허니팟을 위한 원격 키스트로크 모니터링의 설계

이상인°, 박재홍, 강홍식
인제대학교 컴퓨터공학부
dclpai@dclpai.com, mr8kor@kornet.net, hskang@nice.inje.ac.kr

A Design of Remote keystroke monitoring For Honeypot

SangIn Lee°, JaeHong Park, HeungSeek Kang
Dept of Computer Engineering, Inje University

요 약

허니팟은 공격자들이 쉽게 공격할 수 있는 시스템이나 네트워크를 구성하여, 악성해커나 스크립트 키드들이 어떻게 시스템을 침입하고 공격하는지 감시할 수 있도록 구성되어 있는 시스템을 말한다. 일반적으로 허니팟은 방화벽과 로그 기록 등으로 감사기능을 수행하는데, 악성해커는 그 로그마저 복구할 수 없도록 삭제하는 경우도 있기 때문에 독립적인 추적 시스템이 필요하다. 본 논문에서는 LKM(Linux Kernel Module)기법을 이용한 키로거를 통해 공격자가 세션 상에서 입력하는 모든 키보드 내용을 기록하여 공격자의 행동을 쉽고 빠르게 분석하는 원격 키스트로크 모니터링 시스템을 설계해 보았다.

1. 서 론

최근 인터넷이 비약적으로 발전함에 따라 시스템의 위협도 크게 증가하였다. 그렇기에 공격자가 어떻게, 어떤 방법을 이용하여 시스템에 침입하는지 알기 위해 허니팟을 이용하여 공격자를 감시하는 기법이 등장하게 되었다. 허니팟이란 공격자의 공격패턴을 분석하기 위하여 만들어진 일련의 가상 혹은 실시스템을 말한다. 이를 이용하면 해커들의 행동을 예측하여 그에 대한 방어를 할 수 있고, 현재 시스템의 취약점을 알아낼 수도 있다. 즉, 허니팟의 최종 목표는 해커가 침입하고 공격하기 어려운 시스템을 구축하는 것이다. 또한 공격목표를 허니팟으로 돌리기 위한 허니넷을 구축함으로서 실제 데이터가 존재하는 시스템의 공격 성공률을 줄일 수 있다.

허니팟의 기본적인 구조는 방화벽(Firewall), 침입탐지 시스템(IDS), 로그 시스템 등으로 구축되며, 이 중 로그 시스템은 해커의 행동을 파악하기 위한 중요한 요소이다. 로그 시스템을 통해서 공격자가 어떤 명령어를 사용하였고, 어떤 기법을 사용하였는지 알 수 있기 때문이다. 여기서 중요한 것은, 공격자는 관리자가 해당 시스템의 방어를 위해 설치해놓은 도구들을 먼저 제거 후 공격을 시행한다는 것이다. 이를 위해 관리자가 수행해야 할 것은 해당 시스템을 이중 구조의 백업시스템으로 구성해야 한다는 것이다.

본 논문에서는, 사용자가 입력한 모든 내용을 로그로 만든 후 해당 로그를 원격에서 모니터링 할 수 있는 도구를 설계하는 것을 목표로 한다. 2장에서는 관련연구에 대해 서술하고, 3장은 원격 키스트로크 모니터링의 설계에 대해서 서술하였다. 끝으로 4장에서 결론을 맺는다.

2. 관련 연구

2.1 허니팟(Honeypot)

허니팟이란 원하는 시스템을 구축하고, 네트워크에 연결하여 공격자를 기다리는 것이다. 공격자는 허니팟 시스템을 조사하고, 공격할 것이다. 공격자의 모든 행동이 끝난다면, 그 행동을 추적해나가는 것이 허니팟이다.[1]

허니팟은 높은 상호작용을 하는 허니팟과 낮은 상호작용을 하는 허니팟으로 나누어진다. 높은 상호작용을 하는 허니팟은 시스템의 모든 서비스를 가상으로 만들어 공격자의 움직임을 파악하는 것이고, 낮은 상호작용을 하는 허니팟은 기본적인 시스템을 구축해놓고, 침해당한 시스템의 내용으로 공격자를 추적하는 것이다.

낮은 상호작용을 하는 허니팟에서 공격자의 행동을 추적하기 위해서는 기본적으로 방화벽, 로그 시스템 등을 구축해 놓아야 한다. 방화벽 로그는 공격자가 시스템에 침입하기 전에 무엇을 조사하였는지 알 수 있으며, 시스템 로그가 지워졌을 경우 방화벽 로그를 통해 공격자를 추적할 수 있다. 여기서 말하는 로그 시스템은 공격자가 허니팟 시스템에서 어떤 행동을 하고, 어떤 프로세스를 사용하였는지 알 수 있게 해주는 시스템을 말한다. 하지만 공격자가 어떤 시스템이든지 침입하고 나서 바로 로그 시스템부터 파괴하기 때문에 독립적인 로그 시스템을 구축하여야만 한다.[2]

2.2 리눅스 키보드 드라이버의 구조

리눅스 키보드 드라이버는 [그림 1]과 같이, 키보드를 눌렀을 때, 스캔 코드를 키보드 드라이버에 전송하는 기능을 수행한다. handle_scancode() 함수는 스캔 코드를

분석하고, kbd_translate() 함수를 거쳐 translation -table을 사용하여 키 코드를 호출한다. 키 코드는 적절한 키 맵에서 키 심볼로 변환한다. 위와 같은 핸들링을 한 다음, 문자를 얻어 tty queue에 넣는다. 그런 다음, 프로세스가 유저의 입력을 원할 때 프로세스의 stdin은 read() 함수를 호출하는데 sys_read()는 입력문자를 읽어 적절한 tty를 통해 프로세스에게 문자를 전달한다.[3]

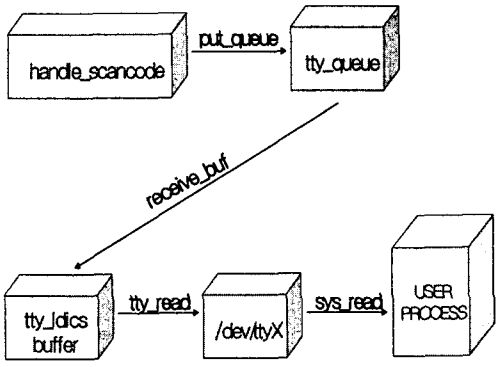

그림 1 리눅스 키보드 드라이버의 구조

2.3 커널 기반의 키로거(Kernel based Keylogger)

키로거는 접속자가 입력한 모든 키보드 내용을 기록하는 도구다. 최근에 커널 기반의 키로거를 제작하는 기법이 등장하였다. 커널 기반의 키로거는 두가지 기법으로 나누어 진다. sys_read와 sys_write 시스템 콜을 가로채는 기법과 tty 버퍼 프로세싱 함수를 가로채는 기법이다. 하지만 시스템 콜을 가로채는 기법을 이용할 경우 시스템이 느려지거나 불안정해지기 때문에 tty 버퍼 프로세싱 함수를 가로채는 기법이 더 좋은 방법이다.[4]

커널 기반의 키로거는 인터럽트 핸들러와 입력 프로세싱 함수를 가로채는 방법을 이용한다.

인터럽트 핸들러는 scancode와 키보드 상태를 읽는데, 이러한 키보드 이벤트는 0x60(Keyboard data register)와 0x64(Keyboard status register)를 통해 읽고 쓸 수 있다. 하지만 인터럽트 핸들러는 플랫폼에 의존적이기 때문에 조심해야 한다.

함수를 가로채는 방법은 위 그림 1에서 나타낸 5개의 함수(handle_scancode(), put_queue(), receive_buf(), tty_read(), sys_read())를 가로채는 것이다. 이 방법을 통해 X나 콘솔의 키스트로크 로깅도 가능하다. 또한 receive_buf() 함수를 가로채 로컬과 원격 세션을 동시에 로깅 할 수 있다. 그리고 커널에서 tty_struct와 tty_queue 구조체는 tty가 오픈될 때마다 동적으로 할당되기 때문에 tty나 pty가 호출 될 때 receive_buf()함수를 동적으로 호출하여 sys_open 시스템 콜을 가로채는 방법을 이용하였다.

아래 표는 위의 방법을 이용하여 구현한 키로거의 기능이다.

- tty와 pts를 통해 로컬과 원격 세션을 동시에 기록.
- tty/session이 독립적으로 로깅.
- 키보드의 특수 키를 모두 지원.
- 라인편집키 지원.
- 타임스탬프 로깅.
- 다양한 로깅 모드

표 1 커널 기반의 키스트로크 기능

2.4 Linux Kernel Module (Process, File Hiding)

키로거가 공격자에게 발견이 된다면 당연히 제거된다. 또한 로그 파일이 공격자에게 발견된다면 먼저 파일을 삭제 하거나 공격자에 대한 로그를 지운다. 하지만 공격자가 발견하지 못하도록 프로세스와 파일을 은닉할 수 있다. 이 기법은 LKM(Linux kernel Module)기법을 이용하는 것이다. 이 기법은 백도어(Backdoor)에 자주 쓰이는 방법이다.

커널의 구조는 단일커널(monolithic kernel)과 마이크로커널(micro Kernel)의 형태가 있다.

단일 커널은 커널의 모든 기능적인 요소들이 자신의 내부 자료구조와 함수들에 모두 접근할 수 있는 하나의 거대한 프로그램이지만 마이크로 커널은 커널의 각 기능적인 부분들이 별도의 단위로 쪼개지고, 그 사이에 엄격한 통신 매커니즘으로 연결되어 있는 구조를 가지고 있다. 마이크로 커널 구조는 커널 자체가 적고 컴팩트하기 때문에 필요시 언제라도 커널 모듈의 로드와 언로드가 가능하다.

이러한 LKM은 시스템이 부팅된 후 언제라도 커널에 동적으로 링크를 시킬 수 있으며, 로드된 모듈은 다른 보통 커널 코드처럼 커널의 한 부분이 된다.

이러한 LKM은 리눅스, FreeBSD, 솔라리스 등 많은 유닉스 시스템에서 그 기능이 제공되고 있다.
모듈은 커널코드와 똑같은 권한과 책임을 진다. 다르게 말하면, 유닉스 커널 모듈은 모든 커널코드나 디바이스 드라이버 커널을 망가뜨릴 수도 있다.

lsmod : 커널에 현재 로드되어 있는 모듈 나열 insmod : 지정한 모듈을 커널로 삽입 rmmod : 지정한 모듈을 커널로부터 제거

표 2 리눅스 시스템에서 커널 로딩과 관련된 명령어

프로세스는 /proc 아래에 process pid로 구성된 디렉토리를 가진다. 커널이 각 프로세스를 다룰 때, proc안의 모든 디렉토리를 검색하여 해당하는 pid를 task_struct로 바꾸어 process를 컨트롤한다. 여기서 process를 리스팅하고자 할 때, getdents() 함수를 가로챈 다음, pid로부터 메모리상의 task_struct를 얻어 키로거 프로세스 이름과 task 이름을 비교하여 일치하면 다음 엔트리로 점프하게 한다.

로그 파일을 은닉하기 위해서는 open() 함수를 가로채서 로그 파일 이름이 검출되면, 파일이 존재 하지 않는

다는 에러를 발생시켜 프로세스를 중단시킨다.

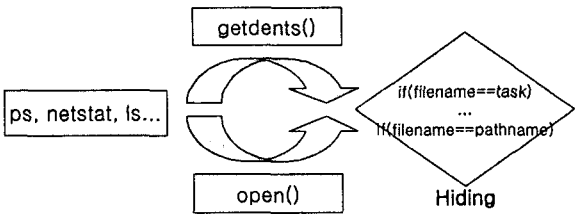

그림 2 프로세스와 로그파일의 은닉

3. 원격 키스트로크 로그 시스템

3.1 원격 키스트로킹 모니터링의 문제점

현재까지 존재하는 허니팟의 가장 큰 문제점은 공격자의 키로거 발견시에 일어날 수 있는 부분에 대한 취약점일 것이다.

허니팟 시스템에 구축되어 있는 키로거는 리눅스 커널 모듈(LKM) 기법을 이용하여 프로세스와 파일을 감출 수 있다. 하지만 리눅스 커널 모듈에 의해 은닉되어진 프로세스와 파일일지라도 최근 공격자들의 성향에 대한 자료를 살펴보았을 경우, 다수의 프로세스 삭제에 대한 패턴을 보이고 있다.

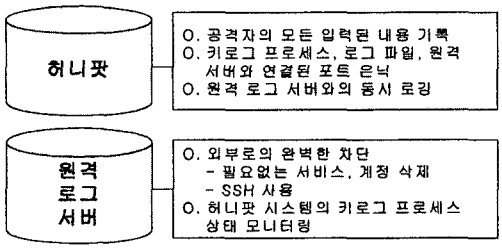

그림 3 허니팟과 원격 로그 서버

만일 공격자가 허니팟의 키로거를 발견한 후, 키로거 프로세스와 로그 파일을 삭제하게 된다면 허니팟에 대한 존재의미가 없어지게 된다.

이것을 방지하기 위해 키로거 프로세스와 로그 파일은 반드시 감추어 져야 할 필요가 있다. 또한 원격 서버를 구성하여 키스트로크 로그를 동시에 저장해야할 필요성이 있다. 즉, 공격자가 허니팟 시스템의 모든 데이터를 제거하는 경우에 대한 최선의 보안책이 가능해지게 된다는 것이다. 또한 원격 키로거 프로세스를 통해 관리자가 키로그에 대한 정보를 모니터링 함으로써 현재 공격자가 하고 있는 행동에 대한 모니터링을 통해 공격자의 추적에 상당히 높은 가능성을 보여주게 된다.

원격 키스트로크 로그 시스템은 해당하는 시스템의 로그 파일에 대한 보안을 위해 외부로부터 완벽한 차단이 요구된다. 이것은 공격자의 로그시스템에 대한 침입으로부터의 보안을 위해서이다. 그러므로 필요 없는 모든 서비스(xinetd, RPC 서비스 등)를 종료시켜야 하며, 필요없는 계정은 삭제해야 한다. 또한 SSH(Secure Shell)을 설치함으로써 스니핑(sniffing)을 방지해야 한다.

3.2 원격 키스트로크 모니터링의 설계

3.1절에서 언급한 현존하는 허니팟의 문제점은 본 논문의 원격 키스트로크 모니터링 시스템을 이용해서 해결할 수 있는 방법을 제시하였다.

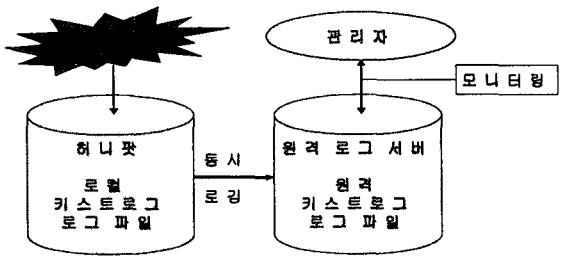

그림 4 원격 키스트로크 모니터링의 구조

그림 4와 같이 공격자는 허니팟 시스템을 조사, 침입하여 목적을 위한 행동을 하게 된다. 공격자의 모든 행동은 허니팟 시스템과 원격 로그 서버에 동시에 기록 하게 된다. 허니팟 시스템의 로그 파일은 공격자에 의해 변조, 삭제 당할 위험이 있으므로 신빙성이 높은 정보가 아니다. 그렇기 때문에 관리자는 원격 로그 서버의 키스트로크 로그 파일을 모니터링 함으로써 신뢰 있는 정보를 얻을 수 있다.

허니팟의 최종목적은 공격자의 공격패턴 분석이다. 허니팟은 허니팟 시스템에 접속하는 것 자체가 공격의 일부이기 때문에 키스트로크를 모니터링 하는 것은 공격자의 공격 패턴을 분석하는데 큰 도움이 될 것이다.

4. 결론 및 향후 연구

허니팟을 구축하는 가장 큰 목적은 공격자의 공격기법에 관한 정보를 얻고, 이것에 대해 연구하는 것이다. 즉, 허니팟 시스템의 가장 큰 목적을 위해서는 공격자의 행동 패턴 분석은 큰 화두이다. 본 논문에서는 공격자의 행동을 쉽고 빠르게 분석할 수 있도록 원격 키스트로크 모니터링을 설계하였다. 이를 통해 관리자는 시스템에 대한 관리 및 감시를 더욱 안전하게 수행할 수 있게 된다.

5.참고문헌

[1] How to Build A Honeypot, Lance Spitzner, 2002.06.09

[2] Honeypots, Lance Spitzner, 2003.05.29

[3] The Linux Keyboard driver, Andries Brouwer,1995.06.01

[4] Writing Linux Kernel Keylogger, rd, 2002.06.19

[5] Complete Linux Loadable Kernel Modules, pragmatic, 1993.03

[6] Honeypots: Tracking Hackers, Lance Spitzner, 2002-09-01