미행에 의한 부정 행위자 신분확인+

김 상욱*, 박 보석*, 장 희진*, 김 건우*, 박 정현**, 임 채호**
경북대학교 컴퓨터과학과 컴퓨터미디어 연구실*, 한국정보보호센터**

Illegal User Identification Using Trace Mechanism+

Sangwook Kim*, Boseok Park*, Heejin Jang*, Gunwoo Kim*, Junghyun Park, Cheho Lim****
Computer media Laboratory Computer Science Department, Kyungpook National University*,
Korea Information Security Agency**
Email: {swkim, parkbs, janghj}@cs.kyungpook.ac.kr*, {parkjh, chlim}@kisa.or.kr**

요 약

미래의 정보전에서는 자신의 정보 시스템에 대한 침해 방지 ,복구 등의 수동적인 형태의 보호뿐 아니라 상대방의 정보 하부 구조(Information Infrastructure)에 대한 공격같은 적극적인 형태의 보호가 요구된다. 기존의 시스템/네트워크 보안 도구만으로 앞으로의 정보전에 대비하는 것은 불가능하다. 그러므로 침입 탐지뿐 아니라 부정 행위자 신분 확인을 위한 구금, 미행 등의 진보된 해킹 방지 기술이 요구된다. 본 논문에서는 미행에 의한 부정 행위자 신분 확인 시스템을 제시한다. 신분 확인 시스템은 부정 행위자에 대한 신분정보를 생성, 유지함으로써 침해 사고를 예방하고 부정 행위자에 대한 역공격도 가능하게 한다. 또한 부정 행위자의 접속 경로에 따른 미행을 통하여 침입자의 행위를 실시간으로 분석, 보고함으로써 보다 정확한 신분 확인이 가능하다.

1. 서 론

차세대 정보전에서 상대방으로부터 자신의 중요한 정보 시스템을 보호하기 위한 새로운 보호 기술이 요구된다. 시스템/ 네트워크 상에서 불법적인 행위를 하는 부정 행위자에 대한 신분 정보를 생성, 유지함으로써 침해 행위의 예방뿐만 아니라 부정 행위자에 대한 공격에 의한 보호가 가능하다. 이러한 신분 정보는 부정 행위자의 구금과 미행을 통해 획득할 수 있다. 본 논문에서는 구금, 미행 기술 등을 이용하여 부정 행위자 신분 확인을 수행하는 부정 행위자 신분 확인 시스템을 제시한다.

부정 행위로부터 주요한 정보 시스템을 보호하기 위해 다양한 보안 도구가 개발되었다. 그러나 취약점 점검, 침입 탐지, 방화벽 등의 보안 도구는 수동적인 보호는 제공하지만 능동적인 대처를 수행할 수는 없다. 그러므로 네트워크/호스트 레벨에서 지원되는 보안 도구와 해커트랩 소프트웨어가 동시에 제공되어야 한다. 호스트 레벨의 모니터링 도구의 예는 Tripwire[1]와 ttywatcher[2]가 있다. IP-watcher[3], Tcpdump[4], Netlog[5], SNIF는 네트워크 레벨의 모니터링 도구의 예이다. 이러한 도구는 사용자 행위를 감시하고 그들의 행위를 로그한다는 면에서 모니터링 도구일 뿐만 아니라 트랩이라 할 수있다. 그러나 단지 이들 시스템을 사용하는 것만으로는 부정 행위자 신분 확인을 위한 시간 확보와 미행, 시스템과 모니터링 행위의 보호가 불가능하다.

본 논문에서 제시하는 시스템은 부정 행위자에 대한 가상 환경을 제공함으로써 실제 시스템을 보호하고 역추적을 위한 시간을 확보하도록 한다.또한 부정 행위자 침입 경로와 악의적 행위에 대한 증거보존은 가능한 침입을 막고 오랜 기간에 걸친 습관과 이동경로의 분석은 해킹 시나리오 데이터베이스 생성을 가능하게 한다.

2. 신분확인 시스템

부정 행위자에 대한 신분 정보를 획득하기 위한 기본 전제는 부정 행위자의 호스트에 직접 접근 가능해야 한다. 그러나 기존의 역추적을 이용한 신분 확인은 모든 호스트에 역추적 모듈이 있어야 하는 제한이 있다. 따라서 역추적뿐만 아니라, 부정행위자의 접속 경로를 미행하여 호스트들에 직접 접근함으로써 보다 정확한 신분 정보 및 증거를 수집할 수 있다.

신분확인 시스템은 가상 잠복 모드, 미행 모드, 복제 모드, 탐문 모드로 동작된다. 침입 탐지 시스템에 의해 부정 행위자로 판명될 경우, 가상 잠복 모드로 전환하여 부정 행위자가 가상의 공간에서 해킹하는 동안 부정행위자의 모든 행위를 감시하면서 역추적한다. 부정행위자가 다른 호스트로 이동할 때 미행 모드에서 부정행위자의 기본

+ 본 연구는 한국정보보호센터 '99 국책과제의 일부임.

인증 정보와 이동 경로를 획득한다. 이들 정보를 기반으로 복제 모드와 탐문 모드로 전환한다. 복제 모드에서는 부정행위자의 이동 경로에 또 다른 거점을 확보하기 위해서 신분확인 시스템을 복제한다. 탐문 모드는 부정행위자에 대한 현재 호스트에서의 신분 정보와 행위 정보를 수집하고, 백도어와 같은 호스트 보안 취약점을 검사한다. 그림 1 은 부정 행위자 신분 확인 시스템의 동작을 나타낸다.

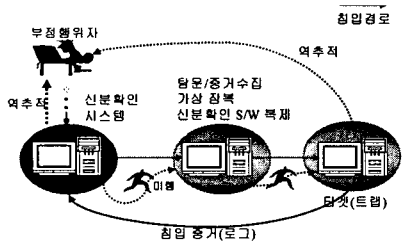

그림 1. 부정 행위자 신분확인 시스템 동작

3. 부정행위자 미행 메커니즘

3.1 가상 잠복

신분 확인 시스템에서 가상 잠복 모드에서는 실시간 침입 탐지 시스템에 의해 부정 행위자에 의한 침입이 감지되었을 경우, 시스템에는 피해없이 부정 행위자의 침입 방법, 침입 패턴, 침입 경로 등의 부정 행위자의 신분 확인을 위한 정보를 획득한다. 그림 2 는 가상 잠복 모드의 개요이다.

침입이 탐지되었을 경우, 신분 확인 시스템은 가상 잠복 모드로 전환된다. 가상 잠복 컴포넌트는 크게 가상쉘과 쉐도우 운영체제로 구성된다.

가상쉘은 부정행위자에게 가상 환경을 제공하기 위한 커널과 사용자 간의 인터페이스이다. 응용프로그램 레벨에서 커널을 제어하는 것이 불가능하므로 커널에 대한 입출력이 가상쉘에서 필터되고 변경된다. 필터된 명령이 시스템에 영향을 주는 시스템 명령인 경우 쉐도우 OS 를 통해 커널에 대해 수행함으로써 시스템에 피해를 주지 않으면서 부정 행위자에 대한 로그를 확보하고 또한 부정 행위자 신분 확인을 위한 시간을 얻을 수 있다. 가상쉘은 기존의 쉘을 캡슐화한다. 쉐도우 OS 는 ASCII 코드 또는 바이너리 파일의 실행 요청에 대해 가상 파일 시스템을 지원한다. 부정행위자의 파일 컴파일 또는 실행과 같은 사용자 명령을 커널로 직접 전달하기 전에 쉐도우 OS 는 텍스트 파일의 내용 또는 바이너리 파일의 경로를 변경한다. 지정된 디렉토리 이상의 디렉토리에 대한 접근을 거부하는 chroot()에 의해 부정행위자의 실제 파일 시스템으로의 접근을 막고 가상 파일 시스템을 제공한다. 이런 메커니즘을 통해 실제 파일 시스템은 보호된다.

3.2 미행

부정 행위자로 의심되는 사용자의 신분 확인을 위해 역추적을 통해 침입 경로를 알아낼 뿐 아니라 침입자임을 확신할 수 있는 증거 수집과 타겟 호스트에 대한 피해를 최소화하기 위해 사용자의 행위를 추적할 필요가 있다. 부정 행위자로 의심되는 사용자가 신분확인 소프트웨어가 있는 호스트를 중간거점으로 하여 다른 호스트에 침입하는 경우 그 경로를 추적하기 위해 미행한다. 미행의 결과 부정 행위자의 이동 경로와 행위 로그를 신분확인 시스템이 획득하게 된다.

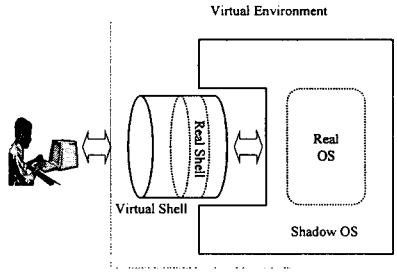

그림 2. 가상 잠복 모드 개요

원격 모니터를 통해 터미널을 모니터링한다. 로드가능한 커널 모듈과 두 개의 스트림 디바이스로 구성된다. 원격 모니터는 키보드 또는 마우스, ttyp*, 콘솔로부터의 상향 스트림(upstream)을 복제하여 하나의 스트림 디바이스로 보내고 ttyp*, console 로부터의 하향 스트림(downstream)을 복제하여 또 다른 스트림 디바이스로 보낸다. 터미널로부터 획득된 정보들은 관리자가 미리 설정한 정책에 따라 행위 필터(Activity Filter)를 거쳐 원하는 행위만 추출되어 행위 분석기를 한다. 추출되는 행위로는 telnet 등을 이용한 다른 호스트로의 이동, 부정 행위의 수행, 새로운 트랩의 생성, 악의적 프로세스의 생성 등이 있다. 미행을 위한 컴포넌트의 외부의 자기 보호를 위해 부정 행위자가 인식하지 못하는 사이에 부정 행위자 미행할 수 있도록 한다.

3.3 복제

신분확인 시스템이 이미 설치된 호스트에 대해 터미널을 사용하지 않을 경우 부정 행위자에 대한 더 이상의 추적은 불가능하다. 정확한 신분 확인을 위해 더 많은 정보를 획득할 필요가 있다. 이를 위해 또 다른 신분 확인의 거점을 확보할 필요가 있다. 미행의 결과 획득된 인증 정보로써 루트 권한 획득이 가능한 경우 부정 행위자 행위를 추적하기 위한 잠복모드 환경설정과 부정 행위자에 대한 위한 탐문모드 즉 다음 미행을 위한 환경설정은 신분확인 시스템의 모드를 복제함으로써 쉽게 이루어진다. 획득된 정보로써 루트 권한 획득이 불가능한 경우는 타겟 시스템내의 취약점을 이용하여 일반 사용자 또는 루트 권한을 획득한 후 신분 정보 획득을 위한 환경을 설정한다.

3.4 탐문 및 역추적

탐문은 미행한 시스템에서 부정행위자의 신분확인을

위해서 부정행위자가 미리 설치한 백도어 및 다른 시스템에서 불법적으로 획득한 파일들을 탐문한다. 역백도어는 신분확인 시스템이 부정행위자의 정보를 비동기적으로 원격에서 감시하기 위한 공격용 백도어이다.

무결한 시스템의 중요 시스템 파일들(/bin/login, /usr/etc/in.telnetd, /usr/etc/in.ftpd, /usr/etc/in.tftpd, /usr/ucb/netstat 등)을 비교하여 백도어의 유무를 검사한다. 그리고 파일의 이름은 다르지만 메시지 다이제스트가 일치하는 파일을 검색하여 백도어가 존재하는지 검사한다. 그 외의 시스템 변화는 디렉토리 접근시간과 변경시간으로 백도어를 검색한다.

이것은 부정행위자가 현재의 모든 세션을 닫고 다른 경로를 통하여 우회 침입했을 때 부정행위자 행위를 감시하기 위해 공격용 백도어를 설치한다. 부정행위자를 속이기 위해서 트랩은 부정행위자가 세션을 닫을 때 설치된다. 부정행위자가 로그인 했을 때 트랩을 구동하기 위해서는 사용자의 환경 파일을 변경하여 쉘 스크립트가 트랩을 구동하도록 하거나, crontab을 사용하여 주기적으로 트랩을 구동한다. 부정행위자의 패스워드 변경, 백-도어 설치, 루트 권한 획득 등과 같은 이벤트를 감지하여 신분확인 메인 시스템에 덤프한다.

추적은 부정행위자가 다른 시스템에서 임의로 다운로드 받은 파일을 검색한다. 다른 시스템에서 도난된 파일이 있는지 검사하여 침입 증거를 확보한다. 파일의 마지막 서명부를 분석하여 파일 원본이 있던 시스템을 찾는다. 파일의 원본이 있던 시스템의 가상 환경에서 파일에 디지털 서명을 수행한다. 실행 파일일 경우 마지막 부분에 서명부를 임의로 붙일 수 있다. 텍스트 파일일 경우에는 텍스트를 가장한 서명부를 파일의 임의 부분에 붙인다.

역추적은 수동적 역추적과 능동적 역추적으로 구분된다. 수동적 역추적은 지역 네트워크의 패킷을 분석하고 tty 정보를 종합하여 출발지 호스트를 추적한다. 필터링된 패킷의 상호 링크를 분석하여 출발지 호스트의 링크를 알아낸다. 능동적 역추적은 추적용 유인 소프트웨어를 사용한다. 따라서 부정행위자가 유인용 프로그램을 다운 받아 가도록 유인한다. 유인 프로그램들에 추적 모듈을 탑재하여 부정행위자의 정보를 획득한다. 네트워크 자원의 사용이 부정행위자에 노출됨을 방지하기 위해서 유동 포트 할당(dynamic port allocation) 기법을 사용한다.

4. 시나리오 및 구현

부정행위자는 일반적인 단계로 시스템에 침입하므로 다음과 같은 침입 시나리오에서 신분확인 시스템은 부정행위자의 신분정보 및 침입증거를 수집한다.

① 가상환경에 접속 시도: 부정행위자의 유인을 위해서 널 패스워드 계정이나 guest 계정을 열어둔다(가상 잠복).

② 가상 환경의 취약점 스캔 : 버퍼오버플로우 취약을 가지는 서비스, 유인을 위한 suid를 가지는 취약점을 생성한다(가상 잠복, 미행).

③ 실행 파일 작성 : 쉘코드를 만들거나 이미 작성된 침입용 실행파일을 다운 받는다(가상 잠복, 미행).

④ 가상환경의 취약점을 이용하여 루트 권한 획득 : 해킹 프로그램을 실행하거나 잘못 설정된 suid 프로그램을 실행하여 루트 권한을 획득한다(가상 잠복, 미행).

⑤ 백도어 또는 스니퍼 작성 및 인스톨 : 재 침입하기위해서 백도어를 설치하거나 스니퍼를 설치하여 시스템의 인증정보를 획득한다(가상 잠복, 미행, 탐문).

⑤ 다음 호스트에 접속 : 획득한 인증 정보나 서비스 취약점을 이용하여 다른 호스트에 접속한다. 접속이 성공하면 루트 권한을 획득하기 위해서 해킹 프로그램을 작성하거나 다운받아서 실행한다(미행, 복제, 탐문).

신분확인 시스템은 c 언어로 Linux 커널 2.2.x 에서 구현된다. 패킷 수집을 위해서는 pcap 라이브러리를 사용하며, 네트워크에서 데이터 및 제어를 숨기기 위해서 crypt API를 사용한다.

5. 결 론

본 논문에서 미행에 기반한 부정 행위자 신분 확인 시스템을 제안하였다. 부정 행위자에게 가상 환경을 제공함으로써 시스템에 대한 피해를 최소화하고 신분 확인을 위한 시간을 확보할 수 있도록 한다. 또한 부정 행위자 이동 경로를 따라 미행하고 탐문함으로써 보다 정확한 신분 정보를 제공한다. 이는 부정 행위자에 대한 증거 수집, 해킹 시나리오 데이터베이스 생성을 가능하게 하여 침해 행위를 방지한다.

신분확인 시스템 자체의 취약점이 있을 경우 해킹 도구로 악용될 수 있으므로 부정행위자로부터 신분확인 시스템을 보호하기 위한 연구가 필요하다. 그리고 해킹 시나리오가 일정한 틀을 가지지 않으므로 시나리오에 포함되지 않은 행위에 대한 처리가 필요하다.

참고문헌

[1] Kim Gene and Spafford E.H., "The design of a system integrity monitor: Tripwire," *Technical Report CSD-TR-93-071, Dept. of Computer Science, Purdue University*, West Lafayette, Indiana, November 1993.

[2] Russel D. and Gangemi G., Computer Security Basics, O'Reilly & Associates, 1991.

[3] N. Michael, "Monitoring and Controlling Suspicious Activity in Real-time with IP-Watcher," *Proceedings of Annual Computer Security Applications Conference*, December 1995.

[4] McCanne S. and Jacobson V., "The BSD Packet Filter: A New Architecture for User-level Packet Capture," *Proceedings of the 1993 Winter USENIX Conference*, January 1993.

[5] Safford D. and Schales D., "The TAMU Security Package: An Ongoing Response to Internet Intruders in an Academic Environment," *Proceedings of the Fourth UNIX Security Symposium*, October 1993.