은닉형 악성코드 분석을 위한 행위 추출연구 동향

황호*, 문대성*,**, 김익균**
*과학기술연합대학원대학교(UST)
**한국전자통신연구원
e-mail:{kcats, daesung, ikkim21}@etri.re.kr

A Study of Research Issue about Behavior Extraction Technique for Evasive Malware

Ho Hwang*, Dae-Sung Moon*,**, Ik-Kun Kim*
*Dept of Information Security, Korea University of Science and Technology(UST)
**Network Secuirty Research Laboratory, Electronics and Communications Research Institute(ETRI)

요 약

오늘날의 백신은 일반적으로 시그니처 기반 탐지법을 이용한다. 시그니처 탐지기법은 악성코드의 특정한 패턴을 비교하여 효율적이고 오탐율이 낮은 기법이다. 하지만 알려지지 않은 악성코드와 난독화 기법이 적용된 악성코드를 분석하는데 한계가 있다. 악성코드를 실행하여 나타나는 행위를 분석하는 동적분석 방법은 특정한 조건에서만 악성행위를 나타내는 은닉형 악성코드(Evasive Malware)를 탐지하는 데 한계를 지닌다. 본 논문에서는 은닉형 악성코드에 적용된 기법에 관하여 소개하고 나아가 이를 탐지하기 위한 방법에 관한 기술동향을 소개한다.

1. 서론

오늘날의 백신은 기하급수적으로 증가하는 악성코드를 탐지하기 위해 시그니처 기반 탐지법을 이용한다. 시그니처 탐지기법은 악성코드의 특정한 패턴을 비교하는 효율적이고 오탐율이 낮은 기법이다. 하지만 알려지지 않은 악성코드와 난독화 기법이 적용된 악성코드를 분석하는데 한계가 있다. 이에 대응하기 위해 실행한 악성코드의 행위를 분석하는 동적 분석 방법은 특정 조건을 만족해야 악성행위를 하는 은닉형 악성코드(Evasive Malware)를 탐지하는데 한계를 지닌다[1].

본 논문에서는 은닉형 악성코드에 대해 알아보고 악성행위를 추출하기 위한 노력을 소개한다. 2장에서는 은닉형 악성코드와 분석환경에 대해 소개한다. 3장에서는 은닉형 악성코드를 탐지하기 위한 연구를 소개하고, 4장에서 결론을 맺는다.

2. 은닉형 악성코드

은닉형 악성코드는 특정한 조건을 만족해야 악성행위를 나타내는 악성코드로, 방아쇠 기반 악성코드(Trigger Based Malware)라고도 불린다. 특정한 조건은 악성코드 제작자가 정의하며, 공격 대상의 환경을 지칭하거나 분석환경을 회피하는 용도로 사용된다. 그러므로 악성코드가 자신이 분석환경에서 실행된다고 판단하면 악성코드는 정상적인 행위만 나타낼 것이다. 이는 악성코드 동적 분석이 다수의 악성코드를 반복적으로 실행하기에 적합한 가상환경에 진행되고, 가상환경은 실제 환경과 차이가 있는 특징을 이용한다.

가상 환경은 방식에 따라 표 1과 같이 가상화(virtualization), 에뮬레이션(emulation), 하이퍼바이저(hypervisor)로 나뉜다. 3가지 방식은 각각 에뮬레이션은 Qemu 기반인 Anubis, 하이퍼바이저는 Xen 기반인 Ether, 가상화는 VirtualBox 기반 Cockoo sandbox가 있다[2-4].

<표 1> 가상화 방식에 따른 분석도구

가상화 방식	프로그램	분석환경
에뮬레이션 (emulation)	Qemu	Anubis
하이퍼바이저 (hypervisor)	Xen	Ether
가상화 (virtualization)	VirtuaBox	Cockoo sandbox

은닉형 악성코드가 분석환경에서 실행여부를 판단하는 조건을 크게 세 종류로 나뉜다 [5-6]. 첫째는 프로세스, 레지스트리, 드라이버의 존재를 확인하거나 IDT(Interrupt Descriptor Table)과 GDT(Global Descriptor Table)와 같은 주소와 같이 실제 환경과 분석환경의 구성의 차이를 확인한다. 둘째는 RDTSC(Read Time Stamp Counter) 명령

어과 같이 실제 환경과 가상환경의 성능 차이를 비교한다. 셋째는 동적 분석의 자동화를 고려하여 마우스 이벤트와 같이 사용자의 입력을 기다리는 방법이 있다. 이와 같이 악성코드 제작자는 다수의 조건을 혼합하여 모두 만족하였을 때 악성행위를 나타나도록 하여 악성 행위의 추출을 어렵게 만들 수 있다.

- ■ 실제 환경과 분석환경의 구성 차이
  - 프로세스(vboxservice.exe), 드라이버(VBoxMouse.sys)
- ■ 실제 환경과 분석환경의 성능 차이
  - RDTSC, VM exit overhead
- ■ 사용자의 입력을 대기
  - 마우스 이벤트, 키보드 입력 등

3. 은닉형 악성코드 분석 방법

본 논문은 은닉형 악성코드를 탐지하는 방법을 3가지로 나눠서 소개한다. 사용자의 입력을 모방하는 방법과 다양한 환경을 구성하는 방법, 그리고 Taint analysis를 이용한 방법이다.

Fleck과 Joo는 사용자의 입력을 모방하는 방법으로 악성행위를 추출하는 방법을 제안했다 [6-7]. Fleck[6]은 키보드, 마우스와 같은 사용자의 이벤트를 기록하고 재생하여 사용자의 입력을 확인하는 악성코드의 행위를 추출했다. Joo[7]는 사용자 이벤트를 목록화하고 이 중 하나를 추출하여 입력하는 퍼징(fuzzing)을 이용했다.

Kirat과 Lindorfer는 은닉형 악성코드가 동작할 다양한 분석환경에서 행위를 추출하고 유사도를 분석하는 방법을 제안했다[8-9]. Kirat[8]은 2장의 3가지 방법(에뮬레이션, 가상화, 하이퍼바이저)과 어떠한 분석도구로 설치하지 않은 실제 환경(bare metal)으로 나눠 악성코드를 실행하였다. 다른 분석환경에서 disk와 network의 변화를 판단하여 악성행위의 추출여부를 판단했다. Lindorfer[9]는 운영체제 이미지와 모니터링 방식을 조합한 악성행위 추출법을 제안했다. 언어와 설치 소프트웨어를 다르게 한 운영체제 이미지와 anubis나 driver와 같이 악성코드의 행위를 기록하는 방법을 조합하였다.

사용자의 입력을 모방하는 방법과 다양한 환경을 구성하는 방법은 두 가지 문제를 가진다. 첫째, 분석환경에 준비한 조건이 악성행위의 동작과 관련이 없을 가능성이 높다. 둘째, 사전에 악성코드에 대한 정보를 모르는 상태에서 현실적으로 악성코드 제작가가 사용한 모든 조건을 준비할 수 없다. 다양한 조건을 준비하면 할수록 하나의 악성코드의 분석에 필요한 분석환경의 수가 증가하여 자원의 낭비를 초래한다. 특히, 특정 날짜에 동작하는 악성행위를 추출하기 위해 날짜를 다르게 분석환경을 구성한다면 엄청난 자원의 낭비를 초래할 수 있다. 그러므로 지능화 된 악성코드를 분석하기 위해서 실용적이고 범용적인 악성코드 행위추출 방법이 필요하다.

Peng과 Hwang은 정보의 흐름을 추적하는 taint analysis 기법을 이용하여 악성행위를 추출하는 방법을 제안했다[10-11]. 악성코드가 받은 입력을 추적하여 조건문에 도달하였을 때 실행경로들을 모두 탐색하기 위해 forced execution을 이용했고 메모리 충돌(memory corruption)이 발생했을 때 즉시 복구해주는 기법을 사용했다. 이 방법은 사용자의 입력을 모방하는 방법과 다양한 환경을 구성하는 방법이 환경을 사전에 모두 준비할 수 없는 단점을 해결할 수 있다. Peng[10]은 악성코드가 특정 조건을 기준으로 나뉘는 실행경로를 work list방식으로 탐색하는 방법을 제안했다. Hwang[11]은 악성코드의 입력과 관련된 실행경로를 파이프라인화하여 탐색하는 방법을 제안했다.

taint analysis를 이용한 악성행위 추출법은 두 가지의 문제점을 가진다. 첫째, forced execution을 사용하여 도달할 수 없는 경로를 실행하거나 jump table(점프 테이블)의 경우 실행경로를 탐색하지 못할 가능성이 존재한다. 둘째, solver를 이용하지 않고 forced execution을 했더라도 여전히 정보를 추적하기 위해 taint analysis를 하므로 실행경로의 길이에 비례해 명령어가 추가되고 정보가 많아져 분석속도가 느려지는 문제를 가지고 있다.

4. 결론

난독화가 적용된 악성코드의 증가에 대응하기 위해 악성행위를 추출하는 범용적인 기법이 필요하다. 이에 본 논문은 악성행위를 추출하기 위한 기존의 연구에 대해 소개했다. 다양한 환경을 구성하는 방법과 사용자의 이벤트를 기록하고 모방하는 방법은 악성코드 제작자가 정의 할 모든 조건을 대비할 수 없다는 한계를 가진다. 또한, 서버와 통신을 하거나 특정 날짜에만 작동하는 악성코드와 같이 환경과 이벤트와 관계없는 조건에서 악성 행위를 하는 악성코드는 악성행위 추출이 불가능한 한계를 가진다. 그러므로 특정 조건에 영향을 받지 않는 taint analysis를 이용하는 악성행위를 추출법이 지속적으로 연구되어야 한다. 하지만 악성코드 제작자가 의도하지 않은 경로에 도달하거나 정상적인 실행흐름에서 벗어날 수 있는 문제를 가진다. 뿐만 아니라 taint analysis을 이용한 악성행위 추출법은 프로그램의 크기나 실행경로의 수에 따라 추적해야 하는 정보의 양이 급속하게 증가하는 문제도 가진다. 앞으로 위 두 문제가 해결되어 실용적이고 범용적인 악성행위 추출법에 관한 연구가 진행되길 기대한다.

Acknowledgement

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 수행하였음.[No. B0101-15-1293, 다중소스 데이터의 Long-term History 분석기반 사이버 표적공격 인지 및 추적 기술개발]

참고문헌

[1] D. Brumley, C. Hartwig, Z. Liang, J. Newsome, D. Song and H. Yin, "Automatically identifying trigger-based behavior in malware," In Botnet Detection , pp. 65-88, Springer US, 2008.

[2] U. Bayer, I. Habibi, , D. Balzarotti, , E. Kirda, and C. Kruegel, A View on Current Malware Behaviors. In LEET, 2009.

[3] A. Dinaburg, P. Royal, , M. Sharif and W. Lee, "Ether: malware analysis via hardware virtualization extensions." In Proceedings of the 15th ACM conference on Computer and communications security, pp. 51-62, 2008.

[4] Cuckoo Sandbox. http://www.cuckoosandbox.org.

[5] Sudeep Singh, "Breaking the Sandbox", Sep. 2014.

[6] D. Fleck, , A. Tokhtabayev, A. Alarif, A. Stavrou and T. Nykodym, "Pytrigger: A system to trigger & extract user-activated malware behavior," In Availability, Reliability and Security (ARES), pp. 92-101, 2013

[7] J. U. Joo, I. Shin and M. Kim, "Efficient Methods to Trigger Adversarial Behaviors from Malware during Virtual Execution in SandBox," International Journal of Security and Its Applications, vol. 9, no. 1, pp. 369-376. 2015

[8] D. Kirat, G Vigna and C. Kruegel."Barecloud: bare-metal analysis-based evasive malware detection", In Proceedings of the 23rd USENIX Security Symposium, pp. 287-301, 2014

[9] M. Lindorfer, C Kolbitsch and P.M. Comparetti. "Detecting environment-sensitive malware," In Recent Advances in Intrusion Detection, pp.338-357, 2011

[10] F. Peng, Z. Deng, X. Zhang, D. Xu, Z. Lin, Z and Z. Su, "X-force: Force-executing binary programs for security applications", In Proceedings of the 2014 USENIX Security Symposium, pp.829-844, 2014

[11] H. Hwang, D.S. Moon, I.K. Kim, "Efficient Exploring Multiple Execution Path for Dynamic Malware Analysis," Journal of The Korea Institute of Information Security & Cryptology, vol. 26, no. 2, pp. 377-386, 2016