안드로이드 환경에서의 유저모드 루트킷의 위협

정준권*, 한선희*, 정태명**
*성균관대학교 전자전기컴퓨터공학과
**성균관대학교 정보통신공학부
e-mail:{jkjung, shhan}@imtl.skku.ac.kr, tmchung@ece.skku.ac.kr

A Threat of Usermode Rootkits on Android Environment

Jun-Kwon Jung*, Sun-Hee Han*, Tai-Myoung Chung**
*Department of Electrical and Computer Engineering, Sungkyunkwan University
**School of Information and Communication Engineering, Sungkyunkwan University

요 약

스마트폰의 사용이 늘어나면서 스마트폰의 취약점을 노리는 악성코드들도 많이 발생되고 있다. 특히 악성코드를 숨겨주는 루트킷이 최근 캐리어IQ사태를 통해 이슈가 되면서 루트킷에 대한 관심이 늘어가고 있다. 루트킷은 동작방식에 따라 유저모드 루트킷과 커널모드 루트킷으로 나눌 수 있는데 PC처럼 운영체제를 통해 자원 및 프로세스를 제어하는 스마트폰도 루트킷의 위협에 안전하지 못하다. 본 논문은 PC환경에서 동작하는 루트킷의 동작원리를 파악하고 스마트폰 환경 특히 안드로이드 환경의 유저모드 루트킷의 동작과 위협을 주지시키고자 한다.

1. 서론

스마트폰은 전세계 모바일 시장을 점유해 나가고 있다. 2010년 4분기에 아이폰은 약 3500만 대, 안드로이드폰은 약 7500만 대의의 판매량을 기록하였고, 전체 스마트폰의 판매량은 1억5천만대 이상을 기록했다[12].

스마트폰은 3G, WiFi등 무선인터넷망을 자유롭게 사용할 수 있고, 그 중 대부분의 WiFi망이 비밀번호 설정과 같은 최소한의 보안대책조차 없이 제공되고 있다. 이러한 노출된 환경에서는 개인정보를 노리는 해커들이 쉽게 스마트폰으로 침입할 수 있게 된다. 다양한 악성코드들이 제로데이(Zero-day), 스팸 메시지, 심지어는 앱을 유통하는 마켓도 활용하여 스마트폰에 침입하고 동작하고자 한다. 이때 루트킷은 악성코드들이 장기간 활동할 수 있도록 악성코드의 존재를 숨겨주는 역할을 한다.

이러한 루트킷은 1990년대 초에 등장했으며, 이름이 알려지기 시작한 것은 2005년 소니BMG가 CD에 루트킷을 심었던 사건이 밝혀지면서 부터다. 이 사건 이후 현재까지 다양한 루트킷이 개발, 발견되고 있고, 최근에는 캐리어IQ사가 스마트폰에 사용자 몰래 루트킷을 심어 상당한 개인정보를 수집한 사실이 드러나면서 해커들의 관심이 루트킷에 쏠리고 있다.

루트킷이 침투 된 단말은 일반적인 안티 바이러스 소프트웨어를 사용해도 이를 탐지할 수 없게 된다. 공격자는 루트킷을 통해 피해자의 단말에 키로거, 애드웨어 등을 설치하거나 공격자가 원하는 동작을 수행 시킨다. 또한 악성코드를 감염시켜 단말을 봇넷으로도 활용하여 디도스같은 추가적인 보안 위협을 가할 수도 있다. 이러한 위협은 PC처럼 운영체제를 활용하여 동작하는 스마트폰에서도 나타날 수 있다. 하지만 문제는 스마트폰용 루트킷을 탐지하거나 제거하는 툴이 거의 없다는 점에 있다. 따라서 본 논문은 현재 시스템에서의 루트킷과 탐지기법을 조사하고 스마트폰 특히 안드로이드 환경에서의 루트킷 위험성을 강조하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 루트킷의 정의와 현재 전반적인 시스템에서 발견된 루트킷들의 동작체계를 살펴보고, 3장에서는 현재 PC 환경에서 동작하는 루트킷, 특히 유저모드 루트킷이 어떻게 동작하는지를 살펴본다. 4장에서는 스마트폰 환경에서의 루트킷의 위협성과 안드로이드 환경에서의 유저모드 루트킷의 구현 가능성을 알아보고 5장에서 결론과 향후 연구방향에 대해 설명한다.

2. 관련연구

본 장에서는 루트킷의 정의와 루트킷을 활용한 악성코드, 그리고 전반적인 시스템에서의 루트킷의 분류를 설명한다.

2.1 루트킷

좁은 의미로 루트킷은 시스템의 탐지를 피하기 위해 루트 권한을 획득하여 자신을 숨기는 프로그램을 의미한다[7].

그리고 넓은 의미로 루트킷은 자신의 존재를 숨긴채로 OS의 권한을 획득하여 비정상적인 동작을 수행하는 소프트웨어들을 의미한다. 이러한 루트킷이 가지는 세가지 목표는 다음과 같다[1].

1. Run : 루트킷은 자신이 침투한 컴퓨터를 아무런 제한 없이 사용할 수 있어야 한다. 대부분의 컴퓨터 시스

템은 액세스 콘트롤 리스트(ACL)라는 자원 접근 제어 메커니즘을 통해 잘못된 권한 접근을 방지한다. 때문에 루트킷은 시스템의 취약점이나 사회공학적 방법을 활용하여 자신이 원하는 동작을 수행할 권한을 획득한다.

2. Hide : 루트킷은 컴퓨터에 설치된 보안 프로그램에게 탐지되지 않아야 한다. 루트킷은 획득한 루트권한을 이용해 보안 프로그램들이 사용하는 시스템 명령을 위조하여 자신을 숨기게 된다.

3. Act : 루트킷은 은닉 외에도 배포자가 원하는 동작을 수행할 수 있어야 한다. 루트킷 배포자는 루트킷을 심는 것에 만족하지 않고 비밀번호를 훔쳐내거나 악성코드를 심는 등의 행위를 루트킷을 통해 수행한다.

공격자들은 악성코드에 루트킷을 포함하여 악의적인 행위, 즉 스팸 메일 발송 또는 봇넷 구성 등을 시도한다. 루트킷은 자체의 위협보다 루트킷을 활용하는 악성코드의 활동을 감추고 유지시키는 것에 초점이 맞춰진다.

2.2 루트킷을 활용한 악성코드

루트킷을 활용한 악성코드는 여러 가지 악의적의 목적을 가지고 활용될 수 있다. 루트킷이 스팸메일을 보내거나 봇넷을 활용한 DDoS 공격 등 다양한 기능을 하는 악성코드들을 숨겨주는 역할을 담당하고 있다. <표 1>은 대표적인 루트킷 악성코드인 Rustock, TDL3, Stuxnet를 소개한다[2,4,5,6].

<표 1> 루트킷을 활용한 악성코드

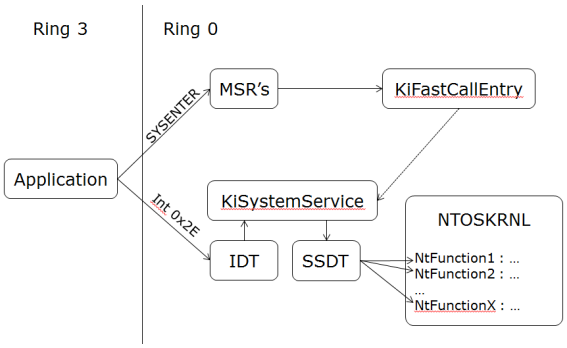
악성코드	악성행위	루트킷 기술
Rustock	스팸메일 발송 (약 300억 개)	- 윈도우 시스템 드라이버로 위장하여 실행된 후 원래의 드라이버 파일로 복구
TDL3	봇넷(DDoS)	- IRP 후킹 - 파티션 되지 않은 디스크 부분에 은신
Stuxnet	산업시설 파괴	- USB를 통해 개별 네트워크로 침입 - 기존에 탈취한 인증서를 활용하여 정상적인 드라이버로 위장

각각의 악성코드들은 포함된 루트킷을 활용함으로써 자신을 은닉하여 장기간 활동할 수 있다. 즉 악성코드에 포함된 루트킷이 악성코드를 오래 지속시키고 더욱 확산시켜 악성코드에 의한 피해를 증가시키게 만든다.

2.3 루트킷의 분류

소개된 악성코드들에 포함된 루트킷은 악성코드를 각종 탐지 시스템에 노출되지 않도록 숨겨주는 역할을 한다. 루트킷은 정상적인 응용 프로그램 및 DLL 파일을 조작하여 정상적인 파일로 위장하거나 커널의 관리자 권한을 비정상적인 방법으로 획득하여 탐지 프로그램이 루트킷을 찾지 못하게 한다.

루트킷은 동작 원리에 따라 유저모드 루트킷과 커널모드 루트킷으로 나눌 수 있다[7]. 유저모드 루트킷은 정상적인 응용프로그램 자체로 위장하거나 정상적인 응용프로그램에 침투하여 자신을 은폐한다. 유저모드 루트킷은 IAT (Import Address Table)와 EAT(Export Address Table)라는 시스템 명령을 매핑한 테이블에 접근한다. 유저모드 루트킷은 여기에 원하는 명령어의 주소를 삽입 또는 변조하거나 특정한 명령의 인라인 후킹을 하게 된다. 이를 통하여 유저모드 루트킷은 자신을 존재를 숨기고 추가적인 동작을 위해 포함시킨 코드를 실행 시킬 수 있다. 커널모드 루트킷은 커널의 권한을 비정상적인 방법으로 탈취하여 정상적인 드라이버나 서비스 함수로 위장하여 특정 프로세스나 파일, 레지스트리 키 등을 감추게 한다[8]. 유닉스 환경에서의 커널모드 루트킷은 LKM(Loadable Kernel Module)으로 위장하여 운영체제를 속인다. 윈도우 환경에서는 SSDT (System Service Dispatch Table)을 후킹하거나 정상적인 드라이버 sys파일을 위조하는 방법 등으로 커널모드 루트킷이 동작한다. (그림 1)은 윈도우 환경에서 Ring 3 권한의 응용프로그램이 Ring 0 권한의 커널 동작을 요청하는 경로를 보여주고 있다. 루트킷은 경로의 일부를 후킹하여 권한 획득 및 은닉행동을 한다.



(그림 1) 윈도우 환경에서의 서비스 콜 경로[10]

3. PC 운영체제상의 유저모드 루트킷

루트킷은 동작목적이 서로 동일하지만 동작원리 및 구조는 운영체제마다 서로 다르다. 본 장에서는 대표적인 운영체제인 윈도우와 리눅스 상의 유저모드 루트킷의 종류와 동작체계 및 탐지 체계를 설명한다.

3.1 윈도우 시스템상의 유저모드 루트킷

윈도우는 PC용 OS 점유율이 90% 이상인 운영체제이다[9]. 그만큼 윈도우의 보안체계를 뚫기 위한 시도가 많았고, 루트킷도 다양한 종류가 발견되었다.

윈도우 응용프로그램은 API를 이용하여 운영체제에게 필요한 기능을 요청한다. 이런 점을 활용하여 유저모드 루트킷은 시스템 응용프로그램을 공격한다. 만약 응용프로그램이 디렉토리 내의 파일을 나열하기 위해서는 Kernel32.dll 파일의 API를 활용해야 한다. 응용프로그램이 API를 사용하기 위해 Kernel32.dll을 메모리에 탑재하고 각 함수

<표 2> 운영체제별 유저모드 루트킷

운영체제	침투방식	탐지방식
Windows	IAT를 후킹하여 루트킷 프로세스 실행	IAT에 등록된 DLL파일을 하나하나 추적하여 분석
	응용프로그램의 바이너리 앞부분을 교체	프로세스의 jmp의 동작을 후킹하여 정상적인 시스템 메모리 영역에서 동작하는지 검사
	레지스트리 및 메시지 후킹을 통해 루트킷 DLL 로드	미리 저장해 놓은 레지스트리 구조 비교 및 IAT의 현재의 DLL 리스트와 비교
Linux	시스템 명령어 바이너리를 교체하여 특정 폴더 및 프로그램 은닉	시스템 명령어 파일의 무결성 검증
	시스템 라이브러리를 수정 후 해당 데몬을 재시작하여 시스템 명령어를 오작동 시킴	파일시스템 또는 동작 프로세스 전수 조사

들의 메모리 주소는 IAT에 저장하도록 커널에 요청을 하게 된다. IAT에 저장된 함수가 호출되면 이에 대응되는 커널레벨 함수를 레지스터에 로드하여 커널의 동작을 수행한다. 유저모드 루트킷은 IAT를 후킹하거나 특정 함수의 바이너리를 교체하거나 시스템 DLL을 로드할 때 루트킷 DLL을 덧붙인다[10].

3.2 리눅스 시스템상의 유저모드 루트킷

리눅스 환경에서의 루트킷은 시스템 명령어를 교체하여 백도어나 트로이잔을 숨겨주는 역할을 한다. 예를 들어 루트킷은 ls를 미리 조작해 놓은 파일로 교체하여 백도어가 설치된 폴더를 감추어 보이지 않게 할 수 있다. 또한 inetd와 같은 시스템 데몬의 conf파일을 수정하여 특정한 프로세스를 감출 수도 있다.

FreeBSD나 솔라리스같은 유닉스기반의 운영체제는 리눅스와 거의 동일한 구조를 가진다. 따라서 각 운영체제별로 세세한 구조는 다를지라도 동작원리는 동일하다. <표 2> 는 두 운영체제별 유저모드 루트킷의 동작원리와 탐지방식을 소개하고 있다[7,10].

스마트폰은 피쳐폰과는 달리 PC처럼 운영체제를 탑재하여 동작하므로 파일시스템이나 프로세스, 어플리케이션 개발언어 등이 PC와 같거나 유사하다. 따라서 스마트폰 환경에서도 PC환경에 존재하는 루트킷과 비슷하게 동작하는 루트킷이 등장한다. 4장에서는 스마트폰 환경에서 동작하는 루트킷의 동작원리를 살펴보고자 한다.

4. 스마트폰 시스템에서의 루트킷

PC환경과 마찬가지로 스마트폰 환경도 루트킷으로부터 결코 안전하지 않다. 본 장에서는 스마트폰 환경에서의 유저모드 루트킷의 위협 가능성과 유저모드 루트킷의 동작구조를 소개한다. 본 논문에서는 안드로이드 환경을 기준으로 설명한다.

4.1 스마트폰 유저모드 루트킷의 위협

최근들어 시큐리티 업체로부터 캐리어 IQ 사건과 같은 스마트폰 루트킷 감염에 대한 보고가 나타나고 있다. 이외에도 특히 커널모드 루트킷에 대한 연구가 다양하게 발표되고 있다. 최근에는 안드로이드용 커널의 시스템 콜 테이블을 이용한 루트킷 기술을 소개하고 안드로이드의 취약성을 경고한 논문이 발표되었다[3]. 이를 비롯하여 스마트폰 루트킷의 정보는 대부분 커널모드 루트킷에 집중되어 있다. 그 이유는 첫째로 커널모드 루트킷이 유저모드 루트킷보다 탐지 및 제거가 어렵기 때문이고, 둘째로 현재 다른 시스템에서도 커널모드 루트킷에 대한 연구가 주를 이루고 있기 때문이다.

유저모드 루트킷은 커널모드 루트킷에 비해 침입 범위가 제한적이고 탐지가 쉽다. 그러나 현재 스마트폰 환경에서는 유저모드 루트킷이 이런 제약이 있다 하더라도 위협적이다. 그 이유는 다음과 같다.

- 은닉을 목적으로 하는 루트킷의 특성상 일반적인 접근방법으로 탐지하기 어렵기 때문에 현재 보안 체계로는 다양한 방식으로 접근하는 악성코드의 침투를 사전에 완벽하게 차단할 수 없다. 만약 취약점을 통해 루트킷이 활동을 시작하게 되면 사용자는 피해를 입고 있다는 사실조차 알기 어렵다.
- 스마트폰 환경은 PC환경보다 유저모드 루트킷의 위협에 더 노출되어 있다. PC환경은 IceSword, GMER, RootkitRevealer등 다양한 루트킷 탐지 툴이 있는 반면, 스마트폰 환경에 맞춰 개발된 루트킷 탐지 툴이 적어 활동 중인 유저모드 루트킷을 찾아내기 어렵다.
- 유저모드 루트킷을 활용한 악성코드의 피해가 지속되고 있다. 2011년 3월 Smiscer Rootkit을 활용한 악성코드를 통해 약 250,000개의 시스템이 감염되었고, 약 520만 달러의 피해가 발생되었다[11]. Smiscer Rootkit은 DLL 인젝션, 레지스트리 조작 등 유저모드 루트킷 기술들을 활용한 악성코드이다. 정상적인 프로세스가 읽어 들일 DLL 파일에다 DLL 인젝션을 활용하여 API의 처리 흐름을 조작한다. 또 변조한 레지스트리 드라이버를 추가한 후 커널에 접근하여 정상적인 디스크 드라이버로 위장한다. 감시 프로세스가 드라이버에 접근하면 위조된 결과, 즉 정상적으로 보이는 결과를 반환하여 자신을 숨기게 된다.

4.2 안드로이드 환경에서의 유저모드 루트킷의 동작 가능성

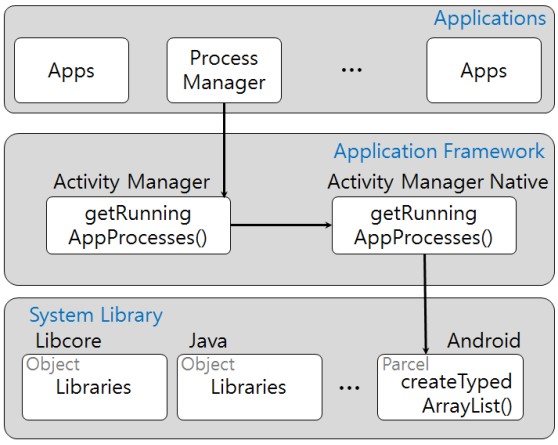
안드로이드 환경은 운영체제의 소스코드가 공개되어 있어 다른 스마트폰 운영체제에 비해 시스템의 구조와 동

작체계를 분석하기 용이하기 때문에 루트킷에 더 노출되어 있다. 그만큼 다른 스마트폰 운영체제보다 안드로이드 환경이 루트킷을 제작하기에 유리하다. 본 절에서는 현재 수행중인 프로세스를 감추는 유저모드 루트킷이 안드로이드 환경에서 어떤 식으로 동작할 수 있는지 소개한다.

하나의 앱이 동작하기 위해서는 한 개 또는 여러 개의 Activity가 동작해야 한다. Activity의 관리를 담당하는 객체인 ActivityManager는 Activity 생성, 종료, 동작 등을 제어하게 된다. ActivityManager의 메서드 중에서 현재 실행중인 프로세스 리스트를 가져 오는 getRunningAppProcesses()는 ActivityManagerNative를 거쳐 안드로이드 framework.jar 라이브러리 안의 Parcel 객체의 createTypedArrayList()를 호출하게 된다. 이 메서드는 프로세스 간의 통신을 지원하는 역할을 한다. Parcel.java 소스안의 메서드를 특정한 이름을 가진 프로세스 리스트를 제외하고 반환 하도록 수정하여 빌드를 한 후 framework.jar를 교체하게 되면 작업관리자에서 특정한 프로세스를 감출 수 있게 된다. (그림 2)는 작업 관리자 앱이 프로세스 리스트를 보기 위한 시스템 라이브러리를 호출하기 위한 경로를 보여준다.

만약 안드로이드 사용자가 이 루트킷을 포함한 악성앱을 다운받는다면 이 앱을 실행 후 종료를 시켜도 실제로 악성코드의 프로세스는 종료되지 않고 동작할 수 있다. 따라서 사용자는 악성코드의 동작을 알아차릴 수 없기 때문에 악성코드는 장기간 스마트폰에 남아 있으면서 악성 메시지 발송, DDoS, 봇넷 등 주기적인 악성행위 또는 공격자의 원격제어에 의한 피해를 입게 될 수 있다.

또한 유저모드 루트킷을 통해 일반적인 앱이 접근할 수 없는 시스템 라이브러리를 변조하게 되면 동일한 권한을 가진 백신 앱도 변조된 라이브러리를 활용하게 되므로 탐지를 회피할 수 있게 된다. 덧붙여 앞 장에서 언급한 Smiscer Rootkit처럼 유저모드 루트킷 기술에 적절한 커널모드 루트킷 기술을 접목하여 활용한다면 탐지 뿐 아니라 제거도 어려운 루트킷이 될 수 있다.



(그림 2) 안드로이드 환경에서의 라이브러리 호출 경로

5. 결론 및 향후 연구

본 논문에서는 스마트폰 환경에서 유저모드 루트킷의 동작 가능성을 분석하였고, 이를 통해 스마트폰 환경에서 유저모드 루트킷이 다른 악성코드와 함께 동작함으로써 스마트폰 보안을 위협하는 존재가 될 수 있음을 시사하였다. 하지만 루트킷은 자신이 직접 악성행위를 하지 않는다. 루트킷의 위협성은 악성코드의 동작과 존재를 숨겨주어 악성코드에 의한 피해를 증대 시킨다는 점에 있다. 유저모드 루트킷은 PC환경에서 커널모드 루트킷에 비해 탐지가 쉬운데다가 충분한 탐지, 대응체계 있어 주목해야할 필요성이 줄어들었다. 하지만 스마트폰이라는 새로운 플랫폼이 생기고 이에 대응하는 체계가 부족한 상황에서 유저모드 루트킷에 대한 대응체계가 필요하다. 따라서 향후에는 스마트폰, 특히 안드로이드 환경에 맞춰 동작하는 유저모드 루트킷을 탐지, 제거할 수 있는 체계에 대해 연구할 계획이다.

ACKNOWLEDGEMENT

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 중점연구소지원사업(2011-0018397)으로 수행된 연구로 인한 결과물임을 밝힙니다.

참고문헌

[1] Spencer Smith, John Harrison, “Rootkit Whitepaper”, Symantec, 2012. 02

[2] THOMAS MARTIN ARNOLD, “A comparative analysis of rootkit detection techniques”, The University of Houston Clear Lake, 2011. 05

[3] Dong-Hoon You, Bong-Nam Noh, “Android platform based linux kernel rootkit”, Malicious and Unwanted Software (MALWARE), 2011 6th International Conference on, pp. 79-87, 2011. 10

[4] M86 Security, http://www.m86security.com/labs/traceitem.asp?article=1362, 2012. 02

[5] Nicolas Falliere, et, al, “W32. stuxnet dossier”, Symantec, 2011. 02

[6] Dr.WEB, “BackDoor.Tdss.565 and its modifications (aka TDL3)”, 2009. 11

[7] Anton Chuvakin, “An Overview of Unix Rootkits”, iDefense Labs, 2003. 02

[8] David Harley, “The Root of All Evil? - Rootkits Revealed”, ESET, 2006. 09

[9] Marketshare, http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8&qpcustomd=0, 2012. 02

[10] Greg Hoglund, James Butler, “루트킷, 윈도우 커널 조작의 미학”, 에이콘, pp. 97-140, 2007. 11

[11] Ahnlab, “ASEC Report Vol.14”, 2011. 03

[12] Gartner, http://www.gartner.com/it/page.jsp?id=1924314, 2012. 02