모바일 안드로이드 운영체제를 공격하는 커널 기반 악성코드 탐지방법 연구

정기문*, 김진숙**
한국과학기술정보연구원
e-mail : kmjeong@kisti.re.kr, jinsuk@kisti.re.kr

A Study of Detection Method for Kernel based Malwares in Mobile Android OS

Kimoon Jeong*, Jinsuk Kim **
*Dept. of Supercomputing Service Enter, Korea Institute of Science and Technology Information
** Dept. of S&T-SEC, Korea Institute of Science and Technology Information

요 약

스마트폰에 주로 사용되고 있는 안드로이드 OS 는 다양한 악성코드로 인해 금전적 피해, 데이터 유출 및 통제권한 상실 등과 같은 많은 피해를 당하고 있다. 침해 위협을 가중시키고 있는 모바일 악성코드 중 심각한 피해를 유발하는 커널 기반의 루팅(Rooting) 악성코드는 일반적인 탐지 방법으로는 찾아낼 수 없는 어려움이 있다. 본 논문에서는 커널 기반에서 동작하는 루팅(Rooting) 악성코드를 탐지하기 위한 방법을 제안한다. 스마트폰 어플리케이션이 실행될 때마다 생성되는 모든 프로세스의 UID 를 확인하여 비정상적으로 사용자(User) 권한에서 관리자(Root) 권한으로 변환되는지를 확인하는 방법이다. 제안하는 방법을 활용하여 알려지지 않은 악성코드로 인한 안드로이드 OS 의 피해를 최소화할 수 있을 것으로 기대된다.

1. 서론

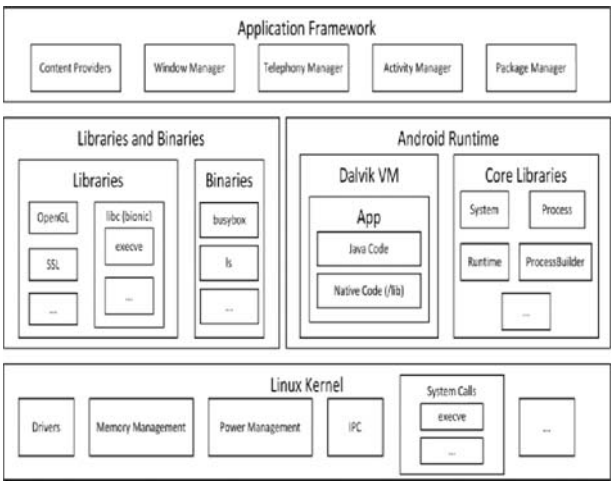
최근 스마트폰의 급속한 보급으로 인해 스마트폰을 대상을 하는 보안위협도 급속히 증가하고 있다.

특히 안드로이드 OS 의 경우 악성코드에 의해 스마트폰의 개인정보가 유출되거나 사용자가 인지하지 못한 상태에서 시스템이 장악되는 등의 다양한 공격에 많이 노출되어 있다. 이러한 피해는 공격자가 스마트폰 단말기의 관리자 권한으로 명령을 실행시킬 수 있는 취약점을 이용한 공격에 의해서 가능한데 모바일 안드로이드 운영체제에 기본적으로 탐재된 리눅스 커널의 취약점을 공격하는 루팅(Rooting) 악성코드에 의해서 발생하게 되면 사용자가 인지할 수 없을 뿐만 아니라 탐지하는 것이 매우 어렵다는 문제가 있다.

모바일 안드로이드 악성코드를 탐지하기 위하여 시스템에서 데이터 흐름을 추적하거나 스마트폰의 다양한 파라미터 값을 추출하여 점검하는 방법 등이 지속적으로 제안되고 있으나 커널 기반의 루팅 행위를 탐지하지 못하고 있다. 리눅스 커널에 보아을 적용하는 방법도 제안되고 있으나 부분적인 루팅 행위만 탐지할 수 있는 한계점을 가지고 있다.

본 논문에서는 모바일 안드로이드 운영체제에 심각한 피해를 초래할 수 있는 루팅 악성코드를 탐지하기 위하여 비정상적으로 사용자(User) 권한에서 관리자(Root) 권한으로 변환되는지를 확인하는 방법을 제안한다. 제안된 방법이 모바일 안드로이드 커널 기반 공격에 대한 대응방안으로 기여할 것으로 예상된다.

2. 관련 연구



(그림 1) 안드로이드 OS 구조

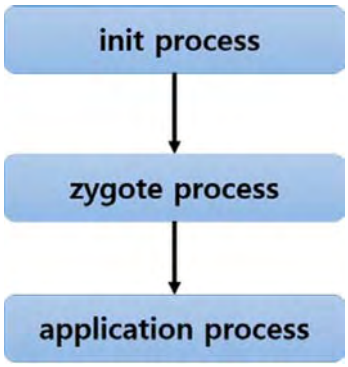
안드로이드 운영체제는 (그림 1)과 같은 형태로 구성되어 있다. 『Application Framework』 레이어는 자원을 효과적으로 사용하기 위하여 다양한 관리 도구로 구성되어 있다. 『Libraries and Binaries』 레이

어는 C 언어로 개발되어 있어 다양한 함수들을 빠르게 실행시킬 수 있는 기능을 제공한다. 『Android Runtime』 레이어는 운영체제와 Dalvik 가상머신을 위한 핵심 라이브러리로 구성되어 있다. 특히 Dalivik 가상머신은 다중 가상머신을 효과적으로 운영할 수있도록 개발되어있다. 마지막으로 『Linux Kernel』 레이어는 운영체제 관리를 위한 드라이버, 하드웨어를 직접적으로 운영하기 위한 드라이버 등으로 구성되어 있다.

모바일 안드로이드 운영체제를 공격하는 악성코드에 대응하기 위하여 시스템 레벨에서 데이터 흐름을 추적하는 TaintDroid[1], 스마트폰의 다양한 파라미터값을 추출하여 점검하는 Andromoly[2] 등이 있다. 하지만 이러한 방법은 실행시 커널 기반의 루팅(Rooting) 행위를 탐지하지 못하는 한계점이 있다. 루팅(Rooting) 악성행위에 대응하기 위하여 리눅스 커널기반의 보안을 적용한 SEAndroid[3], HwanTaek[4]과 같은 연구가 진행되었지만 부분적인 탐지만 가능하였다.

3. 제안하는 방법

일반적인 안드로이드 어플리케이션이 실행될 때 구동되는 프로세스는 (그림 2)와 같은 단계로 실행된다. 관리자(root)권한인 init process 가 어플리케이션 실행을 위한 zygote process 를 생성시키고, zygote process 는 실행하고자 하는 어플리케이션을 사용자(user) 권한으로 실행시킨다.



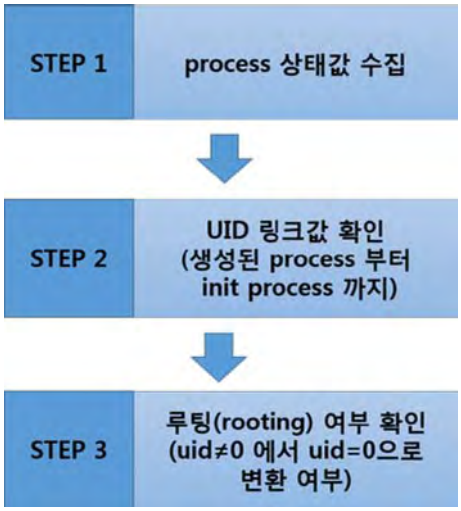
(그림 2) Rooting 악성코드 탐지방법

이 과정에서 각 프로세스의 uid 값의 변화를 살펴보면 다음과 같다. init process 의 uid = 0 이고, zygote process 의 uid = 0 이다. 반면 user 권한으로 실행되는 application process 의 uid≠0 으로 일반적 4~5 자리 단위의 정수가 된다. 또한 해당 application process 가 fork 로 생성하는 새로운 프로세스 또한 uid≠0 인 된다.

반면 관리자 권한을 확보하려 시도하는 루팅(rooting) 악성코드는 일반적인 어플리케이션과 달리 프로세스 uid≠0 인 상태에서 root 권한인 uid = 0 인 새로운 프로세스를 반드시 생성하게 된다.

따라서 새롭게 생성되는 모든 프로세스들을 감시하면서 uid 값을 추적해 uid≠0 인 user 권한 프로세스 상태에서 uid = 0 인 root 권한 프세스가 생성되면 악의적인 커널기반 루팅(rooting) 행위가 발생하였음을 탐지할 수 있다.

제안하는 루팅(rooting) 악성코드 탐지방법은 이와 같은 루팅(rooting) 행위의 특징을 이용한다. 어플리케이션 및 명령어 등이 실행될 때 마다 생성되는 모든 프로세스에 대해서 비정상적인 방법으로 root 권한으로 실행되지를 확인하는 것으로 (그림 3)과 같은 3 단계의 절차를 통해 탐지할 수 있다.



(그림 3) Rooting 악성코드 탐지방법

4. Conclusion

본 논문에서 우리는 루팅(rooting) 악성코드를 탐지하기 위한 방법을 제안하였다. 어플리케이션이 실행될때 마다 생성되는 모든 프로세스를 확인하여 비정상적으로 user 권한에서 root 권한으로 변환되는지를 확인하는 것이다. 제안 방법을 통해 rooting 악성코드 탐지분야에 기여할 것으로 예상되며 향후 rooting 의 종류를 더 다양하게 구분할 수 있는 연구를 진행할 예정이다.

참고문헌

[1] William Enck et al, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," the USENIX Symposium on Operating Systems Design and Implementation, Oct. 2010.

[2] Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, Yael Weiss, "Andromaly:a behavioral malware detection framework for android devices," Journal of Intelligent Information Systems, vol.38, Issues.1, Feb. 2012.

[3] SEAndorid, http://seandroid.bitbucket.org/

[4] Lee, Hwan-Taek, Minkyu Park, and Seong-Je Cho. "Detection and prevention of LeNa Malware on Android." Journal of Internet Services and Information Security (JISIS) 3.3/4: 63-71.

[5] Enck W, Ongtang M, McDanael P, "Understanding Android Security," IEEE, Jan.-Feb. 2009.