보안에 강한 리눅스 서버 구축 및 운영

손현민, 김홍기, 주낙근
동신대학교 컴퓨터학과

The Construction and Operation of Linux with Robust Security

Hyun-Min Son, Hong-Gi Kim, Nak-Keun Joo
Dept. of Computer Science, DongShin University

요 약

인터넷에 의한 정보의 공유 및 각종 서비스의 제공이 점차 증가함에 따라 해킹으로 인한 피해 또한 증가하는 추세이다. 최근 많은 사람들이 운영체제로 막강한 기능을 제공하는 리눅스를 서버로 구축함에 따라 리눅스가 네트워크환경에 가장 이상적인 운영체제로 자리잡아 가고 있는 시점에서 리눅스 서버의 운영상의 보안문제가 심각하게 대두되고 있다. 그래서 본 논문에서는 최근 대표적인 해킹기법과 그에 대응하는 보안기법들을 분석해서 보다 안전한 리눅스 서버를 구축하고 운영할 수 있는 방법들을 제시하였다.

1. 서론

오늘날 서버의 운영체제로 유닉스, 윈도우NT, 리눅스 등이 널리 사용되고 있다. 이들 중에서 리눅스는 운영체제의 핵심인 커널을 비롯한 거의 모든 프로그램의 소스가 공개되어 무료로 사용 가능한 운영체제이다. 따라서 세계의 수많은 사람들이 리눅스를 연구하여 안전한 운영체제로 발전시켜 왔다. 그 결과 리눅스를 운영체제로 사용하는 서버가 급속도로 증가하게 되었고 누구나 손쉽게 설치하여 운영할 수 있는 운영체제로 발전하였다. 하지만, 그 사용이 증가함에 따라서 리눅스 서버의 운영상의 보안문제가 심각하게 대두되고 있다. 이는 리눅스 자체의 보안결함 때문이라기보다는 시스템관리자의 관리소홀이나 인터넷 프로토콜 자체의 보안취약점 때문에 발생되는 경우가 대부분이다. 물론 리눅스 자체의 버그에 의한 침입도 있긴 하지만 버그들이 발견된다 할지라도 짧은 시간 내에 패치 버전이 나올 수 있다는 점이 리눅스의 또 하나의 장점이기도 하다. 이 세상에 보안에 완벽한 시스템은 존재하지 않는다. 단지 불법 침입을 어렵게 함으로써 침입시도를 저지하고 침입자를 빨리 발견함으로써 시스템을 안정화시키는데 보안의 목적이 있다고 볼 수 있다.

본 논문에서는 리눅스 기반의 서버 구축시 현재까지 알려진 보안상의 문제점을 최소화시키고 네트워크 서비스를 효율적으로 관리하여 불필요한 서비스에 의한 침입을 차단하기 위한 방법과 현재 가장 많이 쓰이는 해킹기법 및 그에 대한 해결책을 연구함으로써 보다 더 보안에 강한 리눅스 시스템을 운영하는 방법을 제시하고자 한다. 이를 위해 본 논문에서는 먼저 2장에서 리눅스 서버 환경에서의 보안 문제를 살펴본 후 3장에서 대표적인 해킹기법과 보안도구들의 기능에 대해 요약 정리하고 4장에서 보안도구를 이용해서 안전한 리눅스 서버를 구축 및 운영하는 방법과 마지막으로 5장에서 결론을 기술하도록 한다.

* 이 연구는 정보통신부 대학 S/W 연구센터 지원사업 연구비에 의하여 연구되었음

2. 인터넷과 리눅스 서버 환경에서 보안 위협

네트워크상에서 보안을 위협하는 대표적인 기법은 패킷을 감시하고 잡아내는 스니퍼와 시스템 보안상의 허점을 찾아내는 스캐너, 패킷을 믿을 수 있는 호스트에서 보낸 것처럼 위조해서 위조된 호스트로 인증을 받는 스푸핑 기법 등이 있다. 이러한 공격방법들은 대부분 TCP/IP의 보안 취약성을 이용하는데 TCP/IP 자체가 보안에 대해 취약한 이유는 인터넷이 라우터들의 집합이기 때문에 데이터 전송시 송신측과 수신측 이외에 데이터를 라우팅 하는 다른 조직들의 라우터에 의존해서 데이터가 전달되며 각각의 라우터에 대한 제어를 송/수신측에서 관리할 수 없기 때문이다. 이러한 이유로 인터넷구조의 결함이 동반된 리눅스

서버의 보안개념은 광범위한 보안정책을 요구하고 있다.

3. 리눅스 해킹과 보안 도구 분석

최근에 해킹 기법은 더욱 복잡해지고 프로그램화되는 등 자동화되고 있으며 단지 해킹도구 만을 사용하는 것으로도 강력한 해킹을 시도할 수 있다. 따라서 본 장에서는 지금까지 발표된 공격도구를 분석하고 이러한 대표적인 공격기법을 차단할 수 있는 보안도구를 조사함으로써 서버를 보다 안전하게 관리할 수 있도록 한다.

3.1 최근 해킹 기법과 동향

과거의 해킹 동향은 패스워드 크랙이나 알려진 취약점을 통한 루트권한 획득이 주된 공격방법이었으나 최근의 해킹동향은 보다 다양해지고 고난도의 공격유형을 이루고 있다. 특히 야후나 아마존 등에서의 해킹 방법은 인터넷서비스 업체나 전자상거래업체들에게는 심각한 보안위협이 되고 있다. 오늘날에는 과거와 같이 중요한 정보나 자료의 유출보다는 시스템 마비를 유도해서 서비스를 정지시키는 해킹기법이 보다 더 다양하게 전개되고 있다. [1]

3.2 대표적인 해킹 도구

해킹시도의 측면에서 다양한 도구들이 있는데 여기에서는 최근의 대표적인 해킹기법과 도구들에 대해서 기술한다.

3.2.1 스니핑

이더넷 포트를 감시하면서 지나가는 패킷 흐름에서 passwd, login, su와 같은 것이 있으면 그 후의 내용을 저장해두고 이를 통해 암호화되지 않은 패스워드를 획득하는 기법이다. 스니퍼가 이용하는 프로토콜의 취약점은 이더넷상에서 통신하고자하는 두 호스트간에 배타적인 경로가 설정되는 것이 아니라 패킷이 모든 호스트에 전달되는 점이다.

일반적으로 이더넷에서는 패킷의 목적지와 같은 호스트만 그 패킷을 받아들이고 나머지 호스트들은 그 패킷을 무시하는데 스니퍼는 시스템의 네트워크 인터페이스를 무차별 모드(promiscuous mode)로 전환하여, 네트워크 트래픽의 실제 목적지에 관계없이 지나가는 모든 패킷을 감시하고 잡아낼 수 있다.

스니퍼 공격에 대한 방어로는 시스템의 네트워크 인터페이스를 검사하는 방법이 있는데 ifconfig 명령을 사용해서 네트워크 인터페이스가 무차별 모드(promiscuous mode) 상태인 것을 확인해보면 된다.

해결책으로는 패스워드가 평문인 상태로 네트워크를 통해 전달되지 않도록 하는 방법이 있는데 일반적으로 보안쉘(ssh : secure shell) 프로그램을 사용하여 암호화된 상태로 자료가 전송되도록 하는 방법과 패스워드 대신에 일회용 패스워드(s/key)를 이용하는 방법이 있다.[2]

3.2.2 분산 서비스 거부 공격

서비스 거부 공격(DOS)이란 공격자가 한 개의 시스템에서 많은 양의 패킷을 집중적으로 보내서 상대 호스트의 하드웨어나 소프트웨어 등을 무력하게 만들어 호스트에서 적법한 사용자의 서비스 요구를 거부하도록 만드는 일련의 행위를 말하며 분산된 여러 시스템에서 목표시스템을 집중 공격하는 분산서비스거부공격(DDOS : Distributed Denial of Service)은 대규모 네트워크의 많은 호스트에 데몬이 설치되어 서로 통합된 형태로 패킷을 범람시켜 심각한 네트워크 성능저하 및 시스템 마비를 유발시킨다. [3]

3.2.3 스캐너

스캐너(scanner)란 시스템 보안상의 허점을 찾아내는 보안도구로서 개발되었으며 필요에 따라서 다양한 스캐너가 존재하지만 시스템 스캐너와 네트워크 스캐너의 두 가지 범주로 분류할 수 있다.

시스템 스캐너는 주로 관리자가 서버 환경 설정 시 발생할 수 있는 보안상의 결점을 찾기 위해 이용되며 네트워크 스캐너는 이용 가능한 서비스와 포트를 조사함으로써 원격공격자들이 이용할 수 있는 잘 알려진 결함들을 찾아낸다. 이와 같이 스캐너는 보안도구로서의 의미를 가지고 있지만 크래커들에 의해 공격도구로써 사용되기도 한다. [4]

3.2.4. 스푸핑

스푸핑(spoofing)은 서로 신뢰할 수 있는 호스트에서 패킷을 보낸 것처럼 위조해서 인증을 받는 방법으로 호스트명 기반이나 주소기반의 인증방법을 무너뜨리는 모든 방법을 의미한다.

일반적으로 호스트 기반의 네트워크 접근제어방식을 사용하는 보안도구들이 보편화되어 있으나 이러한 도구들이 IP주소를 이용한 인증방법을 사용한다는 점이 TCP/IP 보안상에 심각한 문제점을 야기시키고 있다. 발신지주소 인증방법은 발신지 호스트에서 IP의 발신지 주소를 설정할 수 있고 TCP/IP에서는 패킷이 변경되었는지 판단할 방법이 없다는 결함을 가지고 있으며 스푸핑은 이러한 결함을 이용해서 신뢰할 수 있는 호스트로 위장해서 목적 호스트에 연결하는 기법이다. [5,6]

3.3 해킹방지 및 탐지 도구

본 장에서는 리눅스 서버를 보다 안전하게 구축 및 운영하는데 꼭 필요한 도구들의 종류와 기능을 분석하여 리눅스 시스템 관리자가 보안취약점을 제거하고 해커의 침입을 사전에 방지하기 위한 방법들을 소개한다.

3.3.1 파일시스템 무결성 검사

파일시스템의 무결성을 검사하는 도구로서 대표적인 것이 트립와이어(tripwire)가 있는데 이것은 파일과 디렉토리 무결성(integrity)을 검사하는 도구로서 이전에 만들어진 데이터베이스에 저장된 파일과 디렉토리에 관한 정보와 현재 존재하는 파일과 디렉토리

를 비교하여 다른 점들을 찾아 기록하여 중요한 시스템 파일이나 디렉토리의 변경을 감지하고 해커의 침입을 알아낼 수 있도록 하는 파일시스템 무결성 검사 도구이다. 트립와이어(tripwire)는 서버 구축 후 바로 실행해야 그 이후의 변동사항을 쉽게 파악하고 침입을 발견하는데 용이하게 사용된다. [7]

3.3.2 보안쉘

rlogin이나 rsh와 같은 BSD 서비스들의 보안 취약점을 해결하기 위해 나온 원격 로그인 프로그램으로 패킷을 전송할 때 암호화시켜 전송하므로 스니퍼 같은 해킹 도구를 사용하여 패킷이 도청되더라도 비교적 안전하다.

보안쉘(SSH)은 BlowFish, Triple DES, IDEA, RSA 등의 다중 알고리즘을 지원하기 때문에 보다 유연하고 확장성이 뛰어난 시스템을 만드는데 용이하게 사용된다. [8]

3.3.3 TCP 랩퍼

TCP 랩퍼(wrapper)는 Telnet, ftp, rlogin과 같은 TCP 서비스를 보호해주는 보안 프로그램으로서 각 서비스에 접근제어를 적용할 수 있다. TCP 랩퍼(wrapper)는 inetd가 서버를 호출할 때 tcpd가 먼저 그 요청을 가로채어 연결요구를 평가해서 그 요구가 규칙에 맞는 정당한 요구인지 판단한 후 정당한 요구이면 요구된 서비스를 시작하고 그렇지 않는 경우에는 그 연결요구를 무시한다. 기본적으로 TCP 랩퍼는 /etc/hosts.allow와 /etc/hosts.deny 두 파일을 설정함으로써 승인된 호스트와 승인되지 않은 호스트들을 명시할 수 있다. TCP 랩퍼는 실제 패킷 필터를 사용하지 않고도 방화벽과 비슷한 기능성을 제공하므로 방화벽이 필요 없이 네트워크 접근제어가 필요할 때 유용하게 사용할 수 있다. [9]

3.3.4 세인트

세인트(SAINT)는 인터넷 상에 있는 시스템의 네트워크 서비스를 이용한 해킹공격에 대해 기존에 알려진 보안상의 취약점들이 존재하는지 진단하는 도구이다. 이는 하나의 시스템 또는 임의의 네트워크 상의 모든 시스템에 대해 보안상의 취약점을 진단할 수 있다. 세인트(SAINT)는 최근에 알려진 CGI 기반의 웹 공격, 서비스 거부 공격, POP서버 공격, 보안쉘(SSH)의 허점을 이용한 공격, 원격 버퍼 오버플로 등 거의 모든 취약점을 분석할 수 있는 도구로 사용되고 있다.

일반적으로 취약점 분석도구에는 여러 가지가 있으나 SAINT와 OLSD/SP 도구를 같이 사용하면 대부분의 보안상의 취약점들을 점검할 수 있다. [10]

3.3.5 ipchains

ipchains는 IP방화벽과 리눅스 커널에서 회계(accounting)규칙을 설치하고 유지하며 검사하는데 사용된다. ipchains는 외부에서 들어오는 패킷과 나가는 패킷에 대해 엄격한 규칙을 적용할 수 있으며 다양한 정책을 수행할 수 있도록 제공해준다. 커널은 세 가지의 규칙 목록('방화벽 사슬' 또는 간단히 '사슬'이라고 부른다)을 가지고 시작한다. 이 세 가지 사슬은 '입력(input)', '출력(output)', 그리고 '전달(forward)'이다. 패킷이 들어오면(예를 들어 이더넷 카드를 통해) 커널은 패킷의 운명을 결정짓기 위해 입력(input) 사슬을 사용한다. 만약 이 단계에서 살아남았다면 이번에는 패킷을 어디로 보내야 할지 결정한다. 만약 패킷이 다른 머신으로 가야 한다면 전달(forward)사슬을 살펴보고 마지막으로 패킷이 떠나려 하는 순간 출력(output)사슬을 점검한다. 서버의 환경에 맞게 이 세 가지 사슬 규칙을 정해준다면 방화벽보다 더 훌륭한 기능을 수행할 수 있다. [11]

3.3.6 실시간 불법 침입시도 자동탐지 도구

실시간 불법 침입시도 자동탐지 도구(RTSD : Real Time Scan Detector)는 네트워크 취약점 검색 공격을 실시간으로 탐지하고 대응할 수 있는 도구이며 어떠한 스캔 도구(SAINT, SSCAN, NMAP 등)로 특정 호스트를 공격할 때 RTSD는 그 호스트가 설치되어 있는 네트워크 내에 하나의 호스트에 설치되어 이러한 스캔 공격을 탐지하며 탐지된 스캔 공격은 특정한 메일로 정보가 보내진다.[12]

4. 안전한 리눅스 서버 구축 방법

리눅스 서버 구축 방법에는 여러 가지가 있을 수 있으나 보안에 중점을 둔 서버 구축 시 일반적인 보안 취약성을 고려하고 유용한 보안도구를 사용하여 보안에 강한 리눅스 서버를 구축하고자 한다면 다음과 같은 사항들을 고려해야 한다.

첫째, 리눅스를 설치할 때 옵션에서 불필요한 서비스를 선택하지 않아야 한다. 리눅스 설치시 전부설치(everything)옵션으로 설치했다면 수많은 서비스를 제공하게 된다. 서버 관리자는 서버의 사용용도에 따라 불필요한 서비스를 제거해야 보안 취약점을 감소시킬 수 있다. 서비스가 설치되었다고 하더라도 /etc/inetd.conf 파일에서 불필요한 서비스는 주석처리를 해줌으로써 불필요한 서비스 제공을 막을 수 있다.

둘째, 리눅스는 새 버전이 발표된 후 곧바로 패치 버전이 나오는 경우가 있으므로 보안에 관계되는 버그 패치는 바로 설치해줄 필요가 있다.

셋째, 스니퍼 공격에 의해 패킷이 크래커에게 필터링된다면 평문으로 전달된 암호문의 경우 안정성을 보장할 수 없다. 이를 막기 위해 보안쉘(SSH)을 사용하여 패킷을 암호화하여 설사 패킷이 이더넷 상에서 크래커에게 노출된다 할지라도 패킷의 내용을 보호할 수 있도록 해야 한다.

넷째, 보안에 적극적이지 못한 서버 관리자들은 대부분 크래커의 침입조차 알지 못하는 경우가 많다. 만약 그들이 백도어를 설치해 놓았다면 더 이상 그 서버는 신뢰할 수 없게 된다. 하지만 크래커의 침입을 알아냈다 하더라도 백도어를 찾기란 그리 쉬운 일은 아니므로 트립와이어(tripwire)를 설치하면 중요한 시

스템 파일이나 디렉토리의 변경을 쉽게 찾을 수 있어 크래커의 침입을 쉽게 알아낼 수 있다.

다섯째, 방화벽을 필요로 하지 않는 네트워크 접근 제어를 할 때는 TCP랩퍼(wrappers)를 사용하고 실제로 방화벽 기능성이나 패킷 필터링 같은 기능을 원한다면 ipchains를 사용하도록 한다.

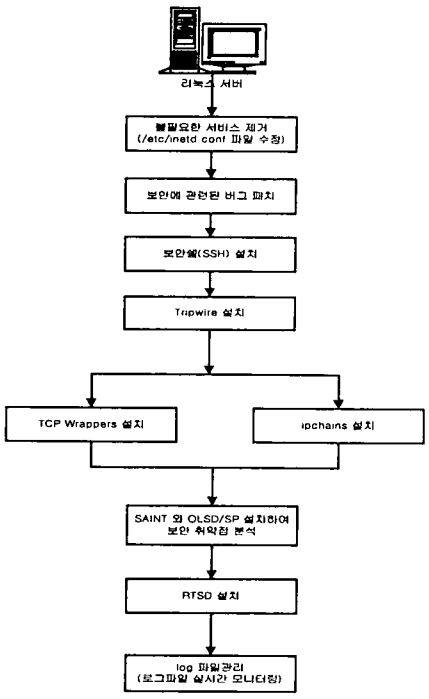
여섯째, 취약점 진단 도구인 세인트(SAINT)와 OLSD/SP를 사용하면 일반적인 리눅스 보안 취약점을 분석할 수 있으며 그에 대한 보안대책의 범위를 줄일 수 있다.

일곱째, 다양한 스캔공격의 예방책으로 RTSD를 설치하면 메일이나 핸드폰을 통해 실시간으로 시스템에 대한 부적절한 접근이 관리자에게 보고가 되므로 그 호스트로부터의 해킹시도를 막을 수 있다.

여덟째, 리눅스 시스템에는 다양한 로그 정보를 가진 파일과 관련 명령어들이 있는데 이들 로그 파일들은 /var/log 디렉토리에 존재하게 된다.
다양한 로그 정보들을 검토함으로써 외부로부터의 불법적인 접근 시도 또는 접근 사실을 알 수 있다.

이와 같은 사항들을 고려해 리눅스 서버를 구축한다면 보다 더 보안에 강한 리눅스 서버를 구축할 수 있을 뿐만 아니라 보안 취약성을 제거할 수 있다. 또한 관리자가 보안대책을 세우는데 편리성을 제공하기도 한다.

다음 [그림 1]은 유용한 도구를 이용한 보안에 강한 리눅스 서버 구축도이며 <표 1>은 도구를 다운받을 수 있는 사이트를 정리해 놓았다.



[그림 1] 보안에 강한 리눅스 서버 구축

<표 1> 보안 도구 다운로드 사이트

보안도구명	다운로드 사이트
Tripwire	ftp://cert.org/pub/tools/cops
SSH	http://www.ssh.org
TCP wrapper	ftp://ftp.porcupine.org/pub/security/index.html
ipchins	리눅스 커널 2.1.102이상 버전에 포함
RTSD	http://www.certcc.or.kr/
SAINT	http://www.wwdsi.com/saint/downloads

5. 결론

본 논문에서는 리눅스 운영체제를 사용하여 서버를 구축하고 운영하는 관리자에게 보다 안전한 서버구축 및 운영을 위한 방법을 제안하였다. 물론 보안에 완벽한 시스템은 존재하지 않는다. 그러나 본 논문에서 제시한 방법을 사용하여 서버를 구축하고 운영한다면 귀중한 정보와 지식을 보다 안전하게 관리할 수 있을 것으로 생각한다.

[참고문헌]

[1] 임채호, "최근 해킹사례에 대한 해설", http://www.certcc.or.kr/concert
[2] Robert Graham, "Sniffing(network wiretap, sniffer) FAQ", http://www.robertgraham.com/pubs
[3] Hyunwoo Lee, Sangyoub Lee, Hyunchul Chung, "분산 환경에서의 서비스거부 공격 분석보고서", http://www.certcc.or.kr/paper
[4] Hyunchul Chung, "sscan 분석 보고서", http://www.certcc.or.kr/paper
[5] 한국정보보호센터, "IP spoofing" http://www.certcc.or.kr/advisory/tr/IPspoof.html
[6] Hyunchul Chung, "TCP Connection Hijacking 공격 및 대책", http://www.certcc.or.kr/paper
[7] Jay Beale, "Tripwire - The Only Way to Really Know", http://www.securityportal.com
[8] http://www.tigerlair.com/ssh
[9] ftp://ftp.porcupine.org/pub/security
[10] 신 훈, "SAINT 분석 보고서", http://www.certcc.or.kr/paper/
[11] Paul Russell, "Linux IPCHAINS-HOWTO", http://www.linuxdoc.org
[12] Hyunwoo Lee, Sangyoub Lee, Hyunchul Chung, Yunjong Jeong, Chaeho Lim, "Analysis of Large Scale Network Vulnerability Scan Attacks and Implementation of the Scan-Detection tool", http://www.certcc.or.kr