리눅스 기반 침입 방지를 위한 로그 분석 방법 연구★

임성화* · 이도현** · 김점구***

요 약

보안성 향상을 위한 안전한 리눅스 시스템은 자료의 불법적인 유출과 위·변조를 막고 사용 원칙에 위배되는 행위의 추적을 위한 감사(audit)능력을 가지고 있어야 한다. 또한 시스템 관리 및 운영자의 책임과 사용자의 행위를 명확히 구분 지울 수 있는 로그관리가 반드시 이루어 져야 할 것이다. 본 논문에서는 리눅스 시스템의 보안 로그를 분석하여 침입차단 및 탐지에 활용하는 방법을 제안하였다. 이를 통해 시스템의 침입차단 상태와 침입탐지 상태, 그리고 파일 시스템의 무결성 변화를 실시간 확인하여 신속히 시스템의 문제를 해결할 수 있어 시스템의 신뢰성 향상에 크게 기여하게 될 것이다.

Methodology of Log Analysis for Intrusion Prevention based on LINUX

Sung-Hwa Lim* · Lee Do Hyeon** · Kim Jeom Goo***

ABSTRACT

A safe Linux system for security enhancement should have an audit ability that prohibits an illegal access and alt ernation of data as well as trace ability of illegal activities. In addition, construction of the log management and moni toring system is a necessity to clearly categorize the responsibility of the system manager or administrator and the u sers′ activities. In this paper, the Linux system′s Security Log is analyzed to utilize it on prohibition and detection of an illegal protrusion converting the analyzed security log into a database. The proposed analysis allows a safe manag ement of the security log. This system will contribute to the enhancement of the system reliability by allowing quick response to the system malfunctions.

Key words : LINUX, Intrusion Prevention, Ldg analysis, Firewall, Security

접수일(2015년 3월 9일), 수정일(1차: 2015년 3월 23일), 게재확정일(2015년 3월 28일)

★ 본 논문은 2014년 산학협동재단 학술연구비 지원으로 연구되었습니다.

* 남서울대학교/멀티미디어학과
** (주)유아이넷/연구소 선임연구원
*** 남서울대학교/컴퓨터학과

1. 서 론

빠른 속도로 발전하는 인터넷과 이 속에서 발생하는 보안 문제에 대처하고, 시스템과 네트워크를 안전하게 유지하기 위해 하루가 다르게 지속적으로 발견되는 위험요소에서 자신의 리눅스 시스템과 네트워크를 보호하기 위해 관리자는 항상 업데이트된 보안 정보를 가지고 시스템을 유지·관리해야 한다[4]. 이후 보안 강화를 위한 침입차단과 침입탐지용 보안 도구를 반드시 설치해서 시스템에서 발생하는 보안 로그를 감사(audit)하고 확인해야 한다.

리눅스·유닉스 계열이 윈도우 계열에 비해 다른 점은 로그에 대한 정책을 세분화하여 설정 할 수 있고 많은 양의 로그를 남길 수 있다는 점이다. 따라서 이 로그 정보를 효율적으로 모니터링하고 관리하는 것이 서버 관리자의 중요한 임무중의 하나이다. 그러나 끝도 없이 쌓여 가는 로그를 일일이 분석한다는 것은 불가능에 가까운 것으로 이를 효과적으로 관리해야 할 필요성이 제기 된다[4].

또한 현재 나와 있는 로그 분석 도구들은 대부분 시스템 내에서 발생하는 로그들을 데이터베이스화하지 못하고 관리자에게 콘솔상태에서 문자로 알려 주거나 메일을 통해 통보해 주는 기능만을 하게 된다. 그리고 파일로 저장되는 로그는 일정한 시간이 지난 후 자동으로 삭제될 것이고, 엄청난 양의 메일은 일일이 확인하지 못한다면 무용지물이 될 것이다.

본 논문은 보안 관리자가 효율적으로 시스템의 보안관리를 할 수 있도록 리눅스 기반 불법적인 침입을 사전에 탐지 및 차단하고 보안 로그를 분석하는 방법을 제안하여 이를 토대로 보안 로그 데이터베이스를 구축하고자 한다.

2. 관련연구

2.1. 리눅스 로그파일

리눅스에서 로그파일은 일반적으로 /var/log에 시스템의 모든 로그를 기록 및 관리하고 있다. 로그파일은 운영하는 서비스에 따라 차이가 조금씩 있지만 일반적으로 syslog.conf의 설정에 따라 달라진다. /var/log의 디렉터리 내에 있는 중요한 로그파일을 분석해보면, 우선 boot.log는 부팅 및 각종 서비스 시작 및 중지에 대한 기록을 가지고 있고, 부팅 시에 에러나 조치사항을 살펴보려면 이 파일을 참조한다. cron의 경우 cron 활동 관련 기록을 담고 있고 시스템의 정기적인 작업에 대한 로그를 가지고 있다. /etc/ 밑에 있는 각종 cron 관련 파일들이 시간별, 일별, 주별, 월별로 정기적으로 운영체제에서 작업을 해야 할 것을 수행하고 작업내용과 과정 등을 /var/log/cron파일에 기록하게 된다. 그리고 messages의 경우 커널 에러, 리부팅 메시지, 로그인 실패 등 시스템 콘솔에서 출력된 결과를 기록하고 syslog에 의하여 생성된 메시지도 기록되게 된다. 운영 체제에서 보내주는 메시지는 주로 콘솔을 통해 실시간으로 보여주게 된다[5].

2.2. Syslog

일반적으로 시스템내의 로깅 기능은 syslog를 이용하여 어떻게 사용할 것인가를 지정하게 된다. 시스템 로깅 프로그램은 시스템의 부팅 시 초기에 실행되어지고 이에 대한 설정은 /etc/syslog.conf를 이용하게 된다. syslogd는 리눅스에서 관리되는 모든 로그파일들을 관리, 설정하는 데몬 유틸리티라는 말이 된다. syslogd에 관련 파일들과 데몬 그리고 각 실행방법들에 대한 내용은 아래의 <표 1>과 같다.

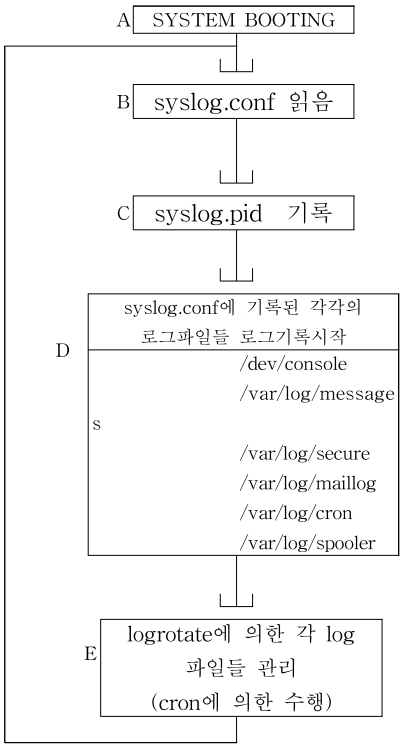
<표 1> Syslogd 관련파일과 데몬과 실행방법

구분	위치 및 실행 방법	설명
로그데몬 (위치)	/sbin/syslogd	로그데몬의 위치 및 데몬프로그램
로그데몬 설정 파일	/etc/syslog.conf	로그데몬의 설정파일, 각종 로그파일들의 설정 및 저장위치지정
로그데몬 PID 파일	/var/run/syslogd.pid	syslogd 데몬의 PID 파일
로그데몬 실행	/etc/rc.d/init.d/syslog start	로그데몬 실행방법
로그데몬 종료	/etc/rc.d/init.d/syslog stop	로그데몬 종료방법
로그데몬 재시작	/etc/rc.d/init.d/syslog restart	로그데몬 재시작방법

로그 데몬의 흐름도는 [그림 1]과 같다. 각 단계별 내용으로 먼저 A단계의 경우 syslogd 실행 또는 재시작으로 시스템이 부팅 되면서 처음으로 시작하게 된다. 또는 /etc/rc.d/init.d/syslog start 라는 명령의 수행으로/sbin/syslogd의 데몬프로세스가 수행된다. 다음

으로 B단계에서는 syslog.conf를 읽고 /sbin/syslogd 데몬이 실행이 되면서 /etc/syslog.conf 파일을 읽어들이게 되며, 이 /etc/syslog.conf 파일에는 시스템에서 사용하는 대부분의 로그파일들에 관한 설정이 되어있다. 그리고 C단계에서의 syslogd.pid를 기록하게 되고, /sbin/syslogd가 데몬이므로 이 프로세스의 실행번호(Process ID: PID)를 /var/run/syslogd.pid에 기록하게 된다. 다음 단계인 D단계의 경우 syslog.conf에 기록된 각각의 로그파일들의 로그기록을 시작하고 /sbin/syslogd의 실행과 함께 syslog.conf 파일에 설정되어 있는 각각의 로그파일들(messages, secure, maillog 등)에 로그가 기록되기 시작한다. 마지막으로 E단계에서는 logrotate에 의한 각 log 파일들을 관리하게 된다 [5].

[그림 1]의 전체적인 흐름을 보게 되면 리눅스 파일시스템이 어떻게 실행되어져서 로그파일에 저장이 되며 또한 logrotate에 의해서 어떻게 관리가 되어지는가를 쉽게 파악할 수 있을 것이다.



[그림 1] 로그 데몬의 실행 흐름도

2.3. 로그파일 분석과 관리

시스템관리에 있어서 가장 중요한 일 중 하나가 로그파일분석과 관리일 것이다. 시스템에 문제가 발생했을 경우 가장 먼저 시스템관리자가 확인해야 할 작업이 로그분석 작업일 것이다. 시스템이 침입이나 해킹을 당했을 때 해킹의 흔적과 기록을 확인하기 위해서 제일 먼저 서버관리자는 로그파일에 의존하여, 누가(ID), 어디서(IP Address), 어떻게(port) 들어와서 어떤 작업을 통해 불법적으로 침입을 했는지, 또는 정보를 유출하고 변경했는지를 확인하게 되고 이 모든 것을 전적으로 로그파일을 참조하여 이루어지게 된다. 시스템 내에 관리자가 관리해야 할 로그파일에는 기본적으로 시스템에 남겨지는 기본적인 로그들(syslogd에 관한 로그들)만도 여러 가지이지만 만약 웹 서버를 운영하고 있다면 웹 로그가 있을 것이고 메일서버를 운영한다면 메일로그파일이 존재할 것이다.

시스템 관리자는 시스템에 어떤 로그파일이 존재하며 해당 로그가 어떤 데몬(또는 프로세스)에 의해서 관리되어 지고 어떻게 로그가 남겨지는가를 정확히 알고 있어야 한다. 또한 로그파일 시스템이 어떤 경로로 남겨지게 되는지에 대해서도 정확히 알고 있어야 필요할 때 이를 신속하게 확인할 것이다[5]. 리눅스 시스템에서 제공하는 로그파일의 종류는 앞에서 언급했듯이 기본적인 로그파일만 10여개 정도이고, 보안 툴(예. Fcheck, Portsentry) 등을 설치하고 나면 이들에 관한 로그파일들이 새로 생성되게 된다. 기본적인 로그들은 syslogd에 의해서 제어가 되며, syslogd의 설정파일인 /etc/syslog.conf 파일을 수정함으로써 이 파일들의 저장위치와 저장파일명을 변경할 수도 있다. 본 논문에서 분석프로그램이 대상으로 하는 /var/log/messages 파일의 내용은 다음과 같다.

```
Nov 24 04:02:27 linux syslogd 1.4.1: restart.
Nov 26 22:08:48 linux sshd(pam_unix)[19027]: session
Nov 26 22:08:55 linux 11월 26 22:08:55 su(pam_unix)[1
501)
Nov 27 00:59:25 linux sshd(pam_unix)[19027]: session
Nov 27 00:59:25 linux 11월 27 00:59:25 su(pam_unix)[1
Nov 27 03:52:47 linux portsentry[736]: attackalert: T
pc.net/142.177.140.135 to TCP port: 1080
Nov 27 03:52:47 linux portsentry[736]: attackalert: H
 with string: "ALL: 142.177.140.135"
```

해당 로그파일의 의미는 월, 일, 시간으로 나누고 해당 프로그램이나 데몬의 이름과 프로세스 넘버가 오게 되고 사용자 로그인의 경우 해당 사용자의 uid가 기록된다. 메시지 부분은 해당 로그가 생성될 때 어떠한 이벤트가 발생했는지를 자세하게 보여 준다.

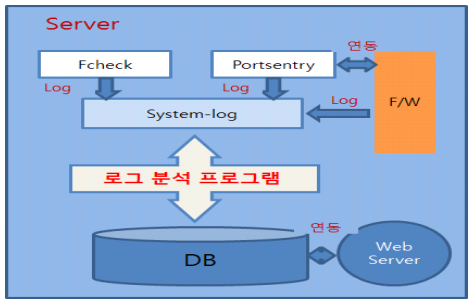
로그파일의 관리적 측면에서 살펴보면 로그 파일의 종류는 참으로 많다. 따라서 이런 로그파일들이 디스크공간을 차지하는 비율 또한 높다고 할 수 있다. 실제로 “파일 시스템 풀(Filesystem Full)” 로 인해서 시스템이 다운된 경우가 발생한다. 로그 파일의 크기가 늘어나서 불필요한 하드디스크 공간이 낭비되고 심지어 서비스가 중지될 정도로 심각한 상황이 초래될 수가 있다. 더 나아가 서비스하는데 투여되어야할 시스템 자원이 로그파일을 기록하는 데에만 집중되어 정작 서비스 자체의 속도나 질이 형편없이 떨어질 가능성도 크다.

리눅스에서 로그로테이션을 담당하는 패키지가 log rotate이다. logrotate는 계속 커지는 파일을 효율적으로 관리하기 위한 프로그램이다. 자동으로 로테이션을 운영하고, 압축, 제거, 메일로 보내주기 등의 작업을 하게 된다. logrotate는 초기 리눅스 설치 시에 자동으로 cron에 등록되어 주기적으로 실행된다[4]. 리눅스 로그파일 관리 툴인 logrotate을 이용하면 다음과 같은 작업을 할 수 있다. 첫째, 로그 파일을 rotate 시킨다. (지정한 양이나 기간이 지나면 다른 파일로 대체한다.) 둘째, 로그파일을 압축해서 보관한다. 세 번째로 작업 시 에러가 발생했을 경우 지정한 메일주소로 상태를 보고한다. 따라서 시스템관리자는 로그파일에 대한 중요성을 인지하고 있음과 함께 관리에도 만전을 기해야 할 것이다.

3. 보안 로그 분석 방법 설계

본 논문에서 제안하는 보안 로그 분석 방법은 리눅스 기반 공개용 보안도구를 이용하여 침입탐지 시스템을 구현하고 침입차단 시스템과 연동 하여 시스템의 보안 상태를 인터넷으로 실시간 확인할 수 있는 불법행위 탐지 시스템이다.

[그림 2]는 본 논문에서 제안한 시스템의 전체구조를 도식화한 그림이다.



[그림 2] 시스템 운영도

제안 시스템의 특징은 다음과 같다. 첫째, 공개된 보안 프로그램을 이용 침입탐지 시스템과 침입차단 시스템의 구축과 연동이 가능하다. 둘째, 기존 시스템 내에 파일로 저장하던 보안 로그를 직접 제작한 프로그램을 통해 분석하여 데이터베이스를 구축한다. 셋째, 추후 구축된 데이터베이스와 웹 서버를 연동하여 원격에서 웹을 통해서도 모니터링 할 수 있게 확장 가능하다.

3.1. 파일 무결성 변화 탐지(Fcheck)

fcheck는 파일과 디렉터리 또는 파일 시스템의 추가 및 삭제 그리고 변경사항을 모니터 할 수 있는 공개 보안도구 이다. 시스템 내부에 크래커의 침입으로 중요 시스템 파일의 변경이나 삭제, 백도어 프로그램의 설치 등으로 인해 발생 할 수 있는 보안 문제를 주기적인 모니터를 통해 감지할 수 있는 리눅스 기반 호스트 IDS 프로그램이다. fcheck는 모니터 하고자 하는 파일 시스템이나 디렉터리 파일 등을 정책파일(policy file)에 등록하고 데이터베이스 파일(system snapshots)을 생성하여 이후 데이터베이스 파일을 기준으로 시스템의 무결성을 검사하게 된다. fcheck는 standard PERL로 만들어 졌고, 기본 적인 문법은 Tripwire을 참고하고 있다. 또한 GNU 라이센스를 따르고 있어 누구나 자유롭게 사용할 수 있다. fcheck는 데몬이나 자동실행을 지원하지 않으므로 명령어를 직접 실행하거나 cron에 등록하여 주기적으로 시스템 변경사항을 체크하여야 한다.

fcheck 사용 시 통상 파일의 추가, 삭제 및 변경이

빈번한 /home 이나 /var등과 같은 파티션 보다는 /etc / 나 /sbin/ 등 중요 시스템 파티션을 설정하여 검사하는 것이 보다 효율적인 운영 방법일 것이다. 또한 fcheck를 사용할 때 특별히 추가 설치해야 할 프로그램이나 라이브러리 모듈 등은 없다.

3.2. 포트스캔 탐지(Portsentry)

Portsentry는 모든 시스템에서 서비스하거나 또는 열린 모든 포트를 모니터링 하는 호스트 IDS이다. TCP, UDP로 오는 포트 스캔을 감지 할 수 있고, 침입 차단 프로그램과 연동 하여 이를 방어를 할 수 있다. 포트스캔이 직접적인 공격은 아니지만 불법적인 침입을 위한 사전 단계로 SYN, NULL, FIN Scan등을 감지할 수 있고, 방어하는 방법은 아래와 같다.

가. TCP Wrappers를 이용한 xinetd를 사용하는 daemon들의 보호.
나. ipfwadm, ipchains, iptables을 이용하여 공격자로 오는 모든 패킷을 거부.
다. /var/log/message에 로그 기록

Portsentry의 동작은 항상 데몬으로 포트스캔을 실시간으로 검사하게 된다. 포트스캔이 들어오게 되면 시스템로그에 자동으로 기록되고 이때 침입차단 시스템인 iptables와 연동 하여 해당IP의 모든 패킷을 막게 되고 TCP Wrappers와도 연동하여 xinetd가 관리하는 모든 데몬 서비스의 접근이 차단 되게 된다.

3.3. 제안 보안 로그 분석 방법

본 논문에서 제안하는 로그 분석 프로그램의 특징은 다음과 같다. 첫째, 표준 PERL로 만들어져 수정 및 구조가 간단하다. 둘째, 특정 프로그램이나 로그에만 사용하는 것이 아닌 다양한 부분에 응용되어 사용될 수 있다. 셋째, 시스템 설정에 의해 메시지가 시간 단위로 messages.1으로 백업되는 공백시간 사이를 체크하여 완전한 분석이 이루어지도록 했다. 넷째, DB에 분석된 값이 중복 입력되지 않도록 lastline이라는 파일을 만들어 최초 분석 시 전체를 분석하고 lastline 파일을 만들어 이후 분석 시에는 이 값을 비교하여 중복되는 불필요한 작업을 줄였다. 다섯째, cron에 바로 등록해 주기적인 로그 분석이 가능하다.

프로그램의 전체 동작과정을 보여주는 다음 알고리즘은 프로그램이 시작되면 이전에 작업한 시스템 로그 파일의 마지막 줄을 기록파일로부터 읽어 들인다. 만약 이전 작업내용이 없다면 최초 작업이므로 분석을 바로 시작하게 되고, 이전 작업내용을 확인하고 로그 파일의 라인을 한 줄씩 호출하여 패턴과 비교하게 된다. 패턴과 일치하는 부분을 만나면 해당 라인을 DB 형식에 맞게 세분화하여 저장하게 된다. 마지막으로 현재 작업한 시스템로그의 마지막을 기록파일에 저장하여 다음 분석 시에 사용한다.

```
# include "이전 작업 로그 파일"
main()
{
  if (최초 작업이 아니면)
    {    do
         시스템 로그 라인 분석;
     while ( 이전 로그 != 시스템 라인 로그)
          }
       로그 분석 시작;
       메시지 라인 분석;
       로그 분석 DB 저장;
}
```

다음은 로그분석 프로그램의 소스분석에 관한 설명이다[8].

```
$lastline = '';
$lastlinefile = 'lastline';
open LAST, $lastlinefile or die "can't open $lastlinefile";
$lastline = <LAST>;
close LAST;
```

lastline 파일을 읽어서 $lastline 변수에 할당하는 부분이다.

```
$start = 0;
if (!$lastline) {
    $start = 1;
}
```

처음 $start 변수의 값을 0으로 설정하고, lastline 파일의 값이 없다면 $start의 값이 1이 된다. 즉, lastline 파일에 값이 없다는 것은 perl을 처음 실행했다거나 이전에 로그가 없다는 것을 말한다. 로그가 없거나 perl이 처음 실행할 경우 start를 1로 주어 로그 분석 if문을 실행하게 된다.

/var/log/messages.1 파일과 /var/log/messages 파일을 각각 배열로 만들어 2번 실행하게 한다. 이유는 시스템 설정에 의해 messages 파일이 주기적으로 백업되므로 이때의 시간 간격 동안 침입이나 무결성에 문제가 발생할 경우 /var/log/messages파일만 분석한다면 /var/log/messages.1로 넘어간 로그 부분을 분석할 수 없게 된다. 그래서 foreach 문을 사용하여 각각의 파일에 대해 2번 실행하게 된다.

```
foreach $log (@log) {
    open LOG, $log or die "can't open $log file";
    while ($line = <LOG>) {
        chomp $line;
        $curline = $line;
    }
    close LOG;
}
```

foreach 문 안의 while 문은 LOG 파일 할당자에 할당된 messages 파일의 라인 한줄 한줄을 \값을 제거하고 $curline 변수에 넣게 된다.

```
if ($lastline eq $curline) {
    $start = 1;
    next;
}
```

중요한 부분으로 $lastline 과 $curline 의 값이 같다면 $start 값을 1로 주고 next로 while문을 다시 실행한다. 이 부분이 의미하는 것은 messages 파일의 처음라인 부터 계속 읽어 오다가 lastline파일의 값과 messages 파일에서 읽어온 값이 같아지는 부분에서 $start = 1로 주고 next를 하여 다음 while문이 실행할 경우 다음 if문을(start=1 이고 eq가 아니므로) 실행하여 로그를 분석하게 된다. 결과 적으로 lastline 파일의 값과 messages의 마지막을 비교해서 다음 줄부터 분석해 DB에 저장하게 된다.

```
($month,$day,$time,$host,$msg)=$line=~
/^(\w+)\s+(\d+)\s+(\S+)\s+(\w+)\s+(.*)/;

$category = 'fcheck' if $msg =~ /fcheck/;
$category = 'iptables' if $msg =~ /iptables/;
$category = 'portsentry' if $msg =~ /portsentry/;
```

이 부분은 로그 파일을 직접 분석해서 DB에 넣기 위한 변수에 값을 할당하는 부분이다. 제어문자가 아닌 일반문자 1개 이상인 것과 공백, 숫자 등을 $month 등의 변수에 넣고 $msg 변수에서 해당 보안 프로그램의 로그 부분을 찾아서 $category의 값을 설정하게 된다.

```
$sth = $dbh->prepare("INSERT INTO log2
(category,month,day,time,msg)
VALUES ('$category','$month',$day,'$time','$msg')");
$sth->execute;
```

statement 핸들러로 db 핸들러를 이용 db에 해당 값을 입력하는 부분이다.

```
open LAST, "> $lastlinefile" or die "can't
open $lastlinefile file";
print LAST $curline;
close LAST;
```

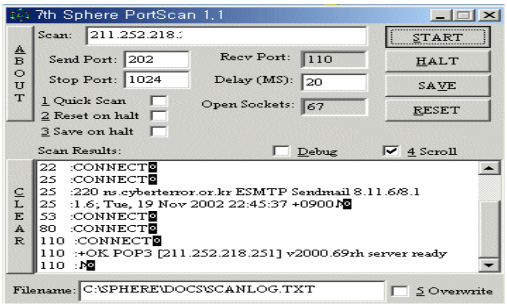
마지막으로 $lastlinefile가 가리키는 lastline파일에 messages의 마지막 부분인 $curline을 > 로 입력한다. 이 값을 이용해서 다음에 분석 프로그램을 실행할 경우 이 값을 가지고 /var/log/messages.1 파일과 /var/log/messages과의 값과 비교하게 된다.

3.4 로그 분석 프로그램 구현

로그 분석 프로그램의 실행은 명령어 라인에 직접 입력해야 한다. 하지만 관리자가 매 시간 매분 프로그램을 직접 실행 할 수 없으므로 cron에 등록해서 분석하도록 한다. 매 15분마다 로그 파일을 분석해서 데이터베이스에 결과를 저장하도록 한다. 하루나 주 단위로 로그를 분석하고자 할 때는 간단한 셸 프로그램을 이용해서 실행 권한을 주고 /etc/cron.daily 나 weekly에 넣어둔다.

```
'0,15,30,45 * * * * ~/project.pl'
```

포트스캔(Portsentry) 분석은 윈도우즈용 Sphere PortScan 1.1을 이용해서 포트스캔을 다음과 같이 실행했을 경우 크게 세 가지의 로그를 확인 할 수 있다.



[그림 3] Sphere PortScan 1.1

Portsentry자체 로그와 TCPwrapper의 연동부분 그리고 iptables와 연동 하여 차단된 IP와 침입시도 상황을 Portsentry 영역(category)을 선택 후 시스템에 대한 포트스캔 로그를 확인할 수 있다. [그림 3]의 Sphere PortScan 프로그램은 해킹사전단계에서 침입 대상시스템의 정보 수집을 위해 윈도우 기반의 대표적 포트스캐너이다.

[그림 4] Sphere PortScan 1.1의 사용방법은 Scan에 IP주소를 기입하고 Send Port에 시작 Port number와 Stop Port에 Scan하고자 하는 Port까지의 숫자를 적고 Start를 누르면 간단히 Scan할 수 있다.

```
portsentry[11721]: attac
kalert: TCP SYN/Normal scan from host: 211.62.3    /211.62.3    to TCP port: 15
```

[그림 4] PortScan침입 로그의 확인

[그림 5]는 IP(211.62.3*.***)에서 tcp에 15번 포트로 포트스캔을 실행한 로그를 확인 할 수 있다. Portsentry는 포트스캔을 실행한 해당 ip를 Tcpwrapper에 deny에 등록하여 xinetd가 관리하는 서비스 데몬에 접속할 수 없도록 차단한다.

```
portsentry[1345]: attack
alert: Host 211.62.36.4 has been blocked via wrappers with string: "ALL: 211.62.36.4"
```

[그림 5] Tcpwrapper의 접속거부 IP확인

서비스 접근을 차단하고 침입차단 프로그램인 Iptables과 연동 하여 해당 IP를 등록하여 입력되는 모든 패킷을 차단하게 된다.

```
portsentry[1345]: attack
alert: Host 211.62.3    has been blocked via dropped route using command: "iptables -I INPUT -s
211.62.36.4 -j DROP"
```

[그림 6] IPTables의 침입차단 IP확인

Iptables 로그는 접근이 차단된 시스템이나 port로의 연결을 모니터링해서 트래픽을 검사하고 연결을 막는 작업을 모니터 할 수 있다.

```
kernel: iptables block:
IN=eth0 OUT= MAC=00:50:8b:d3:76:f0:00:e0:52:c4:7f:00:08:00 SRC=211.237.2  .    DST=211.62.3 .
LEN=48 TOS=0x00 PREC=0x00 TTL=120 ID=8998 DF PROTO=TCP SPT=63984 DPT=23 WINDOW=16384
RES=0x00 SYN URGP=0
```

[그림 7] IPTable의 로그확인

[그림 7]은 소스(SRC)=211.237.*.*에서 목적지(DST)=211.62.*.* 의 23번 포트로 접근한 것을 차단한 로그이다.

4. 비교 분석

4.1. 기존 로그 분석 프로그램

가. Logchek

Logchek는 실시간으로 로그 파일을 분석하여 사전에 해킹 시도 등 비정상적인 상황이라고 정의된 내용이 로그에 남을 경우 해당 로그의 내용을 관리자에게 메일로 발송해 준다. 실시간 로그체크는 logtail이라는 프로그램을 이용하는데, 이 프로그램은 프로세스에 상주하여 실행되면서 체크한 파일의 끝부분을 기억하고 있다가 새로운 로그의 내용이 추가될 때마다 계속적으로 로그체크를 실행하게 된다.

나. Swatch

Swatch역시 Logchek와 유사하게 로그파일을 모니터링 하다가 사용자가 지정한 패턴이 확인되었을 때 해당 내용을 메일로 발송하거나 벨소리를 이용해 경고

를 나타낸다. 또한 특정 스크립트를 실행하도록 할 수 있는 기능이 포함 되어있다.

다. Colorlog

이름에서도 알 수 있듯이 복잡한 로그파일을 메시지의 내용에 따라 색을 다르게 하여 보여주는 프로그램으로서 사용방법이 매우 쉽다. Colorlog는 모니터 할 메시지를 적절한 색으로 설정한 후 실행하게 되며 보안에 문제가 되는 부분만 다른 색으로 보여주어 로그파일을 한눈에 쉽게 관리할 수 있도록 해 준다[4].

4.2. 제안시스템과의 비교

<표 2>에서와 같이 로그 분석방식의 차이에서는 모든 프로그램이 공통적으로 로그파일 내의 특정 단어나 문장의 패턴을 미리 만들어 두고 이와 일치하는 부분을 파싱하는 방식으로 동일한 방법을 가진다.

통지(Notify) 방식의 차이에서는 본 논문 제안 프로그램만이 유일하게 데이터베이스를 지원하고 있고 Logcheck 경우 e-mail만을 지원하며 Swatch 경우 e-mail과 경고음을 그리고 Colorlog는 단순히 로그파일내의 패턴과 일치하는 부분을 색깔로만 구분하여 나타낸다. 추가적으로 본 논문 제안 프로그램의 경우 웹 모니터링 프로그램과 연동 하여 모니터링 서비스를 지원할 수 있고 추후 데이터베이스를 이용하여 다양한 분석이 가능하다. Logcheck과 Colorlog의 경우는 추가적인 기능이 없으며 Swatch의 경우 추가적인 스크립트 파일을 실행할 수 있도록 지원한다.

<표 2> 분석방식에 대한 비교

로그분석방법	분석방식	통지방식	확장성
제안 방법	패턴분석 개별DB화	e-mail 경고음	DB연동 웹 모니터링
Logcheck	패턴분석	e-mail	없음
Swatch	패턴분석	e-mail 경고음	없음
Colorlog	패턴분석	Text의 색 구분	없음

결과적으로 해킹이나 불법적인 침입은 실시간 발견이 어렵다. 일정 시간이 지난 이후 침입이 확인되었을 때 침입당시의 로그를 메일이나 시스템 내에 파일로 저장할 경우 주기적으로 파일이나 메일은 삭제되므로 이후 비교 분석할 자료가 없어 보안 문제에 대처할 수 있는 방안이 줄어들게 된다. 또한 감사 자료가 없어지게 되는 결과를 가져오게 된다.

본 논문에서 제안하는 보안로그 분석과 로그데이터베이스 구축 시스템을 이용한다면 이러한 문제를 상당부분 줄일 수 있다. 또한 추가적으로 로그를 모니터링 해야 할 서버가 많을 경우 로그서버를 별도로 운영하여 모든 로그를 하나의 서버에 집중시켜 한 번에 관리할 수도 있다. 본 논문에서 제안한 시스템과 로그 관리프로그램을 이용한다면 대형 네트워크에서 효율적인 로그 관리가 가능해질 것이다.

5. 결 론

본 논문에서 제안한 로그 분석 프로그램은 Linux환경 하에서 침입으로 의심되는 Log에 대해 Log파일 내에 있는 내용만을 분석하여 메일이나 콘솔화면으로 관리자에게 알려주는 기존의 방식을 로그들에 대한 DB 구축을 통하여 보다 더 안전하게 모니터링 할 수 있는 LAP(로그분석 프로그램)을 구현함으로써 새로운 분석 시스템을 제시하였다.

기존의 다른 LAP를 이용한 로그분석 방식과 비교하여 본 논문의 LAP을 이용한 분석이 침입대응에 보다 효과적이라는 것이 다른 기존 LAP과 비교 분석을 통한 결과로 확인되었다. 로그 정보, 특히 본 논문에서 제안하는 침입 차단과 침입탐지, 파일시스템의 무결성 변화에 관련한 보안로그는 데이터베이스화하여 저장될 필요성이 있으며, 이를 이용하여 다양한 기법을 통한 분석이 가능해질 것이다.

본 시스템을 이용하여 향후 분석 영역을 넓히고 해당 영역의 분석패턴을 세분화 하여 리눅스 전반에 관련한 로그를 분석해 모니터 할 수 있도록 만들고, 시스템의 보안성 유지를 위해서는 항상 최신의 보안정보를 가지고 시스템을 패치 하고 문제점을 보안해야 할 것이다[3]. 여러 가지 종류의 취약점 분석 툴을 이용하여 주기적으로 네트워크와 시스템 내부를 점검하고 관련 로그를 모니터링 해야만 문제점에 빠르게 대처 할

수 있을 것이다. 물론 보안에 완벽할 수 있는 시스템은 없겠지만 본 논문에서 제안한 시스템을 사용한다면 귀중한 정보와 자산을 보다 안전하게 관리하는데 큰 도움이 될 수 있을 것이다.

참고문헌

[1] 리눅스 공동체 세미나 준비팀, 제5회 리눅스 공동체 세미나 강의록, 한빛미디어, 2011.

[2] 송관근, 리눅스에서 로그파일을 이용한 불법행위 탐지, 호서대학교 대학원, 2011.

[3] 반장호・홍석범, 리눅스 보안과 최적화 완벽솔루션, 한빛미디어, 2012.

[4] 박성수, 리눅스 서버관리 실무바이블 v1.5, 수퍼유저 코리아, 2012.

[5] 신현준, 리눅스 서버 구축 & 보안, 사이버 출판사, 2012.

[6] James Stanger Ph D,"Hack PROOFING LINUX", SYGRESS, 2012.

[7] 신영진, 국가・공공기관 침해사고 현황 및 대응실태, 제5회 정보보호 심포지엄, SIS2013, 2013.

[8] 이광준, 해킹에 대한 분석과 보안에 관한 연구, 국민대 산업기술대학원, 2001.

[9] 최길준 외 2명, 해킹과 보안 내가 최고, 영진. 2011.

[11] 한국정보보호센터, CERTCC-KR 통계, 2013.

[12] Anonymous, "Maximum Linux Security", SAMS, 2011.

[13] Mohammed J. Kabir,"Red Hat Linux 7 Server", powerbook, 2011.

[14] Mohammed J. Kabir,"Red Hat Linux Security and Optimization“, Openna, 2011.

[15] Paul Russell, "Linux IPCHAINS-HOWTO", http://www.linuxdoc.rog..

[저 자 소 개]

임 성 화 (Sung-Hwa Lim)

1999년 2월 아주대학교 정보및컴퓨터공학부 학사졸업
2001년 2월 아주대학교 정보통신공학과 석사졸업
2008년 2월 아주대학교 정보통신공학과 박사졸업
2013년 9월~현재 남서울대학교 멀티미디어학과 조교수

email : sunghwa@nsu.ac.kr

이 도 현 (Do Hyeon Lee)

2001년 2월 한양대학교 전자전기공학부 학사
2003년 8월 한양대학교 전자통신전파공학과 석사
2011년 2월 한양대학교 전자통신컴퓨터공학과 박사
2011년 4월 ~ 2014년 2월 남서울대학교 IT융합기술사업단 연구교수
2014년 3월 ~ 현재 ㈜유아이넷 책임연구원

email : dohyeon@gmail.com

김 점 구 (Jeom Goo Kim)

1990년 2월 광운대학교 전자계산학과 이학사
1997년 8월 광운대학교 전자계산학과 석사
2000년 8월 한남대학교 컴퓨터공학 박사
1999년 3월~ 현재 남서울대학교 컴퓨터학과 교수
IT융합연구소장

email : jgoo@nsu.ac.kr