관리자를 위한 리눅스 서버 시스템 모니터링 및 제어 솔루션 구현

윤효준* · 박병호* · 최용석* · 김종수* · 성현경*

*상지대학교 컴퓨터정보공학부

Implementation of Linux Server System Monitoring and Control Solution for Administrator

Hyo-Jun Yoon* · Byung-Ho Park* · Yong-Seok Choi*

Chong-Soo Kim* · Hyeon-Kyeong Seong*

*School of Computer Information and Communication Eng. Sangji University

E-mail : yhj0852@naver.com, eden200@sangji.ac.kr, choi-ys72@daum.net

cskim@sangji.ac.kr, hkseong@sangji.ac.kr

요 약

리눅스 서버는 웹, FTP, SSH등 여러 가지 서비스를 제공한다. 이러한 서비스를 받는 사용자들이 이것을 이용하여 해킹을 시도하고 있다. 그렇기 때문에 서버 보안에 대한 대책이 필요하다. 본 논문에서는 다중의 리눅스 서버의 대한 각 종 서비스 로그를 분석하여 리눅스 기반이 아닌 윈도우즈 기반에서 다중 리눅스 서버 시스템을 모니터링 및 제어할 수 있는 솔루션을 개발하였다.

ABSTRACT

Linux server offers various kinds of service including web, FTP, and SSH. The users of these kinds of service are trying to hack by making use of it. That's why some countermeasures are required for the security of the server. In this thesis, each type of service log of multiple Linux server was analyzed, and a solution was developed to monitor and control the multiple Linux server system not based on Linux but based on Windows.

키워드

Linux server, Multi-Linux, Linux based on Windows

Ⅰ. 서 론

리눅스 서버를 이용하여 웹, FTP, SSH등 여러 가지 서비스를 사용자들은 제공받을 수 있다. 또한 이러한 서비스를 받는 사용자들이 서비스 본래의 순기능을 사용하지 않고 역기능을 사용하려는 불특정 다수가 존재할 수 있기 때문에 해킹의 우려가 커질 수 있다. 그렇기 때문에 서버 보안에 대한 대책이 필요하며 그 중 각 종 서비스에 대한 로그를 분석하여 그에 맞게 대응하는 방법이 있을 수 있다.[1-3] 하지만 이러한 대응 방법은 관리자 입장에서 1대의 서버가 아닌 N대의 리눅스 서버를 관리하고 그에 대한 로그를 분석해야 한다면 이는 텍스트 기반으로 복잡하게 나열된 서비스 로그를 분석하고 그에 맞게 대응하는 것은 시간적으로 비효율적이다.[4] 이러한 N대의 리눅스 서버의 로그를 쉽게 분석하고 관리자에게 분석 정보를 시각적으로 보여줌으로써 한 눈에 알아볼 수 있게 되면서 그에 대한 시스템 제어를 할 수 있는 대응 방법이 필요하게 된 것이다. 현재 윈도우즈 서버로 관리하는 곳이 많기 때문에 N개의 리눅스 서버를 리눅스 서버로 통합 제어하지 않고 윈도우즈 기반의 서버에서 관리할 수 있

도록 할 것이다.[5]

II. 관련연구

2.1 Linux Log File

HTTP, FTP, SSH의 3가지 서비스를 대상으로 프로그램이 동작하므로 이것을 바탕으로 4가지 로그파일을 분석한다. 분석할 대상의 첫 번째 로그파일은 /var/log/wtmp 파일이다, 이 파일은 ssh(Secure Shell) 접속 성공의 관련된 정보를 추출할 수 있다. 이 파일은 2진 형태로 이루어져 있기 때문에 파일 자체로는 분석이 불가능 하며 last 명령어를 통해서 정보를 확인할 수 있다.

2.2 Pro*C/C++

Pro*C/C++를 이용하기 위해서는 SQL 문장이 삽입된 C 프로그램을 (.pc) 작성하고 이 프로그램 소스파일을 Pro*C/C++로 먼저 처리하면 삽입된 SQL 문장들이 C 코드로 재 생성되어 (.c) 새로운 소스파일이 나온다. 이것을 가지고 통상적인 C프로그램처럼 컴파일 해주고 링킹하는 과정을 거치면 오라클과 연동할 수 있는 실행 가능한 프로그램이 되는 것이다. 일반적으로 오라클 9i버전인 경우에는 설치과정을 그대로 진행할 경우 Pro*C/C++가 같이 설치된다.

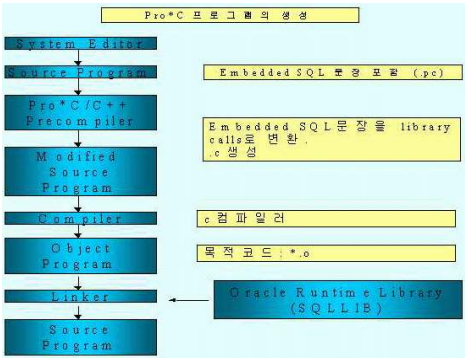

그림 1. Embedded SQL Program의 생성
Fig. 1. The creation of Embedded SQL Program

III. 다중 리눅스 시스템 구조

본 솔루션은 다중의 리눅스 서버를 윈도우즈 기반에서 제어가 가능해야 하므로 클라이언트/서버 구조가 되며 아래 표1과 같이 클라이언트 모듈은 리눅스 개발 환경으로 서버 모듈은 윈도우즈 개발환경으로 구축해야한다. 서버 모듈의 개발도구는 비주얼 스튜디오 2008을 사용하였으며 MFC통해 사용자에게 GUI를 제공하기 위해서이다. 데이터베이스관리프로그램은 Oracle10g가 사용되며 서버 프로그램과 데이터베이스간에는 ODBC를 통해서 접근이 가능하도록 구성한다. 클라이언트 모듈의 개발도구는 Pro*C/C++을 사용하며 이는 unixODBC를 사용하지 않고 데이터베이스에 접근이 가능하도록하며 C언어로 클라이언트 모듈을 구현하기 위해서이다. 또한 클라이언트와 서버와의 데이터통신은 TCP 데이터 통신을 사용하며 GNU Socket 라이브러리와 Windows 소켓 라이브러리를 통해 구성한다.

표 1. 개발 환경 및 개발 도구 모음
Table 1. Statistics for CM and taken out moves

	server	client
Operating System	Windows XP SP3	CentOS 5.2
Development Tool	Microsoft Visual Studio 2008 v9.0.30729.1 SP	Pro*C/C++ Release 10.2.0.1.0
DBMS	Oracle 10g	Oracle 10g

2.1 클라이언트/서버 모듈 전체 구조

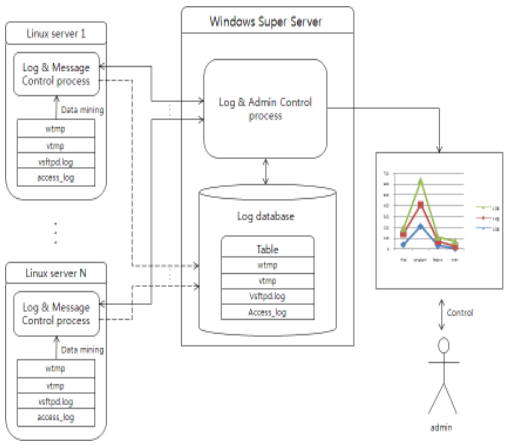

그림 2. 클라이언트/서버 모듈 전체 구성도
Fig. 2. Client / Server module of the entire configuration

전체 구조는 그림6과 같으며 서버측면에서 볼 때 현재 서버 관리 프로그램에 마운팅된 리눅스 서버들을 1:1 또는 1:N의 쉘 명령어를 통한 제어를 하며 마운팅된 서버들의 리스트를 대상으로 데이터베이스에 저장된 추출된 로그 데이터를 통하여 사용자에게 그래프 및 테이블로 출력한다.

클라이언트 측면에서는 각각의 리눅스 서버는 클라이언트 모듈 통하여 서버 모듈에 마운팅되어 쉘명령어를 통해 제어를 받는다.

2.2 서버 모듈 구조

서버 모듈은 실시간 관제가 가능해야하므로 타이머를 통해서 실시간으로 서비스 별 접속 결과를 그래프 또는 테이블로 보여준다. 서버 모듈은 TCP소켓을 통해 클라이언트 모듈이 마운팅될 수 있도록 항상 대기하고 있으며 마운팅시 그 클라이언트의 정보는 소켓리스트와 접속 데이터를 산출하는 월별리스트에 추가되며 그 리스트를 통해 클라이언트를 제어할 수 있다. 또한 사용자로부터 디스플레이 요청시 멀티 쓰레드로 동작하여 한 작업으로 인한 지연상태를 방지한다.

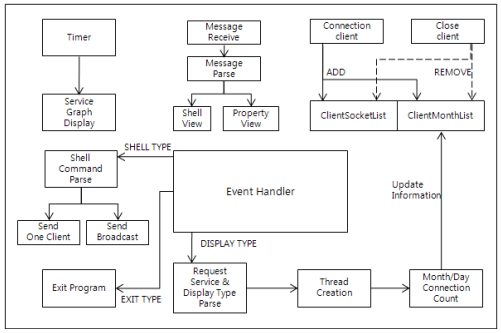

그림 3 서버 모듈 구성도
Fig. 3. Server module configuration

2.3 클라이언트 모듈 구조

클라이언트 모듈은 리눅스 환경에서 실행되며 HTTP, FTP, SSH 서비스에 대한 로그를 분석하여 데이터베이스에 전송하고 서버에 마운팅되어 시스템제어를 동시에 받을 수 있어야한다. 그렇게 때문에 하나의 프로세스로 구성되지 않고 총 5개의 프로세스가 생성되어 구동된다. 각각의 리눅스 서비스에 대한 로그 분석 데이터베이스 업데이트 기능은 최초 모듈 실행시 현제 서비스에 대한 로그를 분석하여 서버 데이터베이스에 전송하고 1초당 로그의 길이를 감지하여 수정시에 그 변화된 길이만큼 로그를 분석하여 서버 데이터베이스에 업데이트한다.

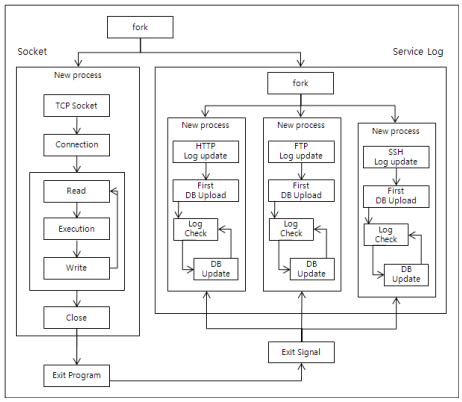

그림 4. 클라이언트 모듈 구성도
Fig. 4. The client module configuration

Ⅳ. 모니터링 및 제어 솔루션 구현

그림4는 구현된 서버 어플리케이션이다. 현재 RFLab 리눅스 서버가 마운팅되어 HTTP 서비스에 대한 11월의 전체 접속량을 보여주고 있다. 또한 쉘커멘더를 통해서 RFLab서버에게 쉘 명령어를 보냄으로써 제어가 가능한 것을 알 수가 있다.

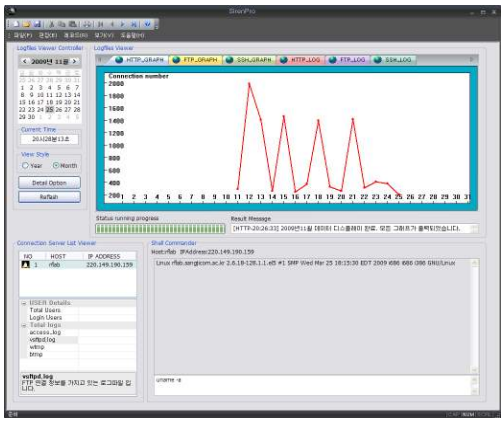

그림 5. RFLab서버의 11월 HTTP 접속 정보화면
Fig. 5. November RFLab server HTTP connection information screen

또한 본 어플리케이션은 다 중의 리눅스 서버 접속도 가능하기 때문에 아래의 그림5와 같이 Sangjicom서버와 RFLab서버의 2009년 FTP접속에 대한 그래프를 나타내고 있으며 Sangjicom서버로 1대1 쉘 접속을 통하여 그 서버 시스템의 정보를 Property창에 출력하고 있다.

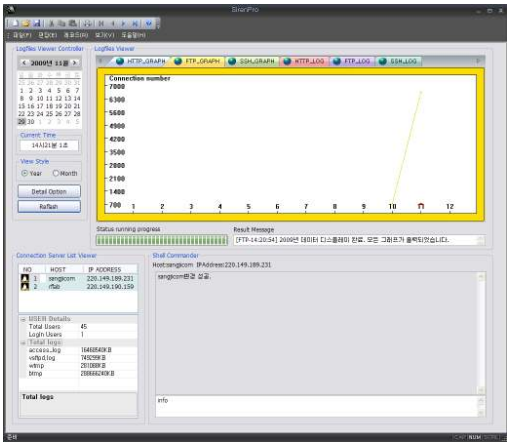

그림 6. Sangjicom서버와 RFLab서버의 2009년 FTP 접속 정보화면

Fig. 6. RFLab Sangjicom servers and serverFTP connection information screen 2009

클라이언트 어플리케이션은 텍스트기반으로 구성되어 있으며 최초 실행시 서버에 마운팅되며 데이터베이스 자동로그인 후 업데이트를 수행하고 있다. 업데이트는 1초당 한번씩 로그를 체크하여 전송하며 현재 상태를 텍스트로 화면에 출력한다.

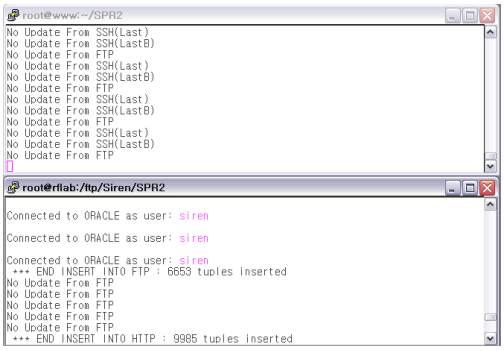

그림 7. RFLab서버와 Sangjicom서버의 클라이언트 어플레케이션이 실행되는 화면

Fig. 7. Sangjicom RFLab server and client applications on the server is running, screen-les keyisyeon

V. 결론 및 향후 연구 방향

본 논문에서 구현된 어플리케이션을 통해서 관리자는 다 중의 리눅스 서버에서 제공되는 HTTP, FTP, SSH 서비스에 대한 접속량을 그래프 및 테이블로 보여준다. 또한 관리자는 그 결과를 통해서 쉘 명령어를 이용해 각각의 리눅스 서버에 보안 정책 및 시스템을 제어할 수가 있다. 그렇기 때문에 기존의 텍스트형식의 로그파일을 분석하고 시스템을 제어하는 방식보다 시간적으로 더 높은 효율성을 보장한다. 또한 브로드캐스트 메시지 전송으로 인하여 멀티 프로세싱이 가능하다. 하지만 이러한 기능은 윈도우즈 기반의 어플리케이션으로 구현되었기 때문에 리눅스 서버에서는 불가능하다. 그렇기 때문에 플랫폼에 독립적으로 수행될 수 있도록 JSP같은 웹프로그래밍언어를 사용하여 웹 상에서 볼 수 있도록 어플리케이션을 구현한다면 플랫폼에 대해 독립적으로 수행될 수 있을 것이다.

참고문헌

[1] 김태용 저, CentOS 리눅스 구축관리실무, 슈퍼유저코리아

[2] Stergios Spanosa, Apostolos Melionesb, George Stassinopoulosa " The internals of advanced interrupt handling techniques: Performance optimization of an embedded Linux network interface" Computer Communications Volume 31, Issue 14, 5 September 2008, Pages 3460-3468

[3] K. Salah, A. Kahtani "Performance evaluation comparison of Snort NIDS under Linux and Windows Server" Journal of Network and Computer Applications, Volume 33, Issue 1, January 2010, Pages 6-15

[4] Michael Athanas, Michael Ogg "An evaluation of PCs for high energy physics under Windows NT and Linux" Computer Physics Communications, Volume 110, Issues 1-3, May 1998, Pages 225-229

[5] http://otn.oracle.com 의 Pro*C/C++ Precompiler Programmer's Guide 참고

[6] http://blog.naver.com/atonikkaz/10016381569