리눅스 커널 백도어 침입자 추적대응시스템 설계 및 구현

전 완 근*

요 약

본 논문에서는 리눅스 커널 백도어 침입자에 대한 추적과 커널 백도어의 공격에 대한 대응방법을 다룬다. 해커는 일반적으로 시스템에 침입하여 접속로그파일을 지우거나 위조된 주소정보를 사용하기 때문에 현재 로그기반의 침해사고 분석방법으로는 침입자를 추적하는데 한계가 있다. 이에 대한 해결책으로 DeFor 시스템을 제안한다. 이 시스템은 삭제된 로그복구와 전체 하드디스크 이미지 증거 분석을 통하여 침입자 위치를 추적하고, 신속하게 대응함으로써 해킹 피해를 최소화할 수 있다.

Design and Implementation of the Linux Kernel Backdoor Intruder Tracing-Response System

Wankeun Jeon*

ABSTRACT

This paper is about the method that chases the Linux kernel backdoor intruder and copes with the kernel backdoor attack. We have a limit to trace the hacker with the current log analysing method because the hacker generally removes the log file and use the forge IP information. I propose the solution to solve the problem with the DeFor system. Through the restoration of the deleted log file, analysis of it and full HDD image, promptly quick response, it is possible to trace hacker spot and reduce hacking damage.

Key words : Hacker, Intruder, Linux Kernel backdoor

* 한서대학교 대학원 정보보호공학과

1. 서 론

컴퓨터, 네트워킹 기술의 발전과 인터넷 문화의 보급으로 인해 지식·정보화 사회로의 이전이 급속하게 변하고 있다. 인터넷은 수많은 정보의 공유와 사회·문화·경제 교류의 확대, 편리한 생활 제공 등으로 다방면에서 삶의 질을 높여주고 긍정적인 효과를 미치고 있으나 설계상의 개방적인 특성과 네트워크 프로토콜(TCP/IP Protocol)의 근본적인 문제점 및 정보시스템 자체의 취약성 등으로 인해서 악의적인 불법 침입에 의한 접근 정보의 오용 및 도청, 위조 및 변조 행위들이 이루어지고 있다. 이러한 문제점들은 개인뿐만 아니라 기업과 정부의 정보 시스템 의존 비율이 확대됨에 따라서 국가 기반 구조의 마비까지 생겨나는 사태가 발생되는 등 큰 사회 문제로까지 발전되고 있으며, 전쟁을 위한 도구로까지 이용되고 있다.

인터넷 이용자의 수와 접속된 호스트들의 수가 증가함에 따라 불법적인 침입과 악의적인 의도의 시스템 훼손 시도가 늘고 있다[1, 2].

해킹수법들은 점차 자동화, 지능화, 대중화, 분산화, 대규모화, 고속화, 은닉화, 범죄화되고 있다[2-4]. 또한 해킹공격에는 점차로 간단하고 자동화된 툴들을 제작해서 많은 피해를 입히고 있으며, 다수의 에이전트들로부터의 분산 공격이 이루어지고 있다. 현재 심각한 피해를 입히고 있는 공격들은 웜바이러스 형태를 띠고 있는데, 이러한 웜바이러스들은 단시간 내에 해당 지역이나 국가의 기간망을 마비시킬 수 있는 피해를 줄 수 있다. 유해 코드들은 여러 가지 기법들을 복합한 형태로 지능화되고 있으며 하루에도 수개씩 제작되고 있다. 종류별로는 매크로 바이러스, 트로이목마가 주종을 이루었고, 전자우편으로 자동 발송되는 웜도 급증하고 있는 추세로 나타나고 있다[2-6].

이밖에도 해킹이나 매우 다양한 기능을 지닌 백도어가 제작되고 있다[7]. 백도어는 일반적으로 자동화된 해킹 공격 툴인 루트킷에 대부분 포함되며 시스템 해킹 후 시스템 관리자 모르게 남겨놓는 프로그램이다.

이처럼 시스템 해킹 후 다시 침입을 하기위하여 심어놓는 대표적인 백도어로는 용도를 기준으로 하여 크게 3종류로 분류할 수 있다. 첫째 부류에는 login, inted, rlogind, tcpd, telnet 백도어 등과 같이 원격접근을 위한 것, 둘째 부류에는 w, who ps, top, netstat, ls 백도어 등과 같이 내부사용흔적을 감추기 위한 것, 마지막으로 shell 백도어 등처럼 일반사용자가 쉽게 관리자의 권한 획득을 목적으로 하는 것이다[8].

과거에는 위와 같이 일반적인 응용프로그램 수준의 백도어가 주류어서 tripwire와 같은 무결성 검사 툴로 쉽게 탐지하여 대응할 수 있었다.

그러나 최근 LKM(Loadable Kernel Module) 해킹기법을 이용한 리눅스나 솔라리스용 커널 백도어가 매우 다양하고 교묘히 만들어지고, 한 차원 높은 커널 레벨수준의 백도어가 시스템에 설치되고 있어 기존의 탐지 툴이나 방법을 이용하여 커널 백도어를 찾아내는 것이 점점 어려워지고 있다.

특히, 리눅스 운영체제의 경우에는 소스가 누구나 볼 수 있도록 공개되어 있기 때문에 해커가 마음만 먹으면 언제든지 해킹모듈을 삽입하여 해킹도구로 사용될 수 있는 문제점도 있다[9].

기존의 chkrootkit과 같은 커널 백도어 탐지 툴이 있기는 하였지만 대부분 단순 탐지위주의 툴로 침해사고 이후 또는 시스템 관리자의 시스템 점검용으로 사용하여 침입자를 추적하거나 침입시점의 공격을 실시간으로 탐지하지 못하였다.

뿐만 아니라 또한 피해시스템의 로그 분석위주로 하여 로그가 삭제되었을 경우에는 더 이상의 심층적인 분석이 이루어지지 못하였고 만일 로그가 삭제가 안 되었더라고 잔존하는 로그 내 IP 정보가 위조되었을 경우에는 침입자에 대한

추적을 더 이상 하지 못하는 문제가 있다.

이러한 문제점들을 해결하기 위하여 본 논문에서는 피해시스템에서 삭제된 로그와 자료를 복구하여 시스템을 심층 분석하고 커널 백도어 침입탐지시점의 침입자를 추적 및 대응할 수 있는 시스템을 설계 구현하였다. 개발에 적용된 방법은 수준 높은 해커들이 가장 많이 사용하는 LKM 기법과 컴퓨터 포렌식의 데이터 복구기능을 일부 활용하였다.

본 논문에서는 먼저 역추적시스템의 필요성을 살펴보고 리눅스 커널 백도어의 공격 및 탐지 툴을 분석하여 커널 백도어의 동작원리를 파악하고 최종적으로 탐지 및 대응방법을 도출하고 3장에서는 침입시점에서의 탐지 즉시 침입자를 추적하고 신속하게 대응하여 피해를 최소화할 수 있는 DeFor[1]시스템을 설계 구현하고 4장에서는 시험분석을 통하여 기존의 도구와 비교분석한 후 5장에서 결론을 맺는다.

2. 관련 연구

2.1 역추적시스템

침입차단시스템의 보안 정책은 주로 네트워크 주소와 프로토콜 정보들로만 정책을 수립하고 있기 때문에 상세한 차단이 어려워 많은 우회기법을 갖고 있으며 침입탐지시스템의 경우도 정해 놓지 않은 패턴을 벗어나는 경우도 많고 오류탐지가 많았다. 일반적으로 침입차단 시스템은 패킷이 시스템을 통과하는 동안 패킷을 일일이 검사하게 되어 있어 대규모 데이터센터의 경우 심각한 성능저하가 발생하기도 한다. 또 통상적으로 데이터 경로에서 각 패킷을 검사하기 때문에 데이터 처리량을 감소시킬 뿐 아니라 각 네트워크의 오류 지점이 될 위험이 있다. 그리고 침입차단시스템은 접근제어 개념의 보안제품으로 세부적인 침입에 대한 방어가 어렵고, 침입탐지시스템과 같이 탐지를 위주로 하는 수동형 보안 시스템은 복구에 드는 시간과 그 비용이 존재하고, TCP Reset과 Firewall signaling으로 대응 처리를 하고 있지만 공격을 정지시킬 수 없다는 문제점들을 안고 있다[10].

즉 현재까지 개발되어 사용되고 있는 대부분의 보안 강화 시스템들은 해커의 해킹 시도 자체를 제한하는 것이 아니라, 해커가 해킹을 시도하는 경우, 이를 조금 더 어렵게 만드는 수준에 지나지 않았다. 해커의 해킹 시도를 능동적으로 대처하지 못하고, 수동적으로 방어하는 수준인 것이다. 또한 인터넷상에 동작하는 여러 보안 강화 시스템들은 매우 다양하기 때문에 해커의 해킹에 대한 상호 협력을 통한 대응이 거의 불가능한 상황이다. 이와 같은 현재의 보안 시스템 환경 때문에 해킹 시도는 날로 증가하고 있고, 이를 효과적으로 방어하지 못하는 것이 현실이다.

이와 같은 문제점을 해결하기 위해 해커의 해킹 시도 자체를 제한할 수 있는 능동적인 해킹 방지 시스템을 개발하고자 하는 노력이 시도되었고, 능동적인 해킹 방지를 위해 가장 시급한 것이 바로 역추적 시스템임을 주지하게 되었다[11]. 이에 새로운 개념의 추적모델을 제안하였다.

2.2 커널 백도어 공격 및 탐지 기법 연구

커널 백도어는 일련의 rootkit 기능을 가진 코드가 커널 내부에 설치되는 형태를 취하고 있다. 리눅스 커널 백도어로 많이 사용되고 있는 Knark, Adore, Rkit, RIAL 같은 툴을 분석한 결과, 이 툴들은 주로 시스템 콜 테이블, 프로토콜 핸들러, 커널 모듈 리스트 변경과 같은 기법을 사용하였다. 그리고 특정 프로세스, 파일, 네트워크 정보들을 숨기기 위해 커널 백도어는 좀 더 하위 레벨의 기법을 사용한다. 또한 커널 백도어 탐지 툴로 많이 사용되고 있는 Chkrootkit, Kstat, Car-

1) DeFor = Detection + Forensics의 합성어

bonite, StMichael의 기능을 분석해보면 실제 커널 백도어의 공격기능 중 일부만 탐지할 뿐 모든 기능을 탐지하지 못하였다. 이에 따라 원격지에서 간단히 패킷만을 전송하여 root 권한의 쉘을 생성시키거나 특정 프로그램을 실행시킬 수 있는 커널 백도어 공격과 네트워크 프로토콜 핸들러에 대한 변조 등을 추가로 탐지하고 추적할 수 있는 시스템 개발이 필요하다.

3. 리눅스 커널 백도어 침입자 추적대응시스템 설계 및 구현

이 장에서는 커널 백도어 공격 탐지시점의 침입증거를 확보하여 침입자를 추적하고 신속하게 대응하는 시스템을 설계 구현하였다. DeFor 시스템은 커널 백도어 침입탐지, 백업 및 분석기능의 포렌식시스템, 예방 및 복원 기능을 수행하는 대응시스템으로 구성된다. DeFor 시스템은 해커가 커널 백도어 공격을 실시하는 즉시 백업 및 분석 시스템과 대응시스템이 가동되어 공격위치를 추적하고 공격 형태에 맞게 신속하게 대응한다.

3.1 침입탐지시스템

커널 백도어 침입탐지시스템은 2장에서 연구한 결과를 토대로 커널백도어의 공격기법인 시스템 콜 테이블, 프로토콜 핸들러, 커널 모듈 리스트의 변경 시도 등을 탐지하고, 그 시점에서 이미지 백업을 원격 요청한다. 그리고 침입탐지가 감지되면 침입시도상황을 시스템관리자나 수사기관에게 연락 또는 신고가 가능하도록 메일을 사용한다. 침입탐지시스템에서는 침입시점에서의 커널 백도어 탐지가 무엇보다도 중요하다. 해커의 공격은 일반적으로 최초 공격 전후로 하여 집중적으로 이루어진다. 이에 따라 공격자의 근원지를 추적하는데 침입시도 당시의 네트워크 접속 상황과 시스템 로그인 정보 등은 매우 결정적인 증거이다.

침입탐지시스템의 세부 실행단계는 다음과 같다.

1단계 : 시스템 표준모듈과 시스템 콜 테이블목록을 침입탐지시스템에 등록한다.
2단계 : 침입탐지시스템을 실행시켜 커널 백도어 공격을 감시한다.
3단계 : 침입탐지시스템에 커널 백도어 공격이 탐지되면 공격유형을 파악한다.
4단계 : 특정모듈 로딩 시도할 경우 만일 모듈이 허가되어있지 않았으면 8, 9, 10단계를 수행한다.
5단계 : 공격유형이 시스템 콜 변경일 경우 시스템 콜을 후킹하여 비정상적 시스템 콜이면 8, 9, 10단계를 수행한다.
6단계 : 공격유형이 모듈 리스트 변경시도일 경우면 8, 9, 10단계를 수행한다.
7단계 : 공격유형이 프로토콜 핸들러 변경일 경우면 8, 9, 10단계를 수행한다.
8단계 : 공격탐지시간정보를 수집한다.
9단계 : 백업시스템에 원격백업을 요청한다.
10단계 : 시스템 관리자에게 메시지 전송한다.

3.2 백업(정보수집)시스템

커널 백도어 침해탐지시스템으로부터 백업요청이 오면 바로 피해시스템에 대한 정보수집에 들어가게 되는데 시스템, 프로세스, 네트워크 연결정보와 같은 휘발성정보 정보추출을 위한 1차 백업과 상세정보추출을 위한 하드디스크 전체를 백업하는 2차 백업과정으로 나눈다.

1차 백업과정에서는 시스템명령어를 이용하여 커널 백도어 공격 탐지시점 당시의 시스템 및 응용프로그램상의 현재 실행중인 프로세스 및 네트워크 연결 정보 등을 수집한다. 이때 커널에서 직접 원천적인 데이터를 추출하는 방법을 사용하여 변조되지 않은 정보를 수집함으로서 침입자가 피해시스템에 설치한 루트킷과 같은 악

성프로그램에 의해 변조된 정보를 가져오게 될 가능성을 줄인다. 그리고 1차 수집된 자료를 분석시스템으로 보내진다.

2차 백업에서는 피해시스템의 증거를 훼손하지 않고 복사된 정보를 분석하기 위하여 피해시스템의 하드 디스크 전체에 대하여 파티션 별로 분석시스템으로 복사한다. 백업된 전체 이미지를 분석시스템으로 전송하기 전에 변조유무를 확인을 위해서 MD5 해쉬값을 부여한다.

백업시스템에서는 운영체제정보와 같은 시스템 정보 및 환경 설정 내용, 로그 파일, 데이터 디렉터리, 기타 응용 프로그램 관련 파일 등이 수집된다.

다음은 백업시스템의 단계별 실행루틴이다.

1단계 : 백업시스템은 침입탐지시스템에서 요청한 원격 백업요청을 수락한다.
2단계 : 백업시스템에서 원격으로 피해시스템에 연결한다.
3단계 : 1차 백업이면, 시스템명령어를 이용한 시스템 및 네트워크 정보수집, 커널레벨 프로세스 및 네트워크 정보수집, 중요 파일 수집후 1차 수집결과를 분석시스템에 전송한다.
4단계 : 2차 백업이면, 전체 HDD 이미지 백업하여 이미지에 해쉬값 부여한 후 분석시스템에 전송한다.
5단계 : 백업을 종료한다.

3.3 분석시스템

분석시스템은 해킹 침입 및 공격 받은 시스템을 분석하고 피해시스템에 남겨진 증거들을 기반으로 공격방법을 추론하고 공격자의 위치를 추적한다. 분석시스템에서는 백업시스템으로부터 이송된 증거파일에 대한 분석에 앞서 증거물의 무결성을 확보하기 위해 해쉬알고리듬을 이용하여 원본 디스크이미지와 디스크이미지의 해쉬값이 일치하는 지를 간단히 확인한다. 무결성 검사과정을 거친 후 복구와 1차 및 2차 분석을 하게 된다.

1차 분석에서는 주로 시스템, 네트워크, 휘발성 정보 등을 주로 분석한다. 분석은 시스템의 현재 실행 프로세스 및 네트워크 연결정보와 커널 레벨 프로세스 및 네트워크 연결 정보를 비교하여, 숨겨진 프로세스의 연결된 파일을 찾고 숨겨진 네트워크 연결을 검출하고 침입자의 IP 주소를 파악한다.

2차 분석은 정밀분석으로 하드디스크 이미지나 복구된 로그파일로부터 삭제되거나 숨겨진 데이터나 파일이나 디렉터리 정보 등을 추출하여 공격 루트킷이나 백도어 기타 공격에 이용된 S/W 등을 통하여 공격방법이나 공격도구 등을 점검한다.

위 분석을 통하여 파악된 IP 주소는 Whois 조회를 통하여 자동으로 침입자의 위치를 판별하게 된다. 그리고 분석결과는 시스템 관리자에게 전송할 수 있도록 한다.

다음은 분석시스템의 단계별 실행루틴이다.

1단계 : 백업시스템으로부터 전송된 이미지에 대해 무결성 검사를 한다.
2단계 : 분석시스템에서 증거 이미지를 복원한다.
3단계 : 1차 증거자료 분석을 통해 커널 레벨의 프로세스 및 네트워크 정보와 현재 실행중인 시스템 및 네트워크 연결정보를 비교하고, 숨겨진 네트워크와 프로세스 정보를 분석하여 비정상 프로세스와 공격 IP 확인한다.
4단계 : 2차 증거자료 분석을 통해 삭제된 로그 및 파일 복구, 주요 시스템 로그 내 공격 IP, 삭제된 공격도구 등을 확인한다.
5단계 : 공격 IP에 대해 Whois 조회한다.
6단계 : 분석결과를 시스템 관리자에게 전송한다.

3.4 대응(예방 및 복원)시스템

커널 백도어의 대응시스템은 커널 백도어 예방 및 복원 시스템으로 이루어지는데 커널 백도어의 침해되지 않은 시스템에 설치하여 사용할 수 있다. 이 시스템은 커널 백도어 탐지시스템의 알고리듬을 활용한다. 이 시스템은 크게 커널 백도어 설치의 예방, 커널 백도어에 의한 시스템 자원의 변경을 복원하는 기능을 제공한다. 여기서 커널 백도어에 의한 시스템 자원의 변경 복원은 시스템 자원을 원래의 상태로 환원하면서, 커널 백도어의 고유의 기능을 수행하지 못하도록 한다.

커널 백도어 대응 시스템에서의 커널 백도어 탐지모듈의 기능에 탐지해야 될 커널 백도어의 유형이 포함되어있기 때문에 복원방법은 그에 따라 각각 다르게 대응한다. 대응시점은 침입자 추적 및 초기 이미지 분석을 위한 1차 백업(정보수집)이 마무리된 후 대응모듈이 작동한다. 시스템관리자는 신규 커널 백도어의 공격방법이 추가분석이 될 경우에 공격패턴과 대응방법이 추가한다. 즉 허가된 커널 모듈의 등록을 통해 로드 될 수 있는 모듈 리스트를 관리하며, 어떤 커널 모듈 로딩시 일차적으로 리스트를 검색하고 리스트에 존재하지 않는 모듈일 경우 이를 차단한다. 만약 로드 되는 커널 모듈이 시스템의 표준 커널 모듈의 이름과 동일할 경우 로드 되며, 이때는 이 모듈의 행위를 감시함으로써 대응이 가능하다.'로드된 커널 모듈이 시스템 콜 테이블의 값을 변경한다면 2차적으로 변경된 시스템 콜 테이블의 값을 복원한다. 또한 모듈이 시스템이 관리하는, 이미 로드된 커널 모듈의 리스트를 변경하거나 특정 모듈을 모듈 리스트에서 끊는 행위를 한다면 이 또한 복원하여 그러한 행위를 통해 이루어질 모듈 숨김 기능을 방지한다.

커널 백도어 대응시스템은 다음과 같이 모두 7단계 루틴으로 실행된다.

1단계 : 침입대응시스템에 커널 백도어 공격이 탐지되면 공격유형을 파악한다.

2단계 : 특정모듈 로딩 시도할 경우 허가되지 않은 로드 모듈일 경우 6단계를 수행한다.

3단계 : 시스템 콜 변경여부 확인하여 변경되었을 경우에는 7단계를 수행한다.

4단계 : 공격유형이 모듈 리스트 변경여부 확인하여 모듈리스트가 변경되었을 경우에는 7단계를 수행한다.

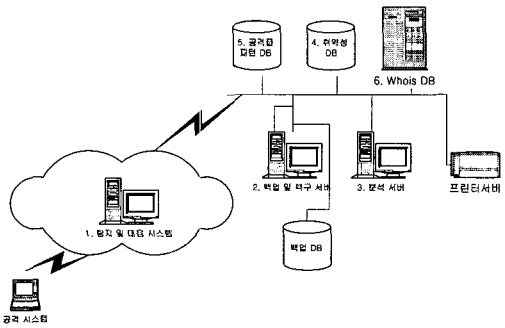
5단계 : 공격유형이 프로토콜 핸들러 변경여부 확인하여 프로토콜 핸들러 변경되었을 경우에는 7단계를 수행한다.

6단계 : 커널로의 모듈 적재를 방지한다.

7단계 : 변경된 부분을 원상태로 복원한다.

4. 시험분석

이번 장에서는 시험분석을 위해 (그림 1)과 같이 X86 리눅스 커널 2.4.2 환경에 탐지 및 대응시스템을 설치하고, 공격시스템에서 공격목표 시스템으로 해킹공격을 하면 침입탐지시스템에서 침입상황을 탐지하여 즉시 해킹피해시스템에서 이미지 백업을 받아 분석 가능한 시스템으로 전송하고 그리고 이와 동시에 침입사실을 시스템 관리자에게 메일 통보하도록 환경을 구축하여 시험하였다.



(그림 1) 시험분석 환경

먼저 시스템관리자는 커널 백도어 침입탐지를 위해 미리 공격 툴에 대한 패턴을 지정하고 커널에 추가될 모듈의 정보를 탐지시스템에 저장하였다. DeFor 시스템에 침입탐지시스템이 설치된 상태에서 커널 백도어 설치시도가 탐지되는 즉시 침입이 시스템 관리자에 메일을 통하여 알려지고 이미징 모듈 원격 동작요청을 하여 1차, 2차 백업(정보수집)시스템이 제대로 동작되는지 시험하였다. 백업(증거수집)시스템은 피해시스템에 커널 백도어로부터 공격이 탐지되는 즉시 네트워크와 프로세스 정보를 확보하고, 인멸 및 손실될 수 있는 잠재된 컴퓨터 범죄의 증거들을 백업하는 이미징 모듈이 동작하여 이때 1차적인 정보만을 수집 후 2차 하드 전체 이미징 작업이 이루어지기 전에 대응모듈이 동작하여 보다 더 빠르게 피해시스템에 대한 복원과 대응을 할 수 있도록 하였다.

분석시스템에서는 복구데이터와 로그파일을 분석하여 침입자를 추적하고 침입에 사용된 공격 도구 등을 잘 찾아내는지 확인하였다.

실제로 DeFor 시스템을 컴퓨터 수사에 적용하여 국내 모 은행 피싱(Phishing)사고와 관련 삭제된 접속로그를 복구 분석한 결과 불법 호스트가 미국 소재지의 개인 사용자 PC의 것으로 하였고, 지워진 tmp 내 파일을 복구하여 분석한 결과 SucKIT을 공격도구로 사용한 것을 확인하였다. 또한 DeFor 시스템은 커널 백도어가 커널 모듈 필터링을 통해 커널 백도어 Knark 에 의한 침입을 원천적으로 봉쇄할 수 있었다.

본 논문에서 구현된 DeFor는 탐지시스템과 백업시스템에서 수집된 정보를 기반으로 복구 및 분석 등 포렌식 절차를 수행한다. 분석시스템은 피해시스템의 휘발성 정보를 LKM을 이용하여 커널에서 수집된 원천 정보와 비교분석한다. 그리고 시스템의 바이너리 파일 무결성 검사 기능을 제공한다. 해커들이 자신의 흔적을 감추거나 백도어 설치를 위해 사용하는 루트킷 검출에 있어 일반 루트킷 뿐만 아니라 커널 루트킷까지 검출한다. 또한 파일시스템의 분석과 지워진 파일의 복구분석이 가능하다. 또한 커널 백도어에 대한 탐지 및 복원기능을 제공한다. <표 1>은 본 논문에서 구현된 DeFor 시스템과 기존의 커널 백도어 탐지 툴의 기능을 비교한 결과이다.

〈표 1〉 기능 비교

구분	Chkrootkit	Kstat	Cabonite	StMachael	DeFor
예방탐지	없음	없음	없음	없음	있음
시스템콜 테이블 변경탐지·복원	있음	있음	없음	있음	있음
프로토콜핸들러 변경탐지·복원	없음	없음	없음	없음	있음
커널 모듈리스트 변경탐지·복원	없음	없음	없음	있음	있음
탐지 신고기능	없음	없음	없음	없음	있음
침입자추적	없음	없음	없음	없음	있음
파일복구 기능	없음	없음	없음	없음	있음
원천휘발성 정보수집기능	없음	없음	없음	없음	있음
무결성 검사기능	없음	없음	없음	없음	있음

5. 결론 및 향후 연구과제

해커의 침입영역은 운영 중인 네트워크 서비스, 운영체제, 응용 프로토콜, 응용프로그램 등이 될 수 있으며 이에 대한 공격방법과 형태는 해커의 능력에 따라 자유롭게 결정한다.

최근 인터넷상의 공개된 리눅스 소스와 해커들의 고난도 프로그래밍 기술로 인하여 나타난 기법 중의 하나가 커널 백도어다. 이와 같이 운영체제의 커널에 백도어를 심는 해킹공격에 기업 내 시스템이 공격을 받을 경우에는 대부분 서버관리자에게 속수무책으로 당할 수밖에 없는 실정이었다.

해킹으로 입은 손해에 대한 책임을 누군가가 감당하여야 하나 현재로서는 커널 레벨의 해킹을 시도한 자를 찾아 검거한다는 것은 거의 불가능하다. 왜냐하면 해커가 추적을 피하기 위해 관련 로그파일에 남아있는 흔적을 완전히 제거

할 수 있기 때문이다.

기존의 해킹피해시스템 분석에 있어서 문제점은 해커에 의해 삭제된 로그파일로 인하여 침입자의 공격방법과 공격 툴, 공격행위 등을 알아낼 수가 없다는 점이다. 삭제된 로그파일은 해킹의 처음부터 중간과정을 거쳐 끝까지 전 과정을 밝혀내는 데에는 중요한 실마리를 제공하는 역할을 한다. 침입탐지시스템이나 침입차단시스템 등도 로그정보를 기반으로 하기 때문에 침입자의 IP 주소를 속여서 공격할 경우에는 침입자 추적에 한계가 있었다.

본 논문에서는 커널 백도어에 대한 공격의 위험성 인식하여 기존의 해킹 피해분석시스템의 문제점을 지니고 있었던 로그기반의 분석기법의 한계와 기존 백도어 탐지 툴이 지니고 있는 단순기능을 넘어 침입탐지 즉시 원천정보를 수집하고, 삭제된 로그나 하드디스크의 데이터를 복구하는 컴퓨터 포렌식 기법을 적용하여 피해시스템을 심층 분석함으로써 고난이도의 해킹기술을 지닌 침입자를 추적하고자 하였다.

본 논문에서 설계 구현한 시스템은 삭제된 로그파일과 디렉터리를 복구하고 복원된 하드디스크의 이미지를 심층 분석하여 침입자의 위치를 빠른 시간에 추적하고 신속하게 대응할 때 피해를 최소화할 수 있다. 따라서 정보수집시간과 하드디스크 이미징 작업시간을 최대한 단축하고 분석시간을 단축하기 위해 분석 알고리즘을 단순화할 필요가 있다. 또한 새로운 유형의 커널 백도어 공격에 대한 탐지방법은 계속 연구되어져야 할 것이다.

참 고 문 헌

[1] 한국전산원 정보화통계 자료, http://stat.nca.or.kr.

[2] 한국정보보호진흥원, 해킹 바이러스 통계 및 분석자료, http://www.krcert.or.kr.

[3] David Dittrich, "Distributed Denial of Service (DDoS) Attacks/tools", http://staff.washington.edu/dittrich/misc/ddos/.

[4] Jose Nazario, "The Future of Internet Worm", http://www.crimelabs.net/docs/worm.html, Jul. 2001.

[5] 한국전자통신연구원, 차세대 해킹 기술 및 네트워크 안정성 분석 연구 보고서, 2001. 12.

[6] 정보통신부, 2003년 통계발표자료, http://www.mic.go.kr/jsp/mic_p/.

[7] Stephen Specht and Ruby B. Lee, Taxonomies of Distributed Denial of Service Attacks, Tools and Countermeasures, Princeton University Department of Electrical Engineering Technical Report CE-L2003-003, May 2003.

[8] 한국정보보호진흥원, 2001 해킹 바이러스 현황 및 분석보고서, http://www.krcert.or.kr.

[9] Buanzo, "Detecting and Understading rootkits, an Introduction and just a little-bit-more", September, 2003. http://www. net-security.org/dl/articles.

[10] http://www.juniper.net/products/intrusion/prevention.html.

[11] Buchholz, Thomas E. Daniels, Benjamin Kuperman, Clay Shields, "Packet Tracker Final Report," CERIAS Technical Report 2000-23, Purdue University, 2000.

전 완 근

1998년 한서대학교 컴퓨터정보학과(이학사)
2000년 한서대학교 전산학과(이학석사)
2005년 한서대학교 정보보호공학과(공학박사)