시스템 및 네트워크 모니터링 도구를 이용한 리눅스 서버의 보안성 강화

정성재, 현상완, 소우영, 송정길
한남대학교 컴퓨터공학과

Security Enhancement for Linux Server Using System and Network Monitoring Tools

Sung-Jae Jung, Sang-Wan Hyoun, Woo-Young Soh, Jung-Gil Song
Dept. of Computer Engineering, HanNam University

요 약

오픈 소스 운영체제인 리눅스는 UNIX 운영체제에 대한 연구에서 시작되어 최근 상용버전 업체의 대두로 질적, 양적 성장을 하게 되었다. 그러나, 리눅스는 다른 UNIX계열이나 마이크로소프트사의 Windows 2000계열 서버에 비해 업데이트가 쉽지 않아 리눅스기반 시스템 관리자는 다른 계열의 운영체제에 비해 좀 더 숙달된 관리 기술이 요구된다. 특히 인터넷을 통한 자동화된 업데이트를 지원하는 윈도우계열에 비하면 많은 노력이 필요한 실정이다.

본 논문에서는 리눅스 기반 시스템 관리자들이 리눅스 서버 보안 강화를 위한 한 방편으로 시스템 및 네트워크 모니터링 도구를 이용한 방법을 제시하고자 한다. 많은 모니터링 도구중에서 공격자들이 흔히 시도하는 방법에 따라 모니터링 도구들을 제시하고 특징이나 역할을 설명하여 리눅스 서버 보안 강화를 위한 도구 선택에 있어 하나의 방향을 될 수 있도록 하고자 한다. 이러한 도구의 제시는 리눅스 서버 관리자들이 보안 강화시 보다 신속하고 정확하게 대처가 가능할 것이다.

1. 서론

리눅스는 전 세계의 개발자, 연구원, 학생들이 함께 개발한 오픈 소스계열의 운영체제이다. 리눅스가 많은 개발자와 프로그래머들이 함께 개발한 운영체제여서 현재 리눅스의 모태가 된 UNIX에 필적할 만할 성능을 나타내고 있다. 그러나, 오픈 소스라는 특징으로 리눅스가 다른 UNIX계열 운영체제나 마이크로소프트사의 윈도우 계열 운영체제에 비해 시스템 운영을 위한 문서나 프로그램, 업데이트 등의 지원이 미흡하다. 특히, 윈도우처럼 자동화된 업데이트 등은 매우 미약하다. 물론 현재 레드햇(RedHat)같은 리눅스 상용화 벤더(Vendor)에서 일부 지원은 하고 있으나, 거의 모든 업데이트 작업은 시스템 관리자의 몫으로 남아 있다. 이러한 문제점으로 리눅스 시스템 관리자는 다른 운영체제에 비해 좀 더 숙달된 기술이 요구되고 관리자의 능력이 곧 서버 보안에 큰 비중을 차지하게 된다. 즉, 보안을 위해 관리자들이 수많은 문서를 찾아봐야 하고 보안과 관련된 응용프로그램들을 찾아서 운영해야 된다. 따라서 본 논문에서는 리눅스 서버 보안 강화를 위한 하나의 방편으로 시스템 모니터링(System Monitor)과 네트워크 모니터링(Network Monitor) 도구들을 이용한 방법을 제시하고자 한다. 리눅스 시스템에 기본적으로 제공되는 명령어나 응용프로그램을 이용하고, 또한 유용한 모니터링 도구를 제시함으로써 리눅스 시스템 보안의 한 방법을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장은 포트(Port)의 개념, 기본적인 점검과 제어에 대해 알아보고, 3장에서는 기본 명령어를 이용한 모니터링, 포트 스캔 감지, 로그, 백도어(BackDoor) 점검, 파일 시스템 무결성 감시, 패킷 모니터링 도구에 대해 알아본다. 4장에서는 스니핑(Sniffing)도구, mrtg, 침입탐지 시스템 등 네트워크 모니터링 도구에 대해 알아본다. 마지막으로 5장에서는 결론 및 향후 연구과제에 대해 제시한다.

2. 포트(Port)에 대하여

2.1 포트의 개념

포트란 시스템과 시스템간 통신시 하나의 물리적인 전송선 위에 웹서비스, 메일 서비스, DNS 서비스 등 여러 개의 응용 프로그램들이 서로 나누어 사용하기 위한 접속 통로라 할 수 있다. 한 시스템내에서 소켓(Socket)을 이용하는 모든 프로세스는 별도의 포트 번호를 부여받고 이러한 번호를 통하여 서버(Server)와 클라이언트(Client) 간에 통신을 하게 된다. 보통 포트 번호는 0번부터 65535번까지 사용하고, 이 중에서 0번부터 1023번까지의 1024개 포트는 Privileged Port 또는 Well Known 포트라고 하여, 많이 사용하는 네트워크 서비스에 고정적으로 할당되어 있다. 즉 웹서비스는 80번, 텔넷은 23번, 메일은 25번 등으로 고정되어 있다. 리눅스 시스템에서 Privileged Port 등 포트번호, 포트 서비스, 프로토콜에 대한 설명은 /etc/services 라는 파일에 명기되어 있다. 1024번 이후의 포트는 Unprivileged Port라 하여 주로 root 이외의 일반 사용자가 특정 서버 포트로 사용할 수 있고, 보통 클라이언트로 사용시에 주로 사용된다.

2.2 포트의 점검 및 제어

현재 운영중인 서버에 어떠한 서비스가 어떠한 포트를 사용하고 있는지 확인할 필요가 있다. 이 경우 가장 쉬운 방법은 telnet 명령을 이용하여 해당 포트로 연결을 해보는 것이다. 그러나, 이러한 방법은 작동중인 서비스가 많은 경우 일일이 다 접속을 해봐야 하므로 상당이 불편하다. 이 경우에는 별도의 포트 스캐닝(Port Scanning) 프로그램을 이용하는 것이 좋다. 현재 가장 대표적인 프로그램이 nmap이다[1]. 이 프로그램을 이용하여 자신의 서버에 불필요하게 작동중인 서비스포트를 확인할 수 있다. 만약 스캔시 관리자도 알지 못하는 포트가 떠 있는 경우가 있다. 이러한 경우 해킹을 당하는 백도어(BackDoor)로 사용하기 위해 포트가 열려져 있을 수도 있으니 해당 포트를 확인하고, 프로세스가 무엇인지 알아내야 한다. 포트에 대한 프로세스의 확인은 lsof와 fuser라는 명령을 이용하여 알아낼 수 있다. 만약 이 프로세스가 정상적인 것이 아니라면 kill, killall 등의 프로세스 제어 명령어를 사용하여 중지시키고, ipchains나 iptables 등의 커널 레벨에서 작동하는 패킷 필터링 프로그램을 이용하여 포트도 제어해야 한다[2].

3. 시스템 모니터링 도구

3.1 명령어를 이용한 시스템 모니터링

netstat는 유닉스계열에 기본적으로 제공되는 네트워크 관련 명령어로 네트워크 연결상태, 라우팅 테이블, 인터페이스 관련 정보를 확인할 수 있다. 특히 netstat -l을 입력하여 현재 시스템에 리슨(Listen)하고 있는 프로그램과 포트에 관한 정보를 확인할 수 있다. 또한 ps(process status)는 시스템의 프로세스 상태를 모니터링할 수 있고, top 명령을 이용하여 CPU 점유율에 따른 프로세스의 상태를 실시간으로 확인할 수 있다.

3.2 tcpdump를 이용한 시스템 모니터링

tcpdump는 네트워크 인터페이스를 거치는 패킷들 중 주어진 조건식을 만족하는 패킷들의 헤더(Header)를 출력해주는 프로그램이다[3]. 보통 특정 포트나 IP에서의 패킷을 모니터링하고 패킷을 출력하여 네트워크 관련 장애를 해결하기 위한 방법을 활용할 수 있다. 또한, telnet을 이용하여 접속하는 사용자들의 패스워드를 알아내는 경우에도 사용할 수 있다.

3.3 포트 스캔 감지 프로그램 활용

원격지 시스템의 정보를 얻기 위해 공격자들이 제일 먼저 사용하는 방법이 포트 스캐닝이다. 이 포트 스캐닝을 통해 해당 시스템에 열린 포트를 알아내고 어떠한 서비스를 하고 있는지 탐지한다. 이러한 이유로 포트 스캔을 당하면 서버 관리자 입장에서는 차후에 침해 가능성이 있다는 것을 의미한다. 따라서 포트 스캐닝을 감지할 수 있는 프로그램이 필요하다. 많은 프로그램이 존재하지만 그 중에서 한국정보보호진흥원의 RTSD라는 프로그램이 유용하다. RTSD는 Real Time Scan Detector의 약자로 공개용 프로그램이고, 실시간으로 클라이언트와 서버의 포트가 상호 반응하는 관계를 보여주고, 포트 스캔시 해당 내용을 서버관리자에게 통보해 준다. 이 외에도 Klaxon[4], Scanlogd[5], Portsentry[6] 등이 있다.

3.4 로그 모니터링 프로그램 활용

리눅스 등 UNIX 계열 운영체제는 거의 모든 내용을 로그로 기록한다. 따라서, 시스템상의 문제나 불법적인 접속시도 등도 로그에 남는데, 끝도 없이 쌓여가는 로그들을 일일이 분석하는 것은 쉽지 않다. 이 때 로그를 관리해주는 프로그램을 이용함으로 좀 더 손쉽게 로그분석을 할 수 있다. logcheck[6], swatch[7],

colorlog[8] 등의 프로그램을 이용하면 실시간 로그체크 뿐만아니라, 메일발송, Beef음 발생, 특정 스크립트 수행, 색상을 통한 표현 등 다양한 형태의 관리가 가능하다.

3.5 백도어(BackDoor) 점검하기

공격자들이 시스템을 공격하여 root 권한을 획득하면 차후 다시 들어오기 위하여 자신만이 알고 있는 백도어를 만드는데, 이러한 백도어 설치 여부를 점검해야 한다. 이 경우 많이 사용하는 프로그램이 chkrootkit이다[9]. 이 프로그램은 명령어의 변조 여부, 호스트 내에서의 바이러스 등을 통한 루트킷 설치 여부를 점검해준다.

3.6 파일시스템 무결성 감시프로그램

chkrootkit이라는 프로그램도 일부 프로그램이 무결한지 체크하지만, 전체 파일시스템에 대해서는 검사하지 않는다. 이 경우 전체 파일시스템에 대해 무결성을 체크해주는 프로그램이 있는데 이 프로그램에 tripwire[10], Fcheck[11], sXid[12] 등이 있다.

3.7 패킷 모니터링 프로그램

tcpdump나 ethereal[13]은 패킷을 모니터링 해주지만 모니터링 결과를 쉽게 보기에는 불편하다. iptraf은 프로그램의 크기도 작으면서 강력한 기능을 제공한다. 이 프로그램은 네트워크에서 나가고 들어오는 모든 요청을 실시간으로 모니터링해주는데 호스트, 포트, 프로토콜 등에 대한 세부적인 정보를 제공하고 IP 트래픽, 인터페이스 통계, LAN 모니터 등 메뉴식으로 쉽게 볼수 있다. 또한 프토토콜별 필터링 기능도 제공한다[14].

4. 네트워크 모니터링 도구

4.1 스니핑(Sniffing) 모니터링 프로그램

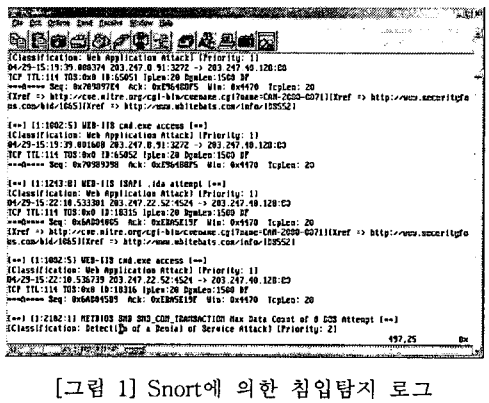
동일 네트워크에 여러 대형 시스템들이 같이 사용할 경우에는 스니핑 공격에 유의해야 한다. 취약한 하나의 시스템에서 관리자 권한을 획득하여 스니핑을 하면 전체 네트워크를 장악할 수 있다. 보통 스니핑이 작동하면 인터페이스카드가 PROMISC 모드로 작동한다. 물론 이 경우 기존의 ifconfig 라는 네트워크 명령으로 확인하고 중단시킬 수 있으나 꼭 PROMISC 모드라 해도 반드시 스니핑이 작동한다고 볼 수 없으니 이 경우 Sentinel[15], Antisniff, ARPWATCH 등의 프로그램을 사용하면 네트워크상에서 스니핑이 작동하는 지 여부를 검색할 수 있다[16].

4.2 MRTG를 이용한 모니터링

MRTG(Multi Router Traffic Grapher)는 SMNP(Simple Network Management Protocol)을 이용하여 서버나 라우터의 트래픽이나 데이터 전송량을 모니터링하여 결과값을 이미지로 생성하고 HTML로 보여주는 프로그램이다. 모니터링 결과를 그림형태로 손쉽게 확인할 수 있고, 웹서버를 통하여 신속히 확인 가능하다. 또한 특정 네트워크와의 접속 속도 등도 모니터링 할 수 있다[17].

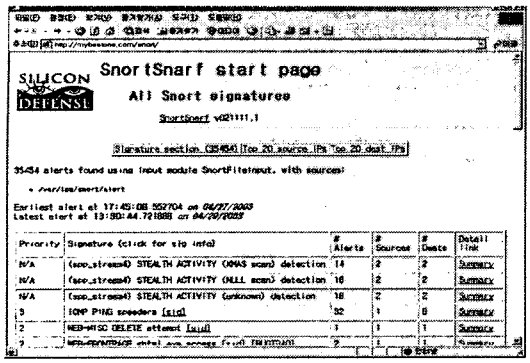
4.3 침입탐지 시스템(IDS) 이용하기

IDS(Intrusion Detection System)란 흔히 침입탐지시스템이라고 한다. 리눅스에서는 공개용 프로그램으로 SNORT라는 프로그램이 유명하고 가장 많이 사용된다. SNORT는 실시간 트래픽 분석, 프로토콜 분석, 내용검색 및 매칭, 침입탐지 Rule에 의거하여 버퍼 오버플로우, 포트스캔, CGI 공격, OS 확인 시도 등 다양한 공격과 스캔을 탐지할 수 있다. 침입탐지 Rule은 보안 커뮤니티나 관련 사이트를 통해 지속적으로 업데이트되고 또한 사용자가 직접 Rule을 작성하여 추가할 수 있도록 설계되어 있어 최신 공격에 대하여 빨리 대처할 수 있다[18].



[그림 1] Snort에 의한 침입탐지 로그

또한, snort는 로그를 접속해서 확인해야 하는데 snort관련 보조 응용 프로그램인 SnortSnarf라는 프로그램을 이용하면 좀 더 손쉽게 로그 확인할 수 있다. 이 프로그램은 snort로그를 관리해줄 뿐만아니라, 결과를 웹페이지형태로 제공하므로, 서버에 접속없이 웹상으로 어디서나 빠르고 편리하게 찾아볼 수 있도록 제공한다[19].



[그림 2] SnortSnart에 웹기반 로그 분석

4.4 원격에서 보안 취약점 점검하기

리눅스 시스템에 대한 네트워크 보안 취약점을 원격지에서 점검가능한데, 이 때 사용되는 프로그램이 SATAN, SAINT, Nessus 등이 있다[20]. 최근에 많이 사용되는 프로그램이 Nessus인데 이 프로그램은 서버와 클라이언트 구조로 이루어져 클라이언트에서 서버로 접속하여 정해진 룰에 따라 취약점을 점검한다. 자체적인 스캔을 위해 nmap을 사용하므로 사전에 nmap을 설치해야 한다. 취약점 점검이 끝나면 모든 정보를 종합하여 어떠한 부분이 취약한지, 또한 어떻게 패치하여 대처해야 하는지에 대한 결과를 리포트로 작성하여 보여주고, 결과보고서를 HTML이나 ASCII 등의 형태로 선택하여 저장할 수 있다.

5. 결론 및 향후 연구과제

리눅스 시스템 관리의 가장 큰 단점은 오픈 소스라는 이유로 윈도우나 다른 UNIX 계열에 비해 시스템 관리를 위한 자동화된 업데이트 체계나 최신의 문서가 부족하여 해당 시스템 관리자가 모든 것을 해결해야 한다는 것이다. 따라서, 리눅스 시스템 관리자는 다른 운영체제에 비해 숙달된 관리 기술이 요구되고, 이러한 실정은 곧 서버 보안에도 영향을 미친다. 리눅스 서버를 침입자의 공격으로부터 막으려면 좀 더 체계화된 연구와 노력이 필요한데, 보안 강화의 한 방법으로 시스템 및 네트워크 모니터링 도구를 이용할 수 있다. 시스템 모니터링은 기본 명령어를 이용한 모니터링, tcpdump, ethereal 같은 프로그램을 패킷 모니터링, 포트 스캔 감지 프로그램을 이용한 모니터링, 로그 모니터링, 백도어 점검, 파일 시스템 무결성 검사 등을 할 수 있고, 네트워크 모니터링은 스니핑 검사, MRTG를 이용한 그래픽한 모니터링, 침입탐지 시스템을 이용한 모니터링, 원격지에서의 보안 취약점 점검 등을 할 수 있다. 3장과 4장에 나열된 도구들을 잘 이용한다면 리눅스 시스템의 보안 강화에 상당한 도움이 되고, 수 많은 모니터링 도구들 중에서 효율적이고 신속하게 도구를 이용한 대처가 가능하다. 그러나, 이러한 도구의 사전지식이 없으면 리눅스가 오픈 소스이듯이 현재 제공되는 시스템 및 네트워크 모니터링 프로그램들도 대부분 모두 오픈 소스여서 관리자가 직접 찾아보고 설치해야 하고 또한 설치나 사용법도 쉽지 않다. 물론 각각의 모니터링 프로그램이 본 논문에 나열된 것들보다 훨씬 다양하고 많은 관계로 논문에 제시된 프로그램을 이용하는 것이 현존하는 최고의 도구를 이용하여 리눅스 서버 보안을 강화하였다고 말할 수는 없다. 아울러 위에 제시된 프로그램들이 모두 독립적인 프로그램으로 모두 관리자가 직접 설치해야 한다는 단점이 있다. 이렇게 다양한 도구를 하나로 통합하여 다양한 기능을 하는 하나의 프로그램에서 수행할 수 있는 통합 보안 도구를 연구할 필요성도 제기된다.

[참고문헌]

[1] http://www.insecure.org/nmap
[2] http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html
[3] http://www.tcpdump.org
[4] http://www.eng.auburn.edu/users/doug/second.html
[5] http://www.openwall.com
[6] http://sourceforge.net/projects/sentrytools
[7] http://swatch.sourceforge.net
[8] http://www.resentment.org
[9] http://www.chkrootkit.org
[10] http://www.tripwire.org
[11] http://www.geocities.com/fcheck2000/fcheck.html
[12] http://marcus.seva.net
[13] http://www.ethereal.com
[14] http://iptraf.seul.org
[15] http://www.packetfactory.net/projects/sentinel
[16] http://www.packetstormsecurity.org
[17] http://www.mrtg.org
[18] http://www.snort.org
[19] http://www.silicondefense.com
[20] http://www.nessus.org