Journal of Convergence for Information Technology
Vol. 11. No. 4, pp. 55-63, 2021
e-ISSN 2586-4440
DOI : https://doi.org/10.22156/CS4SMB.2021.11.04.055

그래프 데이터베이스 기반 악성코드 행위 탐지 기법

최도현[1], 박중오[2*]
[1]숭실대학교 컴퓨터학과 학생
[2]성결대학교 파이데이아학부 조교수

Graph Database based Malware Behavior Detection Techniques

Do-Hyeon Choi[1], Jung-Oh Park[2*]
[1]Student, Computer Science, Soongsil University
[2]Assistant Professor, Division of Paideia, Sungkyul University,

요 약 최근 악성코드 발생률은 약 수만 건이 넘는 추세로, 전부 탐지/대응하는 것은 불가능에 가깝다고 알려졌다. 본 연구는 새로운 악성코드 대응방법으로 그래프 데이터베이스 기반 다중행위 패턴 탐지 기법을 제안한다. 기존 동적 분석 기법과는 다른 새로운 그래프 모델을 설계하고, 대표적인 악성코드 패턴(프로세스, PE, 레지스트리 등)의 그래프 연관 관계를 분석하는 방법을 적용했다. 패턴 검증 결과 기본 악성 패턴에 대한 행위 탐지와 기존 분석이 어려웠던 변종 공격 행위(5단계 이상)의 탐지를 확인했다. 또한, 성능 분석결과 5단계 이상의 복잡한 패턴에 대하여 관계형 데이터베이스 대비 약 9.84배 이상 성능이 향상되었음을 확인하였다.

주제어 : 악성코드, 그래프 데이터베이스, 행위 분석, 연관 분석, 패턴 분석

Abstract Recently, the incidence rate of malicious codes is over tens of thousands of cases, and it is known that it is almost impossible to detect/respond all of them. This study proposes a method for detecting multiple behavior patterns based on a graph database as a new method for dealing with malicious codes. Traditional dynamic analysis techniques and has applied a method to design and analyze graphs of representative associations malware pattern(process, PE, registry, etc.), another new graph model. As a result of the pattern verification, it was confirmed that the behavior of the basic malicious pattern was detected and the variant attack behavior(at least 5 steps), which was difficult to analyze in the past. In addition, as a result of the performance analysis, it was confirmed that the performance was improved by about 9.84 times or more compared to the relational database for complex patterns of 5 or more steps.

Key Words : Malware, Grape Database, Behavior Analysis, Association Analysis, Pattern Analysis

1. 서론

2020년 상반기 기준 국내에서 발생한 악성코드는 트로이목마(71%), 커뮤니티(44%), 익스플로러(92%), 정보유출(67%)로 주로 웹 스크립트와 이메일, 애플리케이션을 통한 악성코드 감염이 큰 비중을 차지했다[1,2]. 악성코드 분석 분야는 기존 사람이 직접 통계분석 해야 했던 영역을 자동화하고, 새로운 보안 탐지/분석/예측하는 기술에 관심이 크다[3]. 기존 시그니처(Signature) 기반 탐지의 한계를 극복하기 위해, 내부 바이너리(Binary) 분석자료 수십만 건에서 특징을 추출하고 벡터변환, 군집화, 클러스터링 등 다양한 기계학습 알고리즘을 조합하여 악성코드를 분석하는 형태로 발전하고 있다[4]. 일반적으로 PE, PDF, ELF, HWP, DOC 등 실행 및 문서 파일과 이메일 등 악성코드 감염 비중이 높은 트로이목마(Trojan)와 호스트 파일

*Corresponding Author : Jung-Oh Park(pjo21@naver.com)
Received March 15, 2021
Revised April 6, 2021
Accepted April 20, 2021
Published April 28, 2021

(Host) 등을 대상으로 분석한다[5]. 근 미래에는 정형/비정형 데이터를 어떻게 다양하게 다방면으로 분석/예측하는 것이 핵심 기술이 될 것이다. Gartner Data & Analytics 2019에 따르면 10대 데이터 및 분석 기술로 '그래프 데이터 분석'을 포함했다[6]. Allied Market Research는 2026년까지 그래프DB 시장이 2019년 이후 연평균 24.5% 성장하여 2026년에는 37억 3,100만 달러에 이를 것으로 예상했다[7]. 급속도로 변화하는 데이터 분석 시장에서 그래프DB 기술에 기대가 높다는 것을 알 수 있다.

본 연구는 그래프 데이터베이스 기반 악성코드 행위 탐지 기법을 제안한다. 기존 악성코드 분석 엔진(정적/동적)과는 다른 그래프 이론을 기반으로 유사 변종, 공격 루트 추적, 공격 우회 등 연관 관계를 분석하는 기법이다. 본 논문의 구성은 다음과 같다. 2장은 관련 연구로써 그래프DB와 악성코드 패턴에 대하여 설명한다. 3장은 제안하는 그래프DB 모델과 악성코드 행위 탐지 기법을 설명한다. 4장은 제안 기법의 성능과 안전성을 비교 분석한다. 5장은 결론으로 맺는다.

2. 관련 연구

기존 보안 분야 악성코드 분석의 한계와 그래프DB 분야에서 선행 연구 및 사례를 분석하고, 제안 기법에 적용하는 악성코드 행위 탐지 분석을 위한 공격 패턴을 분석한다.

2.1 그래프DB 사례 및 연구 분석

그래프 데이터베이스(Graph Database)는 그래프 이론을 기반으로 관계(Relation)를 표현 및 저장하고 제어하는 데이터베이스를 의미한다. 소셜, 추천 엔진, 이상 탐지 등 분야에서 조인 및 트래버스(Traverse) 등 쿼리(Query) 성능이 높은 것으로 알려졌다[8,9]. Fig. 1은 네트워크 모델의 노드를 관계를 표현하는 제조업 제품 생산 사례를 나타낸다[10]. 전체 생산 과정에서 관계 분석을 통해 특정 관계 패턴의 비효율성을 추측/분석할 수 있다. 기존 관계형 데이터베이스와 다른 노드를 연결하는 연관 관계를 중심으로 문제를 탐색하는 방식이다. 다른 예로 사용자 A와 B가 특정 상품을 여러 개 구매할 때, 사용자 A가 다음 선택한 상품을 예측하여 B에게 상품을 추천할 수 있다. 노드의 연관 관계 분석한다는 것은 이러한 불량 제품, 상품 추천 등 연관 패턴을 예측하고 판단하는데 적절한 방법을 제공한다.

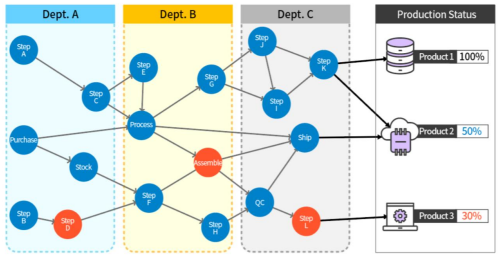

Fig. 1. Performance Management System, Bitnine Use case

최근 수년 이내 관련 연구에는 소프트웨어 코드 분석, 네트워크 장애 분석, GPU 기반 기계학습, 서사자료의 메타분석, 지식 데이터베이스 등이 있다[11-15]. 구글 학술검색 대상 "그래프 데이터베이스" 키워드 기준 수십 건 이하로 타 분야 연구와 비교하여 아직 연구 초기 단계임을 알 수 있다. Fig. 2는 실제 은행에서 이상 거래를 분석하기 위한 그래프 모델 분석 사례를 나타낸다[16].

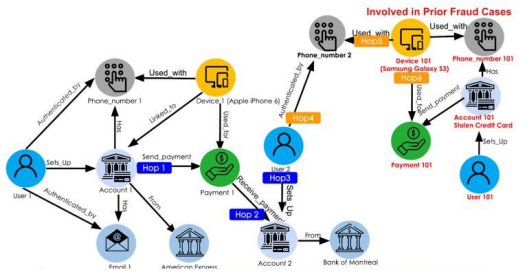

Fig. 2. Frad Detection - Regular vs. Advanced Analytics with Graph

그래프 기반 연관 분석 기법으로 복잡한 이상 거래 패턴을 탐지한다. 은행을 통한 결제에서 사용자의 핸드폰 전화 인증 연결 패턴에 이상징후를 탐지/분석한다. 이외 현석우의 연구에 의하면 우선순위별 그래프 연계 분석이 조직의 중요 자산을 대상으로 정확한 문제점을 파악하는데 최적화하였고, 정우철에 연구에 의하면 기계학습을 기반으로 기존 관계형 데이터베이스보다 빠

른 성능과 예측하기 힘든 이상징후 패턴을 추출할 수 있음을 증명했다[17,18]. 이외 대표적인 사례로 파나마 페이퍼(Panama Paper)로 알려진 조세 피난처 탐사 및 폭로 보도 사례가 있다[19]. 연구/사례를 살펴본 결과 그래프DB는 NoSQL 타입 데이터베이스로써 악성코드 분석을 위한 세부 분야로는 부정을 탐지하는 이상징후 분석이 비교적 유사한 것으로 보인다. 본 연구는 악성코드 데이터에서 의미 있는 패턴을 탐색/분석하는 방법으로 활용하였다.

2.2 빅데이터 기반 악성코드 분석

"잉카 인터넷 시큐리티 대응센터"에 따르면 2020년 말 기준 악성코드 유형 비율 중 트로이목마(73%), 바이러스 또는 웜(22%) 순서로 높은 비중을 차지했다[20]. 트로이목마와 같은 악성코드는 일정 기간 은닉했다가 특정 조건에 해커에 명령에 따라 C&C 서버에 접속하고, 해당 악성코드를 제어/수행한다. 최근 DDoS 공격이나 네트워크 감염 확산과 함께 랜섬웨어(Ransomware)와 같은 복합적인 악성코드로 발전하고 있다[21]. Fig 3과 같이 기존 정적 분석 모델은 직접 프로그램을 실행하는 동적 행위기반 탐지 기술로 발전했다[22]. 가상의 샌드박스(Sand-box)와 같은 가상화(Virtualization) 공간에서 실행을 자동화하는 형태이다.

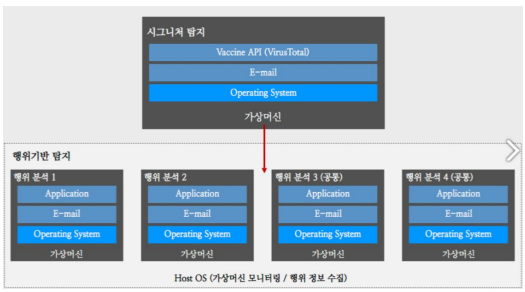

Fig. 3. Virtual Machine Cluster Configuration - Behavior-Based Analysis Example

행위 분석은 일반적으로 문서 파일, 네트워크 통신, 관련 프로세스, 호출한 API 등을 대상으로 이상 행위로 의심되는 악성코드 패턴을 분석한다. 주요 특징에는 PE header, Strings, DLL/API, CPU/Memory, IP주소 및 포트 번호 등의 추출된 대량의 레코드(Record)로 구성된 프로세스 테이블(Table)을 전수 조사한다. 이후 악성 행위 테이블과 조인(Join) 연산을 통해 위협 탐지를 반복하는 구조이다. 문제는 신규/변종 악성코드의 경우 행위 단계(Depth)가 복잡하여 전수조사량이 기하급수로 증가하게 되는데, 이는 관계형 데이터베이스 스키마(Schema) 구조에서 큰 시스템 자원을 요구하게 된다. 이는 최근 일 평균 약 수만 개 이상 발생하는 각종 변종 및 신규 악성코드에 대응에 기존 동적/정적 악성코드 분석에는 한계가 있다는 것을 의미한다. 이러한 한계를 극복하는데 "정보보호 R&D 데이터 챌린지"에 의하면 2018년 이후 최근까지 AI 기반 악성코드 탐지로 매년 95% 이상의 정확도를 도출하여 성능을 입증했다[23]. Fig 4와 같이 머신러닝(Machine Learning) 알고리즘들을 활용하여 데이터의 특성을 추출하고 그룹화 및 분류 문제 등을 적용할 수 있다. 현재 AI 관련 연구들이 회귀(Regression) 또는 분류(Classification)에 서포트 벡터 머신(SVM, Support Vector Machine), 결정 트리(Decision Tree) 등을 일반인 기계학습을 적용하는 단계이다. 인공지능 분석은 고도화된 빅데이터 플랫폼을 활용하여 다양한 데이터의 수집/저장 기능을 전자동화한다. 수많은 신규/변종 악성코드에 대응하기 위해서는 대용량 악성코드 저장/처리에 적합한 전용 데이터베이스의 적용이 필수적인 기술 요구사항이라고 볼 수 있다.

Fig. 4. Machine Learning Grouping Classification - Behavior Analysis Example

3. 그래프 데이터베이스 기반 악성코드 행위 탐지 기법

3.1 악성코드 패턴 정의 및 종류

악성코드의 이상 행위를 분석하기 위해 프로세스(Process), 파일(File), 레지스트리(Registry), 네트워크(Network)를 중심으로 그래프 데이터베이스 모델링을 수행했다. Table 1과 같이 악성코드의 행위는 주로 생성, 종료, 실행, 변조, 삭제 등 특정 네트워크에 연결하여 데이터를 송수신하는 행위 등으로 정의된다.

Table 1. Malicious Code Behavior Analysis Item Example

Target	Action
Process	Create Process, Thread
	Kill Process, Open Process
	Queue APC in Thread
	Add Windows Hook
File	Read, Open, Rename, Delete
	Copy, Create, Modify, Load
	Set File Attribute
Registry	Read, Open Registry Key(Value)
	Delete Registry Key(Value)
	Create, Modify Registry Key
Network	Connect to Socket, URL
	Send Data on Socket

기본 프로세스에 대한 파일명, 경로, 크기 등의 고유 정보를 참조하여 상속 관계를 맺는 이전(Previous) 프로세스 ID와 대상(Target) 프로세스 ID 등 정보를 적재한다. 이외 프로세스 행위로부터 파생되는 파일(설치 및 실행), 레지스트리(키 및 값 변조) 등으로부터 악성 코드 패턴의 특징을 추출한다.

3.2 그래프DB 모델링

그래프 맵 형태로 적재된 데이터는 다양한 형태에 변종 공격의 경로를 객체 간 관계의 흐름 추적을 통해 약성 여부를 탐지할 수 있다. Fig. 5와 같이 그래프 모델에 저장된 레이블만 보고 악성 행위를 판별한다.

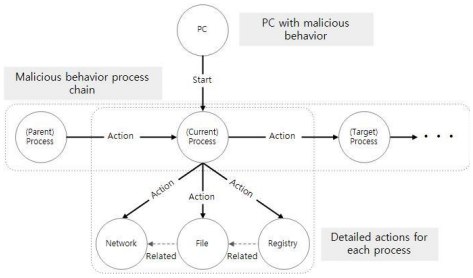

Fig. 5. Malware behavior Graph Modeling Example

프로세스의 경우 생성, 종료, 호출하는 행위와 스레드의 비동기 함수를 실행 및 윈도우(Window)를 후킹(Hooking)하는 행위를 탐지한다. 파일과 레지스트리의 경우 조작 행위(라이브러리 및 API 포함)를 탐지하고, 네트워크는 특정 주소, 소켓에 연결 및 데이터 송수신 행위를 탐지한다. 객체(Node)와 관계(Edge)에 표현된 텍스트는 레이블 이름을 의미하고, 특정 PC에서 수행된 프로세스의 행위(Edge)를 모델링한다. Fig 6와 같이 해당 데이터를 통해 악성 행위의 구체적인 정보를 파악하고, 악성 여부를 판별하기 위해 행위 레이블의 상세 데이터를 속성값(각 고유 식별 정보)으로 정의한다.

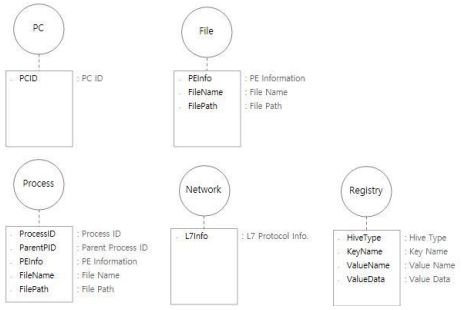

Fig. 6. Malware Behavior Property Modeling Example

3.3 악성코드 패턴 분석 방법

Fig. 7은 기본 프로세스 대상으로 악성코드 패턴을 탐지하는 세부 동작 과정을 나타낸다.

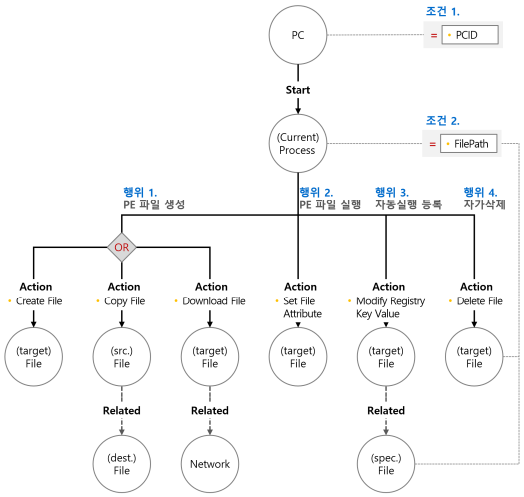

Fig. 7. Malicious Code Behavior Modeling - Basic Detection Example

첫 번째, PE 파일을 생성/복사하는 행위로부터 출발한다. 이외 네트워크를 통해 대상 파일을 내려받는 행위를 탐지한다. 두 번째, PE 파일 실행 행위를 수행하고, 대상 파일에 대한 속성값을 변조하는 행위를 탐지한다. 세 번째, PE 파일을 자동실행 등록, 대상 파일의

레지스트리 키값을 변조하는 특정 파일에 자동실행을 탐지한다. 이외 대상 파일을 자가삭제하는지 확인할 수 있다.

Fig. 8은 랜섬웨어와 같은 다중파일을 조작(파일의 이름 변경, 수정, 복사, 삭제)하는 행위를 탐지하는 세부 동작 과정을 나타낸다. 아래 4가지 행위는 복잡한 형태로 수행되기 때문에 검사 엔진에서 이를 다방면으로 실시간 검사가 요구된다.

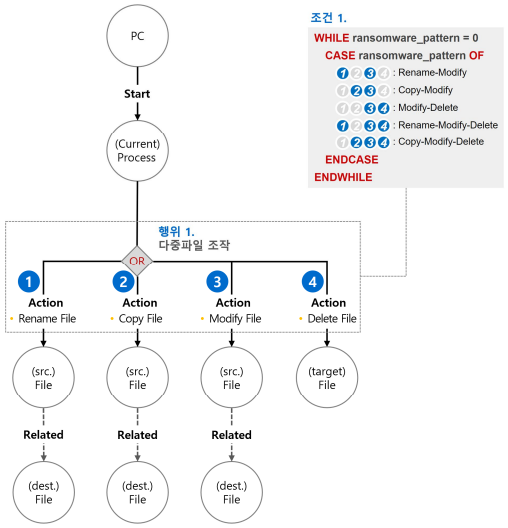

Fig. 8. Malicious Code Behavior Modeling - File Manipulation Example

첫 번째, 파일 이름 변경 후 암호화 등 감염 후 암호화되는 파일 확장자를 통해 랜섬웨어 종류를 파악할 수 있다. 두 번째, 대상 파일 복사 및 변조 등 알려진 호스트 파일 목록을 모니터링하여 백신 프로그램을 무력화와 우회 여부를 검사할 수 있다. 세 번째, 대상 파일와 삭제 등 악성코드 감염 이후 자체 악성코드 흔적을 삭제하는 행위까지 검사한다. 네 번째, 대상 파일을 복사한 뒤, 파일 내용을 변조하고, 해당 파일들을 삭제하는 행위 등이 있다. Fig. 9는 인젝션 당한 프로세스의 의심 행위를 탐지하는 세부 과정을 나타낸다. 특정 PE 파일을 생성한 뒤, 해당 파일을 자동실행 되도록 레지스트리 키 값 변조와 키 생성 행위 등을 검사한다.

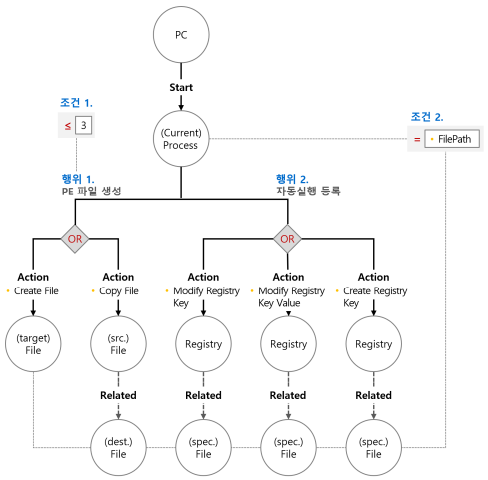

Fig. 9. Malicious Code Behavior Modeling - Registry Example

Fig. 10은 대상 프로세스가 시스템 프로세스에 인젝션을 시도하는 패턴의 세부 과정을 나타낸다. 첫 번째, 대상 프로세스에 스레드(Thread)를 생성과 콜백 함수 호출 등 윈도우 훅을 추가(Windows Hook)하는 행위를 수행하여 인젝션 대기 상태를 점검할 수 있다. 앞서 3가지 모델은 주요 실행 프로세스와 다중파일로부터 연결된 네트워크 정보와 접속 주소, IP주소, 포트 번호 등을 모니터링 한다.

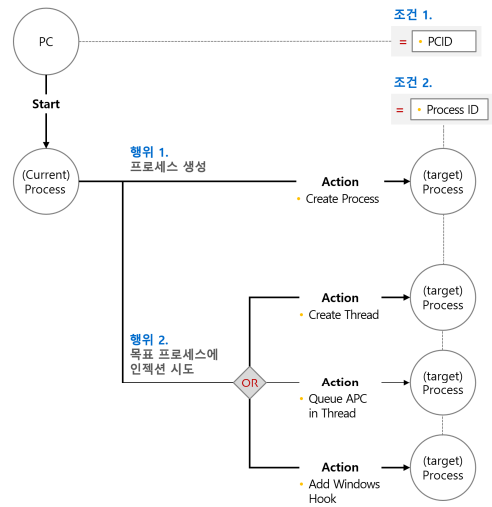

Fig. 10. Malicious Code Behavior Modeling - Injection Example

4. 악성코드 행위 탐지 및 성능평가

1.1 환경 구축 및 설정

본 성능평가에서 구축된 실행 환경은 Table 2와 같이 리눅스 기반의 Amazon S3를 사용하고, Graph DBMS를 구축하였다. 1차 데이터 가공 작업으로 하둡 빅데이터에 저장된 원본 파케이(Parquet) 파일 3.4Gb를 필터링하고 변환하여, 전용 CSV 파일로 추출했다.

Table 2. Test Environment Details

	Spec
OS	Amazon Linux AMI
CPU	40 CPU(Intel Xeon E5-2676 v3)
Memory	157GB
SSD	2TB

1.2 행위 탐지 및 성능평가

제안한 그래프 모델을 기반으로 악성코드를 검증하고, 백신 소프트웨어를 공급하는 A사 등에서 활용 중인 관계형 데이터베이스(RDB)의 행위 탐지 모델과 성능을 비교·분석한다. Fig 11은 악성코드 기본 패턴을 탐지한 결과이다. 기본적으로 행위의 1단계는 시작 객체인 PC와 행위 관계인 Start_event 외의 연결로 시작된다. 1단계를 제외하고 일치(Match)되는 악성코드 기본 패턴은 5, 6단계의 수준까지 연관 관계로 구성된 것을 알 수 있다.

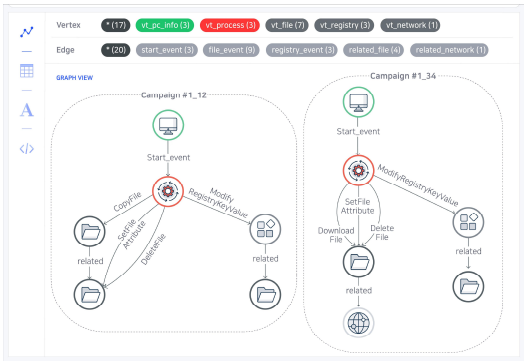

Fig. 11. Malicious Code Default Pattern Detection Result Example

Fig. 12는 기본 패턴 탐지 성능 분석결과를 나타낸다. 제안 모델은 RDB 모델 대비, 5단계 884%, 6단계 8014%로 5단계 9.84배, 6단계 81.14배 성능이 향상되었다.

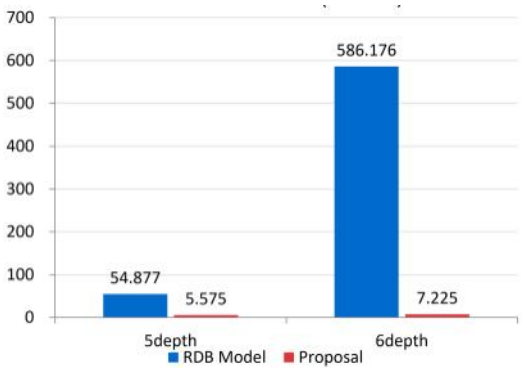

Fig. 12. Malware Basic Pattern Detection Performance Analysis

Fig. 13은 다중파일의 이름을 변경한 뒤 파일 내용을 수정하는 패턴을 탐지한 결과이다. 악성코드 다중파일 조작은 4단계의 수준까지 행위가 연관 관계로 구성된 것을 알 수 있다.

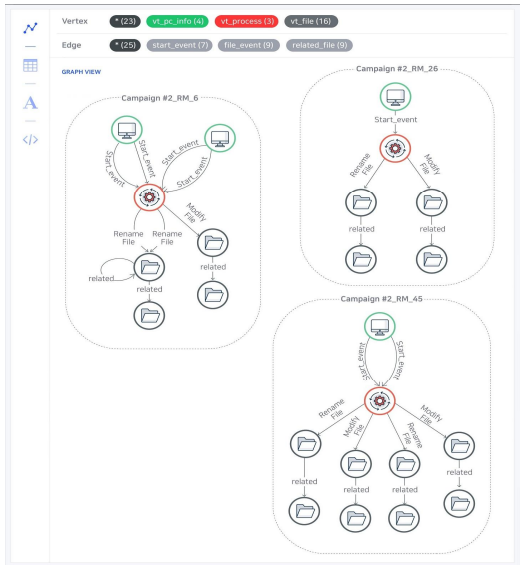

Fig. 13. Malware Multiple File Manipulation Detection Results Example

Fig. 14는 다중파일 탐지 성능분석 결과를 나타낸다. 제안 모델은 RDB 모델 대비, 216%로, 2.16배 성능이 향상되었다.

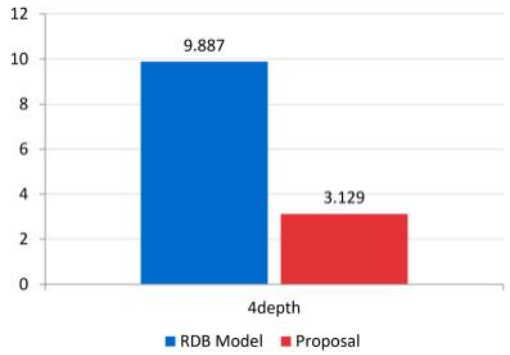

Fig. 14. Malware Multiple File Manipulation Detection Performance Analysis

Fig. 15는 인젝션을 당한 시스템 프로세스가 레지스트리 변조 행위를 수행하는 패턴을 탐지한 결과이다. 레지스트리 변조 행위는 2단계의 수준까지 연관 관계로 구성된 것을 알 수 있다.

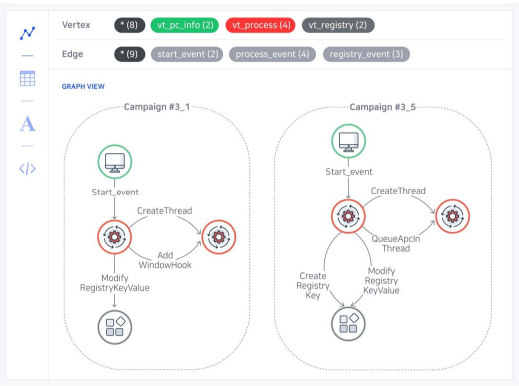

Fig. 15. Malicious Code Registry Manipulation Detection Result Example

Fig. 16은 악성코드 레지스트리 조작 탐지 성능 분석결과를 나타낸다. RDB 모델은 0.0239초에 탐지하였으며, 제안 모델은 0.0208초에 탐지하였다. 15% 성능을 개선, 1.15배 성능이 향상되었다.

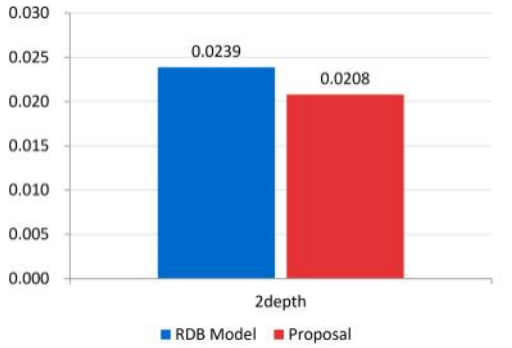

Fig. 16. Malware Registry Manipulation Detection Performance Analysis

Fig. 17은 생성된 프로세스가 시스템 프로세스에 인젝션을 시도하는 행위를 수행하는 패턴을 탐지한 결과이다. 2단계의 수준까지 연관 관계로 구성된 것을 알 수 있다.

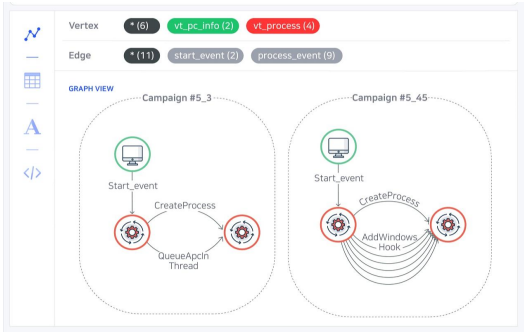

Fig. 17. Malware Process Injection Detection Results Example

Fig. 18은 악성코드 프로세스 인젝션 탐지 성능 분석 결과를 나타낸다. RDB 모델은 0.0241초에 탐지하였으며, 제안 모델은 0.0213초에 탐지하였다. 13% 성능을 개선, 1.13배 성능이 향상되었다. 각 항목의 성능 분석 결과, 제안하는 그래프DB 기반 행위 탐지 기법이 기존 RDB에 비교하여 다중행위 연관 분석에 최적화되어 있다는 사실을 알 수 있다. RDB 모델은 행위의 단계가 복잡해질수록, 탐지시간의 격차가 기하급수적으로 증가함을 알 수 있다.

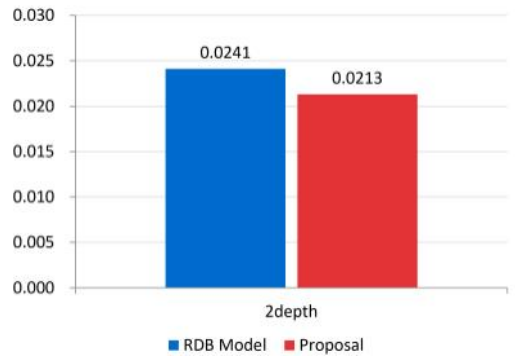

Fig. 18. Malware Process Injection Detection Performance Analysis

5. 결론

제안하는 연구는 기존 전통적인 RDB 모델과는 다른 방식의 그래프 맵을 구현하여, 악성 행위 탐지 및 검증을 수행한다. 새로 설계된 그래프 모델은 악성코드의 다중행위를 직관적으로 표현하고, 행위를 각 단계를 빠르게 추적(Trace)할 수 있고, 악성 행위 연관 분석에 최적화되었다. 기존 RDB 모델의 전수조사 분석은 매우 비효율적으로 실시간 악성코드 분석에는 적절하지 않음을 본 연구에서 증명하였다. 향후, 본 연구는 기본 패턴에서 복잡한 다중행위 패턴으로 분석을 확대하여 새로운 규칙(Rule)을 실시간으로 학습하고, 유사도를 비교 분석하는 기계학습/머신러닝 엔진 개발이 목적이다. 장기적으로는 악성코드 분석 구조(Framework)로써 악성코드 지식 그래프(Knowledge Graph)로 발전시킬 계획이다.

REFERENCES

[1] ESTsecurity. (n. d.). *ESTsecurity. Eastsecurity Security Trend Report No.136. 2021-01 - Malicious Code Statistics and Analysis*(Online). https://www.estsecurity.com/

[2] KISA. (n. d.). *Korea Internet & Security Agency. Malicious Code Hidden Site Detection Trend Report [First Half of 20]*(Online). https://www.boho.or.kr/

[3] K. W. Kook & B. C. Gong. (n. d.). *ITFIND. Trends in Security Technology Development Using Artificial Intelligence - Planning Series (Next Generation Security)*(Online). https://www.itfind.or.kr/

[4] S. J. Kim, J. H. Ha, S. H. Oh & T. J. Lee. (2019). A Study on Malware Identification System Using Static Analysis Based Machine Learning Technique. Journal of The Korea Institute of Information Security and Cryptology. 29(4), 775-784.
DOI : 10.13089/JKIISC.2019.29.4.775

[5] ESTsecurity. (n. d.). ESTsecurity. ESTsecurity Security Trend Report No.137. 2021-02 - Malicious Code Statistics and Analysis(Online). https://www.estsecurity.com/

[6] IDG - CIO Korea. (n. d.). *Gartner, Announcement of Top 10 Data and Analysis Technology Trends in 2019 - Trend 5 : Grape*(Online). https://www.ciokorea.com/

[7] J. S. Seo & H. J. Lee. (n. d.). *Bitnine.. Graph Database Technology Trends and Application Cases*(Online). https://www.itfind.or.kr/

[8] AMAZON. (n. d.). *Amazon Web Services. What Is a Graph Database? - The graph database defined*(Online). https://aws.amazon.com/

[9] W. C. Park. (2020). Is-A Node Type Modeling Methodology to Improve Pattern Query Performance in Graph Database. *Journal of The Korea Society of Computer and Information, 25(4),* 123-131.
DOI : 10.9708/jksci.2020.25.04.123

[10] Bitnine. (n. d.). *AgensGraph Use Case #8. Collaboration/Performance Management System*(Online). https://bitnine.net/

[11] S. C. Sin. (2017). Static Code Analysis based on Graph Database. *Communications of the Korean Institute of Information Scientists and Engineers, 35(2),* 9-13.
UCI(KEPA) : I410-ECN-0101-2017-569-002204936

[12] W. C. Jeong, M. S. Jun & D. H. Choi. (2020). AMI Network Failure Analysis based on Graph Database. *Journal of Convergence for Information Technology, 10(7),* 41-48.
DOI : 10.22156/CS4SMB.2020.10.07.041

[13] D. H. Han & M. S. Kim. (2020). A Matrix Computation Engine and Applications based on Distributed GPUs for Large-scale Machine Learning. *Journal of Computing Science and Engineering, 38(8),* 8-17.

[14] T. R. Kim & J. J. Lee. (2020). A Study on the Structure of Muga by Ontology Method - Focusing on〈Woncheongang Bonpuri〉-. *Korean Folklore Society, 72,* 333-369.
DOI : 10.21318/TKF.2020.11.72.333

[15] J. Y. Kim & K. H. Ro. (2019). Construction of

Knowledge Base Based on Graph Database for College Student Career Advice Using Public Data. *Journal of the Institute of Electronics and Information Engineers of Korea. 56(10),* 41-48. DOI : 10.5573/ieie.2019.56.10.41

[16] RTInsights. (n. d.). *Todd Blaschka and Gaurav Deshpande, How the World's Largest Banks Use Advanced Graph Analytics to Fight Fraud*(Online). https://www.rtinsights.com

[17] S. W. Hyun & T. K. Kwon. (2019). A Study of Effectiveness of the Improved Security Operation Model Based on Vulnerability Database. *Journal of the Korea Institute of Information Security & Cryptology, 29(5),* 1167-1177. DOI : 10.13089/JKIISC.2019.29.5.1167

[18] W. C. Jeong, M. S. Jun & D. H. Choi. (2020). Association Analysis for Detecting Abnormal in Graph Database Environment. *Journal of Convergence for Information Technology, 10(8),* 15-22. DOI : 10.22156/CS4SMB.2020.10.08.015

[19] William Lyon. (n. d.). *Graph Visualization of Panama Papers Data In Neo4j - Revisiting ICIJ's Offshore Leaks In The Face Of The Latest Deutsche Bank Scandal* (Online). https://medium.com/

[20] INCA Internet Security Analysis & Response Center. (n. d.). *October 2020 Malware Statistics*(Online). https://www.estsecurity.com/

[21] T. H. Park, H. W. Lee & W. Shin. (2020). Propagation Modeling of WannaCryptor Wormable Malware. *Journal of The Korea Institute of Information Security and Cryptology, 30(3),* 389-396. DOI : doi.org/10.13089/JKIISC.2020.30.3.389

[22] dyaniworld. (n. d.). *Information Technology/Spear Phishing - A Study on Behavior-Based Discrimination for Spear Phishing*(Online). https://dyaniworld.tistory.com/

[23] S. J. Kim, J. H. Ha, S. H. Oh, T. J. Lee. (2019). A Study on Malware Identification System Using Static Analysis Based Machine Learning Technique. *Journal of the Korea Institute of Information Security & Cryptology, 29(4),* 775-784. DOI :10.13089/JKIISC.2019.29.4.775

최 도 현(Do-Hyeon Choi) [정회원]

· 2008년 2월 : 동서울대학교 컴퓨터소프트웨어학과 졸업
· 2010년 8월 : 숭실대학교 컴퓨터학과(공학석사)
· 2016년 3월 : 숭실대학교 컴퓨터학과(공학박사)

· 관심분야 : Mobile, Network Security, PKI, Virtualization
· E-Mail : cdhgod0@ssu.ac.kr

박 중 오(Jung-Oh Park) [정회원]

· 2000년 7월 : 성결대학교 컴퓨터공학과 졸업
· 2003년 3월 : 명지대학교 전자계산교육 석사
· 2011년 8월 : 숭실대학교 컴퓨터공학 박사
· 2016년 3월~현재 : 성결대학교 조교수

· 관심분야 : PKI, Network security, 암호학
· E-mail : pjo21@naver.com