커널 모드 루트킷 개발 및 탐지기법에 관한 연구

최지원, 문봉교
동국대학교 컴퓨터공학과
e-mail: eternalklaus@dgu.ac.kr

Study on Detection Method and Development of the Kernel Mode Rootkit

Jiwon Choi, Bongkyo Moon
Dept. of Computer Science and Engineering
Dongguk University

요 약

루트킷은 쉽게 말해 루트(root)권한을 쉽게 얻게 해주는 킷(kit)이다. 루트킷은 주로 운영체제의 커널 객체를 조작함으로서 프로세스, 파일 및 레지스트리가 사용자에게 발견되지 않도록 은닉하는 일을 수행한다. 본 논문에서는 루트킷의 은닉 기법중 하나인 직접 커널 오브젝트 조작 기법 (DKOM, Direct Kernel Object Manipulation)에 대해 연구한다. 그동안 루트킷에서 많이 이용되던 DKOM 기법은 작업 관리자로부터 프로세스를 은닉하는 일을 수행하였다. 그러나 본 논문에서는 이를 응용하여 작업 관리자로부터 프로세스를 은닉할 뿐만 아니라 Anti Rootkit 도구까지 우회하는 커널모드 디바이스를 설계하고, 이를 탐지할 수 있는 새로운 방법에 대하여 제안한다.

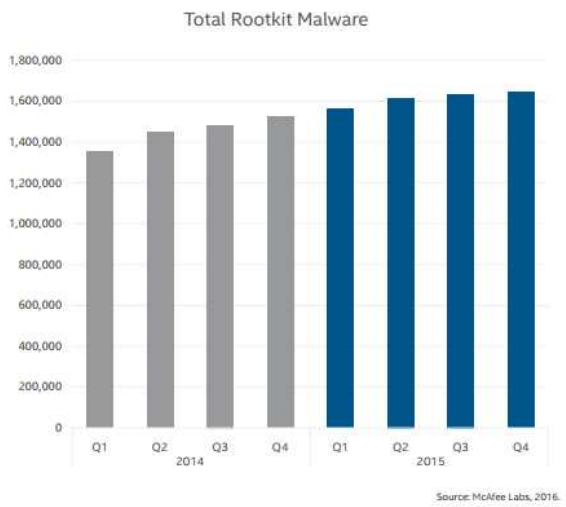
1. 서론

루트킷이란 말 그대로 루트(root)권한을 지속적으로 얻게 해주는 악성코드 킷(kit)이다. 악성코드가 시스템을 감염시킨 후에도 지속적으로 시스템을 공격하기 위해서는, 악성 프로세스의 존재 및 행위를 은닉할 필요가 있다. 이 때 루트킷은 파일이나 레지스트리를 은닉함으로써 악성코드의 탐지를 지연시키거나 어렵게 만드는 일을 수행한다. 이는 피해 시스템에 대해 지속적인 APT공격을 가능케 한다. Mcafee Labs Threats Report 2015에 따르면 루트킷 악성코드는 점차 증가하고 있으며, 2015년에는 1,600,000개를 넘어섰다 <표 1>. 즉, 루트킷이 현재까지의 많은 침해사고의 원인이 되고 있음을 알 수 있다.

DKOM이란 Direct Kernel Object Manipulation의 약자로써, 윈도우 시스템 내의 커널 오브젝트를 직접 조작하여 프로세스를 은닉하는 일을 수행하는 루트킷 기법이다. 프로세스는 커널 상의 EPROCESS라는 구조체에 정보들을 저장하는데, 이러한 커널 객체들 사이의 링크를 조작하여 프로세스 스캔의 우회가 가능하다. 이를 이용하면 시스템에 존재하는 프로세스, 디바이스 드라이버, 파일, 레지스트리 값의 은닉이 가능해진다.

본 논문에서는, 루트킷이 동작하는 영역인 커널 모드에 대해 소개하고, 윈도우 커널 구조체 EPROCESS에 대해 설명한다. 또한 윈도우 디바이스 드라이버(.sys)형태로 제작되는 루트킷의 소스코드에 대한 동작 원리를 소개한다. 결론에서는 이러한 루트킷이 실제로 동작하는 것을 증명하고, 이를 탐지하는 기법에 대하여 제안한다.

<표 1> 루트킷을 이용한 악성코드의 증가 추세 (2016, Mcafee)



2. DKOM 기법의 은닉 프로세스

2.1 유저 모드와 커널 모드

인텔 x86의 CPU는 사용자 접근 제어를 위해 커널 레벨부터 유저 레벨까지 4가지의 Ring Level을 설정한다. Ring Level이 작을수록 시스템에서 높은 권한을 가진다. 유저모드는 Ring Level 3이고 커널모드는 Ring Level 0이다. Ring Level 0(커널 모드) 프로세스는 CPU의 하드웨어에 직접적으로 접근할 수 있다. 루트킷 탐지 툴은 주로 Ring Level 3에서 관리자 프로그램으로써 동작한다. 본

논문에서 제작한 루트킷은 커널 모드에서 Ring Level 0로 동작한다. 루트킷은 커널 디바이스 드라이버(.sys)형태로 제작되어, 커널 오브젝트에 직접 접근할 수 있으며 이의 조작도 가능하다. 따라서 악성 프로세스를 커널 모드에서 은닉시킬 수 있게 된다.

2.2 디바이스 드라이버

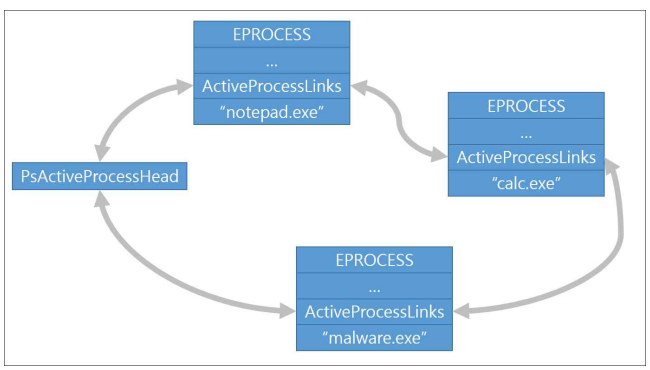
디바이스 드라이버는 시스템에 연결된 장치들을 제어하는 프로그램이다. 애플리케이션이 디바이스를 제어하기 위해서는 각 디바이스별로 각각의 특성에 맞게 설계되어 동작을 제어하는 디바이스 드라이버를 이용하게 된다. 그런데 디바이스 드라이버는 물리적인 장치뿐만 아니라, 확장 서비스를 실행하기 위해서도 사용된다. 확장 서비스 중에는 커널모드로의 접근 서비스 또한 포함된다.

기존의 시스템 서비스를 이용해서는 커널에 접근이 불가능하지만 커널에 접근이 가능한 확장서비스를 디바이스 드라이버 형태로 제작하여 사용한다면 커널로의 접근이 가능해진다. 따라서 디바이스 드라이버는 커널 모드에서 동작하는 루트킷 설계에 있어 핵심이 된다.

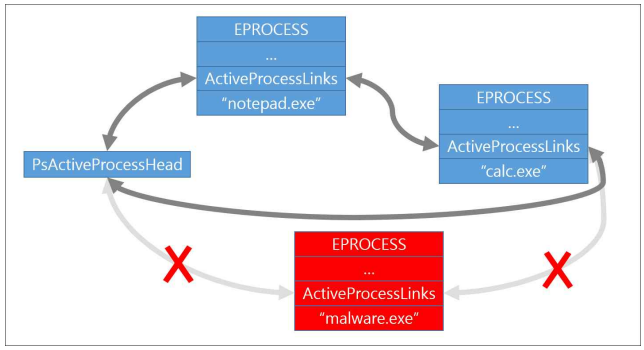
본 논문에서 제작하는 커널 루트킷은, 커널 객체(EPROCESS)를 조작하여 프로세스를 은닉한다. 이를 위해서는 커널에 접근할 수 있는 권한이 필요하다. 그러므로 루트킷은 윈도우에서 sys확장자를 가지는 디바이스 드라이버의 형태로 제작하여, 커널 모드에서 동작 가능하도록 설계한다.

2.3 DKOM

DKOM은 커널 오브젝트들의 링크를 조작하여 프로세스를 은닉하는 기법을 말한다. 예를들어 위에 언급한 EPROCESS 구조체에 존재하는 ActiveProcessLinks의 Flink와 Blink를 조작함으로써 프로세스를 은닉할 수 있다(그림 1 및 그림 2). 먼저 현재 프로세스의 Blink에 연결된 이전프로세스의 Flink를 조작하고, 현재 프로세스의 Flink에 연결된 다음 프로세스의 Bink를 조작한다. 그 다음 현재 프로세스의 Flink와 Blink를 조작한다. <표 2>는 DKOM을 C코드로 구현한 것이다.



(그림 1) 이중 연결리스트로 연결된 EPROCESS 구조체



(그림 2) DKOM기법으로 은닉된 프로세스

이러한 DKOM기법으로, 프로세스 링크들 사이에서 은닉이 가능하며, 작업 관리자와 같은 프로세스 스캔 프로그램의 탐지를 우회할 수 있다. 그 이유는 기존의 프로세스 스캔 도구는 커널의 EPROCESS를 스캔할 때, 단순히 ActiveProcessLinks만을 이용하기 때문이다.

<표 2>는 본 연구에서 개발한 디바이스 드라이버의 일부이다. Process에는 은닉 대상 프로세스의 EPROCESS 구조체 주소가 들어간다. offset_ActiveProcessLinks에는 EPROCESS의 ActiveProcessLinks의 오프셋이 설정된다. 이 오프셋은 Windows OS 종류에 따라 다른데, 본 논문에서는 Windows XP SP3의 오프셋인 0x88을 설정하였다.

<표 2> DKOM기법을 적용한 ActiveProcessLinks

```
offset_ActiveProcessLinks=0x88
// get the address of ActiveProcessLinks
ThisAPL=(PLIST_ENTRY)((PCHAR)Process+offset_Act
iveProcessLinks);
PrevAPL=ThisAPL->Blink;
NextAPL=ThisAPL->Flink;

// cut the previous/next link of ThisAPL
PrevAPL->Flink=ThisAPL->Flink;
NextAPL->Blink=ThisAPL->Blink;

// set the link to ThisAPL itself
ThisAPL->Flink=ThisAPL;
ThisAPL->Blink=ThisAPL;
```

2.4 EPROCESS 구조체

본 논문에서 조작할 커널 오브젝트는 EPROCESS이다. 이는 프로세스의 정보를 포함하는 구조체로써, 다른 프로세스들과 연관되는 링크를 포함한다. EPROCESS 구조체 내의 ActiveProcessLinks는 이전 프로세스와 연결되는 Blink와, 다음 프로세스와 연결되는 Flink를 포함하는 이중 연결 리스트이다. (그림 3)은 커널 디버거(Windbg)를 이용하여 확인한 Windows XP의 EPROCESS구조체의 일부이다. ActiveProcessLinks 구조체는 EPROCESS 구조체

내의 0x88오프셋에 존재함이 확인된다. 이러한 오프셋은 Windows 운영체제 종류에 따라 다르므로, 아래와 같은 커널 디버깅을 통한 오프셋 정보 수집은 루트킷 제작에 있어 반드시 필요한 과정이다.

```
+0x080 RundownProtect     : _EX_RUNDOWN_REF
   +0x000 Count           :  Uint4B
   +0x000 Ptr             :  Ptr32
+0x084 UniqueProcessId    : Ptr32
+0x088 ActiveProcessLinks :  _LIST_ENTRY
   +0x000 Flink           :  Ptr32
   +0x004 Blink           :  Ptr32
+0x090 QuotaUsage         : Uint4B
+0x09c QuotaPeak          : Uint4B
+0x0a8 CommitCharge       : Uint4B
+0x0ac PeakVirtualSize    : Uint4B
+0x0b0 VirtualSize        : Uint4B
```

(그림 3) Windows XP SP3에서의 ActiveProcessLinks 오프셋

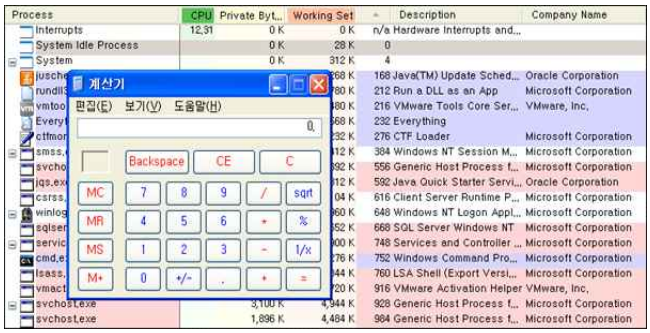
ActiveProcessLinks링크를 절단하는 것만으로는 프로세스 스캐너로부터의 우회만 가능할 뿐, gmer와 같은 안티루트킷으로부터는 우회할 수 없다. 안티루트킷 도구는 프로세스를 스캔할 때, 은닉된 프로세스 탐지를 위해 ActiveProcessLinks외에도, 수많은 EPROCESS 커널 구조체의 링크들을 스캔하기 때문이다. EPROCESS 구조체를 연결하는 이중 연결 리스트는 ActiveProcessLinks 외에도 다수 존재한다. 본 논문에서는 DKOM기법을 이용하여, 프로세스 스캐너 및 안티루트킷 도구가 이용하는 링크들을 모두 절단하였다.

기존에 DKOM을 이용한 루트킷이 ActiveProcessLinks만을 절단하였다면, 본 논문에서는 ERPOCESS 구조체에 존재하는 총 13개의 링크를 절단하였는데, 이는 기존의 루트킷에서는 볼 수 없던 새로운 방법이다.

3. 루트킷 탐지 및 우회

3.1 Process Explorer 탐지우회

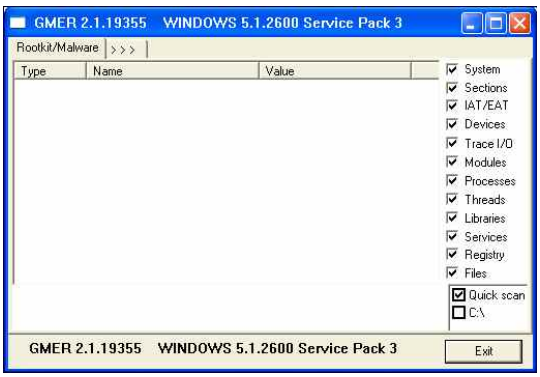
루트킷 디바이스 드라이버는 프로세스의 PID를 입력받고, 입력받은 프로세스를 은닉한다. (그림 4)는 calc.exe (PID:1384)프로세스를 실행한 후, 제작한 루트킷 드라이버를 시스템에 로드하여 은닉한 것이다. 이 경우, 프로세스는 존재하지만 Process Explorer의 프로세스 스캔에서는 1384번 프로세스를 찾아볼 수 없다.



(그림 4) Process Explorer에서의 은닉 수행

3.2 안티 루트킷 탐지우회

gmer란 숨겨진 프로세스, 숨겨진 쓰레드, 숨겨진 모듈, 숨겨진 서비스, 후킹 드라이버 등의 탐지 가 가능한 강력한 안티루트킷이다. 그러나 본 논문에서 제작한 커널 루트킷을 이용해 로 calc.exe를 은닉한 후, gmer를 실행할 경우 (그림 5)와 같이 은닉된 calc.exe 프로세스를 탐지하지 못한다. (은닉된 프로세스가 있을 경우 붉은색으로 프로세스 이름이 표시된다.) gmer의 프로세스 스캔으로부터 은닉한 것이다.



(그림 5) gmer(Anti-Rootkit) 에서의 은닉 수행

3.3 커널디버거 탐지우회

시스템 상에서는 notepad.exe와, calc.exe가 실행되고 있으며, calc.exe(Pid:1384)는 루트킷 디바이스 드라이버를 이용하여 은닉된 상태이다. Windbg 디버거를 이용하여, 커널 모드에서 프로세스를 검색하는 경우 notepad.exe의 프로세스 정보는 검색되나 calc.exe의 정보는 검색되지 않는다. 커널디버거의 프로세스 스캔으로부터 은닉한 것이다.

```
kd> !process 0 0 notepad.exe
Failed to get VAD root
PROCESS 85a5c9f0  SessionId: 0  Cid: 0478    Peb: 7ffd3000  ParentCid: 0640
    DirBase: 156c0460  ObjectTable: e4e43448  HandleCount:  48.
    Image: notepad.exe

kd> !process 0 0 calc.exe
kd>
```

(그림 6) Windbg(커널 디버거)로부터의 은닉 수행

3.4 DKOM을 이용하여 은닉된 프로세스의 탐지

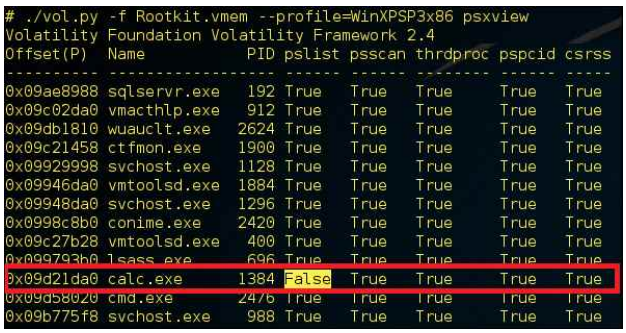
DKOM기법을 이용하여, 프로세스의 구동에 직접적인 영향을 미치지 않는 모든 이중연결리스트를 절단하였다. 이렇게 제작된 루트킷은 작업 관리자뿐만 아니라, 안티루트킷 도구로부터 우회가 가능하므로 기존의 루트킷보다 위험도가 높으며 악용의 소지도 높다. 이 기술이 백도어나 악성코드에 적용된다면 장기간에 걸친 APT공격이 가능하게 될 것이다.

이렇게 은닉된 프로세스를 탐지하기 위해서는 기존의 EPROCESS의 이중 연결 리스트를 스캔하는 것으로는 부족하다. DKOM에 의해 EPROCESS에 존재하는 링크들이 모두 절단되었기 때문이다. 따라서 DKOM기법으로 은닉된 프로세스를 탐지하는 기법을 2가지 고안한다.

첫째로 ETHREAD구조체 스캔의 방법이 있다. 모든 프로세스는 커널상에 EPROCESS구조체를 하나씩 가지고, 프로세스를 구성하는 스레드의 개수만큼 ETHREAD구조체를 가진다. 프로세스의 EPROCESS 구조체가 은닉되었다 하더라도 커널에 존재하는 ETHREAD구조체들을 스캔한다면, 은닉된 프로세스의 탐지가 가능하다.

둘째로 메모리 포렌식의 방법이 있다. 메모리에는 실행파일, 시스템 연관 데이터 구조, 관련된 사용자 활동 정보와 이벤트 등의 동적 정보가 포함되어 있다. 시스템 어떠한 프로세스도 실행되려면 메모리에 로드되어야 하므로, 메모리는 실행되는 모든 프로세스들의 정보를 포함한다. 따라서 메모리를 스캔하여 프로세스 정보들을 추출한다면, 은닉된 프로세스를 포함한 실행중인 모든 프로세스 정보를 얻을 수 있다.

Volatility는 대표적인 메모리 포렌식 도구이다. 먼저, 제작한 루트킷 프로세스가 실행되는 시스템에서 메모리 파일(Rootkit.vmem)을 추출한다. 그리고 추출된 메모리 덤프에서 프로세스를 스캔하면 (그림 7)와 같이 은닉된 calc.exe(Pid:1384) 프로세스 검출이 가능하다.



(그림 7) 메모리 포렌식

4. 결론

본 논문에서 루트킷의 은닉 기법중 하나인 직접 커널 오브젝트 조작 기법 (DKOM)에 대해 연구를 수행하고, 이를 응용하여 작업 관리자로부터 악성코드 프로세스를 은닉할 뿐만 아니라 Anti Rootkit 도구까지도 우회할 수 있는 커널모드 디바이스 기반의 루트킷을 개발하였다. 또한 개발된 우회 루트킷을 탐지할 수 있는 새로운 탐지 방법에 대하여 제안하였다.

참 고 문 헌

[1] Hoglund, Greg, 『Rootkits : subverting the Windows kernel.』 (2006), p12-p66
[2] "Kernel Mode", http://blog.csdn.net/ssmale/article/details/9788535(2016.01.03.)
[3] McAfee, 『McAfee Labs Threats Report - March 2016』 (2016)
[4] 이봉석, 『IT EXPERT 윈도우 디바이스 드라이버.』 (2009), p102-188
[5] 고홍환, 『루트킷을 이용하는 악성코드』, 국가사이버안전센터, p31-45
[6] Oleg Zaytsev, 『Rootkits, Spyware/Adware, Keyloggers and Backdoors: Detection and Neutralization』 (2006), p45-102