.

Table 1. API code for mining

API Code for mining	Search Results
CoinHive.Anonymous	1565 Web pages
WMP.Anonymous	4 Web pages
CryptoNoter.Anonymous	531 Web pages
DeepMiner.Anonymous	226 Web pages
Cryptoloot.Anonymous	531 Web pages
CoinImp.Anonymous	226 Web pages
Client.Anonymous	38 Web pages
DeepMiner.Anonymous	9 Web pages
Client.Anonymous	1872 Web pages
ProjectPoi.Anonymous	40 Web pages

Table 1.은 소스코드 검색 엔진인 PublicWWW[19] 사이트에서 마이닝에 사용되는 자바스크립트를 검색한 후 채굴용으로 사용되는 자바스크립트 API 코드를 수집하고 그 현황을 조사하여 정리한 결과이다. 특히 2019년 이후로는 사물인터넷 기기들에 대한 MikroTik 라우터 취약점을 수백 건 발견할 수 있다. 라우터의 마이닝 코드를 주입하고 연결된 장치에 그것을 연결함으로써 범죄자들은 수익을 극대화할 수 있었다.

이를 확인하기 위해, Censys라는 검색엔진을 활용하여 "Coinhive.Anonymous AND MikroTik" 등을 쿼리할 수 있다. 적절한 검색을 활용하여, 보안 취약점을 활용한 컴퓨터 시스템 및 응용 프로그램의 실시간 공격 파악이 가능하다. 이 과정에서, 포트 및 프로토콜, 유효한 인증서를 수집할 수 있다.

총 48,948,669개의 Top-level domain(TLD) 크롤링 사이트들을 기준으로, .com에서 2353개(0.009%)의 가장 많은 크립토재킹이 발견되었으며 그 뒤로 .ru에서 593개(0.059%), .de에서 254개, .net에서 238개 등이 발견되었다[20]. 이밖에도 632개의 크립토재킹 사이트들을 분석하였을 때 275개에 해당하는 사이트에서 minero.cc/lib/minero.min.js 파일을, 121개의 webminepool.com/lib/base.js 파일을, 96개의 hashing.win/46B8.js을 사용하고 있는 것을 확인하였다. 이러한 스크립트 파일들은 크립토재킹 마이닝이 가능하도록 돕는다[21].

IV. 머신러닝 기반 크립토재킹 탐지 프레임워크

4.1 크립토재킹 탐지 프레임워크

제안하는 프레임워크는 다음과 같다. Fig.1.은 제안 프레임워크를 도시한 구조도이다. 프레임워크는 크게 네 단계로 나누어 살펴볼 수 있다. 1단계(Data Collection, Feature Extraction)는 준비 단계로 데이터를 수집하고 데이터에서 특징을 추출한다. 2단계(Blacklist Filtering, Cryptojacking Detection Model)는 탐지 단계로 이미 알려진 크립토재킹 정보를 담은 블랙리스트를 사용해 1차적으로 필터링하고, 동적, 정적 분석 기반 탐지 모델을 사용하여 2차 필터링을 거친다. 3단계(Learning Module)는 학습 단계로 지속적으로 새로운 데이터를 학습한다. 4단계(Anomaly Detection Alert)는 경고 단계로 결과를 기반으로 이상 행위가 탐지되면 경고한다.

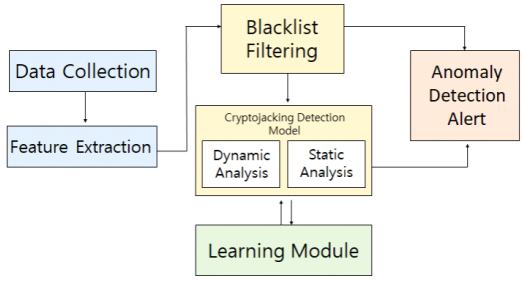

Fig. 1. Machine Learning-based Cryptojacking Detection Framework

1) Data Collection

먼저, Data Collection단계에서 프레임워크에 필요한 동적, 정적 데이터를 수집한다. 동적 분석을 위해서는 크립토재킹 감염 시 나타나는 컴퓨터 성능 지표들을 수집하였다. 성능모니터[22]와 CPUID HW Monitor Pro를 이용하여 필요 데이터를 추출하였다. Windows의 실시간 시스템 성능을 측정할 수 있는 유틸리티인 성능모니터를 사용해 메모리의 Committed Bytes In Use, 프로세스의 ID_Process, Creating Process ID(PID)를 수집하였다. 또한, 컴퓨터 하드웨어 성능을 측정할 수 있는 HW Monitor Pro를 이용하여 코어 온도, TMPIN의 온도, CPU 이용률(CPU Utilization)

을 수집하였다.

정적 분석을 위해서는 정상 데이터로 SRILAB의 자바스크립트 데이터셋을 활용하였다[23]. 해당 데이터셋은 정적 분석 중에서도 악성 문자열이 아닌, 일반 데이터셋으로 활용된다. 본 데이터셋은 프로그래밍을 위한 머신러닝 학습 데이터로 사용되는 자바스크립트 데이터셋이며, 15만개의 자바스크립트 파일로 구성되어있다. 악성 데이터는 Queen's University Belfast의 크립토재킹 자바스크립트를 활용하였다[24]. 해당 자바스크립트 속 내용들은 마이너 스크립트들을 수동 분류한 내용이 포함된다.

Fig.2.은 동적 분석에서 수집한 각 지표가 서로 어떤 상관관계를 보여주는지 시각적으로 표현한 히트맵이다. 그림에 의하면, 서로 같은 지표에 대응하는 대각선은 모두 1로 표현된다. 또한, 서로 밀접한 연관성을 가지는 지표일수록 약 0.7에서 약 0.97에 이르는 높은 수치 및 진한 색상으로 나타나는 것을 알 수 있다. 히트맵을 통해 column 간 상관관계를 알 수 있다. 수집한 지표들이 감염과 얼마나 상관관계가 있는지 알 수 있으며, 상위 지표들 중 일부를 선정하여 실험을 진행하였다.

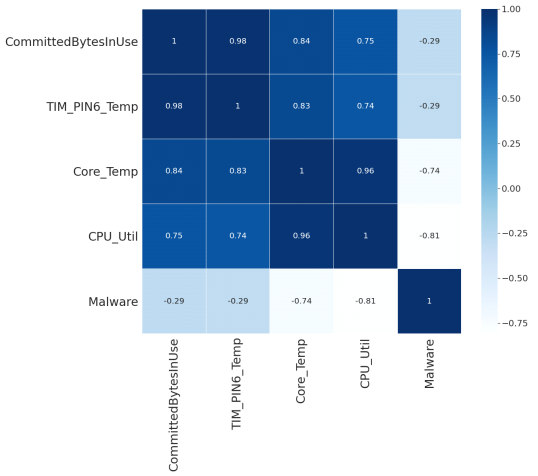

Fig. 2. Correlation Heatmap Representing Associations Between Columns

2) Feature Extraction

Feature Extraction에서는 수집한 데이터를 바탕으로 정적 및 동적 분석에 용이한 특징을 추출한다. 먼저 정적 분석의 경우 자바스크립트 파일의 소스 코드 내용을 읽어 들여 해당 부분에서 의미 있는 단어를 추출하는 것을 말한다. 본 연구에서는 단어의 빈도수를 기반으로 의미있는 단어를 선정한다. 예를 들면 자바스크립트를 토큰화한 뒤, Term Frequency - Inverse Document Frequency(TF-IDF)를 사용해 가중치를 부여하여 정상 자바스크립트와 악성 자바스크립트에 모두 등장하는 단어의 가중치는 줄어들고, 악성 자바스크립트에 포함될 가능성이 큰 단어의 가중치는 높아진다. 동적 분석의 경우 수집한 데이터를 검토하여, 그 중 악성코드 노출시 명확한 변화가 보이는 데이터들을 추출한다. 본 연구에서는 특히 크립토재킹 감염 후 수치가 높아지는 코어 온도와 CPU 이용률을 학습 데이터로 선정하였다. 자세한 내용은 추후 설명한다. 이렇게 정적, 동적 분석에 대한 특징 추출 과정이 독립적으로 수행되어 정확한 결과를 도출하는 데 이용된다.

3) Blacklist Filtering

선행 연구를 통해 얻은 크립토재킹 URL과 스크립트로 블랙리스트를 작성하여 1차 필터를 거친다. Blacklist Filtering은 효율적인 크립토재킹 탐지를 위한 단계이며 머신러닝 기반 탐지 단계 이전에 수행된다. 이 단계에서 기존에 알려진 URL과 스크립트만을 사용하면, 추후 새롭게 탐지되는 생기는 URL 및 스크립트에 한계가 발생할 수 있다. 본 연구에서는 한계점을 보완하기 위하여 머신러닝 기반의 크립토재킹 탐지 모델과 학습 모듈을 사용하여 확장성 및 일반성을 확대할 수 있다.

4) Cryptojacking Detection Modeling

앞선 과정들이 끝난 후, Fig. 1.의 Cryptojaking Detection Model에서는 수집된 데이터는 Dynamic Analysis와 Static Analysis로 나뉘어 각 단계의 특성에 맞게끔 분류한다.

Dynamic Analysis에서는 K-Nearest Neighbors(KNN) 알고리즘을 사용해 코어 온도, CPU 이용률을 학습한다.

Static Analysis에서는 K-means 알고리즘을 적용하여 자바스크립트를 클러스터링한다. 해당 내용을 전체 통계를 바탕으로 빈도수 기반으로 분석을 하게 된다.

5) Learning Module

Learning Module은 새로 들어오는 데이터를

학습하는 단계이다. 선행된 과정이 반복될수록, 더욱 정확한 결과를 획득할 수 있다. 또한, 사용자의 PC 데이터를 내부에서 축적 및 학습하여 정보를 외부에 노출하지 않는 장점이 있다.

6) Anomaly Detection Alert

크립토재킹으로 인한 이상 행위가 탐지될 경우, 사용자에게 경고하는 장치이다.

4.2 머신러닝 기반 크립토재킹 검출 모델

4.2.1 동적 분석

본 연구 논문에서는 효율적인 크립토재킹 동적 탐지를 위해 머신러닝 기반의 지도학습 중 분류 방법을 적용하여 데이터의 감염 여부를 분류한다. 지도학습은 정답이 있는 데이터를 학습하며, 연구에서는 분류(Classification)의 방법을 설정하였다. 분류는 지도학습의 일종으로, 기존에 존재하는 데이터들을 학습하고 일정 카테고리별로 이를 분류하는 과정이라고 할 수 있다. 여기에서는 크립토재킹의 감염 여부를 기준으로 삼을 것이기 때문에, 각각 "악성"과 "비악성(정상)"이 카테고리에 해당한다. 이 과정에서, KNN 알고리즘을 이용하여 새로운 데이터를 결정하는 경우 K개의 점을 선정해 많이 선택되는 범주에 포함되도록 하였다. 그리고 데이터들은 KNN 알고리즘의 적용이 가능하다. 이 기법을 적용하면, 새로운 PID 관련 정보들을 받아왔을 때 인접한 타 PID를 찾아 데이터의 악성 여부를 결정한다. 기존 데이터들은 이미 정상, 악성으로 분류가 되어 있어 분류가 가능해진다. KNN 알고리즘 학습을 기반으로 한 분류를 통해, 위험 PID에 대한 자동화가 가능해진다. 기존의 크립토재킹의 경우, 일일이 CPU 등의 장치 성능을 확인해야 하며 사용자가 위험에 노출이 되어도 피해 사실을 모르는 경우가 대다수였다. 하지만 머신러닝을 접목한 자동화를 통해 고위험군으로 분류된 PID 및 자바스크립트들을 빠르게 종료할 수 있다.

4.2.2 정적 분석

정적 분석은 자바스크립트 데이터셋을 비지도 학습하였다. 비지도 학습은 입력 데이터의 정답을 알려주지 않고 학습을 진행하여 비슷한 데이터끼리 군집화한다. 크립토재킹 스크립트는 공통적으로 발견되는 코드가 등장한다. 예를 들어 'miner', 'anonymous', 'coinhive' 등이 있다. 따라서 크립토재킹 스크립트에 자주 등장하는 단어의 빈도를 측정하여 악성 스크립트임을 판단하였다.

먼저 데이터를 전처리하는 과정을 거쳤다. 스크립트를 단어 단위로 토큰화하고, 불필요한 단어들은 제거하였다. TF-IDF를 사용해 단어의 중요도를 고려하여 단어에 가중치를 부여하였다. 이때, min_df는 0.1로, max_df는 0.95로 지정되었다. 정상 스크립트와 악성 스크립트 모두에게 나타나는 단어의 가중치는 작게 설정된다. 정상 스크립트 또는 크립토재킹 스크립트에만 나타나는 단어는 가중치가 높게 설정된다.

V. 모델 평가 결과 및 분석

5.1 동적 분석 결과

악성 라벨 데이터를 수집하기 위해 홈페이지 중 "WMP.Anonymous" 소스코드가 포함된 여러 사이트를 탐색하여 성능모니터와 CPUID HW Monitor Pro을 통해 9개의 Chrome 인스턴스에 대한 지표를 수집하였다.

이 학습 과정은 Pandas 패키지를 통해, matplotlib 라이브러리를 활용한 pyplot으로 시각적으로 표현했다. 2615개의 열 중 1216개는 정상, 1399개는 악성으로 분류해 두었다.

Fig.3.은 KNN 알고리즘을 적용하여, 새로운 데이터에 대해 악성 및 비악성 분류가 가능함을 보여준다. 그래프에서는 기존에 수집하였던 데이터를 바탕으로, 악성과 비악성의 경계를 가장 효과적으로 분류할 수 있는 지표를 각각 가로, 세로축으로 선정하였다.

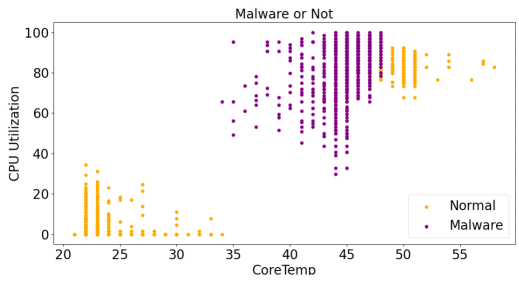

Fig. 3. Categorized Data Using Artificial Intelligence

여기에서 가로축은 CoreTemp, 세로축은 CPU Utilization에 해당한다. 이후 분류된 결과를 점으로 표현하여 악성인 것은 보라색, 비악성인 것은 노랑색으로 나타내었다.

5.2 정적 분석 결과

앞서 수집한 자바스크립트 데이터셋을 활용하여 890개의 정상 스크립트, 890개의 악성 스크립트를 생성하였다. 여기서 활용한 데이터셋은 2018년 7월 12일에 알렉사 탑 백만 웹사이트로부터 수집되었다. 크립토재킹 URL 목록, 스크립트 목록, HTML 파일, 스크립트 파일 등으로 구성되어있으며, 이 중에서 마이닝 스크립트를 학습하였다.

모델은 파이썬(Python)의 사이킷런(Scikit-learn) 라이브러리를 사용하여 TF-IDF로 단어의 가중치를 설정하고, K-means를 사용해 문서의 유사도를 비교하여 Fig.4.와 같이 악성은 보라색 클러스터, 정상은 노란색 클러스터로 군집화할 수 있었다.

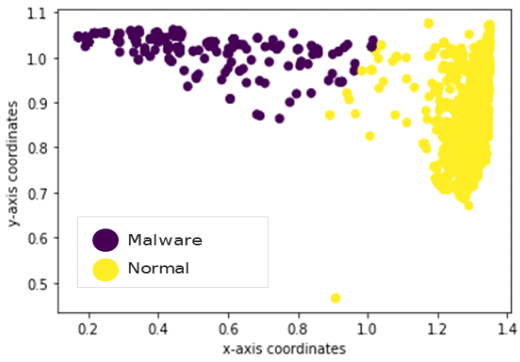

Fig. 4. Clustered Data Using K-means Algorithm

5.3 평가

앞서 언급하였듯, 동적 분석에서는 KNN 알고리즘을 적용하여 평가했다. “Malware”열은 라벨로 지정하였고 나머지 열은 피쳐로 지정해주었다. 사이킷런의 cross_validate 함수를 통해서 모델 평가 시에 여러 지표를 활용할 수 있도록 하였다. k를 1에서 25까지 범위를 늘려 확인하면, Fig. 5.와 같다. Fig.5.는 k가 1~25일 때, 동적 분석 모델의 정확도를 나타낸다. k가 5일 때를 기준으로 정확도가 약 99.6%이다.

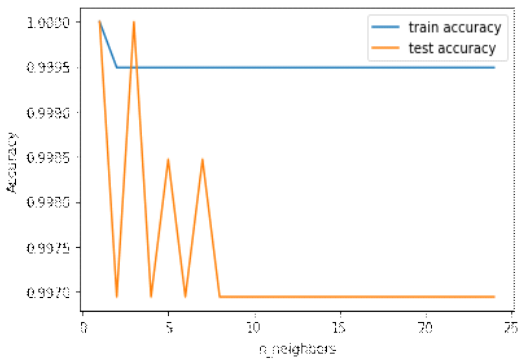

Fig. 5. KNN Cryptojacking Classifier Accuracy

또한, Stratified K-Fold를 이용한 교차 검증을 시도하였다. 해당 방법은 원본 데이터상에서 레이블의 비율을 고려하여 train과 test set 또한 동일한 비율로 학습과 검증하는 것을 돕는다. cross_val_score 함수를 이용하여 데이터셋을 10개로 쪼갠 후 평균 정확도를 확인한 결과, 99.2%이 출력되었다.

정적 분석의 클러스터링 결과를 평가하기 위해 실루엣 계수를 활용하였다. 실루엣 계수는 클러스터링이 잘 이루어졌는지 확인할 수 있는 평가 지표이다. 평가 결과, 정상 클러스터의 실루엣 계수는 0.615, 로 확인되었다. 이진 분류이므로 정상 클러스터가 정확한 군집을 이루는 경우 나머지는 악성으로 분류할 수 있다.

VI. 결 론

본 논문에서는 머신러닝 기반 크립토재킹 탐지 프레임워크를 제안하였다. 크립토재킹 피해는 지속적으로 증가하고 있으며, 크립토재킹 공격을 당하더라도 크립토재킹의 특성상 피해자가 피해를 입은 사실조차 알기 어렵다. 기존에는 크립토재킹에 대응하기 위해서 피해자가 일일이 CPU 사용량을 확인해 감염 증상이 있는지 체크하거나, 알려진 정보를 바탕으로 차단하여 한계가 존재한다. 본 논문에서 제안하는 머신러닝 기반 크립토재킹 탐지 프레임워크는 데이터 수집, 블랙리스트 필터링, 정적 탐지 모델, 동적 탐지 모델 총 네 단계를 거쳐 효과적으로 크립토재킹을 탐지할 수 있다. 결과적으로 해당 단계들을 거쳐 크립토재킹을 빠르고 효율적으로 탐지하는 프로그램의 구현도 가능할 것이다. 이후, 프레임워크의 머신러닝

기반 동적 탐지와 정적 탐지 모델을 구현하고 검증하였다. KNN 알고리즘을 사용한 동적 탐지 모델은 정확도가 99.6%, K-means 알고리즘을 사용한 정적 탐지 모델은 정상 군집의 실루엣 계수가 0.61로 크립토재킹을 탐지할 수 있음을 확인하였다.

정적 분석의 경우에는 전형적인 크립토재킹 공격만 고려했고, 동적 분석의 경우 일반 프로그램에서도 온도나 CPU 사용량 등이 높아지는 경우가 있다. 향후 연구의 한계점을 보완하기 위해 비교군을 넓혀 크립토재킹의 특징적인 지표를 찾고, 지표의 주기성을 파악하여 탐지의 정확도를 높일 수 있는 연구를 진행하고자 한다.

References

[1] T. He, R.M. Aronce, L. Dampanaboina, J. Jose, M. King and E.C. Cohen, "2021 SonicWall Cyber Threat Report," Sonicwall, 2021.

[2] R. Julian, S. Sebastian, D. Tobias, L. Rober, B. Damjan , P. Gerhar and K. Hyoungshick , "The Other Side of the Coin: A Framework for Detecting and Analyzing Web-based Cryptocurrency Mining Campaigns," I*n International Conference on Availability, Reliability and Security*, no. 18, pp 01-10, Aug. 2018

[3] Tanana and Dmitry, "Behavior-based detection of cryptojacking malware" *2020 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT). pp. 0543-0545,* May. 2020

[4] Wenhao Wang, Benjamin Ferrell, Xiaoyang Xu, W. Kevin, Hamlen and Shuang Hao, "SEISMIC: SEcure in-lined script monitors for interrupting cryptojacks", *In European Symposium on Research in Computer Security,* pp. 122-142, Sep. 2018

[5] Geng Hong, Zhemin Yang, Sen Yang, Lei Zhang, Yuhong Nan, Zhibo Zhang, Min Yang, Yuan Zhang, Zhiyun Qian and Haixin Duan, "How you get shot in the back: A systematical study about cryptojacking in the real world," *In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp 1701-1713, Oct. 2018

[6] Petrov, Ivan, Luca Invernizzi and Elie Bursztein, "Coinpolice: Detecting hidden cryptojacking attacks with neural networks," *arXiv preprint arXiv:2006.10861,* June. 2020

[7] antiminer, "AntiMiner", https://github.com/unkn0wn404/MinerBlocker, accessed Jul.13,2021, 2017

[8] minerblock, "MinerBlock", https://github.com/xd4rker/MinerBlock], accessed Jul.13,2021, 2019

[9] nocoin, "NoCoin", https://github.com/keraf/NoCoin/blob/master/src/blacklist.txt, accessed Jul.13,2021, 2018

[10] Radhesh Krishnan Konoth, Emanuele Vineti, Veelasha Moonsamy, Martina Lindorfer, Christopher Kruegel, Herbert Bos and Giovanni Vigna, 2018, "Minesweeper: An in-depth look into drive-by cryptocurrency mining and its defense," *In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1714-1730, Oct. 2018

[11] Muhammad Saad, Aminollah Khormali and Aziz Mohaisen, "End-to-end analysis of in-browser cryptojacking," *In arXiv preprint arXiv:1809.02152*, Sep. 2018

[12] Binance Academy, "CryptoJacking Description", https://academy.binance.com/ko/articles/what-is-cryptojacking, accessed Jul.13,2021

[13] Daily Today, "Cryptojacking to enslave your PC", http://www.digitaltoday.co.kr/news/articleView.html?idxno=2023

02, accessed Jul.13,2021
〔14〕 KrebsonSecurity, “Who and What Is Coinhive?,”https://krebsonsecurity.com/2018/03/who-and-what-is-coinhive/, accessed Jul.13,2021
〔15〕 The Irish Times, “Q&A: What is the story with Coinhive?”, https://www.irishtimes.com/business/technology/q-a-what-is-the-story-with-coinhive-1.3389706, accessed Jul.13,2021
〔16〕 Pandasecurity, “Coinhive, the Monero mining service, is closing down”, https://www.pandasecurity.com/en/mediacenter/news/coinhive-mining-closes/, accessed Jul.13,2021
〔17〕 ZDNet, “Coinhive cryptojacking service to shut down in March 2019”, https://www.tripwire.com/state-of-security/security-data-protection/cyber-security/coinhive-browser-cryptomining-service-dead/, accessed Jul.13,2021, 2019
〔18〕 Malwarebytes, ″Cryptojacking in the post-Coinhive era,″ https://blog.malwarebytes.com/cybercrime/2019/05/cryptojacking-in-the-post-coinhive-era/, accessed Jul.13,2021, 2019
〔19〕 PublicWWW, “PublicWWW”, https://publicwww.com/, accessed Jul.13,2021
〔20〕 Hugo L.J. Bijmans, Tim M. Booij, and Christian Doerr, ″Inadvertently Making Cyber Criminals Rich:A Comprehensive Study of Cryptojacking Campaigns at Internet Scale″, 28th USENIX Security Symposium, pp.1627-1644, Aug. 2019
〔21〕 Said Varlioglu, Bilal Gonen, Murat Ozer, Mehmet F. Bastug, ″Is Cryptojacking Dead after Coinhive Shutdown?″, 2020 3rd International Conference on Information and Computer Technologies (ICICT), pp.385-389, Mar. 2020
〔22〕 Forsenergy, “Windows Performance Monitor Overview”, https://forsenergy.com/ko-kr/perfmon/html/44daefa4-407d-4763-b42f-b613a261da54.htm, accessed Jul.13,2021
〔23〕 SRILAB, “150k Javascript Dataset”, https://www.sri.inf.ethz.ch/js150, accessed Jul.13,2021
〔24〕 J. Burgess (Creator), “CryptoJacking Data (including raw HTML/JS files),” Queen′s University Belfast, CryptoJacking_AlexaTop1m_July2018(.zip), 10.17034/ea782cda-b3ac-4fc3-b78b-c81324453280, accessed Jul.13,2021, Feb 2020

〈저 자 소 개〉

임 은 지 (EunJi Lim) 학생회원
2021년 2월: 성신여자대학교 융합보안공학과 졸업
2021년 3월~현재: 연세대학교 정보대학원 정보보호학과 석사
〈관심분야〉 정보보호, 인공지능, 모의 해킹, 블록체인

이 은 영 (EunYoung Lee) 학생회원
2021년 2월: 성신여자대학교 융합보안공학과 졸업
2021년 3월~현재: 성신여자대학교 미래융합기술공학과 석사
〈관심분야〉 융합보안, 정보보호, 블록체인

이 일 구 (IlGu 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