행위 기반 악성코드 탐지 기술에 관한 동향 연구

김호연*, 최영현*, 정태명**
*성균관대학교 전자전기컴퓨터공학과
**성균관대학교 정보통신공학부
e-mail:*{hykim, yhchoi}@imtl.sklu.ac.kr, **tmchung@ece.skku.edu

A Survey on Behavioral Based Malware Detection Techniques

Ho-Yeon Kim*, Young-Hyun Choi*, Tai-Myoung Chung**
*Dept. of Electrical and Computer Engineering, Sungkyunkwan University
**School of Information Communication Engineering, Sungkyunkwan University

요 약

특정 기업 및 국가를 대상으로 하는 APT(Advanced Persistent Threat)공격의 경우 특정 시스템을 겨냥하여 제작되기 때문에 기존의 시그니처 기반의 악성코드 탐지 방식으로는 해당 악성코드를 탐지할 수 없다. 따라서 알려지지 않은 악성코드를 탐지할 수 있는 행위 기반의 악성코드 탐지 방식이 최근 이슈화되었다. 본 논문에서는 연구되고 있는 행위 분석 기반의 악성코드 탐지 방식들을 분석함으로써 향후 행위 기반 악성코드 탐지 기술 개발 및 연구에 기여하고자 한다.

1. 서론

최근 DDoS공격과 개인정보 침해로 인해 악성코드에 대한 관심이 늘어가고 있다. 악성코드는 공격자가 특정 시스템에 침투하기 위해 직접 개발할 수도 있으나 악성코드 제작 툴을 바탕으로 전문적인 지식이 없는 공격자 또한 변종 악성코드를 생산해 낼 수 있다. 이와 같은 악성코드 제작 툴에 의해 대량의 변종 악성코드가 생성되고 있기 때문에 누적 악성코드의 개체 수는 급증하고 있다. AV-Test에 따르면 2011년 누적 악성코드의 개수를 4,400만 개로 집계하였으며 안티 바이러스 업체인 McAfee는 자사 데이터베이스에 저장되어 있는 악성코드 시그니처 개수를 7,000만 개라고 밝힌 바 있다[4][5]. 이처럼 날로 늘어가는 악성코드에 대응하기 위해 국내·외 안티 바이러스 분석가들은 악성코드 샘플을 수집하여 탐지할 수 있는 시그니처를 추출하고 있다. 하지만 분석가가 일일이 악성코드가 수집될 때 마다 시그니처를 추출하는 일은 매일 수만 개씩 등장하는 모든 악성코드에 대응하기란 현실적으로 불가능하다. 또한 제로데이공격이나 특정 시스템을 겨냥한 APT공격용 악성코드들은 샘플을 수집할 수 없어, 안티 바이러스 제품으로 탐지할 수 없다. 따라서 악성코드를 탐지하는 방식으로 자동화된 동적 분석 방식들이 연구되고 있다. 본 논문에서는 알려지지 않은 악성코드를 탐지하기 위한 행위 기반의 악성코드 탐지 기술들을 분석함으로써 향후 행위 기반의 악성코드 탐지 기술 연구에 기여하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 악성코드 분석을 위한 악성코드 분석방법을 소개한다. 3장에서는 분석된 정보를 바탕으로 악성코드를 탐지하는 방식들을 살펴보고 4장에서 행위 기반의 악성코드 탐지 기술을 비교·분석한다. 마지막으로 5장에서는 본 논문의 결론을 맺는다.

2. 악성코드 분석 방법

시그니처 기반의 악성코드 탐지 방식 또는 행위 기반의 악성코드 탐지방식을 위해서는 악성코드를 사전에 분석하고 이를 구체화한 데이터베이스가 필요하다. 악성코드 분석 방법은 크게 정적 분석(Static analysis)과 동적 분석(Dynamic Analysis) 그리고 이 두 분석 방법을 혼합한 혼합 분석(Hybrid Analysis)로 나눌 수 있다. 본 장에서는 수집된 악성코드를 분석하기 위한 프로그램 분석방법인 정적 분석과 동적 분석에 대해 설명한다.

2.1 정적 분석

정적 분석은 프로그램을 실행하지 않고 디스어셈블러(Dis-assembler)와 디컴파일러(Decompiler)를 이용하여 PE(Portable Execution) 포맷으로 이루어진 악성코드를 역공학(Reverse engineering)을 통하여 분석하는 방법이다. 분석가들은 역공학으로 얻은 어셈블리어를 통해 악성코드가 수행하는 악성행위를 디버깅(Debugging)할 수 있다. 정적 분석의 경우 프로그램을 직접 실행하지 않기 때문에 시스템이 안정적이지만 다양한 이벤트 효과를 적용할 수 없어, 악성코드의 모든 실행과정을 모니터링 할 수 없다는 단점이 존재한다.

2.2 동적 분석

동적 분석은 정적 분석과는 반대 개념으로 악성코드를 독립된 정상적인 시스템에서 수행하며 동적 분석 도구들을 이용하여 시스템의 변화를 모니터링하는 방식이다. 동적 분석의 경우 특정 이벤트 및 시스템에서만 동작하는 악성코드라고 하더라도 가상머신을 이용하여 특정 시스템 환경을 구현할 수 있으며, 다양한 입력을 임의로 적용할 수 있어 정적 분석으로 발견할 수 없는 악성코드를 탐지할 수 있다. 하지만 시스템에서 직접 실행해야 하기 때문에 가상머신이 아니라면 시스템에 손상을 입힐 수 있다. 또한 다양한 안티VM 기술을 통해 가상머신을 우회하는 기능을 적용할 수 있기 때문에 안티VM을 무력화하는 2차적인 기능의 사용, 모든 입력값에 대한 모니터링 등으로 분석에 많은 오버헤드가 발생하는 단점이 있다.

3. 악성코드 탐지 방법

악성코드를 탐지 방법은 악성코드의 분석을 통하여 추출한 시그니처를 기반으로 하는 시그니처 기반의 탐지 방식과 프로그램을 실행하면서 수행하는 행위를 기반으로 악성코드를 탐지하는 행위기반의 악성코드 탐지방식이 존재한다. 또한 악성코드를 탐지하는 위치에 따라서 네트워크 기반과 호스트 기반으로 나눌 수 있다. 다음 <표 1>은 악성코드 탐지 및 대응 기술을 구분을 나타낸 표이다[2].

<표 1> 악성코드 탐지 및 대응 기술 구분

	시그니처 기반	행위 기반
호스트 기반	▪ 분석된 악성코드의 패턴을 이용하여 탐지 ▪ 알려지지 않은 신종, 변종 악성코드 탐지가 어려움	▪ 악성코드의 이상 행위를 기반으로 탐지 ▪ 오탐의 소지가 있음
네트워크 기반	▪ 보유 비정상 트래픽 패턴을 이용하여 봇 트래픽 탐지 ▪ 패턴을 벗어나는 경우 탐지 불가	▪ 네트워크 트래픽을 분석하여 비정상 봇넷 트래픽 탐지 ▪ 오탐의 소지가 있고 탐지율이 매우 낮음

3.1 호스트-시그니처 기반의 악성코드 탐지

호스트-시그니처 기반의 악성코드 탐지 기술은 현재 대부분의 개인용 컴퓨터에서 사용되고 있는 악성코드 탐지 기술로 악성코드에 대응하는 시그니처를 데이터베이스화 하여 호스트 시스템에 감염된 악성코드를 탐지하는 방식이다. 악성코드 검사에 대해 낮은 자원을 소모하고 검사 자체가 시스템에 보안적 위협을 주지는 않는다. 하지만 1장과 2장에서 언급한 바와 같이 알려지지 않은 악성코드에 대해서는 탐지가 어려운 단점이 있다.

3.2 호스트-행위 기반의 악성코드 탐지

호스트-행위 기반의 악성코드 탐지 방식은 동적 분석 도구-예를 들면 Sandbox-를 통해 악성코드를 탐지하는 방식이다. 악성코드의 판단 기준은 악성코드 실행 시 나타나는 시스템의 변화-파일 생성 및 삭제, 레지스트리의 변화 등-를 통해 악성코드를 탐지한다. 행위 자체를 분석함으로써 악성코드를 판단하기 때문에 알려지지 않은 악성코드를 탐지할 수 있으나 오탐의 소지가 있다.

3.3 네트워크-시그니처 기반의 악성코드 탐지

네트워크-시그니처 기반의 악성코드 탐지 방식은 네트워크단에서 시그니처를 기반으로 하여 악성코드를 탐지하는 방식이다. 사용되는 시그니처는 공격패턴 등을 기반으로 하며 침입 탐지 시스템(IDS, Intrusion Detection System), 침입 방지 시스템(IPS, Intrusion Prevention System) 등에 사용된다. 시그니처를 기반으로 하기 때문에 알려지지 않은 악성코드에 대해서는 탐지가 어려운 단점이 잇다.

3.4 네트워크-행위 기반의 악성코드 탐지

네트워크-행위 기반의 악성코드 탐지 방식은 네트워크 트래픽을 모니터링 하고 미리 정의된 룰셋(rule set)에 기반하여 악성행위를 판단, 탐지하는 방식이다. 호스트-행위 기반의 악성코드 탐지 방식과 마찬가지로 오탐의 가능성이 높은 단점이 있다.

4. 행위 기반의 악성코드 탐지 기술

앞서 언급한 바와 같이 시그니처 기반의 악성코드 탐지 기술로는 알려지지 않은 악성코드의 탐지가 불가능하기 때문에 시그니처 기반의 악성코드 탐지 기술과 더불어 행위 기반의 악성코드 탐지 기술의 필요성은 커져가고 있다.

본 장에서는 호스트 단 및 네트워크 단에서의 행위 기반의 악성코드 탐지 기술에 대해 분석한다. 분석은 기본적으로 다음과 같은 형식을 따른다.

- Base: 해당 악성코드 탐지 기술이 수행되는 시점을 나타낸다. Host는 최종 사용자를 나타내며 Network는 IDS와 같은 네트워크 시스템을 의미한다.
- Authors: 본 행위 기반의 악성코드 탐지 기술을 제안한 제안자들을 나타낸다.
- Name: 행위 기반 악성코드 탐지 기술에 대해 명명된 이름을 나타낸다.
- Date: 제안된 기술이 발표된 연도를 나타낸다.
- Input: 악성 행위를 판단하기 위해 시스템에서 수집하는 외부 메타데이터를 의미한다.
- Criteria: Criteria는 'Input'으로 수집한 메타데이터를

기준으로 악성코드의 판단 유무를 결정짓는 정상행위를 나타낸다.
- Target: Target은 탐지하는 악성코드의 종류를 나타낸다.
- Engine Type: Engine Type은 Input으로 수집한 메타데이터를 Criteria와 비교 분석하기 위해 사용되는 분석 방식을 나타낸다.

4.1 PAYL

Wang과 Stolofo는 페이로드 기반의 악성코드 탐지 방법인 PAYL을 제안하였다[6]. PAYL는 악성코드의 행위를 정상적인 행위와 구분을 위해 학습단계를 거쳐 바이트 주기(Byte frequency) 프로파일을 계산한다.
- Base: Network
- Authors: Wang and Stolofo
- Name: PAYL
- Date: 2004
- Input: Byte frequency
- Criteria: Profile
- Target: Malicious behavioral
- Engine Type: Mahalanobis distance

4.2 Data mining

Lee와 Stolfo 는 데이터마이닝을 접목한 IDS를 제안하였다[7]. 본 방식은 데이터마이닝 알고리즘 중 Association rules 알고리즘과 frequent episodes 알고리즘을 기반으로 한다.
- Base: Network
- Authors: W. Lee, S. and J. Stolfo
- Name: Not specified
- Date: 1998
- Input: System Call
- Criteria: Rule set
- Target: Malicious Behavioral
- Engine Type: Data mining

4.3 Behaviour Graph

Clemens Kolbitsch 등은 기존에 최종사용자단에서 사용하는 안티 바이러스 소프트웨어 제품에 추가 적용할 수 있는 호스트-행위 기반의 악성코드 탐지 기술을 제안하였다[8]. 본 탐지 기술은 시스템 콜의 행위 그래프를 기반으로 하여 악성코드를 판단한다. 시스템 콜을 행위 그래프로 표현하는 방식은 테인트 분석을 기반으로 하며 디스어셈블러와 인스트럭션 시퀀스를 Anubis라는 동적 분석 툴을 통해 분석한다[10].
- Base: Host
- Authors: Clemens Kolbitsch *et al.*
- Name: Not specified
- Date: 2009
- Input: System Call
- Criteria: Training dataset
- Target: All kind of Malware
- Engine Type: Behavior graph

4.4 HookFinder

Heng Yin 등은 악성코드가 수행하는 악성행위인 후킹 정보를 모니터링하여 악성행위를 검출하는 HookFinder를 제안하였다[9]. 본 방식은 버클리 대학에서 수행하고 있는 BitBlaze 프로젝트에 일환으로 QEMU를 기반으로 한 동적 분석용 가상머신인 TEMU상에서 수행된다[12].
- Base: Host
- Authors: Heng Yin *et al.*
- Name: HookFinder
- Date: 2008
- Input: Instructions
- Criteria: Tainted data
- Target: All kind of Malware
- Engine Type: Taint analysis

4.5 Behavior-based spyware detection

E. Kirda 등은 행위 기반의 스파이웨어(Spyware) 탐지 방식을 제안하였다. 스파이웨어는 악성코드의 한 종류로 호스트 시스템 내부에서 사용자 정보를 탈취하여 외부로 유출시키거나 현재 시스템을 사용하고 있는 사용자의 습성, 취향 등을 모니터링하는 악성코드를 일컫는다. 해당 방식은 다른 행위 기반의 악성코드 탐지 기술과 마찬가지로 시스템 콜의 사용 여부를 모니터링 한다. 모니터링 된 시스템 콜은 브라우저 관련 시스템 콜이 이며, 해당 시스템 콜의 예상 되는 룰셋을 정의하고 해당 범주를 벗어날 경우 스파이웨어라고 판단한다.
- Base: Host
- Authors: E. Kirda *et al.*
- Name: Not specified
- Date: 2006
- Input: System Call
- Criteria: Rule set
- Target: Spyware
- Engine Type: Hybrid analysis

4.6 E-mail worm vaccine

Stelios Sidroglou 등은 e-mail에 첨부된 파일을 대상으로 해당 파일이 악성코드인지 탐지하는 시스템을 제안하였다[1]. 본 방식은 이메일에 첨부되어 있는 파일만을 가상머신에서 실행하여 해당 파일이 악성코드인지 판단한다.

<표 2> 행위 기반 악성행위 탐지 기술 비교

Base	Authors	Name	Date	Input	Criteria	Target	Engine Type
Network	Wang and Stolofo	PAYL	2004	Byte Frequency	Profile	Malicious Behavioral	Mahalanobis distance
	W. Lee, S. and J. Stolfo	Not specified	1998	System Call	Rule set	Malicious Behavioral	Data mining
Host	Clemens Kolbitsch *et al.*	Not specified	2009	System Call	Training dataset	All kind of Malware	Behavior graph
	Heng Yin *et al.*	HookFinder	2008	Instructions	Tainted data	All kind of Malware	Taint analysis
	E. Kirda *et al.*	Not specified	2006	System Call	Rule set	Spyware	Hybrid analysis
	Stelios Sidiroglu *et al.*	Not specified	2005	Access Registry	Registry feature	Attached file	RAD

사용되는 가상머신은 VMWare를 사용하며 악성코드 판별 유무는 RAD(Registry Anomaly Detection)을 사용한다. RAD는 윈도우즈 레지스트리에 접근하는 정보들을 기반으로 악성행위를 판단하게 되며 본 시스템에서는 프로세스 이름, 쿼리 타입, 접근한 키 값, 응답되는 값, 결과 값을 사용한다.

- Base: Host
- Authors: Stelios Sidiroglu *et al.*
- Name: Not specified
- Date: 2005
- Input: Access Registry
- Criteria: Registry feature
- Target: Attached file
- Engine Type: RAD

5. 결론

본 논문에서는 악성코드의 탐지 방식 중 행위 기반의 악성코드 탐지 방식을 분석을 목적으로 하였다. 현재의 시그니처 기반의 악성코드 탐지 기술로는 탐지가 불가능한 제로데이 공격 또는 특정 시스템을 겨냥한 APT공격에 대응하기 위해서는 행위 기반의 악성코드 탐지 기술에 대한 연구가 필요하다. 본 논문을 통하여 새로운 행위 기반의 악성코드 탐지 시스템을 개발하거나 현재 연구 중인 행위 기반의 악성코드 탐지 기술에 대한 연구를 수행할 때 도움을 줄 수 있다.

향후 연구로는 행위 기반의 악성코드 탐지 기술과 더불어 다양한 악성코드 탐지 기술에 대해 분석하고 이를 통한 새로운 악성코드 탐지 시스템을 개발하는데 기틀을 마련한다.

Acknowledgement

본 논문은 중소기업청에서 지원하는 2011년도 산학연공동기술개발사업(No. 000443010111)의 연구수행으로 인한 결과물임을 밝힙니다.

참고문헌

[1] S. Sidiroglou, *et al.* "An email worm vaccine architecture" Information Security Practice and Experience, pp. 97-108, 2005.

[2] 정현철, "Detection and Response Technology for Botnets" 한국인터넷진흥원, 06. 2008.

[3] 임을규, "악성코드 현황 및 탐지 기술" 정보과학회지, vol. 30, pp. 44-53, 2012.

[4] McAfee, "McAfee Threats Report: Third Quater 2011" McAfee Labs, 2011.

[5] AV-Test, Available: http://www.av-test.org, 2011

[6] K. Wang, S. J. Stolfo, "Anomalous payload-based network intrusion detection" Recent Advances in Intrusion Detection, 2004, pp. 203-222.

[7] W. Lee, S. J. Stolfo, "Data mining approaches for intrusion detection" Proceedings of the 7th Conference on USENIX Security Symposium-Volume 7, 1998, pp. 6-6.

[8] C. Kolbitsch, P. M. Comparetti, C. Kruegel, E. Kirda, X. Zhou and X. F. Wang, "Effective and efficient malware detection at the end host" in Proceedings of the 18th Conference on USENIX Security Symposium, 2009, pp. 351-366.

[9] Z. Liang, H. Yin and D. Song, "HookFinder: Identifying and understanding malware hooking behaviors" Department of Electrical and Computing Engineering, pp. 41, 2008.

[10] Anubis, http://anubis.iseclab.org, 2012.

[11] E. Kirda, C. Kruegel, G. Banks, G. Vigna and R. Kemmerer, "Behavior-based spyware detection" in Usenix Security Symposium, 2006.

[12] BitBlaze: Binary Analysis for Computer Security, http://bitblaze.cs.berkeley.edu, 2012