Linux 플랫폼 상에서의 대용량 로그 처리 기법에 대한 연구

문세광*, 윤한경**

A study for Technique of a large log processing on Linux platform

Sae-Kwang Moon*, Han-Kyung Yun**

요 약

네트워크 보안 장비에서 발생되는 대용량 로그를 syslog-ng을 이용하여 파일로 저장하도록 구현하고, 파일로 저장된 로그 메시지를 데이터베이스로 저장하기 위해 gcc를 이용하여 쓰레드 프로그램을 개발하였으며, 이것을 서버스크립트 언어인 PHP를 이용해 로그의 내용을 파악하는 웹기반 뷰어를 개발하였다. 또한 방화벽에서 발생되는 로그 메시지를 처리하는 패키지 상품과의 비교를 통해 성능을 분석하였다.

ABSTRACT

In the course of research, the program that is able to process files through gcc and imput them into the database has been developed, using Syslog-ng based Unix system. PostgreSQL and PHP are used for database and Web-based server side script respectively.

키워드 : SNMP, syslog, syslog-ng, 대용량 로그 처리

Ⅰ. 서 론

Linux 시스템에서는 서버의 장애 발생시 발생되는 로그를 syslog 데몬을 이용해 저장할 수 있다. 그러나 대용량의 로그가 발생하는 경우에는 몇 가지 문제점이 발생한다. 서버나 네트워크 장비에서 발생되는 로그의 양은 일반적으로 많지 않지만, 방화벽(Fire- wall)에서 발생하는 로그는 1초에 10만 건 이상의 로그가 발생되는 경우가 있다.[1]

본 연구에서는 네트워크 장비에서 보내는 syslog 메시지를 서버에서 받아서 파일로 임시 보관하는 syslog-ng 프로그램을 이용하고, 파일로 저장된 데이터를 로그의 종류별로 구분하여 데이터베이스에 저장하는 gathering 프로그램을 개발하였으며, 서버에서 실행되는 웹 프로그래밍 언어인 PHP를 이용하여 네트워크 관리자가 로그의 내용을 실시간으로 확인할 수 있도록 구성하였다.

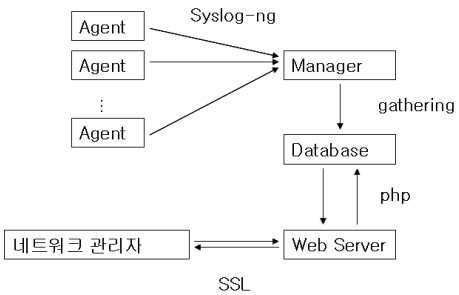

그림1. 데이터 흐름도
Fig.1 Data flow chart

또한 퓨쳐시스템에서 개발된 SecuwayCenter2000 로그 메시지 처리 서버와의 비교를 통해 성능을 분석하였다. [그림1]은 이러

* 한국기술교육대학교 전기전자공학과 석사과정(skmunn@hanmail.net)
** 한국기술교육대학교 정보미디어공학부 교수(hkyun@kut.ac.kr)

한 데이터의 흐름도를 나타내고 있다.

II. 대용량 로그 처리 알고리즘

고성능의 하드웨어 시스템이 아니더라도 syslog- ng에 의해서 만들어진 파일을 멀티프로세싱을 활용하여 데이터베이스에 적재하는 것이 가능하도록 알고리즘을 작성하고, 일반 퍼스널 컴퓨터 사양의 시스템에서 정상적으로 동작되는 것을 확인할 수 있다. [표1]은 테스트한 시스템의 사양을 나타내고 있다.

표1. 하드웨어 사양
Table.1 System hardware

종 류	사 양
processor	Intel Core-II duo 2.2GHz
memory	DDR pc2700 1Gbyte
operating system	Red Hat Enterprise 4
HDD	Sata-II 160G 7200rpm

또한 시스템에서 동작하는 소프트웨어 목록은 [표2]와 같으며, 오픈 소스 프로젝트에 의해 라이센스가 별도로 필요하지 않는 것을 사용하였다.

표2. 소프트웨어 사양
Table.2 System software

종 류	사 양
Database	PostgreSQL
Language	gcc
Web-Server	apache 2.2.10
Script Language	php 5.2.3

네트워크 장비에서 syslog-ng 서버인 manager로 전달되는 로그 메시지를 UDP 패킷을 선택한 것은 SNMP Trap 메시지에 중점을 두었기 때문이다. 현재 사용되는 대부분의 스위치나 라우터는 syslog를 제공한다. 기본적으로 UDP 패킷을 사용하는데, 이것은 로그가 생성되는 시점에 필요한 내용만 전달하면 되고, manager와 agent 사이에 세션 연결을 지속할 필요성이 없기 때문이다. 따라서 agent에서 발송된 로그의 내용이 네트워크의 불안정성으로 인해 manager에 제대로 전달되지 않는 경우가 발생할 수도 있다.[2]

syslog-ng 소스코드를 컴파일하여 설치한 폴더는 /usr/local/etc로 하였으며 syslog-ng.conf 파일을 생성하여 구성 파일을 작성하였다. 다음은 syslog-ng. conf 설정의 일부이다.

```
options {
  sync (0);
  time_reopen (10);
  log_fifo_size (1000);
  long_hostnames (off);
  keep_hostname (yes);
};
source s_sys {
  file ("/proc/kmsg"log_prefix("kernel: "));
  unix-stream ("/dev/log");
  internal();
  udp(ip(0.0.0.0) port(514));
};
destination d_postgres {
file("/logbank/lib/pgsql/

syslog2pgsql/fulllog.$YEAR.$MONTH.$DAY.
  $HOUR.$MIN.$SEC" template("INSERT
INTO
  logs (host,facility,priority,level,tag,adate,
  wdate,program,msg) VALUES ('$HOST','

$FACILITY','$PRIORITY','$LEVEL','$TAG',
  '$YEAR-$MONTH-$DAY $HOUR:$MIN:
  $SEC',now(),'$PROGRAM','$MSG'); \n")
  template-escape(yes));};
filter f_filter3 { facility(authpriv); };
log { source(s_sys);
  filter(f_filter3); destination(d_auth); };
```

option 부분에서는 manager와 agent를 동기화시키기 위해서 sync(0)을 사용하였고, source 부분에서는 manager의 /dev/log 폴더에 로그 메시지가 쌓이도록 설정하였으며, UDP 514번 포트를 그대로 사용하였다.[3] destination 부분에서는 로그 메시지를 받아서 facility 설정에 따른 분류 작

업에 의해 /var/log/ 하위 폴더에 관련 내용을 저장하도록 설정하였다. 로그 메시지를 저장하는 경우에는 데이터베이스에 바로 저장하기 위해서 메시지의 내용을 데이터베이스 쿼리문으로 수정하였다. filter 부분에서는 facility와 로그 레벨에 따라 분류하여 설정하였으며, 이러한 내용을 마지막의 log 부분에 정의하였다.

syslog-ng에 의해 생성된 파일을 처리하기 위한 gathering 프로그램에서 파일을 처리하기 위한 전역변수는 다음과 같이 선언한다. 앞에서 설명한 대로 syslog-ng에 의해 만들어진 로그 파일은 DATADIR로 정의하고, 이것을 임시 폴더로 이동시킨 것을 TMPDIR, gathering 프로그램에 의해 생성된 로그 데이터를 RESULTDIR, 데이터베이스에 최종 삽입하기 위한 파일이 LOGFILE이다.

```
#define DATADIR "/log/gathering/syslog/"
#define TMPDIR "/log/gathering/systmp/"
#define RESULTDIR "/log/gathering/result/"
#define LOGFILE "/log/gathering/
    syslogGathering.log"
```

이렇게 선언한 변수에 대해서 main() 함수 내에서 char 데이터 타입으로 배열을 생성하여 사용하였다.

```
char oldfile[80] = DATADIR;
char newfile[80] = TMPDIR;
char resultfirewall[80] = RESULTDIR;
char resultnormal[80] = RESULTDIR;
```

syslog-ng에 의해 만들어진 파일이 있는지를 검사하기 위해 scandir() 함수를 사용하고, 이 함수에 의해서 폴더 내에 존재하는 모든 파일과 폴더 목록을 알아 낼 수 있다. 만약 파일이 존재하면 부모 프로세스가 자식 프로세스를 만들어 그 파일을 처리하게 하고, 다시 파일이 있는지를 검사하게 된다.

데이터베이스는 PostgreSQL을 사용한다. 먼저 일반 로그 메시지와 방화벽에서 전송된 로그 메시지를 구분하여 logs, logsfirewall에 저장되도록 데이터베이스 스키마를 디자인하였다. 또한 네트워크에서 sys log 메시지를 전송하는 장비를 관리하기 위해 IP 주소를 등록하도록 하였으며, 네트워크의 장비 종류도 데이터베이스에서 관리할 수 있도록 디자인하였다. [표3]은 생성된 테이블 목록이다.

표3. 테이블 목록
Table.3 Table name in database

테이블명	테이블 설명
logs	일반 로그 처리
logsfirewall	방화벽 로그 처리
devchar	네트워크 장비별 IP 관리
device	네트워크 장비 종류

logsfirewall table의 구조를 보면, SNMP의 이론적 배경이 되는 agent IP 주소, facility, 로그레벨, 메시지 내용 등은 logs 테이블과 동일하지만, 방화벽으로부터 전송되는 로그 메시지이기 때문에 NAT (Network Address Translation) 주소, PAT (Port Address Translation)을 사용하는 경우의 PAT 포트, 프로토콜, 방화벽 메시지 내용 등을 저장하기 위한 필드를 추가하였다.

다음은 데이터베이스에서 실행되어야 하는 쿼리문을 파일에 미리 저장해 두었다가 일괄적으로 쿼리를 수행하는 원시코드이다. 시스템의 자원 상태를 확인하는 코드는 생략하고, 일정 시간간격으로 생성된 파일을 확인하여 데이터베이스에 쓰도록 하였다.

```
remove(newfile);
strcpy(psqlnormal,"psql -U logsman -d log <
");
strcat(psqlnormal, resultnormal);
system(psqlnormal);
```

UNIX 시스템에서 쉘 명령어를 사용하여 쿼리문으로 작성된 로그 메시지를 일괄적으로 데이터베이스에 삽입(insert)하는 문장이다.[4] 일반 로그와 방화벽 로그를 따로 저장하기 때문에 전체 코드 안에서는 두 번 수행되게 된다. 위 문장은 백업 파일로부터 데이터베이스를 복구하는 경우에도 많이 사용하는 쉘 명령어이다.[5]

III. 웹기반 뷰어 프로그램 개발

네트워크 관리자가 데이터베이스에 저장된 로그 메시지를 적당한 쿼리를 통해 분석할 수 있도록 프로그램을 개발하였는데, 접근성을 높이기 위해 웹 브라우저를 통해 내용을 확인할 수 있도록 구성하였고, 방화벽에서 생성된 로그 메시지 자체가 보안을 강화해야 하는 항목이기 때문에 SSL(Secure Socket Layer)을 통해 manager에 접근하도록 구성하였다. SSL은 웹서버와 클라이언트 사이에 개인키/공개키 암호화 기법인 RSA를 사용하여 데이터를 암호화하여 전송한다.[6] 일반 웹 서버가 80번 TCP 포트를 사용하여 http://로 접속하는 반면에, SSL은 443번 TCP 포트를 사용하고 https:// 방법으로 접속하여야 한다.

PostgreSQL에 저장된 로그 메시지를 네트워크 관리자에게 효과적으로 보여주기 위해서 전체 구조를 프레임으로 분할하였으며, 왼쪽에 메뉴를 구성하고, 오른쪽 위에 세부 메뉴를 선택할 수 있도록 구성하였다. [그림2]는 이러한 뷰어 프로그램의 전체적인 구조를 보여주고 있다.

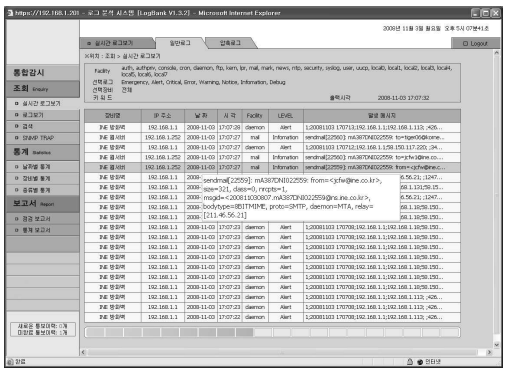

그림 2 뷰어 프로그램 레이아웃
Fig.2 View program layout

실시간 로그 보기는 PostgreSQL 데이터베이스의 특성상 20초 정도의 시간 지연이 발생하도록 구성하였다. gathering 프로그램에 의해 쓰기 작업이 진행 중인 경우에는 약간의 시간 차이를 설정해야 작성중인 테이블에서 조회 작업이 효율적으로 수행되기 때문이다. 일반로그와 압축로그로 분류한 것은 스위치나 라우터와 같은 네트워크 장비에서 발생하는 로그와 방화벽, IDS, QoS와 같은 네트워크 보안 관련 장비에서의 로그 메시지를 따로 관리하기 위한 것이다.

IV. 성능 분석

UNIX를 기반으로 하는 IBM SYSTEM p5 505 모델에서 RPM 패키지로 제공되는 syslogd을 이용하였을 경우에는 방화벽에서 발생되는 로그의 일부가 소실되는 경우가 발생하였다. 퓨쳐시스템에서 생산하는 SecuwayGate2000 제품을 방화벽으로 사용하였는데, 리눅스용 프로그램 juno을 이용하여 사용하지 않는 특정 포트로 다량의 패킷을 전송하였고 따라서 방화벽에서 패킷을 차단하는 것을 확인하였다. 방화벽에서 차단되는 패킷은 로그 메시지를 SNMP manager로 전달하게 되는데, syslogd와 syslog-ng의 차이는 거의 발생하지 않았다. [표4]는 테스트한 서버의 사양을 비교한 표이다.

표4. 시스템 비교
Table.4 System comparison

종 류	Secuway Center2000	Gathering Server
Processor	Intel Core-2 Dual Processor, 2.2GHz	
HDD	SATA-2, 7200rpm 160G	
Memory	DDR2 PC2700 2Gbyte	
Operation System	Windows 2003 Server	Linux (Cent-OS 5.0)
Socket Program	unknown	syslog-ng
Language	unknown	gcc
Database	MS-SQL7.0	PostgreSQL

IBM SYSTEM p5 505 서버는 고성능이고 따라서 많은 비용을 지불해야 하는 반면에, 똑같은 성능을 발휘하는 프로그램을 Gathering 서버에서도 구현할 수 있도록 프로그래밍 하였다. MS-SQL을 비롯한 고성능의 컴퓨터 사양에서는 프로그래밍을 수행하는 경우에 소켓 프로그래밍이나 쓰레드를 사용하지 않고서도 실행되는 결과가 만족스러운 경우가 종종 있다. 그러나 상대적

으로 하드웨어의 성능이 낮은 컴퓨터에서는 실행 상에 속도 저하가 생기는 경우가 많이 발생한다.

퓨쳐시스템에서 생산되는 SecuwayGate2000 제품은 1초에 처리되는 패킷의 양이 20만 건이라고 한다.[1] 즉, 1초에 20만 개의 패킷이 들어와도 보안 정책에 따라 패킷을 통과시킬지, 차단할 지를 판단할 수 있다는 것이다. 또한 이 제품은 SecuwayCenter 2000 제품과 연동하여 방화벽에서 생기는 각종 정보를 관리 서버로 전송하여 데이터베이스에 저장하도록 구현이 가능하다. [그림 3]은 성능 분석을 위한 네트워크 구성도이며, 원활한 테스트를 위해 패킷을 발생시키는 컴퓨터를 Server_A와 Server_B와 같이 두 대로 구성하였다.

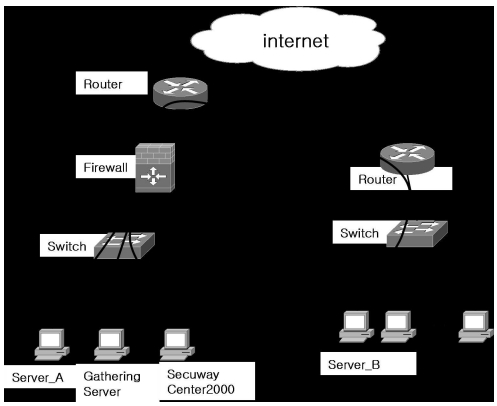

그림 3 네트워크 구성도
Fig.3 Network configuration

방화벽에서 발생되는 로그 메시지를 서버에서 받아서 데이터베이스화 하는 구조를 이루고 있으며, SecuwayCenter2000에서는 퓨쳐시스템 제품이 아닌 다른 네트워크 장비에서 발생하는 로그 메시지를 처리할 수 없다.[1]

본 연구에서는 퓨쳐시스템의 방화벽은 그대로 사용하였지만, syslog-ng를 통해 로그 메시지를 받아서 파일로 생성하도록 하였고, 이렇게 생성된 로그 메시지를 데이터베이스화 하는 Gathering 프로그램을 개발하였다. 패킷을 발생시키는 컴퓨터에서는 리눅스 기반 juno 프로그램을 사용하였으며, Server_A와 Server_B를 통해서 약 5Mbps의 패킷을 1시간동안 발생하도록 프로그래밍 하였다.[5] 이렇게 발생된 패킷을 방화벽에서 처리하게 되고, 이때 발생되는 로그 메시지를 SecuwayCenter2000 서버와 Gathering Server에서 각각 데이터베이스에 저장하도록 하였다. [표 5]는 실행된 결과를 나타낸 결과표이다.

결과표에 의하면 Gathering 서버의 프로세서 사용량이 멀티프로세싱에 의해 정확하게 분배되는 것을 확인할 수 있었던 반면에, SecuwayCenter2000 서버에서는 평균값을 비교했을 때에 약간의 차이가 발생하는 것을 확인할 수 있었다.

표5. 성능 분석 결과표
Table.5 Performance analysis result

분석 항목	Secuway Center2000	Gathering Server
Processor 사용량	49%, 43%	38%, 38%
Memory 사용량	345Mbyte	220Mbyte
file 사용량	Unknown	750Mbyte
데이터베이스 사용량	312Mbyte	452Mbyte

또한 메모리 사용량을 비교해 보면, Secuway Center2000 서버보다 Gathering 서버의 사용량이 더 적은 것을 알 수 있다. 이것은 Windows 2000 서버의 가상메모리 공간과 Linux 서버의 스왑 공간과의 차이 일 수도 있으며, 따라서 결과 값을 알아내기 위해서 프로그램을 실행하기 전 상황과 프로그램을 실행한 후의 상황을 모니터링하여 전체적인 사용량을 결과에 반영하였다.

파일 사용량은 폴더 내에 만들어지는 임시 파일의 량을 측정하였으며, Secuway Center2000의 경우에는 파일 사용량이 얼마나 되는지 알 수 없다.

데이터베이스 사용량은 MS-SQL에서 데이터베이스의 실제 사용 공간을 검사하였고, Gathering 서버의 경우에도 실제 사용 공간을 검사하였다. 데이터베이스 사용공간에 차이가 발생하는 것은 PostgreSQL 데이터베이스의 데이터 스키마의 특성인 것으로 파악된다.

또한 SecuwayCenter2000 서버는 방화벽에서

발생하는 로그 메시지를 직접 받아서 데이터베이스에 바로 삽입하는 구조를 이루고 있는 반면에, Gathering 서버는 syslog-ng에 의해 전송된 로그 메시지를 파일에 임시 저장하였다가 gathering 프로그램에 의해 PostgreSQL 데이터베이스에 삽입하는 구조를 이루고 있다.

V. 결 론

본 연구에서는 이러한 많은 네트워크 보안 관련 장비에 대한 효율적인 관리를 목적으로 각각의 네트워크 보안 장비에서 보내는 로그 메시지를 통합 관리하기위한 방법을 제시하였다. 물론 현재에는 방화벽과 같은 장비와 장비를 관리하기 위한 서버를 별도로 구축하는 것이 아니라, 두 가지를 결합한 네트워크 통합 보안 제품들이 많이 출시되고 있다. 그러나 이러한 장비들은 고성능의 하드웨어를 필요로 하며, 초기 구축을 위한 비용이 만만치 않게 소요된다.

또한 하나의 네트워크 방화벽에서 발생되는 로그를 하나의 통합 서버에서 관리하는 것이 아니라, 네트워크상에 존재하는 모든 장비에서 전송되는 syslog 메시지를 한 대의 서버에서 관리하고자 하는 취지에서 본 연구를 시작하였다.

퓨쳐시스템의 대표적인 방화벽인 SecuwayGate 2000 제품과 이 제품을 관리하기 위한 Secuway Center2000 제품을 선정하여 테스트를 진행하였으며, 하드웨어 사양을 동일하게 유지하면서 로그 메시지를 데이터베이스에 저장하기 위한 방법을 연구하였다. 웜이나 악성코드에 의해 실행되는 대용량의 패킷은 네트워크 보안 제품의 자원을 소진시킬 수 있을 정도로 강력하다. 또한 한 대의 시스템이 아닌 다수의 시스템에서 전송되어지는 많은 패킷을 방화벽은 막아내야 하고, 이것이 방화벽이 존재하는 이유이다. 문제는 방화벽이 패킷을 통과시키거나 차단하는 정책이 올바르게 실행되었는지 판단할 수 있는 로그 메시지를 적절한 방법으로 manager에게 전달해야 하며, 방화벽이 다운되는 경우와 같은 응급 상황에서 로그의 내용을 분석하여 향후 외부로부터의 공격에 대비해야 한다.

이러한 로그 메시지를 효과적으로 데이터베이스에 저장하기 위해 gathering 프로그램을 개발하였고, 데이터베이스에 존재하는 로그 메시지를 조건에 맞게 검색하기 위한 네트워크 관리자용 뷰어 프로그램을 개발하였다. 로그 관리를 위한 통합 시스템을 구축하는데 많은 비용을 지불하지 않아도 되도록, 비교적 일반적인 사양의 서버를 선정하였고, 오픈 소스인 Linux, Apache 웹 서버, PHP, PostgreSQL 등을 이용하였다.

참고문헌

[1] 퓨쳐시스템, 『방화벽/VPN 솔루션』, 2006
[2] Douglas Mauro, Kevin Schmidt, 『Essential SNMP, Second Edition』, O'Reilly, 2005
[3] Balabit IT Security, 『syslog-ng v3 administrator guide』, 2007
[4] ISHII TATSUO, 최준호 역, 『POSTGRESQL』, 영진출판사, 2001
[5] 우재남, 『뇌를 자극하는 Redhat Fedora: 리눅스 서버 & 네트워크』, 한빛미디어, 2005
[6] 김재선, 『About Firewall & Network Security』, 영진닷컴, 2006

저자약력

문세광(Sae-Kwang Moon)
1998. 충북대학교 학사
컴퓨터공학과
2009. 한국기술교육대학교 석사
전기전자공학과 정보통신전공

<관심분야>
네트워크 프로그래밍, 네트워크 보안

윤한경(Han-Kyung Yun)
(미)Southern Illinois Univ. 석사
(미)Southern Illinois Univ. 박사
현재 한국기술교육대학교
정보미디어공학부 교수

<관심분야>
인공지능, HCI, Haptic