IoT 악성코드 분석을 위한 op 코드 카테고리 시퀀스 특징과 기계학습 알고리즘 활용

문성현†, 김영호†, 김동훈‡, 황두성†
†단국대학교 소프트웨어학과
‡아칸소주립대학교 컴퓨터과학과
777bareman777@gmail.com, dudgh1002@naver.com, dhkim717@gmail.com ,dshwang@dankook.ac.kr

Opcode category sequence feature and machine learning for analyzing IoT malware

Sunghyun Mun†, Youngho Kim†, Donghoon Kim‡, Doosung Hwang†
†Dept. of Software Science, Dankook University
‡Dept. of Computer Science, Arkansas State University

요 약

IoT 기기는 취약한 아이디와 비밀번호 사용, 저사양 하드웨어 등 보안 취약점으로 인해 사이버 공격 진입점으로 이용되고 있다. 본 논문은 IoT 악성코드를 탐지하기 위한 op 코드 카테고리 기반 특징 표현을 제안한다. Op 코드의 기능별 분류 정보를 이용해서 *n*-gram 특징과 엔트로피 히스토그램 특징을 추출하고 IoT 악성코드 탐지를 위한 기계학습 모델 평가를 수행한다. IoT 악성코드는 기능 개선과 추가를 통해 진화하였으나 기계학습 모델은 훈련 데이터에 포함되지 않은 진화된 IoT 악성코드에 대한 예측 성능이 우수하였다. 또한 특징 시각화를 이용해서 악성코드의 비교 탐지가 가능하다.

1. 서론

IoT(Internet of Things)는 각종 사물들이 센서를 통해 데이터를 수집하고 공유하는 시스템이다. IoT 기기는 취약한 아이디와 비밀번호 사용, 저사양 하드웨어, 이기종 사물네트워크 연결 등 보안 취약점을 나타내고 있다[1]. 이로 인해 IoT 기기들은 사이버 공격 진입점으로 악용되고 있다.

2016 년 Mirai 악성코드는 수십만대의 IoT 기기를 감염시켜 DDoS(Distributed Denial-of-Service) 공격을 시도했다[2]. 신종 IoT 악성코드는 기존 악성코드를 기반으로 진화되고 있다[3]. Kaspersky 의 DDoS 공격 보고서에 의하면 2021 년에 발생한 Simps 악성코드는 Mirai 와 Gafgyt 를 기반으로 발전했다. 같은 연도에 발생한 ZHtrap 악성코드도 Mirai 를 기반으로 한다[4].

본 논문에서는 소프트웨어의 op 코드(operation code) 기반으로 IoT 악성코드 특징을 추출하고, 기계학습 모델을 활용하여 IoT 악성코드 탐지를 수행한다. 2 절에서 op 코드를 활용한 악성코드 탐지 관련 연구를 소개한다. 3 절에서 특징 추출 방법을 제안하며 4 절에서 데이터셋 구성에 따른 기계학습 모델의 성능을 비교 평가한다. 마지막 절에서 향후 연구 방향과 결론을 서술한다.

2. 관련 연구

Yuxin Ding 외 3 명[5]은 디컴파일된 실행파일의 분기 op 코드에 따라 CFG(control flow graph)를 분석했다. 프로그램의 실행 흐름에 대한 op 코드 시퀀스를 추출하고 *n*-gram 을 적용해서 특징을 표현했다. 정보획득(information gain)을 사용해서 특징 벡터의 차원을 축소하였으며, 의사결정트리(decision tree) 모델에서 94.0% 정확도(accuracy)를 얻었다.

Tran Nghi Phu 외 4 명[6]은 Angr[7]를 사용해 CFG 를 분석하고 Intel80386, MIPS 구조의 op 코드를 Vex IR(intermediate representation)로 변환했다. Vex IR 은 11 가지의 op 코드들로 구분한다. op 코드의 종류만 고려한 것을 level 1, 기능을 포함한 경우 level 2 로 정의했다. CFG 의 실행 경로에 따라 CFDVexlevel 1, CFDVexlevel 2 시퀀스를 추출하고, *n*-gram 을 적용해서 특징을 구성했다. 카이제곱(chi-squared)을 사용

해서 특징 벡터의 크기를 줄였다. SVM 모델로 분석 시 CFDVexlevel 1은 95.98%, CFDVexlevel 2는 97.02%의 정확도를 얻었다.

Hamid Darabin 외 4명[8]은 시퀀스 패턴 마이닝(sequence pattern mining) 알고리즘을 통해 MSP(maximal sequential pattern)를 찾는다. op 코드를 기능별로 6가지 카테고리로 나누고, MSP에서 카테고리 변화에 대한 빈도를 계산하여 특징을 구성했다. SVM, 랜덤포레스트(Random Forest), 의사결정트리, k-NN, 에이다부스트(AdaBoost), 다층 퍼셉트론(multilayer perceptron) 모델에서 모두 98% 이상 정확도를 보고했다.

3. 제안 방법

3.1. Op 코드 카테고리화

전체 데이터셋은 $D = \{P_1, P_2, \dots, P_N\}$으로 정의한다. P_i는 IoT 정상파일 또는 악성코드를 나타내고, N은 데이터의 총 개수이다. D에서 op 코드를 추출하기 위해서 objdump[9]를 사용한다. 추출된 op 코드 시퀀스를 $P_i =< o_1, o_2, o_3, \dots, o_{m_i} >$으로 정의한다. o_j는 op 코드를 나타내며 m_i는 P_i의 op 코드 수이다.

Hamid Darabin이 제안한 방법에 따라 op 코드 카테고리화를 수행한다[8]. op 코드 카테고리는 표준 데이터 처리 명령어(C1), 곱셈 명령어(C2), 분기 명령어(C3), 적재/저장 명령어(C4), 상태 레지스터 접근 명령어(C5), 기타 명령어(C6) 이다(표 1)[10]. 카테고리화 된 op 코드 시퀀스는 $P_i =< cat_1, cat_2, \dots, cat_{m_i} >$으로 재정의한다. cat_j는 o_j의 카테고리이다.

표 1. Op 코드 카테고리 분류

Category	Mnemonic
C1	AND, EOR, SUB, RSB, ADD, ADC, SBC, RSC, TST, TEQ, CMP, CMN, ORR, MOV, BIC, MVN, CDP
C2	MUL, MULL, MLA, MLAL
C3	B, BL, BX
C4	LDM, STM, LDR, STR, LDC, STC, MRC, MCR
C5	MRS, MSR
C6	DMB, DSB, ISB, NOP, SEV, SVC, WFE, WFI, SWI, SWP, ADR, FLDM, YIELD

3.2. 특징 추출

Op 코드 카테고리 기반 2가지 특징 추출 방법을 제안한다. 첫 번째는 P_i에 n-gram을 적용해서 카테고리화 된 op 코드 순서쌍 시퀀스를 생성하며, $NGRAM_i = < t_1, t_2, t_3, \dots, t_{T_i} >$ 으로 표현한다. t_j는 $C^n = \{(c_1, \dots, c_n) | c_i \in for\ every\ i \in \{1, \dots n\}\}$ 중 하나의 값으로 표현되며 T_i는 $NGRAM_i$의 길이이다. $NGRAM_i$에서 각 순서 쌍들의 발생 빈도를 계산하여 특징 벡터 v로 표현한다. 전체 데이터셋 D에 대한 n-gram 특징을 $V = \{v_1, \dots, v_N\}$로 구성시킨다.

두 번째는 P_i에 슬라이딩 윈도우를 적용해서 윈도우 시퀀스를 생성한다. 각 윈도우에서 카테고리 별 빈도수 f_{c_i}를 통해 확률 $p(C_i)$을 계산하고 엔트로피 $e(C_i)$를 구한다.

$$e(C_i) = -p(C_i)\log_2 p(C_i)$$
$$\text{where } p(C_i) = f_{C_i}/window\ size$$

0 ~ 1 사이를 균등하게 l개의 구간으로 나누고, $e(C_i)$를 매칭되는 구간에 누적하여 특징 벡터 u를 구성한다. 전체 데이터셋 D에 대한 엔트로피 히스토그램 특징을 $U = \{u, \dots, u_N\}$으로 표현한다.

4. 실험

4.1. 데이터 구성

악성코드 탐지를 위한 데이터셋은 멀웨어즈(Malwares)[11]에서 ARM CPU 구조에서 동작하는 IoT 정상파일/악성코드 19,890개를 제공받았다. 악성코드는 Dofloo, Gafgyt, Mirai, Tsunami 총 4개의 패밀리로 구성된다(표 2). Kaspersky 분류 기준에 따라 악성코드 패밀리가 분류되었다.

표 2. 데이터셋 D_1

Type	Family	No. of Data
Benign	-	2,593
Malware	Dofloo	844
	Gafgyt	8,582
	Mirai	7,404
	Tsunami	467
Total		19,890

표 3. 데이터셋 D_2

Dataset	Type	Family	No. of Data
Training Set	Benign	-	2,493
	Malware	Tsunami	467
Test Set	Benign	-	100
	Malware	Gafgyt	100

표 4. 데이터셋 D_3

Dataset	Type	Family	No, of Data
Training Set	Benign	-	2,493
	Malware	Tsunami	467
		Gafgyt	8,482
Test Set	Benign	Benign	100
	Malware	Mirai	100

데이터셋 구성에 따라 3가지 악성코드 탐지 실험을 진행한다. 각 실험에 사용된 데이터셋은 D_1, D_2, D_3으로 구분된다. D_1은 전체 데이터셋으로 구성된다(표

2). D_2는 정상파일과 Tsunami, Gafgyt 악성코드로 구성된다(표 3). D_3은 정상파일과 Tsunami, Gafgyt, Mirai 악성코드로 구성된다(표 4).

4.2. 실험 방법

IoT 악성코드는 기능 재사용/개선/추가를 통해 진화해왔다. 제안 방법이 기존의 IoT 악성코드의 특징을 추출할 수 있다면 기계학습 모델에 학습시키지 않은 악성코드에 대한 탐지가 가능할 것이다. 2-gram 특징 $V_{n=2}$, 엔트로피 히스토그램 특징 $(l=6)$ $U_{l=6}$과 데이터셋 구성(D_1, D_2, D_3)에 따른 3가지 분류 실험을 진행한다.

분류 문제 CP_1은 데이터셋 D_1을 이용한 5-식 교차 평가를 수행한다. CP_2는 데이터셋 D_2의 정상파일과 Tsunami를 통해 모델을 학습하고 Gafgyt 악성코드의 탐지 성능을 평가한다. CP_3은 데이터셋 D_3의 정상파일, Tsunami 와 Gafgyt를 통해 모델을 학습한 후 Mirai 악성코드 탐지 성능을 평가한다.

실험 성능을 평가하기 위해 5-NN, SVM, 의사결정트리(DT), 랜덤포레스트(RF) 모델을 사용한다. 성능 지표로 정확도(ACC), 탐지율(true positive rate, TPR), 오탐률(false positive rate, FPR), AUC-ROC(ROC), F1-score(F1)를 사용한다. 탐지율은 실제 악성코드를 악성코드라고 분류한 비율이며 오탐률은 정상파일을 악성코드로 분류한 비율이다.

4.3. 실험 결과

표 5는 분류 문제 CP_1의 실험 결과이다. 모든 모델에 대해서 $V_{n=2}$와 $U_{l=6}$은 정확도 98% 이상을 얻었다. 랜덤포레스트는 $V_{n=2}$에서 탐지율 99.9%와 오탐률 1.5%, $U_{l=6}$에서 탐지율 99.7%와 오탐률 2.5%로 가장 우수한 성능을 보였다.

표 5. 분류 문제 CP_1의 실험 결과

Feature	Model	ACC	TPR	FPR	ROC	F1
$V_{n=2}$	5-NN	0.987	0.999	0.090	0.954	0.971
	SVM	0.989	0.994	0.043	0.976	0.976
	DT	0.990	0.998	0.062	0.968	0.978
	RF	0.997	0.999	0.015	0.992	0.993
$U_{l=6}$	5-NN	0.983	0.997	0.112	0.943	0.961
	SVM	0.986	0.992	0.056	0.968	0.968
	DF	0.987	0.995	0.070	0.963	0.971
	RF	0.994	0.997	0.025	0.986	0.987

표 6은 CP_2의 실험 결과이다. $V_{n=2}$의 5-NN과 SVM, $U_{l=6}$의 5-NN은 CP_1의 실험 결과와 비슷하다. 이외의 모델은 낮은 성능을 보였다. 의사결정트리에서 $V_{n=2}$는 탐지율 83.0%, $U_{l=6}$은 탐지율 49.0%이며 랜덤포레스트에서 $V_{n=2}$는 탐지율 85.0%, $U_{l=6}$은 64.0%의 탐지율을 보였다. $V_{n=2}$ 보다 $U_{l=6}$의 악성코드 탐지 성능이 상대적으로 낮게 분석되었다.

표 6. 분류 문제 CP_2의 실험 결과

Feature	Model	ACC	TPR	FPR	ROC	F1
$V_{n=2}$	5-NN	0.985	0.990	0.020	0.985	0.985
	SVM	0.960	0.940	0.020	0.960	0.960
	DT	0.895	0.830	0.040	0.895	0.895
	RF	0.925	0.850	0.000	0.925	0.925
$U_{l=6}$	5-NN	0.975	0.960	0.010	0.975	0.975
	SVM	0.735	0.530	0.060	0.735	0.723
	DF	0.730	0.490	0.030	0.730	0.713
	RF	0.820	0.640	0.000	0.820	0.814

표 7은 분류 문제 CP_3의 실험 결과이다. CP_1의 실험 결과와 유사하며 분류 문제 CP_2의 실험 결과보다 향상됐다. 의사결정트리에서 $V_{n=2}$는 탐지율이 83.0%에서 91.0%로, $U_{l=6}$은 탐지율이 49.0%에서 96.0%로 증가했다. 랜덤포레스트는 $V_{n=2}$의 경우 탐지율이 85.0%에서 97.0%로, $U_{l=6}$의 경우 탐지율이 64.0%에서 97.0%로 향상됐다.

표 7. 분류 문제 CP_3의 실험 결과

Feature	Model	ACC	TPR	FPR	ROC	F1
$V_{n=2}$	5-NN	0.980	1.000	0.040	0.980	0.980
	SVM	0.975	0.980	0.030	0.975	0.975
	DT	0.940	0.910	0.030	0.940	0.940
	RF	0.980	0.970	0.010	0.980	0.980
$U_{l=6}$	5-NN	0.960	0.980	0.060	0.960	0.960
	SVM	0.970	0.990	0.050	0.970	0.970
	DF	0.940	0.960	0.080	0.940	0.940
	RF	0.975	0.970	0.020	0.975	0.975

4.4. 특징 시각화와 분석

실험 결과를 분석하기 위해서 정상파일과 악성코드의 특징을 시각화 한다. 사용된 op 코드의 수가 달라서 추출된 특징 값이 상이하다. 최소 최대 정규화(min-max normalization)을 적용하여 특징 값을 0~1 사이로 고정시켰다.

그림 1은 정상파일과 악성코드 특징 패턴을 보여주며 (a)는 $V_{n=2}$, (b)는 $U_{l=6}$이다. 컬러맵의 범위를 0~1로 지정했을 때 악성코드의 특징 패턴을 파악할 수 없었다. 분석을 용이하게 하기 위해 0.005 이상인 특징 값을 0.005로 고정시켜 특징을 재구성했다.

그림 1에서 정상파일과 악성코드의 패턴 차이가 보

인다. 그림 1 (b)의 5번째 정상파일의 패턴은 악성코드의 패턴과 유사하게 나타났다. CP_1에서 악성 코드 탐지 성능은 우수했지만, 랜덤포레스트를 제외한 5-NN, SVM, 의사결정트리에서 오탐률이 높았던 이유로 분석된다.

그림 1 (a)의 2, 4번째 Tsunami 특징 패턴은 5번째 Gafgyt 특징 패턴과 비슷하며 (b)의 2, 4번째 Tsunami 특징 패턴과 5번째 Gafgyt 특징 패턴 또한 유사하다. 하지만 Tsunami의 데이터 수는 467개이며 Tsunami 특징 패턴이 모든 Gafgyt 특징 패턴을 대표하지 않아 CP_2에서 낮은 성능을 보인 것으로 예측된다.

그림 1 (b)의 2번째 Gafgyt 특징 패턴과 5번째 정상파일의 특징 패턴이 유사하다. 반면 그림 1 (a)의 2번째 Gafgyt 특징 패턴과 5번째 정상파일 특징 패턴은 구별된다. CP_2에서 $V_{n=2}$가 $U_{l=6}$보다 결과가 우수한 이유로 분석된다.

그림 1에서 Mirai 특징 패턴과 유사한 특징 패턴을 확인할 수 없었다. 하지만 CP_3에서 Mirai 탐지 성능이 우수했다. CP_2에 비해 학습에 사용된 악성코드의 수가 많아서 모델 변별력이 개선되었다.

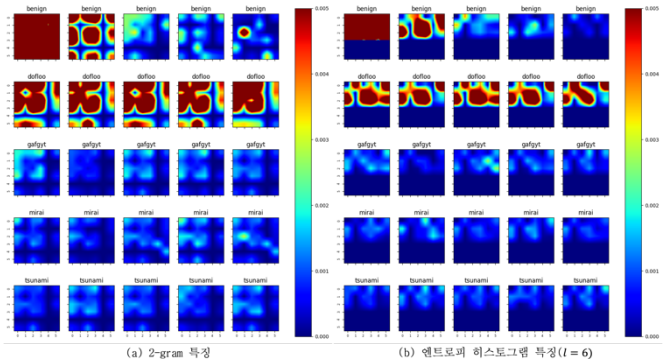

그림 1. Op 코드 카테고리 기반 특징 시각화

5. 결론

본 논문에서는 op 코드 카테고리를 활용해서 n-gram과 엔트로피 히스토그램 특징을 추출했다. 제안 방법은 CFG와 MSP를 찾을 필요가 없기 때문에 특징 추출 시간을 단축시킬 수 있다.

데이터셋 구성에 따라서 3가지 실험(CP_1, CP_2, CP_3)을 진행했다. CP_1에서 전반적인 우수한 성능을 보여줬으며 CP_3에서 모델 학습에 포함되지 않은 Mirai 악성코드를 탐지할 수 있는 가능성을 확인했다. 하지만 CP_2에서 2-gram의 5-NN과 SVM, 엔트로피 히스토그램 특징의 5-NN을 제외한 나머지 모델은 Gafgyt 악성코드 탐지 성능이 낮았다.

Tsunami 특징 패턴 중 일부는 Gafgyt 특징 패턴과 유사하며 Gafgyt 특징 패턴 중 일부는 정상파일 특징 패턴과 유사하다. 학습 데이터가 비교적 적은 CP_2에서 2-gram 특징과 엔트로피 히스토그램 특징은 특정 모델에 의존하는 경향을 보였다. 반면 CP_2보다 많은 데이터가 제공된 CP_3에서는 모델에서 비교 성능이 우수했다.

모델의 변별력 개선을 위해 추가적인 데이터 수집이 요구되며 IoT 기기에서 사용되는 다양한 아키텍처에 적용 가능한 일반화된 op 코드 카테고리와 특징 표현 연구가 필요하다.

Acknowledgement

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임[No.2019-0-00197, 스마트 퍼실러티 환경보호를 위한 신뢰기반 사이버 보안 플랫폼]

참고문헌

[1] 권혁찬 외, "차세대 IoT 네트워크 보안 현황", 정보처리학회지, 제 22 권, 제 2 호, pp. 43-44, 2015.

[2] Chris Williams, Today the web was broken by countless hacked devices [Online], https://www.theregister.com/2016/10/21/dyn_dns_ddos_explained.

[3] Q.-D. Ngo et al., "A survey of IoT malware and detection methods based on static features", KICS, vol. 6, no. 4, pp. 280-286, 2020.

[4] Alexander Gutnikov et.al, DDoS attacks in Q2 2021 [Online], https://securelist.com/ddos-attacks-in-q2-2021/103424.

[5] Yuxin Ding et al., Control flow-based opcode behavior analysis for Malware detection, Computers & Security, vol. 44, no. 1, pp. 65-74, 2014.

[6] Tran Nghi phu et al., CFDVex: A Novel feature Extraction Method for Detecting Cross-Architecture IoT Malware, SoICT, Hanoi, 2019., pp. 248-254.

[7] Yan Shoshitaishvili et al., State of The Art of War: Offensive Techniques in Binary Analysis, IEEE Symposium on Security and Privacy, San Jose, 2016.

[8] Hamid Darabin et al., An opcode-based technique for polymorphic Internet of Things malware detection, Concurrency Computation Practice and Experience, vol. 32, no. 6, 2019.

[9] objdump [Online], http://www.gnu.org/software/binutils.

[10] ARM-infocenter, Armv6-M Architecture Reference Manual [Online], https://developer.arm.com/documentation/ddi0419/e.

[11] Malwares [Online], https://www.malwares.com.