최근접 이웃 방법에 기반한 비정상 프로세스의 검출

정동호[1], 송상철[2], 김상욱[3*)]
[1]한양대학교 인공지능학과 박사과정
[2]한양대학교 인공지능학과 석사과정
[3]한양대학교 컴퓨터소프트웨어학과 교수
{mars9954, sangchul, wook}@hanyang.ac.kr

Suspicious Process Detection Based on Nearest Neighbors

Dongho Jeong[1], Sangchul Song[2], Sang-Wook Kim[3*)]
[1,2]Dept. of Artificial Intelligence, Hanyang University
[3]Dept. of Computer Science, Hanyang University

요 약

매년 급증하는 악성코드(malware)로 인해 기업, 공공기관 등 다수의 PC가 있는 대상까지 피해 사례가 늘고 있다. 악성코드로 인한 침해사고 흔적에서 비정상적인 동작을 한 프로세스를 찾는 기술은 해당 PC의 침해 여부 판단, 사후 대응 등 사이버 보안에 기여할 수 있을 것이다. 본 연구에서는 최근접 이웃 방법을 활용하여 시스템 메모리 데이터에서 비정상 프로세스를 검출하는 방안을 제시한다. 또한 실험을 통해 제안 방법이 정확도 및 여러 지표에서 우수한 성능을 달성함을 보였다.

1. 서론

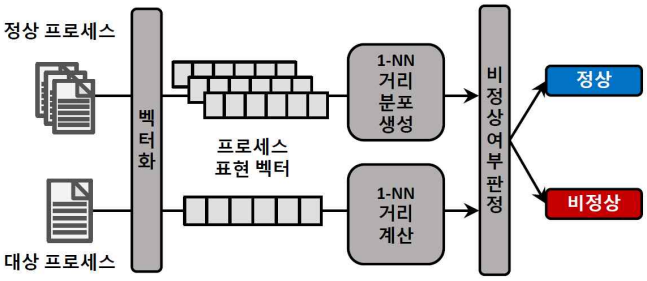
악성코드(malware)는 대상에게 피해를 입힐 목적으로 제작된 소프트웨어이다. 최근 들어, 악성코드의 공격은 기업, 공공기관 등 사회 혼란을 야기할 수 있는 목표물을 대상으로 하며[1], 이러한 경우 다수의 PC에 대해 침해 여부 조사를 해야 하므로 많은 수의 사이버 보안 전문가를 필요로 한다. 또한 악성코드는 시스템에 자신의 존재를 노출하지 않기 위해 정상 프로그램으로 위조하여 공격하기 때문에 보안 전문가의 조사를 어렵게 한다.[2] 따라서 시스템에서 악성코드를 검출하는 기술은 침해 조사, 사후 대응 등을 위해 효과적으로 활용될 수 있다.

본 연구에서는 시스템 메모리에 남은 프로그램의 실행 흔적에서 악성코드로 간주할 수 있는 비정상적인 프로세스를 검출하기 위한 머신러닝 프레임워크를 제안한다. 비정상 프로세스를 검출하는 문제는 같은 프로그램과 대응하는 프로세스들을 비교했을 때 상대적으로 다른 프로세스를 찾는 문제로 생각할 수 있으며, 프로세스 간의 거리를 이러한 프로세스 간의 '다름'으로 택하여 비정상 프로세스를 검출한다. 실험에서는 실세계 데이터셋을 사용하여 제안 방법이 비정상 프로세스 검출에 효과적임을 보였다.

*) 교신저자

2. 제안 방법

제안 방법은 1) 프로세스 표현 벡터화, 2) 1-NN 거리분포 생성, 3) 비정상 여부 판정의 3단계로 나뉜다.



(그림 1) 제안 방법의 개요

먼저, 비정상 여부를 검출하기 위해 각 프로세스를 벡터로 표현한다. 각 프로세스를 그것이 상호작용한 시스템 컴포넌트로 나타내기 위해 멀티핫(multi-hot) 벡터화를 진행한다. 예를 들어, 시스템 메모리에서 추출된 시스템 컴포넌트의 전체 집합이 {'a', 'b', 'c'}라면, 시스템 컴포넌트 'a', 'b'와 상호작용했던 프로세스는 [1, 1, 0]으로 표현할 수 있다.
만약 프로세스의 벡터가 영벡터이거나, 여러 프로세스의 벡터가 같은 경우, 시스템에 영향력이 없거나 중복인 표현이므로 필터링하여 제거한다.

프로세스를 벡터화한 이후, 학습 데이터셋의 프로그램마다 프로세스 간 거리를 기반으로 가장 가까운 이웃을 구한다. 프로세스 간 거리 계산은 자카드 거리

[4]를 사용했다. 이렇게 프로그램 별로 구한 정상 프로세스 간의 거리 분포를 간단히 1-NN(1-Nearest Neighbor) '거리 분포(distance distribution)'라고 한다.

마지막 단계에서는 1-NN 거리 분포를 결정 경계로 하여 주어진 프로세스의 비정상 여부를 판정한다. 판정 대상인 프로세스를 p_t로 표현하고, p_t와 같은 프로그램의 정상 프로세스 중 가장 가까운 이웃과의 거리 값을 d_t라고 하자. 만약 d_t가 사전에 정한 임계값 Θ보다 크다면 비정상 프로세스로, 그렇지 않다면 정상 프로세스로 판정한다. 이 때, 임계값은 검증 단계에서 경험적으로 결정한 값이다.

만약 p_t가 학습 데이터셋 내에 존재하지 않는 프로그램의 프로세스라면 그것이 정상인지 보장할 수 없기 때문에 어떠한 계산 없이 비정상 프로세스로 판정한다.

3. 실험

본 연구에서는 제안 방법의 효과를 입증하기 위해 사이버 보안 전문가가 다수의 PC가 있는 조직에 대한 악성코드 공격 시나리오를 바탕으로 생성한 데이터셋을 사용해서 실험을 수행한다. 전문가는 메모리 분석 도구인 'Volatility'[3]를 사용해 각 PC의 메모리에 남은 프로그램의 실행 흔적인 프로세스와, 시스템 컴포넌트인 DLL, Privilege, File을 추출했다. 각 컴포넌트의 전체 집합의 크기는 각각 2604, 35, 19176이다.

데이터셋은 학습, 검증, 평가 데이터셋으로 구성되어 있다. 학습 데이터셋은 전문가에 의해 정상으로 확인된 프로세스들로만 구성된다. 검증 및 평가 데이터셋은 공격당한 PC에서 추출한 프로세스들과, 학습 데이터셋에는 없는, 처음 등장하는 프로그램의 프로세스들이 일부 섞여 있다. 각 데이터셋 별 프로그램과 프로세스의 수는 표 1에 제시되어 있다.

<표 1> 데이터 정보 요약

구분	클래스	데이터셋		
		학습	검증	평가
프로그램		310	76	137
프로세스	정상	7,066	663	1,131
	비정상	-	4	5
	첫 등장	-	57	102
	합계	7,066	724	1,238

평가 지표로는 정확도(accuracy), 정밀도(precision), 재현률(recall), F1-점수, 실제 정상치 중 모델에 의해 이상치로 판정된 오류율인 False Positive Rate(FPR), 민감도(False Rejected Rate)를 사용했다.[4, 5] 지표 중 FPR, FRR은 오류율이므로 낮을수록, 나머지 지표는 높을수록 좋은 결과를 의미한다.

4. 결과 및 분석

표 2에서는 평가 데이터셋을 대상으로 제안 방법을 적용했을 때 지표별 가장 높은 성능을 제시한다.

실험 결과, 오류율인 FPR과 FRR은 각각 0.001, 0으로 낮은 수치를 보이는 동시에, 정확도 및 재현율은 0.999, 0.978, 1.000으로 높은 값을 보이므로 제안 방법이 비정상 프로세스의 탐지에 효과적임을 알 수 있다.

<표 2> 평가 데이터셋 실험 결과

Acc.	Prec.	Rec.	F1	FPR	FRR
0.999	0.978	1.000	0.976	0.001	0

5. 결론

본 연구는 프로세스 사이의 거리를 비교하여 비정상 프로세스를 검출하는 방법에 대해 다루었다. 제안 방법의 우수성을 보이기 위해 보안 전문가가 구성한 데이터셋을 이용해서 실험한 결과 높은 효과를 확인했으며, 해당 분야에 대한 전문 지식 없이 원본 데이터만을 사용했음에도 사이버 보안 분야의 문제에 좋은 성능을 보였다.

사사

이 연구 논문은 ETRI부설연구소의 위탁연구과제 [2022-010]로 수행한 연구결과입니다. 이 논문의 내용을 발표할 때에는 ETRI부설연구소에서 수행한 위탁결과임을 밝혀야 합니다.

또한 이 논문은 (1) 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원과 (No. 2020-0-01373, 인공지능대학원지원(한양대학교)), (2) 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구입니다 (No. 2018R1A5A7059549).

참고문헌

[1] Kim et al., "A human-in-the-loop approach to malware author classification", In *proc. of* ACM CIKM, 2020, pp. 3289-3292.

[2] Wang et al., "Attentional Heterogeneous Graph Neural Network: Application to Program Re-identification" In *proc. of* SDM, 2019, pp. 693-701.

[3] Volatility 2018. Volatility: Memory Forencis System. https://www.volatilityfoundation.org/.

[4] Han et al., Data Mining: Concepts and Techniques. Morgan Kaufmann, 2006.

[5] Ford et al., "Applications of machine learning in cyber security", In *proc. of* CAINE, 2014, pp.118-123.