# 모바일 기기에서의 동적 악성 코드 검출 시스템에 대한 연구

권동현*, 이하윤*, 조영필*, 백윤흥*,
*서울대학교 전기정보공학부
e-mail:dhkwon@sor.snu.ac.kr

# A Study on Dynamic Mobile Malware Detection Systems

Dong-Hyun Kwon*, Ha-Yoon Yi*, Young-Pil Cho*, Yun-Heung Paek*
*Dept of Electrical and Computer Engineering, Seoul National University

요 약

모바일 기기가 점차 발전하고 사람들 사이에 널리 보급되면서 여러 가지 모바일 응용 어플리케이션들을 통하여 모바일기기는 사람들의 많은 개인정보를 담고 있게 되었다. 하지만 이에 따라 이러한 모바일 기기의 정보들을 노리는 악성 코드를 포함한 어플리케이션 또한 점차 늘어나고 있다. 그래서 모바일 환경에서의 악성 코드를 검출하기 위한 여러 가지 많은 연구들이 진행되고 있는데 이번 연구에서는 별도의 수행환경에서 동적 분석을 기반으로 한 악성 코드 검출 연구들을 살펴보도록 하겠다.

## 1. 서론

최근 모바일 기기의 확산과 더불어 모바일 기기의 보안에 대한 관심이 높아지고 있다. 모바일 기기는 그 특성상 기존 PC와는 달리 사용자가 일상생활에서 거의 항상 지니고 다니기 때문에 사용자의 위치정보, 사진 및 동영상 등 사용자의 개인정보들이 많이 담고 있게 된다. 이러한 모바일 기기의 특성에 따라 그림 1과 같이 실제 국내에서도 이러한 모바일 악성코드로 인한 피해사례가 점차 많이 발생하고 있다.

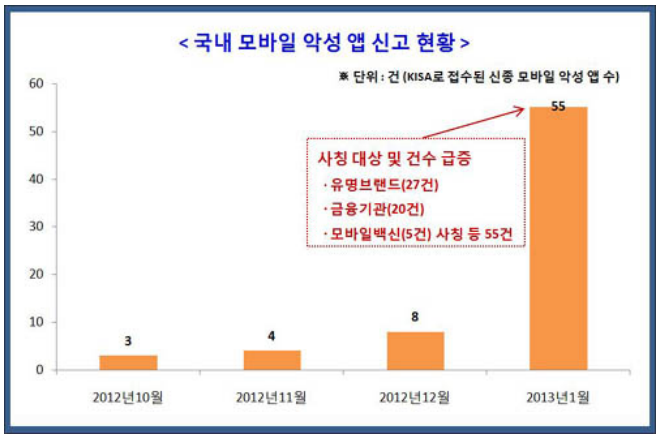


그림 1 국내 모바일 악성 어플리케이션 신고현황[1]

그래서 최근에는 이러한 어플리케이션에 들어있는 악성코드를 검출하기 위한 다양한 연구가 진행되고 있다. 이러한 연구는 크게 정적분석과 동적 분석을 기반으로 악성코드를 검출하고 있다. 먼저 정적분석은 어플리케이션 바이너리를 디컴파일하여 얻은 코드를 바탕으로 해당 어플리케이션의 시스템접근권한, 사용하는 API, 함수들의 호출 그래프 등을 통해 악성코드의 유무를 검출하는 방식이다. 하지만 이러한 정적 분석만으로는 코드난독화를 이용한 공격이라든지 동적 코드 로딩과 같이 수행 중이 아니고서는 어떠한 동작을 할지 알 수 없는 공격의 경우에 효과적으로 검출할 수 없다는 문제점을 가지고 있다.[2] 그래서 최근에는 이러한 정적분석의 단점을 보완하는 동적 분석 관련 연구들이 많이 제시되고 있다. 동적 분석은 수행 중에 불린 시스템 콜, 명령어 호출 순서, 기기의 배터리 잔량 및 프로세서 사용량 등을 통해 해당 어플리케이션의 동작을 감시함으로써 악성코드의 유무를 판단하는 방식이다. 하지만 이러한 동적 분석 방식의 경우에는 악성행위 감시를 시스템 성능 오버헤드가 크다는 문제를 가지고 있다. 그래서 최근 이러한 동적 분석을 기반으로 한 연구에서는 사용자 기기 성능 저하를 막기 위해 별도의 에뮬레이터와 같은 가상머신 수행환경에서 동적 분석을 진행하고 있다.

따라서 본 연구에서는 별도의 수행환경에서 모바일 기기에서의 악성코드를 동적 분석하는 연구들을 비교해보기 위하여 먼저 2장에서는 기존연구들은 어떠한 것들이 있는지 살펴보고 3장에서는 이러한 접근방식들을 비교해 본 뒤, 4장에서 결론 제시하도록 하겠다.

## 2. 별도의 수행환경을 통한 동적 악성코드 검출 시스템

(1) 샌드박스 기반의 접근방식

먼저 샌드박스 기반의 접근 방식의 경우는 에뮬레이터와 같은 가상머신을 하나 할당을 하고, 가상머신에 분석하려는 어플리케이션을 설치한다. 그리고 실제 사용자 입력을 사용할 수 없기 때문에 임의의 입력이나 가상의 사용자 입력을 생성하여 해당 어플리케이션의 동작을 감시하는 방법이다. 보통 해당 어플리케이션을 가지고 있기 때문에 동적 분석뿐만 아니라 정적분석 또한 함께 진행하게 된다. 샌드박스 기반에 접근방식의 기존 연구 및 소프트웨어들은 AASandbox[3], Droidbox[4], Google bouncer 등이 있다.

(2) 미러링 기반의 접근방식

미러링을 이용한 접근방식의 경우, 먼저 샌드박스 기반의 접근방식과 마찬가지로 에뮬레이터와 같은 가상머신을 할당을 하게 된다. 그리고 감시하려는 사용자의 모바일 디바이스 환경을 그대로 해당 에뮬레이터에 복사하게 된다. 실제 사용 시에는 해당 사용자 모바일 디바이스에서의 사용자 입력이 발생할 때마다 이를 가상머신에 그대로 재현함으로써 동작을 감시하는 방법이다. 이러한 방식의 경우 대체로 가상머신을 사용자 디바이스가 사용하는 네트워크의 프록시 서버가 관리할 수 있게 함으로써 사용자 디바이스의 네트워크 트래픽을 그대로 해당 에뮬레이터로 미러링 할 수 있게 한다. 미러링 기반 접근방식의 기존 연구들은 Paranoid[5], Secloud[6] 등이 있다.

## 3. 접근 방식들의 비교와 한계점

앞선 2장에서 제시한 것처럼 샌드박스 기반의 접근 방식과 미러링 기반의 접근방식은 모두 에뮬레이터와 같은 가상머신을 사용하여 동적 분석을 한다는 공통점이 있지만 그 특성은 차이가 있다.

먼저 샌드박스 기반 접근 방식의 장점은 사용자가 해당 어플리케이션을 사용하기 전에 별도의 기기에서 검사를 수행하기 때문에 사용자 기기의 성능 오버헤드가 없고, 수행하는 과정에서 발생할 수 있는 악성코드에 대한 공격을 사용자 기기는 받지 않는 다는 장점이 있다. 하지만 일반적으로 검사시기와 실제 사용자의 사용하는 시기가 다르다는 점과 실제 사용자 입력을 통한 검사가 아니기 때문에 코드 커버리지 문제가 있다.

다음으로 미러링 기반의 접근 방식의 장점은 분석이 사용자가 사용하는 시기와 거의 같은 시기에 할 수 있고, 실제 사용자의 입력을 통해 동작을 감시 할 수 있기 때문에 코드 커버리지 문제와 무관하다는 장점이 있다. 하지만 초기 사용자의 디바이스 환경을 그대로 가상머신으로 보내는 데이터양이 보통 수 기가바이트에 해당 할 정도로 큰 오버헤드가 발생한다는 문제가 있다. 또한 어플리케이션의 수행이 가상머신 뿐만 아니라 모바일 디바이스에서도 수행되기 때문에 가상머신을 통해 악성행위를 탐지하였다 하더라도 디바이스의 악성행위를 사전에 막는 것을 보장하지 못하고 사후처리를 해야 한다는 단점이 있다.

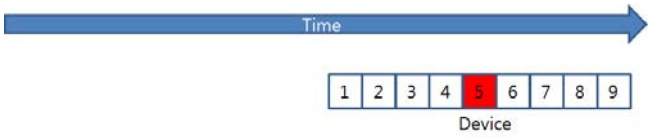


그림 2 모바일 기기에서의 악성코드 수행


그림 3 샌드박스 기반 접근 방식에서의 악성코드 수행

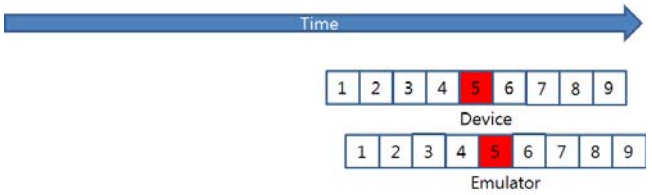


그림 4 미러링 기반 접근 방식에서의 악성코드 수행

그림 2는 악성 코드를 포함한 어플리케이션이 디바이스에 수행되는 모습을 보여준다. 즉 해당 어플리케이션의 일반적인 동작 순서를 1~9이라고 할 때 5번에 해당하는 코드가 악성코드인 예제이다. 이 때 샌드박스 기반의 접근방식에서는 그림 3과 같이 디바이스에서 수행되기 전에 에뮬레이터에서 수행하지만 실제 사용자가 사용하는 순서에 따른 검증을 해보지 못할 수도 있고, 검사하지 못하는 코드가 있을 경우 악성코드를 감지하지 못할 가능성이 있다. 한편 미러링 기반의 접근방식의 경우 그림 4와 같이 거의 디바이스 비슷한 시간에 검사가 이루어지지만 에뮬레이터에서 악성행위를 알아챈 시각이 디바이스보다 늦을 경우 악성 코드에 의한 악성 행위를 사전에 막을 수 없다는 한계를 지니고 있다.

## 4. 결론

별도의 수행환경을 통한 악성 코드 검출 시스템은 제한된 자원을 지닌 모바일 환경에서의 동적 분석을 위해서 필연적으로 있어야 할 것이다. 하지만 이번 연구에서 살펴본 것처럼 기존의 접근 방식들은 각기 다른 장단점을 지니고 또한 한계점을 가지고 있다. 따라서 앞으로의 관련연구에서는 각각의 접근방식을 보완하여 단점을 극복하려는

방향 혹은 새로운 접근방식으로 각 접근방식들의 한계를 극복하려는 방향으로 진행되어야 할 것이다.

## Acknowledgement

본 연구는 교육과학기술부/한국과학재단 우수연구센터 육성사업(과제번호 2012-0000470), 중소기업청에서 지원하는 2012년도 산학연공동기술개발사업(No. C0019562), 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신) [No. 10047212, 1kB 이하 암호문 간의 연산을 지원하는 동형 암호 원천 기술 개발 및 응용 연구] 및 IDEC의 지원을 받아 수행하였습니다.

## 참고문헌

[1] http://www.kisa.kr/notice/pressView.jsp?mode=view&p_No=8&b_No=8&d_No=1045

[2] Rastogi, Vaibhav, Yan Chen, and Xuxian Jiang. "Droidchameleon: evaluating android anti-malware against transformation attacks." Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, 2013.

[3] Blasing, Thomas, et al. "An android application sandbox system for suspicious software detection." Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on. IEEE, 2010.

[4] https://code.google.com/p/droidbox/

[5] Portokalidis, Georgios, et al. "Paranoid Android: versatile protection for smartphones." Proceedings of the 26th Annual Computer Security Applications Conference. ACM, 2010.

[6] Zonouz, Saman, et al. "Secloud: A cloud-based comprehensive and lightweight security solution for smartphones." Computers & Security 37 (2013): 215-227.