creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

조선대학교
CHOSUN UNIVERSITY

[UCI]I804:24011-200000266802

2018년 2월

석사학위 논문

프로세스 마이닝을 이용한 변종 악성코드 탐지 기법

조선대학교 산업기술융합대학원

소프트웨어융합공학과

한 지 훈

조선대학교
CHOSUN UNIVERSITY

프로세스 마이닝을 이용한 변종 악성코드 탐지 기법

Variant Malware Behavior Detection Technique Using Process Mining

2018년 2월 23일

조선대학교 산업기술융합대학원

소프트웨어융합공학과

한 지 훈

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

프로세스 마이닝을 이용한 변종 악성코드 탐지 기법

지도교수 최 준 호

이 논문을 공학석사학위신청 논문으로 제출함.

2018년 2월

조선대학교 산업기술융합대학원

소프트웨어융합공학과

한 지 훈

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

한지훈의 석사학위논문을 인준함

위원장 조선대학교 교수 김 판 구 (인)

위 원 조선대학교 교수 신 주 현 (인)

위 원 조선대학교 교수 최 준 호 (인)

2017년 11월

조선대학교 산업기술융합대학원

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

목 차

ABSTRACT

Ⅰ. 서론 ····· 1
A. 연구 배경 및 목적 ····· 1
B. 연구 내용 및 구성 ····· 4

Ⅱ. 관련 연구 ····· 5
A. 악성 행위 탐지와 분류 방법 ····· 5
B. 프로세스 마이닝 ····· 7

Ⅲ. 프로세스 마이닝을 적용한 악성코드 탐지 방법 ····· 10
A. 시스템 구성도 ····· 10
B. API 함수 기반 속성 추출 ····· 12
C. 악성 행위 API 함수 패턴 ····· 13
D. 악성행위 탐지를 위한 프로세스마이닝 적용 ····· 14

Ⅳ. 실험 평가 방법 및 결과 분석 ····· 22
A. 실험 데이터 셋 ····· 22
B. 성능 평가 ····· 23

Ⅴ. 결론 및 제언 ····· 28

참고문헌 ····· 29

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

표 목 차

[표 2-1] 파일관련 API 함수 패턴 ····· 6
[표 3-1] API 함수 분류 ····· 12
[표 3-2] 악성행위별 추출된 API List 예 ····· 13
[표 3-3] 악성 행위 빈도별 API List ····· 14
[표 3-4] 대표적인 악성행위별 API 함수 ····· 15
[표 3-5] 악성 행위별 빈발 순차패턴 ····· 20
[표 3-6] Trojan 최소지지도 변화에 따른 패턴 수 ····· 21
[표 4-1] 악성행위에 따른 데이터 세트 ····· 22
[표 4-2] 최소지지도별 정탐률, 오탐률 ····· 23
[표 4-3] 악성 행위별 프로세스 예측 정확도 ····· 24
[표 4-4] 악성코드 분류 비교 실험 평가 ····· 25

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그 림 목 차

[그림 1-1] 악성코드 통계 ····· 2
[그림 2-1] 프로세스 마이닝 연구 분야 ····· 7
[그림 2-2] 프로세스 마이닝 흐름도 ····· 8
[그림 3-1] 전체 시스템 구성도 ····· 10
[그림 3-2] 프로세스 마이닝을 적용한 악성행위 API 함수 패턴 ····· 16
[그림 3-3] Ransomware API 함수 패턴 ····· 17
[그림 3-4] Trojan API 함수 패턴 ····· 18
[그림 4-1] Trojan.win32.Tunneler API 함수 패턴 ····· 24
[그림 4-2] Trojan.Win32.BHO.hdz API 함수 패턴 ····· 26

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

ABSTRACT

Variant Malware Behavior Detection Technique Using Process Mining

Jihoon Han
Advisor : Prof. JunHo Choi, Ph.D
Department of Software
Convergence Engineering
Graduate School of Industry
Technology Convergence,
Chosun University

In the past, most hacking and malicious acts via the Internet were carried out without any special purpose. However, recent cyber attacks have been carried out to gain financial and social benefits. In particular, the possibility of exposure to cyber threats such as Ransomware, Cloud, APT, and DDos has increased significantly due to the recent advancement of information technology. In addition, there is a need for continuous action to counter new and unknown variant malicious codes that continuously come out. A typical method for detecting variant malicious codes is heuristic based detection. Heuristic based detection is to predict and detect possible malicious codes in advance based on the malicious codes that have been detected so far. In general, heuristic diagnosis can be classified into signature and behavior-based diagnosis

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

depending on diagnosis time. Signature heuristic diagnosis presented in this paper to detect variant malicious behavior is a technique to diagnose a specific condition (code) that is suspected to be malicious codes. The key to this technique is to detect variants suspect of malicious behavior in case they have similar detection rules, patterns, or codes. However, there are some drawbacks with this technique, namely, the detection time, the error occurrence, and the high possibility of misdiagnosing a normal file as a malicious file. To overcome these drawbacks, we propose a variant malicious code detection technique using process mining. Process mining is to derive a process model from the event logs, which can be very useful for finding processes that are not visualized. Each process module is created by refining the API function data of the malicious file through the static analysis. The code is classified as a variant when it is set as an exceptional process pattern instead of a normal process pattern, based on the results of the process modeling and the features of the similarity between the normal file and the malicious file, the similarity among the classified malicious code groups, and the typical similarity of the malicious code groups.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Ⅰ. 서론

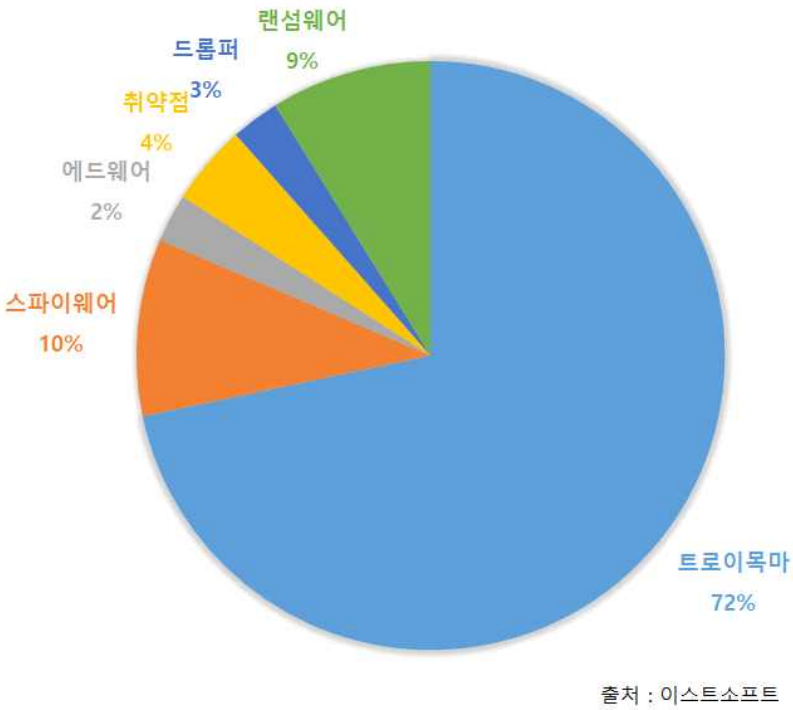
A. 연구 배경 및 목적

최근 변종 악성코드의 확산으로 인한 피해가 나날이 증가하고 있다. 사용자들의 컴퓨터가 감염되고 개인정보들을 감염된 사용자들의 컴퓨터로부터 노출시킨다. 그에 비해 변종 악성코드에 대한 대응은 국내외 안티바이러스 업체들의 기본적인 보안 시스템 만으로는 한계가 있는 실정이다. 이러한 악성 코드는 기술적인 면에서도 더욱 더 위험성을 더해져 현재 시그니쳐를 기반으로 한 기존 악성연구뿐 만 아니라 변종 악성탐지에 대한 탐지 및 분석을 수행하기 위한 연구가 진행 되고 있다[1].
악성코드가 한번 설치가 되면 공격자들은 PC 뿐 만 아니라 모바일 테플릿 등 모든 컴퓨팅 기기들을 완전히 통제 할 수 있으며 이를 이용하여 악의적인 행위(정보유출, 시스템 교란 및 파괴)를 한다.

하지만 수집된 악성코드 샘플 수는 절반에도 못 미치며 최근 새로 생성되는 악성코드는 대부분 변종(mutant)악성코드로 나타나고 있는 추세이며, 변종 악성코드에 대한 대응은 국내외 안티바이러스 업체들의 기본적인 보안시스템 만으로는 한계가 있는 실정이다. [그림 1-1] 은 전체 악성코드에 대한 연도별 통계와 2017년도 악성코드 유형별 비율에 따른 분류이다[2].

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY



[그림 1-1] 악성코드 통계

[그림 1-1]을 통해 최근 악성코드가 유형 중 트로이 목마가 약 72% 가장 높은 비율을 나타내고 있다 . 그 외에도 랜섬웨어, 원격제어 드롭퍼 등 다양한 악성코드 유형이 나타나고 있는 추세이다. 최근에 발견 되는 악성코드들의 대표적인 특징들을 보면, 이러한 악성코드가 많은 사용자들이 주로 사용하는 윈도우 뿐 만 아니라 랜섬웨어(Ransomware)와 같은 악성행위로 인한 심각한 사회적 문제로 자리잡고 있다. 이러한 변종 악성코드는 바이러스 백신의 탐지를 피하기 위해 암호화 또는 난독화와 같은 다양한 방법을 사용한다[3]. 이러한 악성행위들을 탐지하기 위한 다양한 연구가 활발하게 진행되고 있으며, Dissassemble을 통한 파일의 종류, 압축정보, 크기, 헤더 등의 다양한 내용을 확인하거나 여러 가지 특징들을 추출하고 분류를 통해 악성인지 판별이 가능하다[4]. 본 논문에서는 변종악성코드의 행위적 특징과 기존 악성코드와의 연관성을 API 함수패턴에 따라 분류하고 프로세스 마이닝을 적용하는 기법을 제안한다.

윈도우 내에서 악성코드가 실행되기 위해 보여지는 PE파일을 분석하여 악성코드에 삽입된 API 함수를 분류하고, 정상 파일과 악성코드에 따른 유사성 분류를 통

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

한 변종 악성코드와의 연관된 규칙과 패턴을 예측할 수 있는 프로세스 맵을 제시한다. 프로세스 마이닝(Process Mining)은 광범위한 정보 시스템을 사용하는 과정에서 기록된 데이터를 바탕으로 프로세스에 대한 유의미한 정보를 찾는 연구 분야로서, 데이터 중에서도 특히 프로세스와 관련된 데이터에 초점을 둔다 [5].

이에 본 논문에서는 악성코드와 변종 악성코드들이 가지는 API 함수 특징에 따라 변종에 대한 분류방법을 제안하고, 프로세스 마이닝을 통해 패턴을 예측하는 프로세스 맵을 제시한다. 프로세스 마이닝은 많은 로그 이벤트들의 발생 패턴을 설명할 수 있도록 프로세스 모델 개선 작업을 수행하고, 이를 통해 악성행위 반복 실행 패턴을 파악하고, 변종간의 유사성을 식별할 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

B. 연구 내용 및 구성

본 연구의 주요 내용은 변종 악성코드들에서 나타나는 API 함수패턴 정보를 프로세스 마이닝에 적용하여 변종 악성행위를 탐지하는 방법이며 이를 제안하기 위해 다음과 같은 구성한다.

서론에 이어 2장 관련 연구에서는 본 연구의 이론적인 배경이 되는 프로세스 마이닝(ProcessMining)의 기존 연구들에 대해 살펴본다.

3장에서는 본 논문에서 제안하는 악성코드 API 함수 분류와 악성코드가 실행됐을 때 행위적인 특징을 프로세스 마이닝 기법 적용방법에 관해 기술한다. 프로세스 마이닝으로 악성 API 함수들의 행위적인 특징을 통해 유사한 패턴에 따른 변종 악성행위 탐지를 제안한다.

4장에서는 실험환경과 실험에 사용한 각종 데이터 셋에 관해 설명하고, 실험을 통한 비교 및 성능을 평가한다.

마지막으로 5장에서는 결론과 향후 연구에 관해 서술하며 마무리한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Ⅱ. 관련 연구

A. 악성 행위 탐지와 분류 방법

악성코드 분석에 가장 일반적으로 PE파일 내의 사용되는 API 함수 정보를 활용하여 분석을 하게 된다. API 함수는 응용 프로그램 프로그래밍 인터페이스는 프로그램을 구성하고 있는 기본 요소로써 프로그램이 가지고 있는 다양한 특징적인 행위를 가지고 있어 악성코드만의 고유한 행위적 특징들을 추출하고 분석하는데 유용한 방법으로 활용되고 있다. API 함수를 이용하여 해당 파일이 악성 프로그램인지 정상 프로그램인지 구분하기 위한 연구와 악성코드들의 유사한 행위를 하며 새롭게 확산되는 변종 악성코드들을 분류하는 연구들이 진행 되었다[14]

Engin Kirda & Christopher Kruegel는 의심스러운 API 함수 목록을 사용하여 정적으로 스파이웨어를 검색 하고 추출된 API 함수 호출을 의심스러운 기능 목록과 비교하여 하나이상의 샘플 API 함수 호출이 의심스러운 것으로 간주되면 샘플은 스파이웨어로 분류 하였다. 하지만, 추출한 API 함수 중 WriteFile과 같은 양성 샘플이 악의적인 샘플과 동시적으로 나타나 FP(False Positive)는 줄어들고 FN(False Negative)이 늘어나는 오판의 가능성이 있다는 한계점과 Microsoft Internet Explorer의 Tool Bar나 BHO(Browser Helper Object)기반의 스파이웨어에 한정지어 연구하여 변형된 스파이웨어에 대한 API 함수 패턴 연구가 미흡하다는 한계점이 있다[14].

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

악성행위 그룹		주요 API 함수 패턴
Search File to Infact	Behaviour 1	FindClose FindFirstFile FindFirstFileEx FindFirstFileName TransactedW FindFirstFileNameW FindFirstFileTransacted FindFirstStream
Copy/Delete Files	Behaviour 2	CloseHandle, CopyFile CopyFileEx CopyFileTransacted CreateFile CreateFileTransacted
Get File Information	Behaviour 3	GetBinaryType GetCompressed FileSize GetCompressedFile SizeTransacted GetFileAttributes GetFileAttributesEx Transacted GetFileBandwidth Reservation GetFileInformation ByHandle
Move File	Behaviour 4	MoveFile MoveFileEx MoveFileTransacted MoveFileWithProgress
Read/Write Files	Behaviour 5	OpenFile OpenFileById ReOpenFile ReplaceFile WriteFile CreateFile CloseHandle
Change File Attributes	Behaviour 6	SetFileApisToANSI SetFileApisToOEM SetFileAttributes SetFileAttributesTransacted

[표 2-1] 파일관련 API 함수 패턴

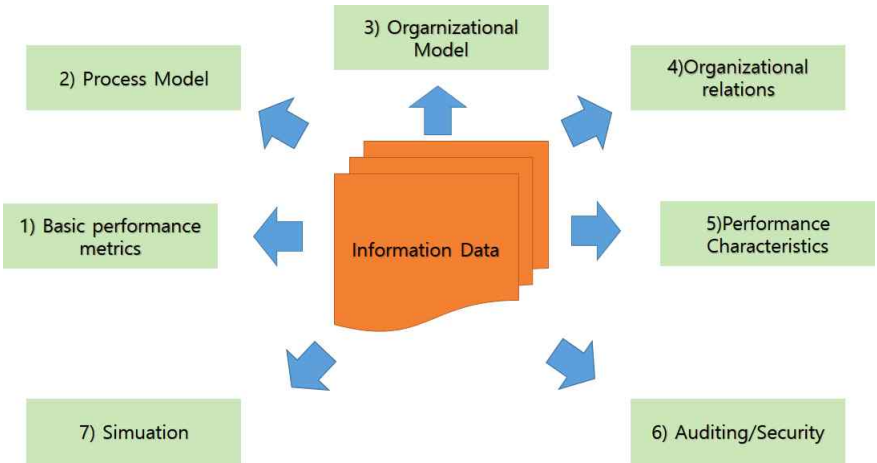
Mamoun Alazab & Venkataraman Sitalakshmi 등은 악성코드 분석 프로그램인 IDA pro를 이용하여, 디스어셈블링을 통한 분석을 진행 하였다. [표 2-1]와 같이 파일관련 행위를 총 6가지 특징을 분류하고, 분류된 악성코드들의 행위적인 특징들을 추출 하였다. 악성 파일로부터 추출한 공통적인 API 함수를 추출하여 악성 코드들만의 특정 문자열만을 가지고 악성코드의 특징을 가진 문자열과 비교 및 분류가 가능하다. 하지만 Windows API 함수 호출을 사용하는 Malware에만 적용되는데 한계가 있다[15].

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

B. 프로세스 마이닝

프로세스 마이닝(Process Mining)은 BPM, ERP, CRM, SCM 등에 저장되어 있는 과거 업무수행 기록(정보시스템 로그)과 다양한 기업의 업무 처리 시스템에서 기록되는 빅데이터의 이벤트 로그를 분석하여 “의미 있는 정보를 찾아내는 것”을 목적으로 하며, 프로세스 성과 측정, 프로세스 모델 도출, 조직 모델 도출, 시뮬레이션 분석 등 다양한 분야로 연구가 진행되고 있다[6, 7, 8]. 이벤트 로그에 포함된 추세, 패턴 및 세부정보를 식별하기 위에 특수 데이터 마이닝 알고리즘(Alpha, Genetic, Heuristic, Fuzzy)이 이벤트 로그 데이터셋에 적용된다. 프로세스 마이닝은 프로세스 효율성 및 프로세스 이해를 향상시키는 것을 목표로 하며 ABPD(Automated Business Process Discovery) 라고도 한다.



[그림 2-1] 프로세스 마이닝 연구 분야

프로세스 마이닝의 가장 핵심적인 기술은 이벤트 로그에서 프로세스 모델을 도출하는 것이며, 이를 통해서 가시화 되지 않은 프로세스를 찾는데 매우 유용하게 쓰일 수 있다. 예를 들어, 사용자가 특정한 상품을 구매하여 배송받기까지의 일련의 이벤트 순서의 값들을 각각 Activity로 분류하고 이벤트 로그를 추출하여 프로세스 마이닝 알고리즘을 적용하면 실제 사용자들이 어떻게 구매가 이루어지는지를 나타내는 프로세스 모델을 빠르고 정확하게 추출할 수 있다. 가장 대표적으로 사용

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

되는 프로세스인 병원 진료 프로세스의 경우, 환자의 특성 정보를 이용하여 환자가 받을 의료 서비스를 예측 할 수 있는 프로세스 모델을 만들 수 있다[9].

이러한 프로세스 모델 도출을 위해 다양한 마이닝 알고리즘이 개발되고 있는데 가장 처음 개발된 알파(Aplpha)마이닝, 노이즈 처리에 적합한 휴리스틱(Heuristic) 마이닝, 복잡한 프로세스 모델을 간단하게 볼 수 있는 퍼지(Fuzzy)마이닝 등이 주로 알려져 있다. 이들은 보통 순차(Sequence), 병렬(AND분기/병합), 선택(XOR), 순환(Loop)을 지원한다. [10, 11, 12] 가장 대표적으로 사용되는 알파 마이닝 기법의 경우, 프로세스 패턴 로그에 나타나는 일련의 순서대로 작업 수행이 되는 상황을 모델링 하여 모델을 도출 할 수 있다.

이와같이 프로세스 마이닝은 기존의 데이터 마이닝(데이터 위주)과는 달리 end-to-end를 포함한 프로세스에 중점을 둔다. 프로세스 마이닝의 시작은 event data 분석을 시작으로 [그림 2-2]와 같이 Case ID, Activity name을 가진다.

Case	ID	Operation	Type
CASE 1	정상파일	GlobalFree	정상
CASE 2	정상파일	VirtualAlloc	정상
CASE 3	정상파일	VirtualFree	정상
CASE 4	Ransomware	ExitProcess	랜섬웨어
CASE 5	Ransomware	GlobalFree	랜섬웨어
CASE 6	Virus	SelectObject	바이러스
CASE 7	Virus	GetModuleFileNameA	바이러스
CASE 8	Virus	CreatePen	바이러스
CASE 9	Virus	GetWindowsDirectoryW	바이러스
CASE 10	Virus	UnhandledExceptionFilter	바이러스
CASE 11	정상파일	RegOpenKeyExW	정상
CASE 12	정상파일	DragQueryFileA	정상
CASE 13	Virus	GetSystemDirectoryA	바이러스
CASE 14	Virus	WriteFile	바이러스
CASE 15	Virus	GetLastError	바이러스
CASE 16	정상파일	HeapFree	정상
CASE 17	정상파일	EnterCriticalSection	정상
CASE 18	Worm	LoadLibraryExW	바이러스
CASE 19	Worm	DeleteCriticalSection	바이러스
CASE 20	Ransomware	GetSystemDefaultLCID	랜섬웨어
CASE 21	Ransomware	GetStartupInfoW	랜섬웨어
CASE 22	Virus	QueryActCtxW	바이러스
CASE 23	Virus	ClipCursor	바이러스
CASE 24	Virus	DeleteFileA	바이러스
CASE 25	Virus	ExpandEnvironmentStringsW	바이러스

[그림 2-2] 프로세스 마이닝 요소

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

프로세스 마이닝은 많은 기업에서도 활용되고 있는데 예를 들어, 공기업 등의 업무 프로세스 분석, 국내외 대형 병원의 환자 건강검진 프로세스 관리를 위해 RFID를 비롯한 최신 IT기술을 바탕으로 스마트 건강검진 시스템을 구축하고 있다.[5] 또 세계 최대의 반도체 생산 장비 업체인 ASML은 생산 장비에 내재된 임베디드 시스템의 내부 프로세스 분석에 프로세스 마이닝 기술을 활용하였다. 필립스의 경우 메디컬 장비용 소프트웨어를 테스트 하기 위해 프로세스 마이닝 기법을 활용하여 제품의 품질 향상에 기여하고 있다[5]. 특히, 아인트호벤 공대의 Aalst교수를 중심으로 프로세스 마이닝의 개념을 만들었고, 오픈소스 프로세스 마이닝 도구인 Prom을 만들어 Plug-in 형태로 공유할 수 있도록 하고 있다. 이 외에도 'Fluxicon'이라는 프로세스마이닝 S/W회사가 만든 프로그램인 Disco프로그램은 Prom과 같은 프로세스 마이닝 툴로써 사용이 가장 편하기 때문에 많은 데이터 분석자가 사용하고 있다[13].

Collection @ chosun

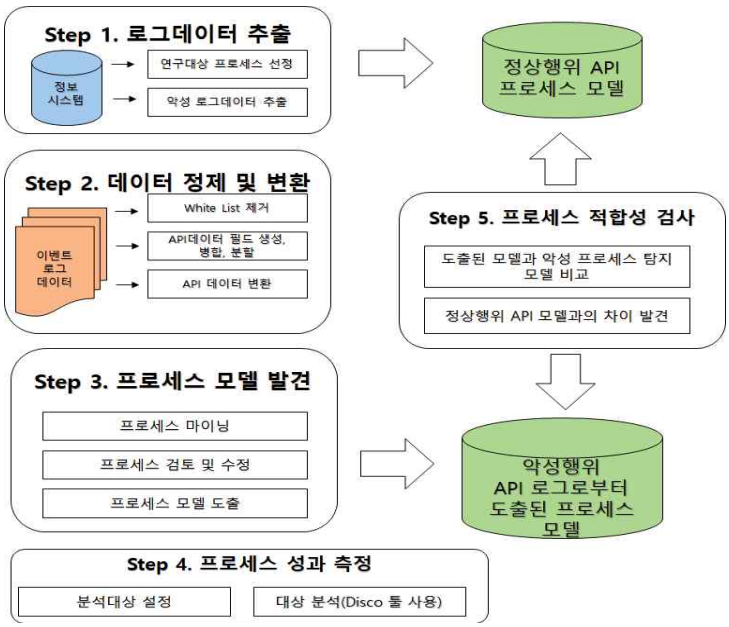
조선대학교
CHOSUN UNIVERSITY

Ⅲ. 프로세스 마이닝을 적용한 악성코드 탐지 방법

본 논문 에서는 악성코드 분석을 통한 유사성이 있는 악성행위 API와 동작 프로세스를 추출하여 프로세스 마이닝을 통해 프로세스를 모델링 한 후, 정상적 행위가 아닌 예외적인 악성 행위패턴에 대한 탐지를 하는 방법에 대해 제안한다.

A. 시스템 구성도

[그림 3-1]은 본 논문에서 제안하는 프로세스 마이닝을 이용한 변종 악성행위 탐지에 대한 전체 시스템 구성도이다.



[그림 3-1] 전체 시스템 구성도

본 논문에서 제안하는 프로세스 마이닝을 이용한 변종 악성행위 탐지는 크게 정적분석을 통한 악성행위 API 함수를 수집, White List제거를 통한 API 함수 필드

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

생성 및 병합을 통한 API함수 데이터 정제, 프로세스 마이닝 적용을 위한 모델링 과정으로 구성되어 있다. 제안하는 프레임워크는 악성코드 데이터 분석 및 프로세스 마이닝을 위한 전처리 과정을 제외하고 목적에 따라 총 네 단계로 구분한다. 단계 1 에서는 악성코드가 가장 많이 호출한 API 함수에 대한 데이터 수집으로 정상 Windows파일, Virus, Worm, Trojan, Ransomware에 대한 API 함수 정보를 추출한다. 단계 2 에서는 악성코드 데이터로부터 프로세스 마이닝에 적용하기 위한 이벤트 로그 추출 단계이다. 로그데이터를 추출 하기 위해서는 먼저 악성행위에 대한 동적분석과 정적 분석을 통해 얻어지는 데이터를 바탕으로 어떤 정보시스템에 어떤 형태로 기록 되는지 파악 해야 한다. 단계 3에서는 프로세스 마이닝 적용을 위한 악성코드들의 이벤트 로그들을 군집화 하는 단계이다. 군집화는 주어진 데이터의 집합을 유사한 성질을 가지는 그룹으로 나누는 방법을 뜻한다. 악성코드는 다양한 공격방법이 혼합되어 있는 구조를 가지고 있으며 악성코드의 API 함수 패턴에 따른 유사성 분류가 가능하다. 단계 4에서는 각 악성데이터 군집으로부터 프로세스 모델을 발견하는 단계이다. 본 연구에서는 데이터 분석 도구로 Disco를 사용하여 Excel 형태로 정제된 파일을 csv 포맷으로 변환하는 과정을 수행 하였다. 단계 5 에서는 예외 프로세스를 감지하는 단계이다. 단계 3 에서 적용한 악성코드 이벤트 로그들의 군집화를 통한 단계 4 에서의 프로세스 모델 과정에서 나타나는 정상적 행위가 아닌 다른 예외 패턴에 대한 정보를 얻을 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

B. API 함수 기반 속성 추출

변종 악성코드들의 속성 및 분류를 위한 탐지는 방법은 정적 분석과 동적 분석을 통해 추출된다. 각 속성에 대한 분류 결과는 [표 3-1] 와 같이 세부적으로 11가지의 속성을 가진다.

Category	API Feature
Process	CreateProcess
	ExitProcess
	TerminateProcess
	OpenProcess
	SearchProcess
	RegisterHotKey
	InformationProcess
File	CopyFile
	MoveFile
	DeleteFile
	OpenFile
	ReadFile
	WriteFile
	SearchFile
Registry	CreateRegistry
	DeleteRegistry
	OpenRegistry
	ReadRegistry
	WriteRegistry
Network	Connect
	Listen
	Send
	Recv

[표 3-1] API 함수 분류

본 실험에 앞서 악성코드 API 함수 분석을 위해 기존에 발견된 악성코드들의 공통적인 특징들을 수집 및 분석하여 악성코드가 많이 사용하는 특정 API 함수들을 추출하였다. 추출된 명령어를 보면 Process와 관련된 문자열은 프로세스를 생성해 조작할 수 있음을 알 수 있고, RegisterHotKey는 특정키의 조합으로 HotKey를 설정하여 특정 Process나 소프트웨어에 설정한 Hotkey를 활용할 수 있음을 알 수 있다. 파일생성과 관련된 API 함수들은 감염 이후 악성코드 파일을 시스템에 위치시

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

키는데, 주로 임시폴더에 최초 생성된 악성코드를 시스템폴더로 복사하거나, 다른 악성코드를 인터넷에서 다운받거나, 악성코드 자신이 가지고 있는 다른 악성코드를 만드는 행위를 한다. 이 외에도 현재의 실행이나 추후의 은밀한 실행을 위해 자신이나 다른 악성코드를 설정하는 악성코드의 유형의 대표적인 Resource Section에 exe나 dll을 포함하는 경우가 많다. 본 논문에서는 악성코드의 API 함수 데이터를 이용하여 논문에서 제안하는 프로세스 마이닝 적용과정을 살펴보고자 한다.

C. 악성 행위 API 함수 패턴

프로세스 모델링을 위한 정상과 악성 API 함수를 [표 3-2]와 같이 작성하고, API 함수로만 악성행위를 탐지하기에는 오탐의 가능성이 높기 때문에 본 논문에서는 악성코드가 가장 많이 호출한 API 함수를 PEID와 Exeinfo 툴을 사용하여 정상파일과 악성 파일에 대한 API 함수를 추출 하였다. 공통된 API 함수를 제거하고, 빈도수에 따라 그룹화 하여 총 1520개의 악성 API 함수와 3754개의 정상 API 함수 데이터를 본 실험의 데이터셋으로 사용 하였다.

행위 분류	API 함수	행위 분류	API 함수
Virus	GetFileType	Virus	GetFileType
Virus	ExitProcess	Virus	SetLastError
Worm	ReadConsoleA	Worm	CreateFileA
Worm	WriteFileEx	Worm	SetFileTime
Trojan	SetLastError	Trojan	TlsGetValue
Trojan	TlsSetValue	Trojan	DeleteFileW
Ransomware	FreeLibrary	Ransomware	ExitProcess
Ransomware	VirtualFree	Ransomware	GetProcAddress

[표 3-2] 악성행위별 추출된 API List 예

추출된 데이터를 보면 GetFileType과 ExitProcess와 같이 정상파일에서 사용하는 API 함수들도 있지만, 정상인 빈도 보다 악성이 빈도가 높은 API 함수들로 분류가 진행되었으며, [표 3-3]와 같이 API 함수들의 빈도수에 따라 정렬하였다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

API 함수
GetModuleFileName
ExitProcess
WriteFile
GetCommandLineA
GetStartipInfoA
CreateFileA
VirtualFree
VirtualAlloc
SetFilePointer
GetFileType
RegOpenKeyExA
SetEndOfFile
RaiseException
GetVersion
FindFirstFileA
GetFileSize
DeleteFileA
CopyFileA
CompareStringA
GetEnvironmentStrings
RegCreateKeyExA

[표 3-3] 악성 행위 빈도별 API 함수

D. 악성행위 탐지를 위한 프로세스마이닝 적용

앞 절에서는 프로세스 마이닝에 적용하기 위한 API 함수들을 분류하였다. 본 절에서는 전처리된 API 함수 데이터를 프로세스 마이닝에 적용과 탐지방법에 관해 서술한다.

프로세스를 도출하여 달성하고자 하는 목표는 프로세스 마이닝을 이용하여 정적 분석과 통한 악성코드를 각각 API 함수에 대한 행위적 특징들을 가시적으로 보여줌으로써 악성행위 탐지 프로세스에 대한 통찰력을 제공하는 것이며, 이를 바탕으로 변종 악성행위와 같은 예외탐지를 파악하는 것이다. 악성 행위에 대한 작업 프로세스는 크게 API 함수정보 및 파일 생성/제거, 레지스트리 생성/변경으로 구분되어 있다. 실험에 앞서 [표 3-4]과 같이 분류된 악성행위별 API 함수를 총 4개의 행

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

위로 구분하였다. 악성행위를 하는 API 함수를 기준으로 삼는 Case1 그룹(전처리된 악성API함수 빈도별 집합)과 파일관련, 프로세스관련, 레지스트리 관련 행위로 그룹화 하였으며, VirusTotal과 웹 사이트에서 수집한 Worm, Trojan, Ransomware, Virus들에 대한 세부적인 분류를 진행 하였다.

Case ID	Type	Activity
case 1	M	GetModuleFileName
case 1	M	ExitProcess
case 1	M	GetCommandLineA
case 1	M	VirtualFree
case 1	M	RegQueryValueExA
case 2	File W	Write File
case 2	File T	FindFirstFileEx
case 2	File W	DeleteFileA
case 2	File V	GetModuleFileNameW
case 2	File R	SetFilePointer
case 3	Process V	ExitProcess
case 3	Process R	GetProcAddress
case 3	Process V	VirtualFree
case 3	Process W	WriteProcessMemory
case 4	Reg W	RegOpenKeyExA
case 4	Reg V	RegQueryValueExA
case 4	Reg R	RegUnLockKeyA
case 4	Reg T	RegQueryValueExA

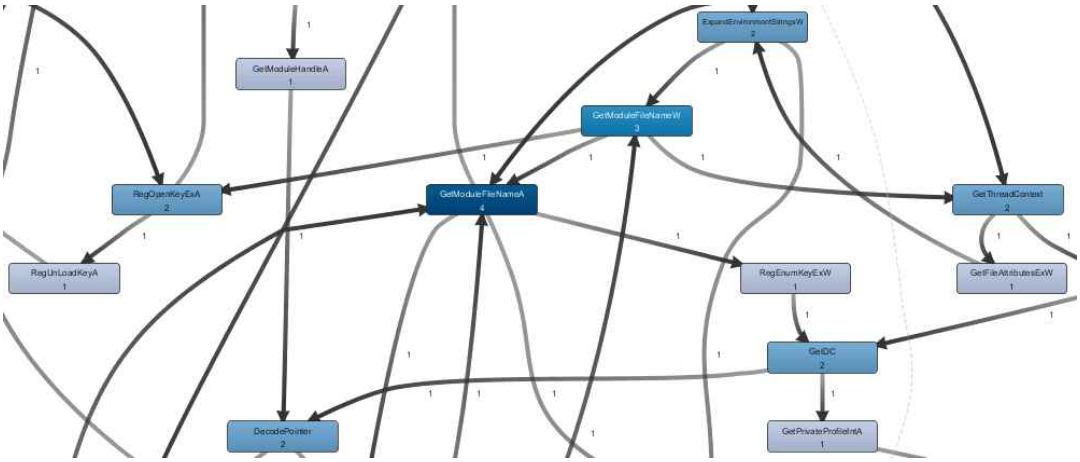
[표 3-4] 대표적인 악성행위별 API 함수

각 악성행위별 이벤트가 기록되어 있는 csv 파일을 불러오면 Case ID, Activity, Resource등을 [그림 3-2] 에 보여지는 각각의 Instance에 따른 항목들은 4가지로 구분 지을 수 있다. Case ID는 단어그대로 각 사건들을 구별 할 수 있는 ID를 지정하는 것이고, Activity는 각 사건들이 어떤 작업을 진행한 사건인지 구분하는 것이다. Resource는 각 작업을 수행하는 대상을 구분 짓는 것이다. 이러한 기본적인 설정을 마친 후에 프로세스 맵을 보여주는데, [그림 3-3]과 같이 프로세스의 빈도를 기준으로 상대적으로 높은 작업일수록 음영이 진하게 보여진다.

또한, 각각의 악성코드 행위에 따라 분류하고 프로세스 마이닝 적용을 위한 데이터셋을 작성 하였다. 수집된 데이터에 대한 악성 행위패턴에 따른 유사도는 Disco

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

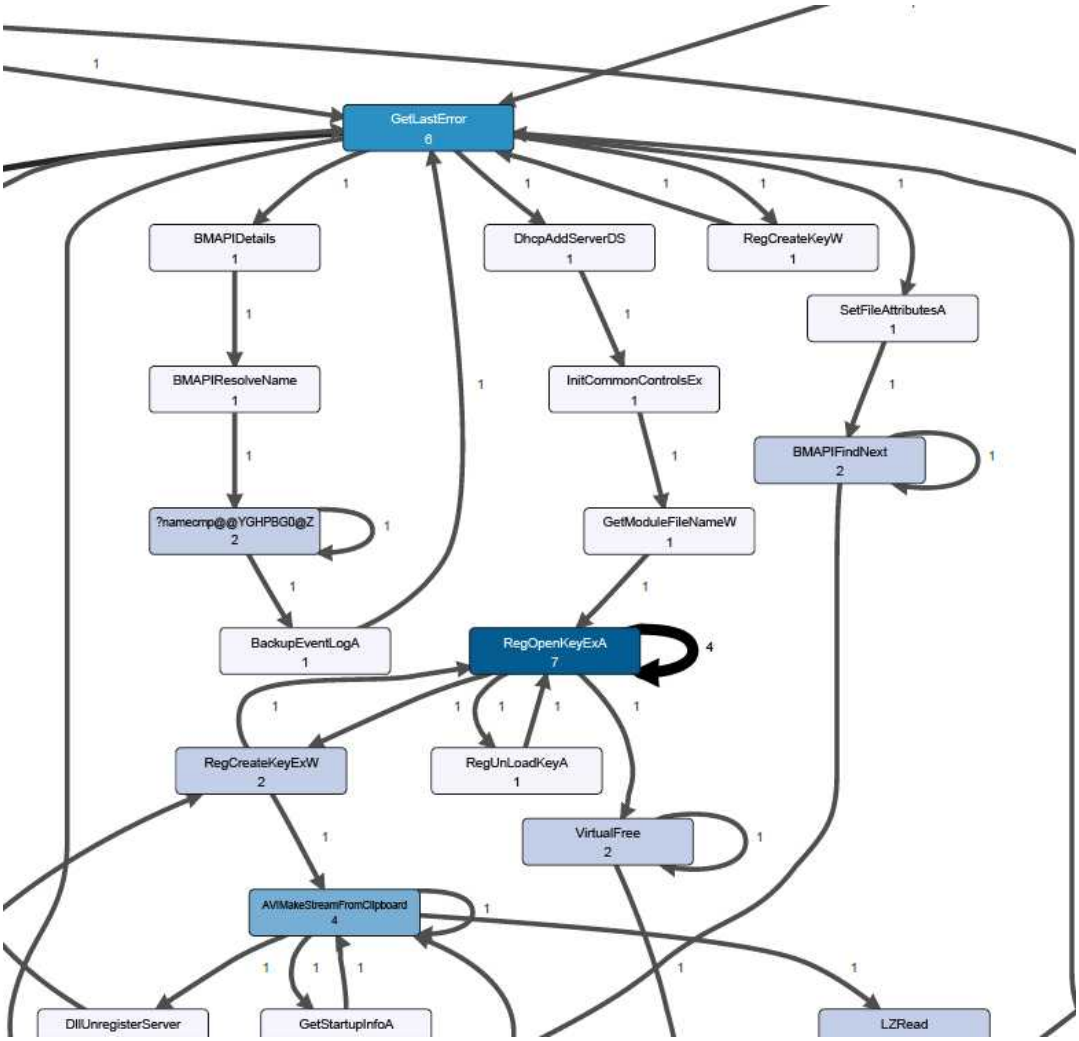


[그림 3-2] 프로세스 마이닝을 적용한 API 함수 패턴

툴을 이용하여 발견된 실제 프로세스 모델을 [그림 3-2] 과 같이 확인 할 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY



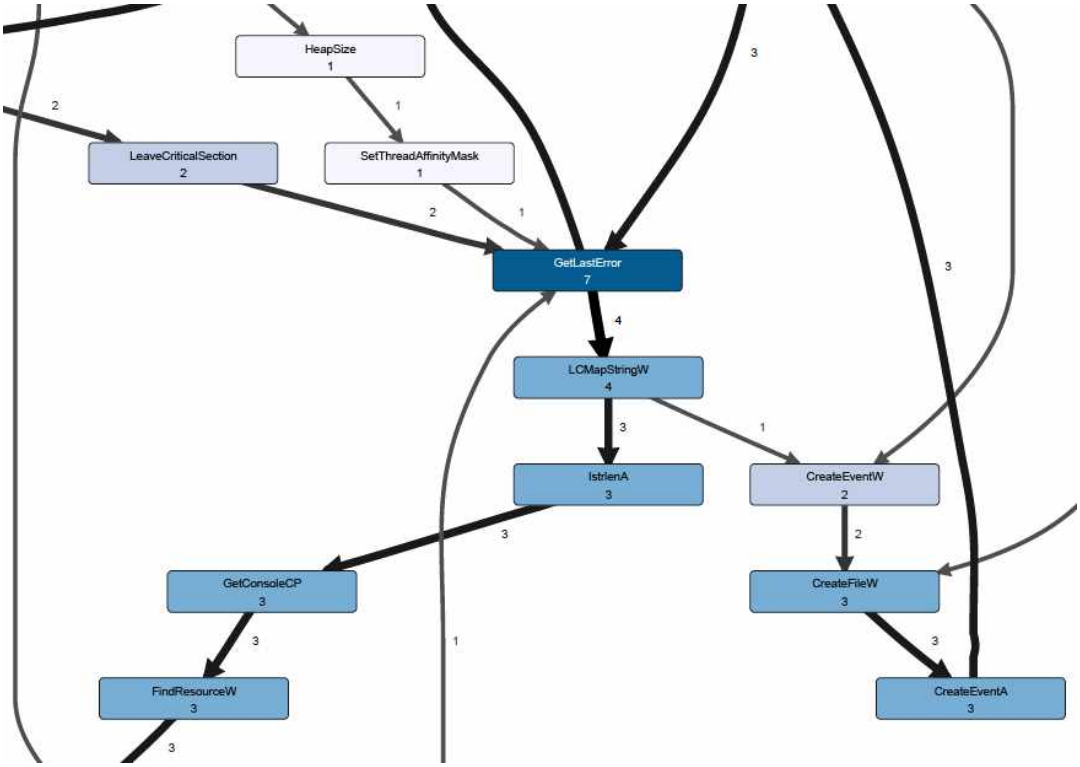
[그림 3-3] Ransomware API 함수 패턴

다른 종류의 악성행위 패턴의 빈도 수에 따라 작업 프로세스가 상대적으로 진한 모습을 나타내고 있다. 직사각형은 Activity를 나타내고 숫자는 이벤트 로그의 빈도수를 나타낸다. Activity는 작업의 선후 관계를 나타내며 활동의 색깔이 진하고 두꺼울수록 더욱 빈번하게 수행되었음을 알 수 있고, 많은 종류의 악성코드들에도 유사한 패턴이 존재함을 확인할 수 있다. Ransomware의 경우 수집된 500개의 데이터가 행위에 따라 공통적으로 API 함수와 관련된 행위 프로세스를 진행하는 것

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

을 알 수 있다. 예를들어, RegOpenKeyExA와 DebugSetMute에 대한 API 함수는 각각 7번의 작업을 수행하였으며, 공통적인 작업 프로세스에 따른 유사행위를 확인 할 수 있다.



[그림 3-4] Trojan API 함수 패턴

특히, Trojan의 경우 Ransomware나 Worm, Virus에 비해 보다 많은 프로세스 행위를 확인 할 수 있다. 이는 다른 종류의 Trojan 악성코드라도 유사한 API 함수 행위를 가지고 진행되는 프로세스를 가지고 있으며, 변종 Trojan악성 행위에 대한 패턴에 대한 프로세스를 적용 할 수 있다. 작업의 빈도가 가장 높게 보여진 VirtualFree 프로세스 패턴을 보면, [표 3-4]와 같이 악성행위 빈도수가 높게 산출되었던 API 함수와 악성코드 샘플들의 작업이 모두 진행되었음을 알 수 있으며, VirtualAlloc(웜에서의 악성프로세스 행위), SetFilePointer(웜에서의 파일기반 악성 행위), GetFileType(파일기반 악성 행위) 프로세스 작업으로 진행됨을 확인 할 수

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

있다. 이를 통해, 악성행위 빈도가 높은 프로세스를 기준으로 불특정한 파일에 대한 프로세스 작업이 동시에 일어날 경우 기존의 유사한 악성행위 패턴임을 알 수 있고, 쉽게 정상인지 악성인지 행위 예측이 가능하다. 이와같은 악성행위 패턴을 추출하기 위해 먼저 정상파일과 악성파일에 대한 API 함수 행위 패턴을 생성한다. 그리고, FP-Growth(frequent iterative patterns algorithm) 빈발패턴 알고리즘을 사용하여 악성 행위별 빈발 순차패턴을 [표 3-5]와 같이 생성하였다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

패턴 종류	생성된 패턴	빈발 개수	추출 패턴개수
패턴 T-1	GetCommandLineA SetFilePointer GetFileType	121	43
패턴 T-2	GetVersionExA GetFileSize RegCreateKeyExA ExitProcess	98	37
패턴 T-3	GetCurrentDirectoryA GetLastError FreeLibrary GetTickCount	57	25
패턴 W-1	RegOpenKeyExA RegCreateKeyExA Write File DeleteFileA	115	58
패턴 W-2	GetModuleFileNameA GetModuleHandleA RegCloseKey CloseHandle ReadFile	43	31
패턴 W-3	GetVersion GetSystemInfo GetDiskFreeSpaceW GetLastError	31	25
패턴 V-1	GetModuleFileNameA CloseHandle WriteFile	86	64
패턴 V-2	GetModuleFileNameA GetLastError GetCurrentProcess CloseHandle	77	51
패턴 V-3	ByHandleExRegCreateKeyExA GetStdHandle	25	15
패턴 R-1	ByHandleEx DeleteFileW CopyFileA	31	47
패턴 R-2	VirtualFree RegUnLoadKeyA GetEnvironmentStrings	22	13
패턴 R-3	ClearEventLogA GetSaveFileNameA ExitProcess	18	9
패턴 X-N	⋮	⋮	⋮

[표 3-5] 악성 행위별 빈발패턴 추출

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

패턴의 종류는 Trojan, Worm, Virus, Ransomware로 구분짓고. 패턴 순서에 따라 숫자를 부여하였다. 예를들어, 패턴 T-2의 경우 Trojan의 악성코드가 GetversionExA라는 행위를 시도할 경우 나타나는 빈발패턴에 대한 정보를 가지고 있다. 패턴의 개수는 API 함수 패턴의 반복 정도와 관련이 있으며, 최소지지도 값이 커질수록 빈발 패턴의 개수가 감소한다. 최소지지도를 높거나 작게 설정하면 행위 패턴의 수에 대한 기준값이 없어 행위패턴에 따른 분석이 어렵게 된다. [표 3-6] 는 Trojan.Win32.Tunneler에 대한 최소지지도 변화에 따른 패턴 수와 대표적인 형태이다.

	최소지지도	패턴 개수	API 함수 행위패턴
Trojan.Win32. Tunneler	0.1	2583	GetVersion CreateWindowExA GetWindowThreadProcessId RegCreateKeyExA CreateWindowExA GetWindowThreadProcessId GetModuleFileNameA CreateWindowExA GetWindowThreadProcessId LoadLibraryA RegOpenKeyExA LocalFree RegCreateKeyExA
	0.2	1658	
	0.3	681	
	0.4	237	
	0.5	173	

[표 3-6] Trojan 최소지지도 변화에 따른 패턴 수

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Ⅳ. 실험 및 평가

본 장에서는 제안하는 악성 행위 패턴 마이닝의 성능을 측정하기 위해 실험환경, 실험 데이터, 실험결과에 대해 기술한다. 수집한 악성코드들을 이용하여 행위 시퀀스 파일을 만들었으며 API 함수를 기반으로 악성코드 행위 패턴에 따른 실험을 진행 하였다.

A. 실험 환경 및 데이터 셋

본 논문의 실험을 위해 윈도우10 운영체제 기반으로 테스트 환경을 구축하고 정상파일, Trojan, Worm, Ransomware, Virus에 대한 API 함수 행위정보를 기반으로 행위패턴 데이터를 수집한다. Vmware을 사용한 가상환경에서의 실험에 사용한 데이터는 [표 4-1]와 같다.

No.	Malware Type	Counts
1	Trojan	500
2	Worm	300
3	Ransomeware	500
4	Virus	300
5	Windows	400
Total		2000

[표 4-1] 악성행위에 따른 데이터 셋

실험을 위한 악성코드 데이터는 VxHaven과 Virus Total로부터 악성코드 샘플을 수집하고, 악성코드 샘플로부터 추출된 행동들을 활용하여 분류하였다[16,17]. 수집한 악성코드는 Exeinfo PE프로그램을 사용하여 packing여부를 확인해 본 결과, UPX로 packing이 된 악성코드로 UPX프로그램을 사용하여 Unpacking 후 BinText를 사용해 Unpacking한 악성코드 API 함수를 추출 하였다. 추출된 데이터를 바탕으로 프로세스마이닝의 대표적인 툴인 Disco를 이용하여 API 함수 패턴을

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

적용 하였다.

B. 성능평가

[표 4-3]의 패턴 결과에서 대체적으로 최소 지지도가 0.1 일 경우 많은 패턴의 개수가 생성되며, 본 실험에서는 프로세스의 복잡도를 최소화 하기 위한 최적치 값을 찾기 위해 최소 지지도에 따른 악성코드 분류 실험을 진행 하였다.

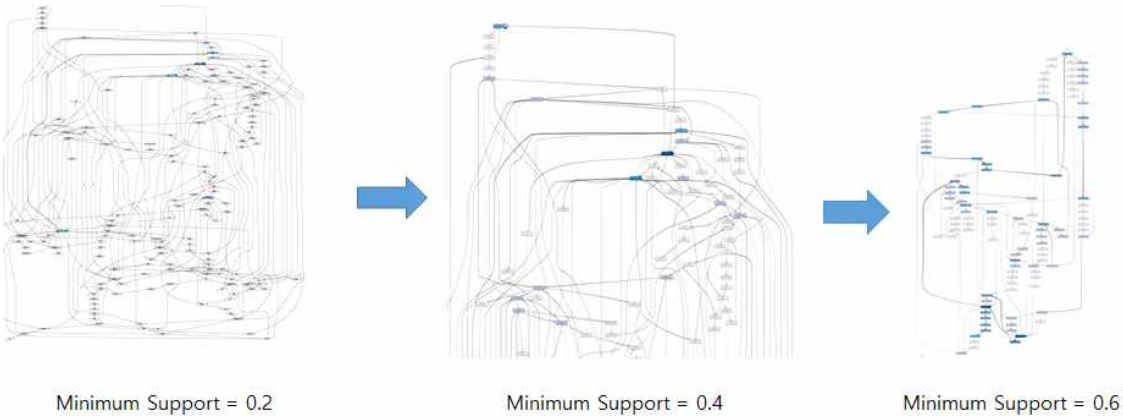
최소지지도	코드 개수	정탐률	오탐률
0.1	2583	74.4%	25.6%
0.2	1658	77.8%	22.2%
0.3	681	84.3%	15.7%
0.4	237	90.2%	9.8%
0.5	173	82.7%	12.3%

[표 4-2] 최소지지도별 정탐률, 오탐률

검증 결과 [표 4-2]와 같이 최소지지도에 따른 정탐률과 오탐률을 확인할 수 있다. 본 논문에서는 프로세스 복잡도의 최소화를 위해 탐지율이 가장 높은 0.4를 최소지지도를 최적치로 두고 처리성능을 평가한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY



[그림 4-1] Trojan.win32.Tunneler 함수 패턴

본 논문에서 제안하는 프로세스 마이닝을 이용한 변종 악성코드 탐지 방법을 평가하기위하여 악성코드 종류에 따라 총 1000개의 악성코드를 API함수를 이용하여 4개(Trojan, Worm, Virus, Ransomware)로 분류 하였다.

Cluster	탐지율	전체빈도	정확도
Trojan	173	237	0.729
Worm	130	180	0.722
Virus	185	265	0.698
Ransome	148	318	0.465
Total	637	1000	0.637

[표 4-3] 악성 행위별 프로세스 예측 정확도

[표 4-3]와 같이 악성 행위 프로세스에 따른 전체 변종 악성행위 프로세스 예측모형의 정확도는 약 63%로 측정되었으며, Trojan과 Worm의 경우 전체빈도에 따른 예측빈도가 약 72%로 나타났으며, 이를 통해 API함수 패턴과 유사한 변종 악성행위를 적절히 구분할 수 있다. 반면, 상대적으로 정확도가 낮게 측정된 Ransomware의 경우 예측의 복잡도와 상대적으로 유사성이 낮은 프로세스 패턴으로 예측모형의 정확도가 낮아 분류에 어려움이 따른 것으로 보여진다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

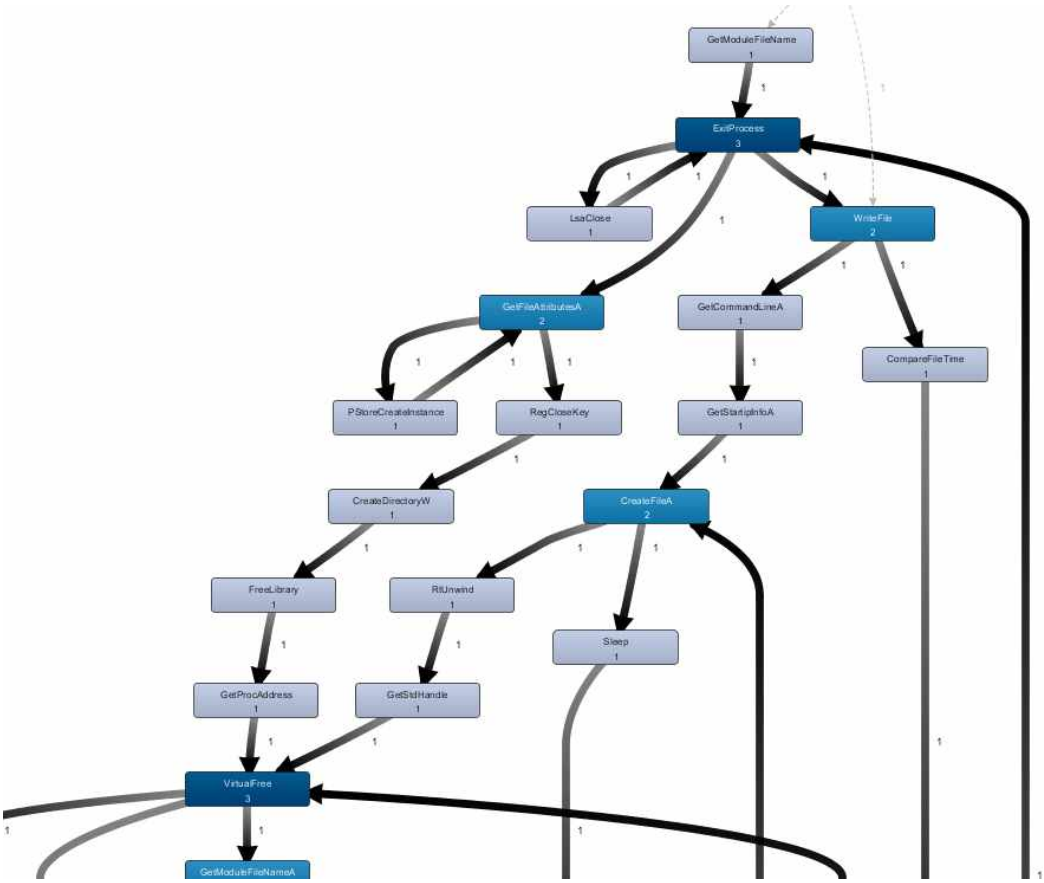
분류	결과	
	제안방법(%)	기존 유사도 분석(%)
Trojan	74%	75%
Worm	84%	81%
Virus	71%	75%
Ransomware	46%	-

[표 4-4] 악성코드 분류 비교 실험 평가

비교평가 결과 패턴의 유사도 비교가 기존의 유사도 분석과 비슷한 결과를 보였다. 하지만 본 논문에서 제안했던 연구방법은 기존의 연구보다 높은 수치의 탐지율로써 성능평가를 하는 것이 아닌 기존의 연구방법으로는 탐지하지 못했던 악성코드를 본 논문에서 제안하는 방법으로 추론이 가능하다는 새로운 연구방향성을 확인할 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY



[그림 4-2] Trojan.Win32.BHO.hdz API 함수 패턴

[그림 4-2]는 기존의 연구방법인 유사도 비교방법으로 오탐지 되었던 Trojan.Win32.BHO.hdz이 본 논문에서 제안하는 방법으로 악성행위를 유추할 수 있다. 프로세스 맵에 보여지는 공통적인 API 함수 특징을 보면 ExitProcess, GetModuleFileName과 같은 악성파일과 정상파일에 공통적으로 나타나는 API 함수에 대한 패턴이 빈번하게 나타나고 있음을 알 수 있다. 이러한 API 함수들은 [표 3-3]과 같이 정상인 빈도보다 악성인 빈도가 높은 API 함수 특징으로 해당 악성코드가 비정상적인 악성 API 함수행위 패턴을 가지고 있음을 알 수 있다. 기존 유사도 분석 방법은 변종 악성코드 추론의 한계성을 보이지만 본 논문에서 제안한 API 함수 패턴을 프로세스 마이닝을 이용한 방법은 비정상 행위 패턴 추출 및 유사한 변종 악성코드에 대한 추론이 가능하다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

또한, API 함수 이외에 특정한 이벤트를 가지는 악성코드 정보를 이용하여 그룹화 및 프로세스를 적용할 경우 보다 상세한 악성행위에 대한 모델링을 할 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

V. 결론 및 제언

본 논문에서는 기존에 알려진 악성코드와 변종 악성코드들이 가지는 행위기반 API 함수들의 특징에 따라 변종에 대한 분류 방법을 프로세스 마이닝을 통해 제안하였다. 기존의 연구에서는 악성코드를 탐지하고 분류하기 위해 유사도 측정을 하고, 정적 및 동적 분석을 통한 결과로 악성코드들의 공통적인 속성을 추출하거나 학습을 기반으로 한 연구가 진행 되었다. 따라서, 본 논문은 기존 악성코드들이 가지고 있는 공통적인 행위패턴에 대한 API 함수를 바탕으로 악성코드들의 행위적인 특성을 기반으로 한 프로세스 맵을 구축하고, 악성코드가 동작하는 동안 발생하는 비정상 행위를 탐지하기 위해 정상 패턴과 비정상 행위 패턴을 추출하고, 이를 이용하여 비정상 행위 패턴을 마이닝 하여 알려지지 않은 비정상 행위를 탐지할 수 있는 방법을 제안 하였다. 본 논문의 향후 연구에서는 특정한 이벤트를 가지는 악성코드 정보를 이용하여 데이터를 만들고 그룹을 세분화 한다면 보다 상세한 악성 행위에 대한 프로세스 모델링을 할 수 있을 것이다. 또한 DDoS(분산서비스공격)과 같은 심각한 보안위협에 있어 공격 패턴에 대한 이벤트로그를 수집하고 프로세스 맵을 모델링 한다면, 그 동안 일어났던 악성 공격에 대한 유사성을 프로세스 작업의 빈번함을 통해 확인 할 수 있을 것이고 추가적인 공격에도 탐지가 가능 할 것이다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

참고문헌

[1] Jixin Zhang, Zheng Qin and Hui Yin, "IRMD: Malware Variant Detection Using Opcode Image Recognition", IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS), pp. 1175 - 1180, 2016.

[2] KISA 한국인터넷진흥원 탐지동향 보고서 https://www.boho.or.kr/data/reportList.do

[3] 장성수 "API 순차성을 반영하는 유사도 분석을 이용한 변종 악성코드 탐지 기법에 관한 연구", 성균관대학교 전자전기컴퓨터공학과, pp. 41-42, 2012.

[4] D. D. Lille, B. Coppens, D. Raman and B. D. Sutter, "Automatically combining static malware detection techniques," 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), pp. 48-55, 2015.

[5] 한국방송통신전파진흥원, "빅데이터 분석을 위한 프로세스 마이닝 기술 동향 방송통신기술 이슈&전망", Korea Communications Agency, 제 62호, 2014.

[6] 권혁진 "프로세스 마이닝 기법을 이용한 스마트 건강검진 시스템 도입효과 분석 연구", 숭실대학교 산업정보시스템공학과, 2015.

[7] 이동하 "프로세스 마이닝을 이용한 조선 산업의 생산 공정 분석", 부산대학교 산업공학과, 2014.

[8] 김현경, 신광섭 "Process Mining 기법을 이용한 물류센터 입출고 프로세스 분석 및 개선", 한국경영과학회지 39권 4호,, pp 1-17, 2014.

[9] 이영우, 이태식. "대형 병원 외래 시스템의 환자 흐름 개선방안의 적용 전략에 관한 연구" 대한산업공학회 추계학술대회 논문집, pp 1006-1016, 2009.

[10] A. K. A. de Medeiros W. M. P. van der Aalst A. J. M. M. Weijters, "Workflow Mining: Current Status and Future Directions", pp 389-406, 2003.

[11] W. van der Aalst, T. Weijters and L. Maruster, "Workflow mining: discovering process models from event logs," in IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pp. 1128-1142, 2004.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[12] Wen,L.,van derAalst,W.M.,Wang,J.,& Sun,J.
"Mining processmodels with non-free-choice constructs DataMining andKnowledgeDiscovery",15(2), pp 145-180, 2003

[13] 매니페스토 UNIST(울산과학기술대) http://www.win.tue.nl/ieeetfpm/

[14] Engin Kirda and Christopher Kruege, Greg Banks, Giovanni Vigna, and Richard A. Kemmerer, "Behavior-based Spyware Detection". Usenix Security, 2006.

[15] Mamoun Alazab, Venkataraman Sitalakshmi, and Paul Watters. "Towards Understanding Malware Behaviour by the Extraction of API calls." Cybercrime and Trustworthy Computing Workshop (CTC), Second. IEEE, 2010.

[16] vxheaven, http://vxheaven.org/(accessed., Oct, 11, 12, 2017)

[17] virustotal, https://www.virustotal.com/(accessed. Oct, 10, 5, 2017)

Collection @ chosun