스마트 업데이트를 이용한 리눅스 보안 시스템

석원홍[0] · 강진석[00] · 강흥식[000]
인제대학교 정보컴퓨터 공학부
babootaeng@naver.com
comdol12@netian.com
hskang@nice.inje.ac.kr

Linux Security System Using Smart Update

Won-Hong Seok[0] , Jin-Suck Kang[00], Heung-Seek Kang[000]
Dept of Information and Computer Engineering, Inje University

요 약

오늘날, 리눅스의 급속한 활용은 상대적으로 해킹사례의 확산으로 이어지고 있다. 이에 리눅스 시스템의 취약점에 대한 패치 파일들이 개발자들에 의해서 빠르게 보급되고 있지만 리눅스 관리자들의 보안에 대한 무관심과 번거로운 업데이트 작업의 기피로 인해 제대로 보안 패치가 이루어지지 않고 있는 것이 현실이다. 본 논문은 이러한 실질적인 문제점들을 해결하고자 지금까지의 번거로운 소프트웨어 패치 작업을 스마트 업데이트 기능을 지닌 LSUS(Linux Smart Update System)을 통해 취약점을 지닌 소프트웨어를 자동으로 패치 함으로써 안전한 시스템을 운영할 수 있도록 하였다.

1. 서론

리눅스는 특정기업의 소유가 아닌 운영체제라는 장점으로 인해, 공개된 후 10년이 채 되지 않아 기존 상용 운영체제들을 위협할 정도로 그 보급과 사용이 급속히 증가하고 있는 운영체제이다. 그러나 리눅스의 보급이 늘어나면서 해킹사례도 빈번해지고 있는 실정이다. 리눅스는 커널소스를 포함한 소프트웨어들이 오픈 소스인 특성 탓에 실제로 운영에 있어 보안에 문제점이 많은 것이 사실이다. 즉, 이것은 해커들이 소스를 분석하여, 버그를 찾아 낼 수 있다는 것을 의미한다. 하지만 이러한 오픈 소스의 특징이 보안이라는 측면에서 절대적으로 불리한 것만은 아니다. 오히려 비공개 소프트웨어들이 설계에 있어서 기능이나 편리함에 비해 보안에 신경을 많이 쓰지 않고 개발되어지기에 더 문제가 될 수 있다. 그에 반해 오픈소스는 이런 비공개소스보다 취약점을 분석하고 대처함에 있어서 신속성이 상대적으로 우수하다. 실제로, 오픈 소스 소프트웨어 취약점은 길게는 이틀, 짧게는 한 시간만에 패치 파일들이 나오고 있다. 그러나 오픈 소스의 이런 장점에도 불구하고 현실에서는, 관리자들의 보안에 관한 무관심과 번거로운 업데이트 기피로 말미암아 효과적으로 패치를 통한 업데이트가 이루어지지 않고 있는 실정이다. 따라서 본 논문에서는 이러한 관리자의 번거로운 소프트웨어 패치 작업을 스마트 업데이트 기능을 지닌 LSUS를 구축하여, 취약점을 지닌 소프트웨어를 효과적으로 패치 함으로써 안전한 시스템을 운영할 수 있도록 하였다.

이에 본 논문의 형식은 다음과 같다. 먼저 2장에서는 스마트 업데이트의 활용에 대해서 알아보고 3장에서 본 논문에서 제안하고 있는 LSUS의 설계와 구현 방안이 서술된다. 이어 4장에서는 구현된 LSUS의 실험 결과를 보인 후 마지막으로 5장에서 결론과 향후 발전 방향에 대해서 서술한다.

2. 스마트 업데이트의 활용

현재 스마트 업데이트는 많은 분야에서 활용되어지고 있다. 대표적으로 백신 프로그램인 V3 시리즈 와 침입탐지 시스템인 세이프존넷에 대해 간략히 소개한다.

V3시리즈는 안철수 연구소에서 개발된 백신 프로그램으로 신종 바이러스를 보다 효율적으로 대처하기 위해서 스마트 업데이트를 통해 사용자는 최신 버전의 엔진을 유지하여, 바이러스를 통한 피해를 최소화 시켜준다.

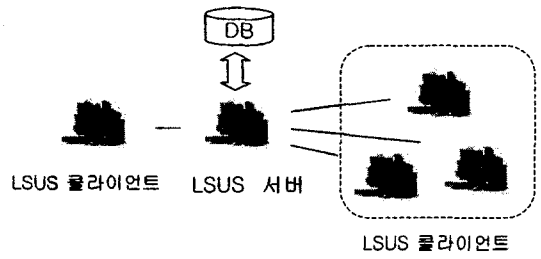
세이프존넷은 실시간으로 네트워크를 감시하여 효율적인 침입탐지/분석 및 대응, 해킹 발생시 시스템을 자동으로 조절하여 대응, 해커의 공격형태 및 패턴을 기억 정보자산에 대한 오용 및 훼손사고를 미연에 방지하는 특징을 지닌 시스템으로 스마트 업데이트 기능을 통해 새로운 대응방법을 자동으로 업데이트 해 줌으로써 고도화되는 해킹기법에 시시각각으로 대응할 수 있다.

이상으로 보안에 관련한 특정한 제품에 대해서 알아보았다. 물론 이러한 보안 관련 분야뿐만 아니라, 사용자에게 편리하게 최신버전을 제공하는 등 다른 여러 분야에서도 스마트 업데이트는 많이 활용되어지고 있다.

3. LSUS 설계 및 구현

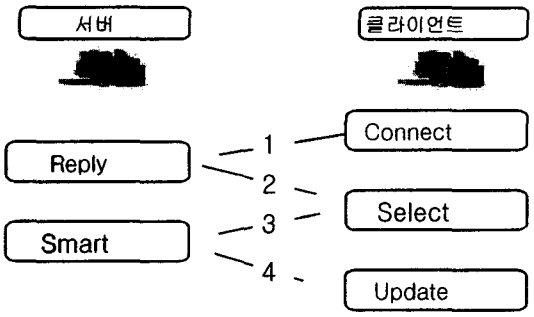
3.1 LSUS 구성도

다음의 [그림 1]은 하나의 LSUS 서버에 여러 클라이언트들이 접속하여 자신의 시스템을 점검한다.



[그림 1] LSUS 전체 구성도

LSUS는 크게 클라이언트 부분과 서버 부분으로 나누어진다. 사용자가 LSUS 클라이언트를 실행시키면, 서버에 접속하여 자신의 시스템 소프트웨어 정보를 서버에게 제공하여, 서버에서 업데이트가 필요한지 판단한 후 업데이트가 필요한 소프트웨어가 발견되면, 해당 소프트웨어의 패치 파일을 제공하여, 업데이트 하는 기능을 가진다.



[그림 2] 서버와 클라이언트간의 실행

[그림 2]에서는 서버와 클라이언트 사이에서 이루어지고 있는 과정을 보여주고 있다. 그리고 이상의 기능들은 아래와 같은 모듈로 이루어져 동작한다.

1) Connect 모듈

클라이언트에 설치되어 있는 LSUS를 실행하게 되면, 서버에 접속을 하기 위한 인증 절차를 간단히 거친 후에 서버에 접속이 된다. 이 때 클라이언트에서는 서버로 자신의 시스템 정보를 보내게 된다. 그리고 서버에서는 클라이언트가 접속되었다는 LSUS 서버에서 파악할 수 있다.

2) Reply 모듈

LSUS서버는 로긴된 클라이언트로부터 받은 정보를 데이터베이스에 저장되어 있는 최신 정보와 비교하여, 보안상 문제점이 보이는 소프트웨어가 발견되면, 클라이언트에게 패치가 필요함을 알려주는 역할을 한다. 그리고 새로운 보안 뉴스나 공지사항 내용을 LSUS 클라이언트 메인 화면에 제공함으로서, 관리자가 좀 더 효율적으로 시스템을 관리할 수 있도록 도와준다.

3) Select 모듈

LSUS 클라이언트는 서버가 패치가 필요하다는 메시지를 받게 된다. 클라이언트에서 확인 메시지를 보내게 되면, 서버에서 패치 파일을 전송하게 된다. 그리고 클라이언트의 옵션 메뉴에서 자동 패치 설정을 체크하였다면, 확인 메시지 없이 스마트 업데이트는 자동으로 실행되게 된다.

4) Smart 모듈

클라이언트에게 필요한 데이트를 전송하는 기능을 가졌다. 클라이언트 옵션 메뉴에서 FTP 나 HTTP를 통해서 받을 것인지 설정할 수 있다.

5) Update 모듈

서버로부터 받은 업데이트 할 패치 파일을 자동으로 처리해주는 모듈이다. 패치 수행결과는 로그파일에 남기고, 현재 진행 중인 업데이트 파일들의 상황을 보여 주는 역할을 한다.

3.2 LSUS 서버 구현

LSUS 서버는 클라이언트의 정보를 제공받아 데이터베이스(Oracle)에 저장되어 있는 보안에 취약점 소프트웨어 정보와 비교하여 패치가 활용한 소프트웨어가 발견되면, 즉시 그에 해당하는 패치 파일을 제공하여 준다. 데이터베이스와 연결을 위해 JDBC를 이용하였다.

3.2.1 LSUS 서버의 기능

- LSUS의 핵심인 업데이트가 필요한 정보와 패치 파일 제공
  스마트 업데이트 기능을 위해서 클라이언트가 접속했을 때 DB의 최신 정보와 비교하여 업데이트 필요시에 클라이언트에게 패치 파일을 제공한다.
- 보안 뉴스 제공
  LSUS 클라이언트 실행 시 메인 화면에서 최신 보안 뉴스를 확인할 수 있다.
- 메일을 통해 업데이트를 권고
  LSUS에는 각 클라이언트의 시스템 정보를 가지고 DB에 저장해 놓고, 주요한 패치를 해야 하지만 LSUS 서버에 접속하지 않을 경우, 메일을 통해서 업데이트를 권고한다.

3.3 LSUS 클라이언트 구현

LSUS 클라이언트는 리눅스 환경에서 구현하였으며, 사용한 언어로는 어떠한 플랫폼에서도 작동하는 자바를 이용하여 구현했다.

3.3.1 LSUS 클라이언트 기능

- 제공받은 패치 파일 설치

서버로부터 패치 파일을 받아 시스템에 설치한다.

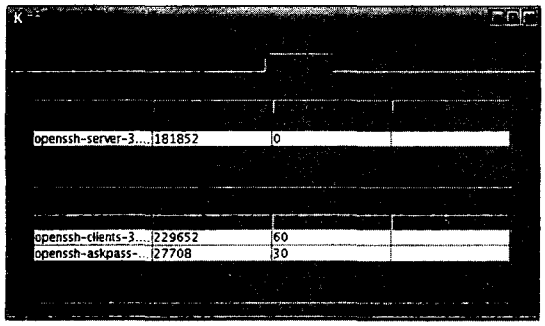
· 업데이트 로그 저장
업데이트 했던 로그파일을 볼 수 있도록 제공한다.

· 패키지 정보 제공
자신의 시스템에 설치되어 있는 소프트웨어들을 관리자가 보기 쉽게 보여주는 기능을 한다.

4. 실험 및 결과

4.1 LSUS 클라이언트 실험 및 결과

Openssh의 취약점을 지닌 소프트웨어를 사용되어지고 있는 시스템에서 LSUS 클라이언트를 실행했을 때, 소프트웨어 패치가 필요하다는 메시지를 받은 후 확인 메시지를 보내게 되면 서버로부터 Openssh 최신파일을 제공받아 업데이트가 이루어지게 된다.

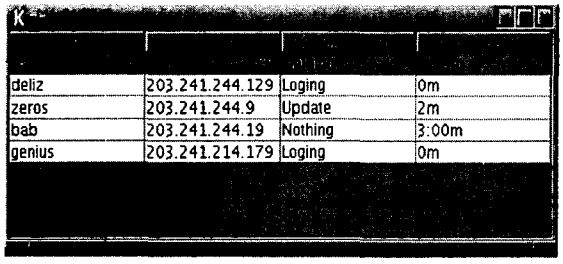


[그림 3] 업데이트 실행 장면

[그림 3]은 바로 이러한 업데이트가 이루어진 후에 추후 관리가 편하도록 LOG 파일로 남겨지는 것을 보여주고 있다.

4.2 LSUS 서버 실험 및 결과

[그림 4]는 LSUS 서버를 실행한 장면이며, 서버에 접속한 클라이언트에 대한 정보를 보여주고 있다. LSUS 서버 관리자는 모니터링을 통해 효율적인 관리를 할 수 있다.



[그림 4] LSUS 서버의 실행화면

5. 결론 및 향후 연구

리눅스는 오픈 소스라는 장점을 지녔음에도 불구하고 관리자의 보안에 대한 무관심과 번거로운 업데이트 작업을 꺼려 제대로 업데이트가 이루어지지 않아 많은 해킹 피해를 당하고 있는 실정이다. 따라서 본 논문에서는 소프트웨어를 재빨리 패치 하여, 해킹 피해를 줄이고 효율적으로 시스템을 운영 위해서 스마트 업데이트를 이용한 보안 시스템인 LSUS를 제시하고 구현하였다. LSUS는 관리자들에게 보다 쉽게 패치를 할 수 있으며, 그 외에도 패키지 정보 보안 뉴스 등을 관리자에게 제공함으로써 관리자에게 시간적인 여유를 제공한다. 그러나 본 시스템은 여전히 몇 가지 해결하지 못 한 문제점을 지니고 있다. 먼저 리눅스 소프트웨어 개발시에 최적화를 위해 다른 모듈들과의 많은 공유를 하고 있다. 이러한 소프트웨어를 설치할 때 의존성문제로 인해 설치가 제대로 이루어지지 않는 경우가 발생하게 된다. 이러한 문제점이 발생하게 되면, 본 시스템은 업데이트가 중지되며, 자동으로 업데이트 할 수 없다는 문제이다. 이 경우에는 LOG 파일에 업데이트하지 못 한 내용이 남기에 관리자가 손수 업데이트를 해 줘야 할 것이다. 또 다른 문제점은 리눅스는 명확한 표준 없이 여러 벤더에서 제작되어지고 있다. 각 벤더에서 나오는 리눅스는 약간의 차이점을 가지게 되어, 소프트웨어관리를 하는 방법이 다른 리눅스들도 볼 수 있다. 현 시스템은 레드햇 계열에 시스템에서만 제대로 업데이트 기능을 한다. 이러한 문제점은 현재 리눅스 표준을 위한 벤더들간의 노력으로 개선되어야 할 것이다. 따라서 향후 연구과제로는 업데이트기능 뿐만이 아니라, 취약점 분석 도구를 추가하여 좀 더 효율적으로 업데이트가 이루어질 수 있도록 추진해 나갈 것이다.

참고 문헌

[1] http://www.kldp.org
[2] http://java.sun.com
[3] http://www.redhat.com
[4] http://www.securityfocus.com
[5] Harvey M. Deitel , Paul J. Deitel, Java (4E) - How to Program, Prentice Hall, 2001.
[6] Elliotte Rusty Harold, Java Network Programming, O'Reilly, 2000
[7] W. Richard Stevens, UNIX Network Programming Vol.1 Prentice Hall, 1997
[8] W. Richard Stevens, "TCP/IP Illustated, Volume1", Addison-Wesley
[9] W. Richard Stevens, "Advanced Programming in the UNIX Environment", Addison-Wesley