Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.5, Oct. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.5.1161

하이브리드 특징 및 기계학습을 활용한 효율적인 악성코드 분류 시스템 개발 연구*

유 정 빈,† 오 상 진, 박 래 현, 권 태 경‡
연세대학교 정보보호연구실

Development Research of An Efficient Malware Classification System Using Hybrid Features And Machine Learning*

Jung-Been Yu,† Sang-Jin Oh, Leo-Hyun Park, Tae-Kyoung Kwon‡
Information Security Lab., Graduation School of Information, Yonsei University

요 약

기하급수적으로 증가하고 있는 변종 악성코드에 대응하기 위해 악성코드 분류 연구가 다양화되고 있다. 최근 연구에서는 기존 악성코드 분석 기술 (정적/동적)의 개별 사용 한계를 파악하고, 각 방식을 혼합한 하이브리드 분석으로 전환하는 추세이다. 나아가, 분류가 어려운 변종 악성코드를 더욱 정확하게 식별하기 위해 기계학습을 적용하기에 이르렀다. 하지만, 각 방식을 모두 활용했을 때 발생하는 정확성, 확장성 트레이드오프 문제는 여전히 해결되지 못했으며, 학계에서 중요한 연구 주제이다. 이에 따라, 본 연구에서는 기존 악성코드 분류 연구들의 문제점을 보완하기 위해 새로운 악성코드 분류 시스템을 연구 및 개발한다.

ABSTRACT

In order to cope with dramatically increasing malware variant, malware classification research is getting diversified. Recent research tend to grasp individual limits of existing malware analysis technology (static/dynamic), and to change each method into “hybrid analysis”, which is to mix different methods into one. Futhermore, it is applying machine learning to identify malware variant more accurately, which are difficult to classify. However, accuracy and scalability of trade-off problems that occur when using all kinds of methods are not yet to be solved, and it is still an important issue in the field of malware research. Therefore, to supplement and to solve the problems of the original malware classification research, we are focusing on developing a new malware classification system in this research.

Keywords: Malware, Classification, Machine Learning, ssdeep

I. 서 론

자동화된 악성코드 생성 도구가 인터넷을 통해 유포됨에 따라 악성코드 출현 개수가 기하급수적으로 증가하고 있다. 2017년 AV-Test 악성코드 동향 보고서에 따르면 DDoS, 스팸 발송, APT 공격 등에 사용된 악성코드는 연간 기준 약 1억 2천만 개에 달한다 [1]. 그러나 전체 악성코드 가운데 신종 악성코

Received(07. 16. 2018), Modified(08. 16. 2018), Accepted(08. 17. 2018)
* 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음 (IITP-2018-2016-0-00304), 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2017-0-0038 80, 차세대 인증 기술 개발)
† 주저자, getchabug@gmail.com
‡ 교신저자, taekyoung@yonsei.ac.kr(Corresponding author)

드는 20% 미만으로 악성코드 대부분이 기존 악성코드의 변종이다.

변종 악성코드는 패턴 매칭 기반 탐지방식을 회피하기 위해 기존 악성코드를 변형한 악성코드이다. 변종 악성코드는 외형을 변형한 다형성 악성코드, 배포될 때 코드가 재작성되는 변성 악성코드로 분류된다. 이에 따라, 다양한 악성코드 특징, 기계학습 알고리즘을 적용한 연구가 진행되고 있다 [5,6,7].

악성코드 특징은 추출 방식에 따라 정적/동적 특징으로 분류된다. 정적 특징은 악성코드를 실행시키지 않고 DLL, 섹션 정보 등을 추출하기 때문에 빠르지만 변종 악성코드에 취약하다. 반면, 동적 특징은 제한된 환경에서 악성코드를 직접 실행시켜 악성코드 행위 정보 (예: API 시퀀스, 레지스트리 정보 등)를 추출하기 때문에 변종 악성코드에 유연하지만 특징 추출에 소요되는 시간이 높은 확장성 문제가 있다.

단일 특징 추출 방식은 각 방식에 따른 고유 한계를 갖기 때문에, 최근에는 이를 완화하기위해 정적/동적 특징을 모두 활용한 하이브리드 연구가 제안되고 있다. 그러나 정확성과 확장성 사이에 존재하는 트레이드오프 문제는 여전히 해결되지 못했으며, 학계에서 중요한 연구 주제이다. 이에 따라 본 연구에서는 기존 악성코드 분류 연구의 트레이드오프 문제 개선을 목표로 한다. 이를 위해, ssdeep 정적 분석도구를 활용해 분류되는 악성코드와 그렇지 않은 악성코드를 나눈 뒤, 정적/동적특징, 지도/비지도 학습 알고리즘을 병렬 활용한다.

II. 기반 기술 소개

2.1 Ssdeep

입력된 2개 이상 바이너리를 CTPH (Context Triggered Piecewise Hashing) 토대로 파일 유사도를 확인할 수 있는 정적 특징 추출 도구이다 [2]. CTPH는 입력된 컨텍스트를 블록 단위로 해싱하고, 해싱 컨텍스트를 토대로 유사도 측정하는 도구이다. 특히 입력 컨텍스트 길이가 다르더라도 유사도를 측정할 수 있다는 장점이 있어, 최근 악성코드 분류 연구에서 활발히 활용되고 있다.

2.2 Forward Stepwise Selection Algorithm

악성코드를 효율적으로 분류할 수 있는 최적의 특징 조합을 선정하기 위해, 단일 특징 정보들의 유사 악성코드 그룹화 정확도를 측정한다 [5]. 나아가, 높은 정확도를 갖는 단일 특징 (f)을 점진적으로 특징 조합에 추가한다. 다음 특징을 조합에 추가했을 때 추가하지 않았을 때와 정확도가 같다면, 해당 특징을 조합에 추가하지 않고 알고리즘이 종료되며 최적의 특징 조합이 생성된다. 특히 모든 특징 조합을 전수로 고려해야 하는 Best Subset Selection 알고리즘 (2^f)에 비해 현저히 적은 특징 조합을 고려해 유사한 성능을 보이기 때문에 효율적이다 ($1 + (f * (f + 1)) / 2$).

III. 시스템 설계

유입되는 대용량 악성코드를 효율적으로 분류하기 위해 새로운 시스템을 제안한다. 제안하는 시스템의 주요 아이디어는 본격적인 악성코드 분류에 앞서, ssdeep을 활용해 유사도 관계를 확인하는 것이다. 본 연구에서는 ssdeep을 활용해 유사 악성코드 간 그룹화되는 결과를 '정적 상관관계 (Static Correlation)'라 정의한다. 1개 이상 유사 악성코드와 정적 상관관계가 확인되는 악성코드의 경우, 정적 특징과 분류 알고리즘

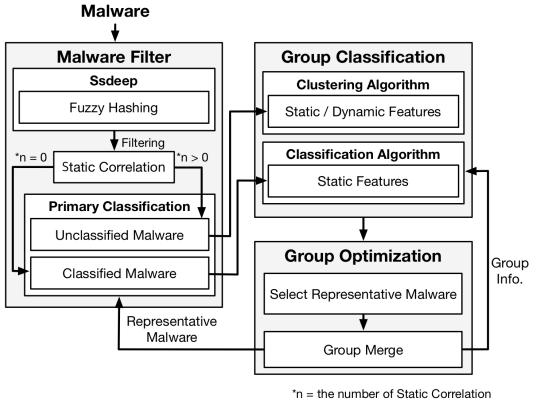


Fig. 1. System Overview

Table 1. Reliable Malware Dataset

Family	#	Family	#
Virut	1,087	turkojan	62
IRCBot	620	zbot	62
Hupigon	571	mabezat	62
Agent	541	mamianune	53
Heur_gen	92	elkern	43
upatre	78	domaiq	22
		Total	**3,293**

을 활용해 빠르게 분류한다. 반면, 정적 상관관계를 확인하기 힘든 악성코드는 정적/동적 특징과 클러스터링 알고리즘을 활용해 분류한다. 제안하는 시스템은 Fig 1.과 같이 악성코드 필터, 그룹 분류, 그룹 최적화 순으로 단계적 수행된다.

3.1 악성코드 필터

본격적인 악성코드 분류가 수행되기 전, ssdeep을 활용해 유입된 악성코드간 정적 상관관계를 확인한다. 배포될 때마다 코드가 재작성되는 변성 악성코드 또는 ssdeep 유사도가 낮게 나오는 악성코드의 경우, 정적 상관관계를 파악하기 힘들다. 이러한 원리를 통해 ssdeep 유사도 임계값을 설정하고, 이를 기준으로 정적 상관관계를 확인한다. 1개 이상 유사 악성코드와 정적 상관관계가 확인되는 악성코드의 경우, 정적 특징과 분류 알고리즘을 활용해 빠르게 분류한다. 반면, 정적 상관관계를 확인하기 힘든 악성코드는 보다 많은 특징을 요구하기 때문에 클러스터링 알고리즘 단계로 전달된다.

3.2 기계학습을 활용한 그룹 분류

본 단계에서는 정확성/확장성 트레이드오프 문제를 최소화하기 위해 악성코드 필터 결과를 토대로 분류 알고리즘과 클러스터링 알고리즘을 병렬 활용한다.

사전 분류된 악성코드. 정적 상관관계를 갖는 악성코드는 정적 특징, 분류 알고리즘을 활용해 분류된다. 분류 알고리즘은 학습 데이터를 토대로 새로 유입된 데이터를 분류하는 지도학습 알고리즘이다. 학습데이터를 기반으로 분류하기 때문에 분류 속도가 빠르지만, 새로운 유형의 악성코드를 유연하게 분류하기 힘들다.

사전 분류되지 않은 악성코드. 상관관계를 갖지 않는 악성코드는 정적/동적 특징, 클러스터링 알고리즘을 활용해 유연한 분류를 수행한다. 클러스터링 알고리즘은 분류되지 않은 데이터 셋을 사용하는 비지도학습 알고리즘이다. 거리 기반 알고리즘 (예: 유클리디안알고리즘, 코사인 알고리즘 등)으로 알려지지 않은 데이터 간 특징 거리를 계산하고 분리된 데이터 구조로 나누는 것을 목표로 한다. 특히, 학습 데이터가 없더라도 악성코드를 군집화할 수 있기 때문에, 새로운 유형의 악성코드를 분류하는데 적합하다. 그러나 군집화 된 악성코드가 어떤 악성코드 그룹인지 확인하기 힘들다.

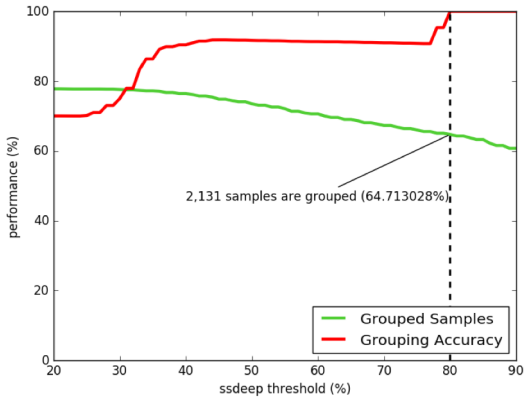


Fig. 2. Ssdeep Group Classification Result

3.3 그룹 최적화

정적 상관관계를 갖지 않는 악성코드는 그룹 클러스터링을 통해 군집화되기 때문에 각 군집이 어떤 악성코드 그룹인지 확인하기 힘들다. 따라서, 군집화된 악성코드에 그룹 정보를 입력해주는 작업이 필요하다. 본 연구에서 제안하는 프레임워크의 확장성을 보장하면서 그룹정보를 입력하는 작업을 수행하기 위해 대표 악성코드를 활용한다. 대표 악성코드는 정적 상관관계를 가져 분류 알고리즘을 활용해 분류된 그룹과 정적 상관관계를 갖지 않아 클러스터링을 통해 군집화 된 악성코드 그룹에서 각각 중심점 (Centroid)과 최근사한 악성코드로 선정한다. 선정된 대표 악성코드들은 클러스터링 알고리즘을 활용해 그룹 병합 작업 수행한다. 그룹 병합 작업은 그룹 정보를 포함한 사전 분류된 악성코드의 대표 악성코드들과 같이 수행되기 때문에, 그룹 정보를 확인하기 힘든 군집화 된 악성코드의 그룹 정보를 식별할 수 있을 뿐만 아니라 그룹 통합을 진행할 수 있다.

IV. 실험 설계 및 결과

4.1 실험 환경 및 데이터 셋 생성

모든 실험은 Intel (R) Core (TM) i5-6600 CPU @ 3.30 GHz, 32 GB RAM and Ubuntu 16.04.2 LTS 환경에서 Python 2.7 및 머신러닝 모듈 Scikit-learn을 활용해서 실험을 수행했다. 또

Table 2. Classification Algorithm Result

Algorithm	Precision	Recall	F-Measure
ExtraTrees	95.27	93.17	94.21
Random Forest	94.48	92.13	93.29
KNN	92.02	90.02	91.01
DecisionTree	89.97	90.94	90.40
SVM	92.16	85.82	88.88

한, 실험에 활용할 데이터 셋 생성을 위해 웹 아카이브 VX Heavens와 악성코드 분석 전문가를 통해 그룹 정보가 입력된 악성코드를 수집했다 [3]. 나아가 신뢰할 수 있는 데이터셋을 실험에 활용하기 위해, 사전에 입력되어 있는 그룹 정보와 Virus Total 스캐닝 결과를 교차 검증함으로써 총 3,293개의 신뢰할 수 있는 악성코드를 생성했다 (Table 1. 참고) [4].

4.2 ssdeep을 활용한 그룹 분류

정확성을 보장하면서 높은 확장성을 갖는 시스템을 구축하기 위해 사전분류를 수행하는 본 단계의 정확한 측정이 필요하다. 즉, 악성코드간 존재하는 정적 상관관계 확인의 기반이 되는 최적의 ssdeep 임계값을 찾는 실험이 필요하다. 이에 따라 본 연구에서는 신뢰할 수 있는 최적의 임계값 선정을 위해 Table 1. 악성코드로 두 가지 실험을 수행했다.

전체 악성코드 그룹. 신뢰할 수 있는 악성코드 전체 (3,293개, 12개 Family)를 대상으로 ssdeep 임계값을 20%~90%로 조정하면서 실험을 수행했다. 결과적으로 Fig 2.와 같이 임계값 80%일 때, 36초 동안 2,131개 악성코드가 100% 정확도로 분류되어 가장 우수한 성능을 보였다. 이는 전체 악성코드의 64.71%로, 35.29% (1,162개) 악성코드만 샌드박스를 활용해 동적 특징을 추출하고, 클러스터링을 통해 그룹 분류 하면 된다는 것을 의미한다.

임의로 선정된 10개의 악성코드 그룹. 전체 악성코드를 대상으로 수행한 실험을 검증하기 위해, 전체 악성코드 가운데 임의로 선정한 악성코드들로 이루어진 10개의 그룹으로 같은 실험을 수행했다. 임의로 선정된 악성코드로 이루어진 10개의 그룹에 각각 임계값을 조정하며 실험을 수행한 결과, 전체 악성코드를 대상으로 한 실험과 유사하게 임계값 80%일 때, 100% 정확도로 분류되었다.

Table 3. Clustering Algorithm Result

Algorithm	Precision	Recall	F-Measure
Birch	91.34	82.00	86.42
Agglomerative Clustering	92.99	79.06	85.46
K-Means	94.48	64.39	76.59

4.3 기계학습을 활용한 그룹 분류

사전 분류된 악성코드. 정적 상관관계를 갖는 악성코드의 경우, 정적 특징 및 분류 알고리즘을 활용하기 때문에 정적 상관관계를 갖지 않는 악성코드에 비해 빠르게 분류된다. 특히, 제안하는 시스템에서 활용하는 분류 알고리즘 부문의 정확성을 고려하기 위해, 시스템에 가장 적합한 분류 알고리즘을 선정하는 연구가 필요하다. 이에 따라, Table 4.와 같이 악성코드 정적특징을 토대로 정밀도와 재현율을 측정해 시스템에 알맞은 분류 알고리즘을 선정한다. 본 연구에서는 다양한 분류 알고리즘 가운데, Table 2.와 같이 다른 원리를 활용하는 ExtraTrees, Random Forest, kNN, DecisionTree, SVM 총 5개의 알고리즘을 선정해 실험에 활용했다. 분류 알고리즘의 성능 측정은 데이터 셋 (2,131개) 개수를 고려해 10-fold cross validation을 활용했다. 성능 측정 결과, ExtraTrees가 12개의 그룹을 정밀도 95.27, 재현율 93.17, 그리고 정밀도와 재현율의 조화 평균을 나타내는 F-Measure가 94.21로 가장 우수한 성능을 보였다.

사전 분류되지 않은 악성코드. 정적 상관관계를 갖지 않는 악성코드의 경우, 정적 특징만을 활용해 분류하기 힘들다. 즉, 악성코드로부터 추출한 정적/동적 특징을 토대로 클러스터링 알고리즘을 활용해 군집화해야 한다. 그러나, 악성코드로부터 추출한 모든 특징을 활용하면 '차원의 저주 (Curse of Dimensionality)' 문제가 발생할 확률이 높다. 이에 따라 본 연구에서는 Table 4.에 언급된 모든 특징을 후보군으로 삼고, Forward Stepwise Selection 알고리즘을 활용해 정적 상관관계를 갖

Table 4. Malware Feature Information in Each Algorithm

Features	Static Features			Dynamic Features		
	Entropy	PESection	Compile Time	ICMP	Host	UDP
				Registry	PCAP	DNS
				Domain	Dropped	Mutex
	API	Opcode 2-gram Frequency		Command	Service Created/Started	File R/W/A/D
Classification Algorithm	✓	✓	✓	-	-	-
				-	-	-
	✓	✓		-	-	-
				-	-	-
Clustering Algorithm	✓	✓	✓	✓	✓	✓
				✓	✓	✓
	✓	-		✓	✓	✓
				✓	✓	✓

지 않는 악성코드 분류에 가장 적합한 특징 조합을 선정한다. 나아가, 제안하는 시스템에서 활용하는 클러스터링 알고리즘의 정확성을 고려하기 위해, 시스템에 가장 적합한 알고리즘을 선정한다.

Forward Stepwise Selection 알고리즘으로 악성코드 분류에 가장 적합한 특징 조합을 선정한 결과, 'File Write', 'File Delete'가 최적의 특징 조합으로 선정되었다. 나아가, 선정된 악성코드 특징 조합을 토대로 다양한 클러스터링 알고리즘을 적용해보고 정밀도와 재현율을 측정함으로써, 시스템에 알맞은 클러스터링 알고리즘을 선정하는 실험을 수행했다. 본 연구에서는 다양한 클러스터링 알고리즘 가운데, 다른 원리를 활용하는 알고리즘 Birch, Agglomerative Clustering, K-Means 총 3개의 알고리즘을 선정해 실험에 활용했다. 또한, Table.1 악성코드 가운데 정적 상관관계를 갖지 않는 악성코드 (1,162)를 활용했다. 성능 측정 결과, Birch가 45개의 그룹을 정밀도 91.34, 재현율 82.00, 그리고 정밀도와 재현율의 조화 평균을 나타내는 F-Measure가 86.42로 가장 우수한 성능을 보였다.

4.4 그룹 최적화

정적 상관관계를 갖지 않는 악성코드의 경우, 클러스터링 알고리즘을 활용하기 때문에 악성코드 레이블링 정보를 확인하기 힘들다. 즉 정적 상관관계를 가져 분류 알고리즘을 활용해 분류된 악성코드와 그룹 통합을 수행하고, 악성코드 레이블 정보를 입력해주는 단계가 필요하다. 본 연구에서는 그룹 최적화 단계에서 발생할 수 있는 확장성 문제를 고려하기 위해 각 그룹의 중심점과 가장 가까운 악성코드를 대표 악성코드로 지정하고, 대표 악성코드만을 활용해 그룹 병합을 수행한다. 나아가, 정적 상관관계를 갖는 악성코드 그룹을 높은 정확도로 가장 많은 그룹을 병합할 수 있는 알고리즘을 선정하기 위해 클러스터링 알고리즘, 분류 알고리즘, 그리고 유사도 알고리즘을 모두 활용해보고 가장 적합한 알고리즘을 선정한다.

분류/클러스터링 알고리즘을 활용해 분류된 결과에서 각 그룹별 중심점에 가장 가까운 악성코드를 선정하고 (분류 알고리즘은 12개의 대표 악성코드, 클러스터링 알고리즘은 45개의 대표 악성코드가 선정되었다) 클러스터링 알고리즘, 분류 알고리즘, 그리고 유사도 알고리즘의 성능을 측정했다. 분류/클러스터링 알고리즘은 각 단계에서 선정한 알고리즘을 활용했고, 유사도 알고리즘 또한 다른 원리를 활용하는 Bray-curtis, Correlation, 코사인 유사도 총 3가지 알고리즘을 선정해 활용했다. 유사도 측정 결과, Fig 3와 같이 클러스터링 알고리즘인 Agglomerative가 그룹 병합 정확도 92.26%로 가장 높은 정확도를 보였다.

V. 관련 연구

Wang 등은 악성코드 분류를 위해 import table 정적 정보, 시스템 상태 동적 정보 등을 활용했다 [6]. 단일 특징을 활용한 연구에 비해 높은 정확도를 보였지만, 감시 스파이웨어만을 대상으로 했기 때문에 다른 유형의 악성코드에 유연하지 못한

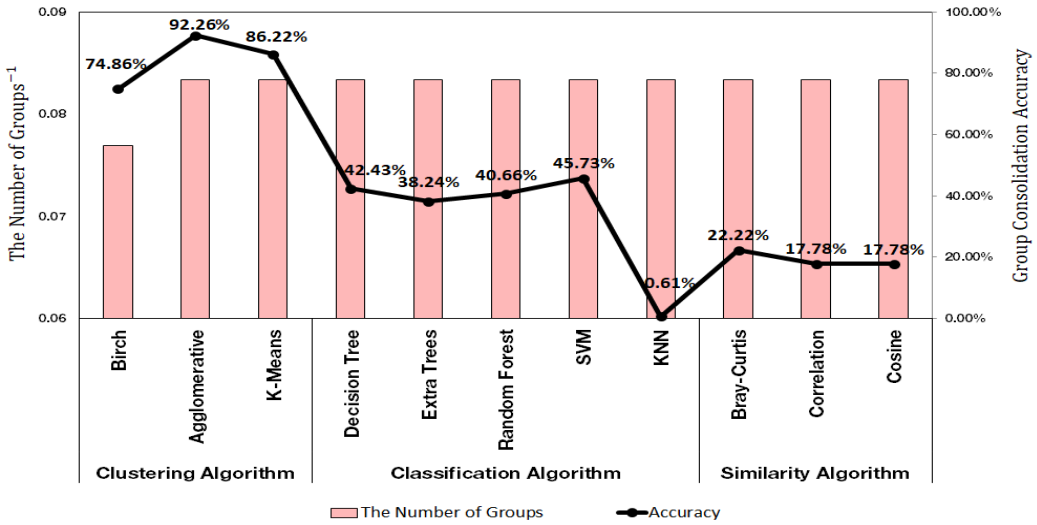


Fig. 3. Representative Malware Group Merge Measurement Result

문제가 있다. 한정적인 유형의 악성코드를 대상으로 한 시스템은 상용화되기 힘들다. 이에 제안하는 시스템은 다양한 유형의 악성코드를 대상으로 성능을 측정했다. 반면, Anderson 등은 다양한 악성코드 유형을 대상으로 2-gram 정적 정보, 동적 명령어 추적 정보를 분류에 활용했다 [7]. 제안된 시스템은 기존 동적 분석에 비해 메모리 소모가 낮고, 다형성 악성코드에 유연하다는 장점이 있지만, 악성코드 특징을 모두 추출하는데 약 5분의 시간이 걸린다는 문제가 있다. 이는 하루 발생하는 악성코드가 약 10만 개에 달한다는 통계적 수치를 참고했을 때, 전체 악성코드를 포괄하기 힘든 확장성을 갖는다. 따라서, 제안하는 시스템은 ssdeep 정적 분석 도구를 활용해 분류할 수 있는 악성코드를 최대한 분류하고, 분류되지 않은 악성코드만을 대상으로 동적 특징을 활용한다.

VI. 결 론

본 연구는 효율적으로 악성코드를 분류할 수 있는 시스템을 제안했다. 특히, 기존 연구와 다르게 ssdeep을 활용해 악성코드간 정적 상관관계를 확인하고 초기분류했다. 이를 통해, 정적 상관관계를 갖지 않는 악성코드만 동적 특징을 활용한다는 점에서 정확성을 유지하면서 확장성을 높였다. 추후 연구에서는 대용량 악성코드에 대한 정확성, 확장성 검증을 수행한다.

References

[1] AVTEST, "AVTEST Annual Report", https://www.av-test.org/en, May. 2018.

[2] SSDEEP, "ssdeep - Fuzzy hashing program", https://ssdeep-project.github.io/ssdeep/, May. 2018.

[3] VX Heaven, http://83.133.184.251/virensimulation.org/, May. 2018.

[4] Virus Total, "VirusTotal - Free Online Virus, Malware and URL Scanner", https://www.virustotal.com/ko/, May. 2018.

[5] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto. "Novel feature extraction, selection and fusion for effective malware family classification," In Proc. Data and Application Security and Privacy (CODASPY), pp. 183-194, Mar. 2016

[6] T.Y. Wang, S.J. Horng, M.Y. Su, C.H. Wu, P.C. Wang, and W.Z. Su. "A surveillance spyware detection system based on data mining methods," In Proc. IEEE Congress on Evolutionary

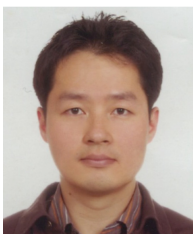
Computation, pp. 3236 - 3241, Jul. 2006.
〔7〕 B. Anderson, C. Storlie, and T. Lane. "Improving malware classification: bridging the static/dynamic gap," In Proc. Artificial Intelligence and Security (AISec), pp. 3 - 14, Oct. 2012.

〈저자소개〉

유 정 빈 (JungBeen Yu) 학생회원
2016년 2월: 대구카톨릭대학교 정보보호학 졸업
2016년 3월~현재: 연세대학교 정보보호 연구실 석사 과정
〈관심분야〉 시스템 보안, 악성코드 탐지, 기계학습

오 상 진 (Sang jin Oh) 학생회원
2018년 2월: 을지대학교 의료IT마케팅학과 졸업
2018년 3월~현재: 연세대학교 정보보호 연구실 석사 과정
〈관심분야〉 정보보호, 악성코드 탐지, 기계학습

박 래 현 (Leo Hyun Park) 학생회원
2017년 2월: 광운대학교 컴퓨터공학 졸업
2017년 3월~현재: 연세대학교 정보보호 연구실 석사 과정
〈관심분야〉 악성코드 탐지, 유저블 시큐리티, 기계학습



권 태 경 (Taekyoung Kwon) 종신회원
1992년 2월: 연세대학교 컴퓨터과학과 학사
1995년 2월: 연세대학교 컴퓨터과학과 석사
1999년 8월: 연세대학교 컴퓨터과학과 박사
1999년~2000년: U.C. Berkely Post-Doc
2001년~2013년 8월: 세종대학교 컴퓨터공학과 교수
2007년~2008년: Univ. Maryland at College Park 교환 교수
2013년 9월~현재: 연세대학교 정보대학원 교수
〈관심분야〉 암호 프로토콜, 유저블 시큐리티, 사물인터넷 보안, 소프트웨어 보안 등