행위 그래프를 이용한 악성코드 유사도 판별법

김지훈*, 손강원*, 조두산**, 윤종희*
*영남대학교 컴퓨터공학과,**순천대학교 전자공학과
e-mail :f13521@naver.com, kwkw91@yu.ac.kr, dscho@sunchon.ac.kr, youn@yu.ac.kr

A Method for Malware Similarity Analysis based on Behavior Pattern Graph

Ji-Hun Kim*, Kang-Won Son*, Doosan Cho**, JongHee Youn*
*Dept of Computer Engineering, YeungNam University
**Department of Electronics Engineering, Sunchon National University

요 약

Malicious(악의적인) + Code 즉, 악의적인코드를 포함한 소프트웨어라는 의미로 줄여 Malware(Malicious + Software) 라고 불리는 악성코드는 최근 네트워크와 컴퓨터의 급속한 발전에 따라 기하급수적으로 증가하고 있는 추세이다. 폭발적인 증가율 추세를 보이고 있는 악성코드의 위협을 대비하기 위해 악성코드에 대한 분석이 필요한데 그 분석의 종류로는 초기분석, 동적 분석, 정적분석으로 나누고 장, 단점을 정리하였다. 또한 악성코드 대량화에 따른 효율적인 분석과 빠른 의사결정을 위한 악성코드 유사도에 대한 연구를 소개하고 API Call Sequence와 분류된 API를 이용한 악성행위 유사도 판별법을 제시하고 실험하였다.

1. 서론

1949년 폰 노이만은 생물학적 바이러스와 같이 자기복제와 증식을 컴퓨터에서도 할 수 있다는 이론을 발표하였다. 이 후 자기복제와 증식을 갖추고 있는 컴퓨터 프로그램을 '바이러스'라고 부르게 되었다. 하지만 바이러스라는 용어만으로는 표현에 제한되는 부분이 많아 그 개념을 포괄적으로 표현할 수 는 개념이 필요하게 되었다. 그로 인해 만들어진 용어가 Malicious(악의적인) + Code 즉, 악의적인코드를 포함한 소프트웨어라는 의미로 줄여 Malware[1](Malicious + Software) 라는 용어가 탄생하였다. 최근 네트워크와 컴퓨터의 급속한 발전에 따라 악성코드의 발생량 또한 급속도로 급증하고 있다. 뿐만 아니라 기존의 악성코드를 변조한 변종 악성코드 또한 기하급수적으로 증가하고 있는 추세이다.

폭발적인 증가율 추세를 보이고 있는 악성코드의 위협을 대비하기 위해 여러 가지 악성코드 분석 방법이 연구되고 있다. 덧붙여 대량분석을 조치로 자동화가 필요하게 되었으며 그 자동화 분석과 악성코드에 대한 유사점이 찾고자 하는 악성코드 유사도에 대한 연구도 활발히 진행되고 있다.

본 논문의 2장에서는 악성코드 분석의 종류를 소개하고 그 장단점을 알아보며, 3장에서는 기존에 진행되었던 악성코드 유사도 분석에 대한 연구를 소개하며, 4장에서는 분류된 API[2] List와 API Call Sequence를 이용하여 그래프를 그리고 그것을 토대로 하는 유사도 판별에 대한 방법을 제시하고 5장의 결론으로 마무리 짓는다.

2. 악성코드 분석의 종류

악성코드의 위협에 대비하고 악성행위를 미연에 방지하기 위해서는 악성코드에 대한 분석이 필요하다. 악성코드의 분석은 크게 세 가지로 분류할 수 있는데, 초기 악성코드의 외형만으로 분석하는 초기분석, 악성코드 파일을 디스어셈블하여 세부적인 동작을 분석하는 정적분석, 악성코드를 직접 실행시켜 분석하는 동적 분석이 그것이다.

먼저 악성코드를 실행시키지 않고, 그 외형만을 보고 분석하는 초기분석단계에서는 수집된 악성코드를 안티-바이러스 엔진을 통해 진달을 하거나 패킹 여부 확인, 관련 함수 정보를 추출하고 프로그램 내의 String 정보를 수집하는 등의 방법이 있다. 대부분의 작업이 관련 Tool을 이용하여 분석되기 때문에 간단하고 어렵지 않으나 악성코드의 동작방식을 유추 할 수 있을 뿐 정확한 동작방식에 대해서 해석하기 어려운 단점이 있다.

두 번째 정적분석이란 악성코드를 실행시키지 않고 분석 도구들을 이용하여 직접 분석하는 방법이다. 주로 악성코드 파일을 디스어셈블하여 세부적인 동작을 분석하는데, 이렇게 프로그램을 역분석하는 기술을 리버스 엔지니어링[3] 이라고 하고 일반적으로 알려진 OllyDbg[4], IDA PRO와 같은 Tool을 이용한다. 정적분석은 세부적인 동작을 분석하는 용이하고 직접 실행시키지 않기 때문에 작업환경에 대한 안전하다는 장점을 가지고 있지만 최근 악성코드들은 패킹이라고 불리는 실행압축[5], 코드 난독화[6]를 통해서 이 정적분석의 어렵게 하고 있다.

마지막 동적 분석이란 악성코드를 실제로 동작시켜 그

행위를 분석하는 방법이다. 악성코드를 실행시킨 직후부터 프로세스, 파일시스템의, 네트워크 등의 변화를 실시간으로 확인하는 방법이다. 동적 분석은 실제로 실행시키기 때문에 코드 난독화와 실행압축을 피해 갈 수 있지만 실행환경에 대한 감염 우려가 있다. 그렇기 때문에 일반적으로 가상머신을 이용한 동적분석을 사용한다.

3. 악성코드 유사도에 대한 관련 연구

악성코드에 대한 위협을 방지하고 대응하기 위해 악성코드 분석에 대한 다양한 방법이 연구되고 있는데 대표적으로 유사도에 대한 연구가 있다. 폭발적으로 규모가 커지고 있는 악성코드에 분석에 대한 대량분석, 자동화, 신속한 의사결정을 위한 1차 분류가 대표적인 유사도 연구의 목적이라 할 수 있다.

국내외 관련연구로 Opcode 통계와 String 유사 알고리즘을 이용한 연구[7], 파일 DNA 기반의 행위 패턴을 분석을 통하여 유사도 비교에 대한 연구[8], 변종 악성코드의 공통 속성을 이용한 탐지에 대한 방법[9], N-Gram을 이용하여 악성코드 유사도 분석에 대한 연구[10] 등이 수행되어 왔다.

4. 행위 그래프를 이용한 악성행위 유사도 판별

4장은 3장에서 소개된 악성코드 유사도에 대한 연구와 더불어 API Call Sequence와 분류된 API를 이용한 악성행위 유사도 판별법을 제시하고자 한다.

4.1 API 수집

악성코드가 주로 사용하는 API를 수집하기 위해서 먼저 x86 PE 기반의 악성코드 샘플 200여개를 표본으로 하여 정적분석을 진행하였다. IDA Pro를 이용하여 단순히 사용된 API를 수집하였으며 약 100000개의 API가 수집되었다.

4.2 API 분류

4.1에서 수집된 100000개의 API를 사용하기에는 그 양이 방대하다. 그렇기 때문에 표본으로 정한 샘플 200여개의 악성코드에서 공통으로 나타나는 API를 분류하였다. 약 1000여개의 API로 축소되었다. MSDN[11]을 참고하여 약 1000개의 API를 종류별로 다시 상위 52개의 카테고리로 분류하였다. 표1은 분류된 API의 상위 52개의 카테고리를 표식화 한 것이다.

Process	Timer	console	Disk	Clipboard	Power_management
Thread	Memory	Keyboard_input	Bitmap	Diagram_box	heap
System	Hadle_and_object	Registry	window	System_shutdown	Pipe
mutex	Error_handler	Variant_manipulation	device	Socket	Device_install
service	Message	resource	hook	Directory_management	Network_management
event	File	volume	cursor	Setup	Display_monitors
Dynamic_link_library	Debugging	Event_logging	Video_capture	Remote_desktop_servicde	Window_station_and_desktop
Security_management	Print_job	cryptography	WinInNet	Remote_access_service	Window_networking
metafile	Mouse_input	shell	time	etc	

표 1 API 52개 상위 카테고리

4.3 행위그래프와 유사그래프

4.2장의 52개의 카테고리는 행위그래프의 노드로 사용된다. 그래프의 간선은 디어셈블하여 API의 Control Flow에 따라서 나타냈으며 20개의 worm[12] 악성코드로 테스트하면 그림 1과 같은 그래프가 그려진다.

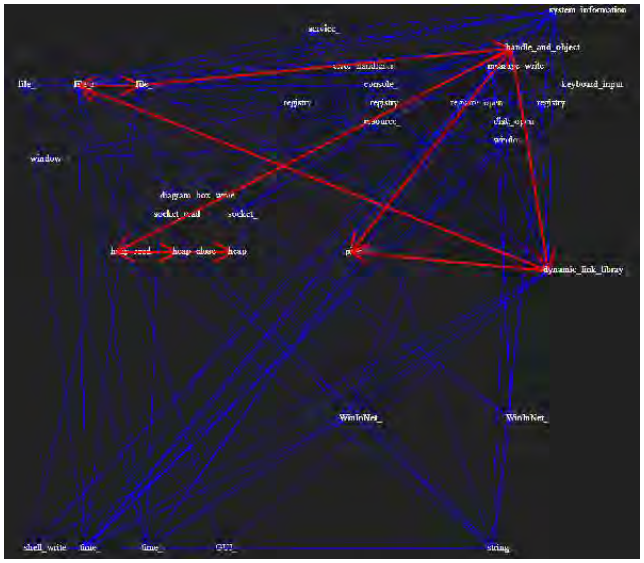

그림 1 worm 행위그래프와 유사그래프

뿐만 아니라 유사도 그래프를 관찰하기 위해 테스트용 worm 악성코드 이외의 worm 악성코드 샘플을 다시 테스트하면 굵은 간선으로 worm에 대한 유사 하위 그래프가 출력 되는 것을 볼 수 있다.

5. 결론

본 논문에서는 악성코드 위협을 방지하기 위한 여러 가지 분석방법을 살펴보았고 폭발적으로 증가하고 있는 악성코드의 대량화와 다양성을 효율적으로 분석하기 위한 악성코드 유사도 방법을 살펴보고 API Call Sequence와 분류된 API LIST를 이용한 유사도 판별법을 제시하였다. 비록 유사한 하위 그래프가 나타났다 하더라도 그 유사도가 정확히 수치화 되지 않아 판별의 신뢰도가 떨어진다고 할 수 있다. 향후, 행위 그래프와 하위 그래프에 대한 유사도가 수치화 한다면 더 판별력을 높일 수 있을 것이다.

참고문헌

[1] Wikipedia, Malware, http://en.wikipedia.org/wiki/Malware/
[2] Wikipedia, API http://en.wikipedia.org/wiki/Application_programming_interface
[3] WIkipedia, Reverse engineering , http://en.wikipedia.org/wiki/Reverse_engineering
[4] Wikipedia, OllyDbg, http://en.wikipedia.org/wiki/OllyDbg
[5] Wikipedia, 실행압축 http://en.wikipedia.org/wiki/Executable_compression
[6] Wikipedia, 코드난독화 http://en.wikipedia.org/wiki/Obfuscation_(software)
[7] KyoungSoo Han, Jae Hyun Lim, Eul Gyu Im, "Malware analysis method using visualization of binary files" Proceedings of the 2013 Research in Adaptive and Convergent Systems Pages 317-321, 2013
[8] Eun-Gyeom Jang, Sang Jun Lee, Joong In Le, "A Study on Similarity Comparison for File DNA-Based Metamorphic Malware Detection", Journal of The Korea Society of Computer and Information,2014
[9] Seong-Bin Park, Min-Soo Kim, and Bong-Nam Noh, "Detection Method Using Common Features of Malware Variants Generated by Automated Tool", 2013
[10] Heejun Kwon, Sunwoo Kim, Eul Gyu Im, "An Malware Classification System using Multi N-gram", Journal of Security Engineerin, 2012
[11] MSDN, https://msdn.microsoft.com/en-us/
[12] Wipipedia, Worm, http://en.wikipedia.org/wiki/Worm