通信情報保護學會論文誌
第 9 券 第 3 號, 1999. 09

시스템 보안 강화를 위한 로그 분석 도구 ILVA와 실제 적용 사례

조 상 현*, 차 성 덕*

ILVA : Integrated audit-log analysis tool and its application

Sang-hyun Cho *, Sung-deok Cha*

요 약

인터넷의 급속한 발전과 함께 정보 시스템의 보안 위협인 침입 사고도 급증하고 있어, 보다 강화된 보안 메커니즘이 요구되고 있다. 시스템 로그 분석은 이런 침입 사실을 탐지하고 침입자를 추적하기 위해 필수적인 과정이나, 로그 자료의 종류와 형태의 다양함으로 인해 자동화된 로그 수집 및 분석이 현실적으로 어려운 상태이다.

우리는 침입 추적에 필요한 로그 자료의 형태를 정의하고 방대한 로그 자료로부터 효율적으로 로그 수집, 분석할 수 있는 도구를 설계및 구현하였다. 이 논문에서는 개발된 도구를 사용하여 실제 침입 추적을 한 경험을 소개하고 도구의 향후 개선 방향을 제시한다.

ABSTRACT

Widespread use of Internet, despite numerous positive aspects, resulted in increased number of system intrusions, and the need for enhanced security mechanisms is urgent. Systematic collection and analysis of log data are essential in intrusion investigation. Unfortunately, existing logs are stored in diverse and incompatible format, thus making an automated intrusion investigation practically impossible.

We examined the types of log data essential in intrusion investigation and implemented a tool to enable systematic collection and efficient analysis of voluminous log data. Our tool, based on RBDMS and SQL, provides graphical and user-friendly interface. We describe our experience of using the tool in actual intrusion investigation and explain how our tool can be further enhanced.

* 한국과학기술원 전산학과 소프트웨어 공학연구실, 첨단정보기술연구센터

I. 서론

인터넷의 발전으로 컴퓨터를 통한 정보 공유가 활성화되고 있는 가운데, 시스템의 운영체제나 여러 응용프로그램이 가질 수 있는 소프트웨어적인 결함에 의해 많은 형태의 침입 유형이 생겨나 다양한 컴퓨터팅 환경에서 경제적인 손실이 발생하고 있다.

시스템 침입의 위험은 크게 비밀성의 손실, 무결성의 손실, 사용가능성의 손실등 3가지 측면으로 구분할 수 있다[1].

비밀성의 손실은 허용되지 않는 비권한자에 의한 자료의 접근및 복제를 말하며, 무결성의 손실은 비권한자의 자료 수정 및 삭제에 의한 손실을 말한다. 사용가능성의 손실은 침입자의 과도한 프로세스 수행등으로 인해 정상적인 작업이 이루어질 수 없는 손실을 의미한다. 이와 같이 시스템에 손실을 줄 수 있는 일련의 작업들을 침입(intrusion)이라고 하고 이러한 침입을 행하는 주체를 침입자(Intruder)라고 한다.

침입자에 의한 시스템 침해를 방지하기 위한 컴퓨터 시스템 보안관리모델[2]에서는 방어전략을 예방, 탐지, 추적, 복구의 4단계로 나누고 있는데, 실제로 많은 시스템 보안 연구들은 예방과 탐지 부분에 비중을 두고 진행되고 있다. 반면에 추적의 경우는 (Investigation)의 연구가 상당히 미흡하여 별도의 자동화된 분석 도구에 의존하기 보다는 직접 관련자료를 확보하여 수동적으로 분석하고 있는 실정이다.

대부분의 시스템이 가지고 있는 내부적인 결점들은 여러 형태의 침입으로 공격받을 수 있는데, 이때 시스템에서 일어나는 여러 상태에 대한 기록을 로그(audit trails 또는 log)라고 한다. 로그는 시스템에 문제가 발생하였을때, 문제의 원인을 확인하고 그러한 행위의 주체를 알아내는데 상당히 중요한 역할을 하고 있다.

시스템 운영체제에서 로그를 자세하게 생성할수록 얻을 수 있는 정보의 양은 많아지나, 그만큼 저장매체에 대한 비용은 많아지며 그러한 로그 자료를 수작업으로 분석하는데는 많은 시간이 필요하게 된다. 따라서 적절한 양으로 로그 자료를 생성하는게 필요하다.

일반적으로 시스템들은 많은 종류의 로그를 생성하고 있는데, 각각의 로그들의 양이 상당히 방대하기 때문에 이를 효과적으로 줄일 수 있는 방법에 대한 연구가 Purdue COAST[3]팀 뿐만 아니라 여러 곳에서 진행되고 있다. 그러나, 로그를 활용할 침입 탐지 시스템들의 요구 사항 즉 어떠한 정보들이 남겨져야 하는지에 대한 명확한 기준이 마련되지 않았기 때문에 로그 요약에 대한 연구들은 미흡한 실정이다. 따라서 많은 양의 로그 자료들을 바탕으로 관리자가 어떠한 정보를 얻어내기 위해서는 로그 자료들 사이의 연관 관계에 따라 관련 정보들을 동시에 검색할 수 있는 시각화된 도구가 필요하다.

이러한 도구를 활용한다면 침입 탐지 시스템과 함께 사용하여 침입 발견후 로그 자료들을 다각적으로 분석하여 침입 경로와 침입 행위들을 확인하는데 많은 도움을 줄 수 있다.

이 논문에서는 침입 경로를 확인하기 위해 로그 자료를 관리자 자신이 직접 수동적으로 검색하는 대신, 여러 형태의 질의를 통해 시스템에 분산되어 있는 많은 양의 로그 자료로 부터 유용한 정보를 제공해 줄 수 있는 자동화된 도구를 구현하여 실제 환경에서 적용해 도구의 장단점을 소개하고자 한다. 2장에서는 침입 탐지 시스템과 로그에 대한 개요를, 3장에서는 시스템 침입이 발생하였을때 관리자가 원하는 정보의 형태가 무엇인지 그리고 로그 분석 도구가 가져야 할 기능들을 명세하였다. 4장에서는 이를 위해 제안된 로그 분석 도구인 ILVA(Integrated Log Viewer and Analysis Tool)를 소개한다. 5장에서는 실제 환경에서 제안된 도구의 활용 성과와 침입 사건에 적용되었을때의 한계점에 대해 논의한다. 끝으로 6장에서는 자동화된 로그 분석도구의 개발의 효과에 이러한 연구의 필요성에 대해 언급한다.

II. 배경

침입 탐지란 시스템의 여러 사용 형태나 로그 기록을 바탕으로 시스템의 불법 침입을 실시간으로 탐지해 내는 것을 말하고, 이러한 기능을 하는 시스템을 침입 탐지 시스템이라고 한다.

침입 탐지 시스템은 단일 호스트기반, 다중 호스트기반, 네트워크기반의 침입탐지시스템으로 나뉘어 지며, 침입 탐지 모델에 따라서는 이상 탐지(Anomaly Detection), 오용 탐지(Misuse Detection)시스템으로 나누어 볼 수 있는데,대체적으로 정보수집, 정보가공 및 축약, 분석등의 세가지 단계로 구성된다[4].

이상 탐지 시스템은 시스템의 상태를 수치화하고

시스템 로그	설 명
messages	여러 종류의 경고 메시지
(p)acct	모든 사용자들에 의해 수행된 명령어
lastlog	사용자의 최근의 접속 기록
loginlog	접속 실패에 대한 기록
sulog	su명령어 사용에 대한 기록
utmp(x)	사용자의 현재 접속 상태를 기록
wtmp(x)	사용자의 과거 접속에 대한 기록

표 1. 일반적인 시스템 로그들

Tab. 1. General System logs

수치의 변화를 통계적으로 분석하여 침입등의 이상 상태를 판단하는 시스템인 반면, 오용탐지시스템은 기존에 알려진 침입패턴을 로그 기록에서 찾아 봄으로써, 침입 탐지가 이루어진다. 이러한 이상 탐지와 오용 탐지 시스템에 대한 연구가 활발히 이루어지고, 도구로 구현되어 여러 도메인에서 시험 운용되고 있다. 또한 이미 제안된 여러 탐지 모델이 적절한지를 객관적으로 판단할 수 있는 정형적인 검증이 부족한 상태이다[5].

대부분의 시스템은 로그 기록 보관의 비용을 고려하여 기본적인 접속 기록만을 보관하는 선에서 그치고 있어 효율적이고 보다 정확한 시스템 상황 분석을 위해서는 기록의 내용이 부족한 편이다. 물론, 최근에 와서는 별도의 로그 생성 프로그램을 설치하여 좀더 향상된 로그 자료기록 메커니즘이 제공되고 있지만, 여전히 충분치 못한 결점을 가지고 있다[5]. 첫째, 기존의 시스템이 갖는 로그자료 형태는 대체적으로 시스템 형태나 운영체제에 의존적인 관계로 되어 있어 침입탐지시스템이 다른 형태의 로그자료를 처리 하기 위해서는 별도의 전처리를 해야한다. 이러한 결점으로 인해 시스템에 독립적으로 동작하는 침입 탐지 시스템 개발과 이러한 시스템을 함께 활용함으로써 얻어지는 네트워크 안정에는 한계가 있다. 둘째, 로그자료가 갖고 있는 정보의 형태나 그 내용의 상세함이 침입 탐지 시스템에게는 불충분 할 수 있다. 어떤 시스템에서의 로그 자료가 작업의 주체(subject)에 대한 정보가 직접 기록되지 않아 별도의 작업이 필요할 수 있다. 예를 들어, CPU나 여러 자원의 사용량을 바탕으로 시스템의 서비스 거부를 확인하고자 할 때, 로그 자료에 이러한 정보가 기록되지 않는다면, 침입탐지 시스템은 무용지물일 수 밖에 없다. 따라서 시스템에 공통적으로 요구되는 로그자료의 표준화가 필요하다.

기본적으로 Unix계열의 운영체제는 (표 1)과 같은 로그 자료들을 생성한다. 이외에도 각 사용자의 프로세스를 기록해 확인할 수 있는 lastcomm등의 로그 자료가 있다. 이런 로그 자료들은 기본적으로 원시적인 형태으로 되어 있어 일일이 사람이 내용을 수동적으로 확인해야 한다. 따라서 상당한 크기의 로그 파일이라면 현실적으로 이 자료들을 살펴보는 것은 어려운 일이다. 그리고, 시스템마다 생성되는 로그 파일들은 각기 다른 포맷을 가지고 있어 이를 표준화하려는 노력이 있어 왔는데 대표적으로 Bishop[6] 표준 로그 포맷을 예로 들 수 있다.

침입 탐지 시스템 개발자들은 운영체제에서생성하는 기존의 로그들을 활용되기 보다는 별도의 로그 포맷을 새로 정의하여 로그 자료를 생성해 사용한다. 침입 탐지 시스템인 ASAX[5]의 경우에도 NADF (Normalized Audit Data Format)이라는 로그 포맷을 제안하고 이를 활용하여 탐지 시스템을 개발하였다. 이처럼 아직까지는 완전히 공인된 표준화된 로그 포맷을 사용하기 보다는 필요한 형태의 로그 포맷을 새로 제각기 정의해서 쓰고 있는 상황이다.

로그를 생성하는 도구로는 기본적으로 일부 시스템 로그 자료를 생성해 주는 syslog와 XDAS[7]와 같이 분산 환경에서 로그 정보를 교환하기 위해 개발된 도구들이 있다. 그런데, 실제 시스템 로그 자료들에 대한 기초적인 분석과 이를 시각적으로 표현해주는 로그 분석 도구 들의 개발은 흔하지 않은데, 대표적인

도구로는 Moitra의 ALVA[8]와 UC-DAVIS의 Audit Workbench[9]가 있다.

국내의 경우에는 Audit Workbench와 유사한 기능을 가지고 있는 경북대의 전산망 종합 보안상황 모니터 시스템[10], 그리고 기초적인 로그자료 뷰어 기능을 가지고 있는 Looker[11]등이 있다.

Audit Workbench에는 시스템에 존재하는 로그 자료들을 시각적으로 표현하기 위해 그래핑, 하이퍼텍스트, 슬라이싱 등을 이용하여 만든 Visual Aud Browser(VAB) 그리고 일련의 사용자 작업을 시간서에 따라 재현해 줄 수 있는 Frame Maker로 구성는데, 이 도구를 사용하여 관리자는 많은 양의 로그 자료로 부터 시스템에 대한 전반적인 정보를 얻을 수 있다. 특히 하이퍼텍스트 형태로 결과를 출력해 줌으로써, 연관된 자료들을 쉽게 따라가면서 확인해 볼 수 있는 장점이 있다.

그러나, 단일 호스트에서 동작하여 네트워크 연결 정보보다는 시스템 자체의 프로세스만을 보여주며, 시스템 상태에 대해 요약된 정보를 제공하기 보다는 단순히 로그 자료를 시각화해 주는 한계점을 갖고 있다.

전산망 종합 보안 상황 모니터 시스템에서는 여러 개의 호스트를 추가하여 로그자료의 시각화를 시도하였는데, 대체적인 내용은 Audit Workbench와 동일하다. 이 시스템 역시 시스템의 프로세스 위주로 로그 자료를 시각화하고 있어 관리자입장에서 특정 시간대에 어떠한 작업들이 일어났는지를 쉽게 확인하기에는 한계가 있다.

로그 분석 도구는 시스템의 이상 현상이 확인되었을 때 원인 분석을 위해 많이 쓰이는데, 위의 여러 도구들은 실제 침해 사고 추적에 부족한 점이 있다.

III. 로그 분석 기법

실제 시스템에 침입이 발생했다는 사실이 확인되었을 때, 관리자는 문제의 시간대에 어떠한 사용자들이 접속되어 있고, 그들이 어떠한 작업을 했는지와, 그들이 어떠한 경로로 시스템에 접속되었는지에 가장 큰 관심을 갖는다. 따라서 로그 자료를 바탕으로 유용한 정보를 관리자에게 제공할 수 있다면, 접속 경로를 확인할 수 있으며, 해당 경로를 막음으로써 일차적으로 재 침입을 예방하는데 많은 도움이 될 것이다.

본 논문에서 소개하고 있는 로그 자료 분석 기법은 먼저 실제 환경에서 로그 자료를 분석할때 일반적으로 어떠한 질문이 이루어지고, 이에 맞는 답을 어떻게 찾을 수 있는지에 초점을 두고 이루어졌다. [12]에서도 이와 같은 방법으로 접근이 이루어지고 있다. 이것은 실제 침입 사건이 발생하여 원인과 침입자를 조사할때 일반적인 질문들을 나열해 본 것이다.

- 특정 시간 동안에 시스템에 접속되어 있던 사용자는 누구인가?
- 누가 특정 파일을 생성하거나, 수정 혹은 삭제하였는가?
- 특정 시간 동안 어떠한 사용자에 의해 어떠한 프로그램이 동작하고 있었는가?
- 누군가 임의의 목적으로 프로그램을 수정하는 경우는 없었는가?
- 의심스러운 사용자의 구체적인 작업내용은 어떠한가?
- 시스템에 접속 시간대와 접속해 온 호스트를 통계적으로 살펴볼때 의심스러운 사용자는 없는가?
- 현재 시스템에서의 의심스러운 사용자를 발견했을 때 그 이전 호스트는 어디였고 해당 호스트의 상태는 어떠한가?

이러한 질문 중에 비교적 간단한 것의 답을 얻기 위해서는 로그 자료를 일일이 살펴봐도 가능하다. 그러나 여러 호스트사이에서의 여러 관련 로그 자료를 통합하여 검색하는데는 이러한 수동적인 조사에는 한계가 있다. 또한 시스템에서 생성되는 많은 로그 자료들 중에 어떠한 자료를 검색해야 하는지와 해당 자료의 크기가 상당히 클 경우 적절한 범위내로 축소시킬 수 있는가는 일반적인 시스템 관리자에게는 어려운 질문일 수 있다. 이런 점을 고려하여 로그 분석 도구가 제공해야 할 기능들을 적어보면 아래와 같다.

(1) 로그 정보 시각화

로그 자료들은 시스템의 운영체제에 따라 각기 다른 형태의 구조를 가지고 있다. 예를 들어 Sun OS계열의 시스템에서는 (표 1)과 같은 내용의 로그를 생성하고 있다. 기본적으로 시스템에는 많은 양의 원시적인 포맷의 로그 자료들이 존재하는데, 이를 일정한 기준이나 사용자 질의에 따라 세분화 하여 시각적으로 보여줌으로써 관리자가 얻고자 하는 정보 형태로 쉽게 표현될 수 있다. 이를 위해 효과적인 로그 자료의 시각화 방법을 제공해야 한다.

(2) 로그 정보의 일관성 검사

분석의 자료가 되는 로그 파일은 일반적으로 단순한 형태의 포맷으로 구성되어 있으므로, 침입자에 의해 변조 혹은 손상되기 쉬운 속성을 가지고 있다. 따라서 로그를 안정적으로 보존하기 위해서는 주기적으로 로그파일의 일관성을 점검함으로써 로그파일의 손상을 확인할 수 있어야 한다.

(3) 여러 시스템의 로그 통합 분석

실제 시스템 환경에서는 단일 호스트 내의 로그 자료들만의 분석뿐만 아니라 다수의 호스트를 통합적으로 연결하여 분석해 볼 필요가 많아진다. 특히 여러 경로를 거쳐 이루어진 침입을 역추적 할 때 더욱 그러하다. 관리자가 하나 이상의 여러 개의 호스트를 통합 관리할 경우 일일이 해당 시스템의 로그 자료를 찾아 따로 분석하지 않고 분석 도구를 한 시스템에서만 동작시켜봄으로써 호스트간의 유기적인 연결 관계를 분석하여 종합적으로 사용자의 행위나 사용자의 접속 경로 등을 조사할 수 있어야 한다.

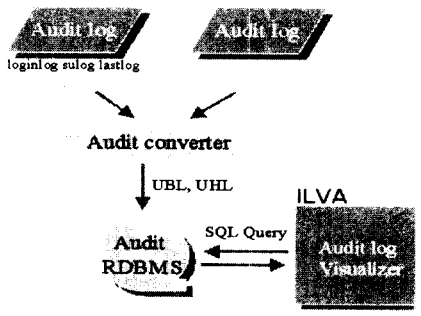

그림 1. ILVA의 구조
Fig. 1. ILVA Architecture

(4) 통계적 오용 탐지기

일반적인 침입 탐지 시스템의 경우 시스템 내의 로그 자료를 통계적으로 분석하여 이상 행위들을 찾아 내고 있는데, 로그 분석 도구 역시 몇몇 로그 자료에 대해서 이러한 분석을 시도해 봄으로써 침입 탐지에도 활용될 수 있다. 특히 호스트내에서의 사용자의 접속 시간대나 사용기간, 접속후 행위등을 바탕으로 사용자의 성향을 종합적으로 분석해 줄 수 있다면 침입자의 오용들을 탐지할 수 있어야 한다.

IV. ILVA의 접근 방법

우리는 시스템에 남아 있는 다양한 형태의 로그자료들로 부터 실제 관리자에게 유용한 정보를 제공해 줄 수 있는 로그분석 도구인 ILVA(Integrated Log Viewer and Analysis Tool)를 구현했으며, 이 도구에서 필요로 하는 기본 로그 자료를 크게 2가지 형태의 로그 포맷으로 정의하였다. 그리고, 로그 분석 도구의 개발에 앞서 실제 시스템 침입 사건이 발생된 환경에서 관리자가 원하는 정보의 형태를 정리해 이것을 바탕으로 로그분석 도구가 제공해야 할 기능을 명세 하였다.

로그 분석 도구인 ILVA는 크게 로그자료 수집, 축약및 변환, 로그 분석의 3가지 단계로 구성된다.

시스템에 산재되어 있는 상당수의 로그 자료중에 분석에 도움이 될 만한 자료를 수집하고, 이들 자료를 선별하여 로그 분석 도구에서 쓰일 수 있는 형태로 변환하여 데이터베이스에 저장한다.

로그를 데이터 베이스에 저장할 경우, 분석 도구의 여러 질의에 대해 빠른 속도로 자료를 검색해 줄 수 있어 통합 로그 분석이 가능하다.

구현된 ILVA의 구조는 (그림 1)과 같은데, 구성 요소로는 로그 수집기, 로그 변환기, 로그 저장소(DB), 로그 뷰어, 그리고 로그 분석기로 구성된다.

4.1. 로그 수집 및 변환

시스템에 침입이 확인되었을 때, 관리자는 해당 시각 어떠한 사용자가 사용하고 있었으며, 그 사용자는 어떠한 작업을 하고 있었는지를 가장 알고 싶어한다. 이 같은 정보는 시스템에서 생성되는 많은 로그 자료중 특히 사용자 접속과 관계되는 기록등에서 얻을 수 있다. 그러나, 이러한 자료들은 포함하고 있는 내용과 그 크기면에서 분석 도구에서 직접 이용하기가 곤란하다. 예를 들어 사용자들의 접속 일시등을 기록하고 있는 wtmp로그의 경우 6개월 정도의 기록량이 17MBytes정도에 달하여 직접 조사하는데는 많은 시간이 소요된다. 따라서 우리는 ILVA에서 사용할 2가지 형태의 로그 포맷을 정의하고 이러한 형태로 로그 자료를 생성하게 하였다.

· 기본포맷 1 : 사용자 호스트 로그
(UHL:User Host Log)

UHL (User Host Log)

ID	Host	Login Date	Logout Date	Usage Time	ports

UBL (User Behavior Log)

ID	Ports	Login Date	Current Date	Commands

그림 2 로그 포맷
Fig. 2. Log Format

사용자가 접속해온 호스트와 접속시각, 유지시간 그리고 포트등에 대한 정보를 가지고 있다.

shcho salmosa.kaist.ac 98/09/14 18:04 18:27 (00:23) pts/19

·기본포맷 2 : 사용자 작업 로그
(UBL:User Behavior Log)

시스템에 접속한 사용자의 작업 모니터링할 수 있게 해주는 것으로 사용자가 단말기에서 실행시킨 명령어들을 기록한다.

shcho_pts/7_98/09/30/13:07_98/09/30/13:09_ helvis all.dat

관리자는 자신의 시스템에 침입하여 작업이 이루어진 시각등을 바탕으로 침입자가 정상적인 계정을 얻었음을 확인할 수 있고, 해당 사용자가 누구인지 알고 싶어한다.
이를 위해서는 특정시간대에 접속한 사용자와 접속해 온 호스트들을 살펴보아야 한다. 이것은 사용자 호스트 로그(UHL)을 분석해 보면 된다. 사용자 호스트 로그는 시스템의 wtmp로그 파일을 분석하여 생성된다.

특정 시각 시스템에 접속한 사용자들의 기록은 SunOS계열의 유닉스 시스템에서는 wtmp로그 파일을 조사하여 알 수 있으나 실제 무슨 작업을 했는가는 lastcomm이나 현재 시각의 로그인 정보들이 기억되는 utmp로그 파일로 부터 얻어질 수 있다. 그런데, lastcomm의 경우 시스템에 대해 실행시킨 프로그램에 의해 만들어진 여러 프로세스들까지 자세히 기록하는 장점이 있지만, 기록되는 로그의 크기와 실제 관리자가 시스템 전반의 프로세스에 대해 지식이 부족한 경우 1차적인 시스템 로그 분석에서는 효과적이지 못하다. 따라서, 좀더 자세하게 어떠한 사용자가 단말기 앞에 앉아 무슨 작업을 했는가를 쉽게 확인하기 위해서는 사용자가 실행시킨 프로그램 단위로 즉 실행 명령어 수준에서 모니터링할 필요가 있다.
물론, alias등의 방법으로 실행하려는 프로그램의 이름을 고의적으로 변경한다면 명령어 수준의 모니터링이 의미가 없다고 생각될 수도 있지만, 이 경우에도 alias등의 방법으로 다른 명령어를 링크하는 작업이 모니터링 될 수 있으므로 위의 약점은 어느 정도 보완될 수 있다.

기존의 Sun OS계열 유닉스 시스템하에 명령어 수준의 시스템 모니터링을 위해서 몇가지 방법을 생각할 수 있는데, 우선 별도의 쉘을 작성하여 모든 사용자가 그 쉘을 사용하게 할 수 있다. 그러나 이 방법은 로그인 후 쉘을 변경하거나 하는 경우 모니터링이 이루어지지 않는 문제가 있다. 결국 커널 수준의 로그 기록 메커니즘의 변경이 필요하게 된다. 다른 방법으로 csh이나 tcsh의 경우 history를 이용하는 방법을 생각할 수 있다. history는 쉘에서 사용자가 실행시킨 명령들을 순차적으로 기록하고 있는데, 이 history기록을 사용자가 로그아웃하기 전에 기록하는 방법으로 명령어 수준 모니터링을 할 수 있다. 그런데, 이 방법 역시 한계점이 있다. 특정사용자가 set history=0를 함으로써 history가 기록되지 않게 할 수 있기 때문이다.
따라서 우리는 현재 시점에서 시스템에서 작업하는

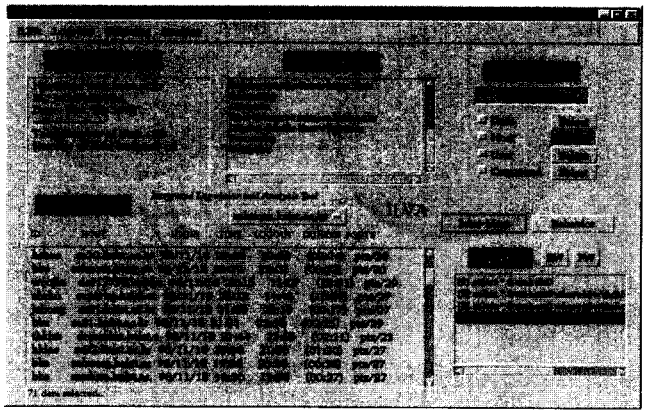

그림 3 ILVA 실행 예
Figure 3. Example of ILVA

사용자들에 대한 정보를 가지고 있는 utmp로그 파일을 주기적으로 수집하여 별도의 로그(사용자 작업 로그:UBL)를 기록하게 하였다. 이를 통해 명령어 수준의 로그기록이 가능하다.

ILVA의 로그 변환기는 시스템의 많은 로그 자료를 바탕으로 사용자 호스트 로그(UHL)와 사용자 작업 로그(UBL) 두 자료를 생성해 낸다. 사용자 호스트 로그는 사용자들의 접속 시간을 기록하고 있는 wtmp로그 파일을 바탕으로 하여 만들어 진다. 사용자 작업 로그는 매분마다 utmp로그 파일을 분석하여 생성된다. 이러한 두 로그 자료는 데이터베이스에 저장되고 분석기에 의해 접근되어 진다. 성격이 다른 시스템에서도 위의 두 가지 형태의 로그자료를 생성해 줄 수 있다면 개발된 도구를 플랫폼에 독립적으로 사용할 수 있다.

ILVA의 초기 버전에서는 로그 데이터베이스로 postgreSQL 버전 6.3[13]을 사용하고 있으나 다른 데이터베이스를 사용해도 큰 수정없이 도구가 작동될 수 있도록 설계되어 있다. 따라서 위에서 정의한 2가지의 형식의 로그 자료만 얻을 수 있다면 이 도구는 플랫폼에 독립적으로 동작할 수 있는 장점이 있다.

4.2. 로그 뷰어 및 분석기

로그 뷰어는 시스템에서 얻어지는 많은 양의 로그 자료를 직접 살펴볼 수 있으며 (그림 3)에 보여진 것과 같이 5가지의 구성요소로 이루어진다.

기본적인 도구의 실행상태및 시스템 정보를 나타내주는 시스템 윈도우, 질의 내용과 처리 상태를 표시하는 상태윈도우, 사용자의 여러 종류의 질의를 선택할 수 있는 선택 윈도우, 질의 결과를 출력하는 결과 윈도우, 그리고 결과 윈도우로 출력되는 결과에 대한 SQL 질의문을 보여주며 질의를 히스토리화(history)하여 이전,이후등 사용자의 선택에 따라 다양하게 질의 순서를 변경 재 실행할 수 있도록 하는 질의 윈도우로 구성된다.

일반적으로 시스템 침입 확인시 확보된 로그 자료를 단순한 텍스트 에디터나 파일 뷰어들을 이용하여 수작업을 통해 일일이 확인하는데, 로그 자료의 양에 따라 검색시간이 비용등이 커지며, 중요한 정보들을 놓칠 우려가 있다. 따라서 이런 점을 고려하여 다양한 질의가 자동적으로 수행될 수 있는 도구가 필요하다. Looker[11]는 기본적인 시스템 로그 자료의 분류 출력의 기능을 갖고 있지만, 기간이나 특정 사용자에 대해서 출력하는 기능만을 갖고 있어 관리자의 다양한 query에 대한 답을 얻기에 미흡했다.

이를 해결하기 위해 분석하려는 로그 자료들을 데이터베이스에 저장하여 초기의 뷰어에서 많은 양의 로그 자료중에서 관리자(사용자)의 질의에 따라 점차로 뷰어의 출력 데이터를 요약해 갈 수 있도록 한다면 로그 분석에 더욱 효과적인 도구가 될 것이다. (그림 3)에 나타나 있듯이 사용자, 호스트, 시간, 명령어등을 이용한 다양한 질의가 가능하다.

로그 분석기는 기본적으로 얻어진 로그자료를 바탕으로 하여 사용자의 접속패턴을 분석하게 되는데, 이

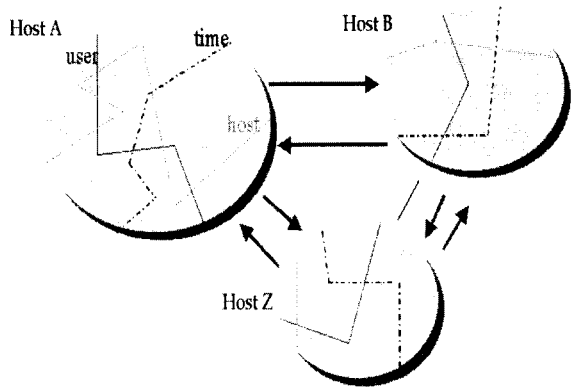

그림 4 여러 호스트간의 통합 로그 분석
Fig. 4. Multi-Host Log Analysis

를 위해 사용자 호스트 로그를 데이터베이스에 기록하여 통계적인 결과를 얻어 낸다. 주로 사용자 호스트 로그를 분석하여 얻어진다. ILVA에서는 사용자가 접속해온 호스트들의 접근 비율을 바탕으로 상대적으로 접속 비율이 적은 이상 접속 기록을 찾아내는 사용자 접속 호스트 분석과 사용자의 평균 작업 시간대를 바탕으로 이상 접속 시간대를 찾아내어 작업 내용을 자세히 살펴볼 수 있는 사용자 접속 시간대 분석이 가능하다.

분석기는 또한 여러 로그 자료들 사이의 연관 관계를 연속적으로 질의할 수 있도록 하고 있어, 여러 시스템 간의 연결을 통한 침입 경로를 확인하는데 큰 도움을 줄 수 있다. 관리자는 특정 시스템으로 부터 접속한 사용자가 이전 시스템에서 어떠한 작업을 했는지를 확인해 보고 싶어할 수 있다. 이때 관리자는 접속해온 호스트가 어디였는지 역으로 추적해서 해당 시스템에서의 작업내용도 확인하고 싶어진다. 이를 위해서는 직접 그 시스템에 접속하여 대상 시스템의 로그 자료를 다시 분석해야 한다. 그런데, 해당 시스템의 일부 로그 파일만을 전송 받아 통합적인 로그 분석이 가능하다면 보다 더 효율적일 것이다. 특히 경로상의 시스템으로 부터 사용자 작업 로그와 접근 호스트 로그가 남게될 경우 이들을 수집하여 비교할 경우 침입 경로를 조사하는데 많은 도움이 될 수 있다.

(그림 4)에서 처럼 A라는 시스템에 침입 흔적이 발견되었을 때, 해당 시스템의 로그에 대해 여러 형태의 질의, 예를 들어 시간, 사용자, 접속 호스트등을 지정하여 범위를 줄이고, 관련있는 다른 호스트가 있는 경우 해당 호스트의 로그를 같은 방법으로 분석해 점점 범위를 축소해 간다면 특정한 사용자의 작업 내용을 확인할 수 있어 여러 시스템을 경유한 침입을 발견할 가능성이 높아진다.

예를 들어 다음과 같은 사용자 접속 기록이 있다고 했을때,

· shcho superbg.kaist.ac.kr 98/11/18 22:13 22:43 (00:30) pts/11

접속해 온 호스트가 superbg임을 확인하고 그 시각 superbg.kaist.ac.kr에서 사용하고 있던 사용자들의 기록과 사용자들의 작업 내용을 확인할 수 있다. 이러한 연속적인 자료접근은 시스템의 관리자가 관리하는 영역내의 모든 컴퓨터의 자료를 바탕으로 하여 단계적으로 침입자의 경로를 추적할 수 있게 해준다.

ILVA의 로그 분석기는 각 사용자의 접속 정보를 '호스트:로그인 일시:지속 시간'의 형태로 정리하여 사용자의 평균적인 접속 성향을 분석해 주기 때문에 침입 사건이 발생하였을 때 통계적으로 사용자의 일반적인 성향과 다른 사용자를 찾아 낼 수 있다. 또한 시스템 접속 시간대를 크게 24시간으로 나누어 각 시간대의 접속 비율을 조사하고 극히 낮은 비율의 접속 기록에 대해서는 해당 시간대에 사용한 사용자의 작업(command)들을 부가적으로 조사한다면 침입자의 작업 내용을 확인할 수 있는 가능성이 높다.

V. 적용 사례 및 한계

SYSTEM	C1	C2
사용자 수	90	95
기록 기간	14개월	8개월
wtmpx로그 크기(KB)	17,928	17,492
UHL 로그 크기 (KB)	299	1,187
축소 비율	1/60	1/14

표 2. 사용자 호스트 로그와 wtmpx 로그 비교

Table 2. UHL vs wtmpx

5.1. 사례 1 : 정상적인 시스템 환경

실제 구현 도구인 ILVA를 KAIST 전산학과 소프트웨어공학 연구실의 일부 컴퓨터에 설치하여 활용해 보았다.

KAIST 전산학과 소프트웨어공학 연구실은 4대의 SunSparc 워크스테이션과 약 30여대 의 PC가 네트워크로 연결되어 있는데, 이중 하나의 워크스테이션에 도구를 설치하였다. 그리고 다른 2대의 워크스테이션에는 로그 변환기만을 설치하였다.

도구가 설치된 머신에서는 기본적으로 시스템의 로그 자료를 분석하여 보여줌으로써 시스템 외부로 부터 여러 접속 호스트와 시스템에 접속실패한 사용자들이나 특정 시간의 시스템 사용자들의 작업 내용들을 쉽게 확인할 수 있었다. 또한 사용자 분석을 통해 사용자들의 평균 접속 시간대와 작업 시간 그리고 해당 사용자의 작업 내용들을 확인할 수 있었다. 또한 사용자의 이전 경로상의 시스템 기록들을 함께 검색할 수 있어서 개발된 도구가 소규모 네트워크의 관리자 입장에게 상당히 유용하게 쓰일 수 있음을 확인할 수 있었다.

그러나, 로그변환기가 설치되지 않은 시스템의 경우 사용자의 작업 내용 검사를 할 수 없는 부족한 면이 있었다. ILVA가 필요로 하는 사용자 호스트 로그와 사용자 작업 로그를 부가적으로 생성하는데 이때의 부가적인 비용은 전체 로그 크기에 비해 상대적으로 적은 비용이어서 로그 분석 도구의 이용이 시스템 전반에 별다른 부담을 주지 않는 것을 확인할 수 있었다. (표 2)에서 보여지듯이 사용자 호스트 로그는 wtmpx로그에 비해 약 수십분의 1정도로 그 크기가 작다. 사용자 작업 로그도 약 1개월간의 시스템 기록인 250KBytes내외로 비교적 작다.

로그 자료를 도구에서 분석하기 위해서는 로그 데이터베이스를 만들어야 된다. 그런데, 실제 실험 결과 데이터베이스에 로그 자료를 저장하는 과정에서 많은 시간이 소요됨을 확인할 수 있었다. (표 3)에서와 같이 2개 시스템의 4개의 로그를 DB에 저장하는데 약 51.2분이 소요하였다. 이것은 실제 침입 사건을 추적하는데 개발된 도구를 이용하기 어려운 단점이 될 수 있다. 따라서 저장 시간을 줄이는 것이 중요한 문제이며, 이를 위해 저장되기 전에 일차적으로 로그 자료를 기간별로 분리하여 저장하는 것이 바람직하다.

(그림 3)은 실제 도구를 사용하여 사용자 작업을 조사한 결과이다. 현재 결과 윈도우로 각 사용자들의 작업 내용이 출력되고 있으며, 선택 윈도우의 여러 옵션을 통하여 결과를 계속적으로 요약해 갈 수 있다.

시스템에 접속한 사용자들의 접속 시간대를 분석해 보면 대체적으로 성향을 확인할 수 있다. (그림 5)는 세 사용자가 특정 시스템에 8개월 동안 접속한 시간대를 나타내고 있다. 세 사용자는 공통적으로 오전 3

로그	UHL		UBL	
	C1	C2	C1	C2
로그 크기 (KB)	299	1,187	235	248
기록 기간 (월)	15	8	1	1
로그 DB 튜플수	5,203	20,573	3,802	4,152
로그 DB 저장 소요 시간 (분)	7.9	31.2	5.8	6.3

표 4. 로그 DB 저장 소요 시간

Table 3. Time to update Log DB

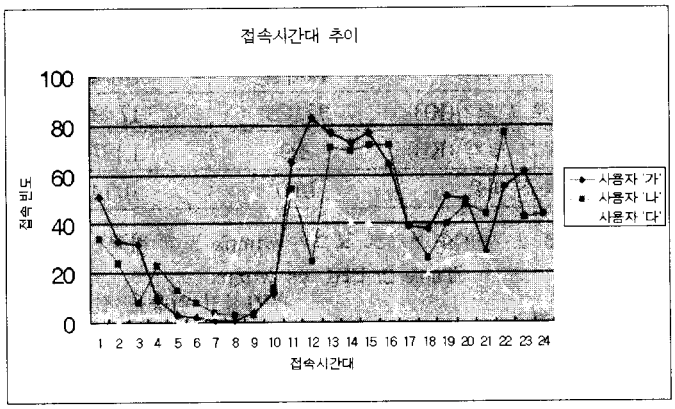

그림 5. 사용자 접속 시간대 분포

Fig. 5. Users' Login Time Distributions

시 부터 오전 7시까지 접속 빈도가 상당히 적고 오전 10시 부터 오후 4시 사이에 접속 빈도가 높다.

한편, 사용자 '가'와 사용자 '나'는 대체적으로 비슷한 성향을 보이는데, 반해 사용자 '다'는 오후 11시 이후부터는 거의 접속을 하지 않는다는 것을 보여주고 있다.

따라서 사용자의 접속 시간대 분석은 사용자 성향을 파악할 수 있는 좋은 자료가 되어 주며, 접속 빈도가 상당히 적은 시간대에 접속한 경우에는실제 정상 사용자의 사용인지 확인해 보는 것이 필요하다. ILVA를 사용하면 접속비율이 낮은 시간대의 사용자가 수행한 작업 내용을 자세히 조사해 볼 수 있다.

ILVA의 시스템 접속 호스트 분석 결과를 보면 주로 접근해 온 호스트에서 잦은 접속이 이루어 지고 있다. 따라서, 접근 호스트에 대한 분포를 살펴 보고 처음이나 상대적으로 낮은 비율의 호스트로 부터 접근해 온 사용자를 좀더 주의깊게 관리한다면 침입 탐지에도 많은 도움이 될 수 있다. C2 시스템의 접속 호스트들을 조사해 본 결과 8개월 동안 약 20,573개의 호스트로 부터 접속이 이루어졌으며 대부분의 접속이 자주 접속해 온 호스트로 부터 이루어졌고, 약 1%의 호스트들은 처음 접속해 온 호스트들이여서 이들로 부터 접속한 사용자의 작업 내용들을 조사해 볼 필요가 있었다.

이와 같이 ILVA를 사용하여 사용자 접속 시간대와 접속 호스트들을 통계적으로 분석하여 의심스러운 접속에 대해 세부적으로 접속후 작업 내용의 조사를 쉽게 할 수 있었다.

```
kp__pts/7__99/01/22|19:41__99/01/22|19:42__ -csh
kp__pts/7__99/01/22|19:41__99/01/22|19:44__ ftp ftp.sodre.net
kp__pts/0__99/01/23|10:35__99/01/23|10:36__ -csh
kp__pts/0__99/01/23|10:35__99/01/23|10:37__ ftp net.polyu.edu.hk
kp__pts/0__99/01/23|10:35__99/01/23|10:47__ telnet ultra2.ultratech-is.net
kp__pts/1__99/01/23|11:44__99/01/23|11:45__ telnet libr0.kaist.ac.kr
kp__pts/1__99/01/23|11:44__99/01/23|11:52__ -csh
kp__pts/1__99/01/23|11:44__99/01/23|11:53__ ./std purple.kaist.ac.kr
kp__pts/1__99/01/23|11:44__99/01/23|12:06__ ./std cs.snu.ac.kr
kp__pts/1__99/01/23|11:44__99/01/23|12:07__ ./std cogsys.kaist.ac.kr
kp__pts/1__99/01/23|11:44__99/01/23|12:12__ ./std cpu.chungnam.ac.kr
kp__pts/1__99/01/23|11:44__99/01/23|12:15__ ./std ai-nt.cse.cau.ac.kr
kp__pts/1__99/01/23|11:44__99/01/23|12:16__ ./std opera. cse.cau.ac.kr
kp__pts/1__99/01/23|11:44__99/01/23|12:18__ ./std www.cba.cau.ac.kr
kp__pts/1__99/01/23|11:44__99/01/23|12:19__ ./std ailab2.cse.cau.ac.kr
```

그림 6. 침입자의 작업 로그 (UBL)

Fig. 6. Intruder's UBL

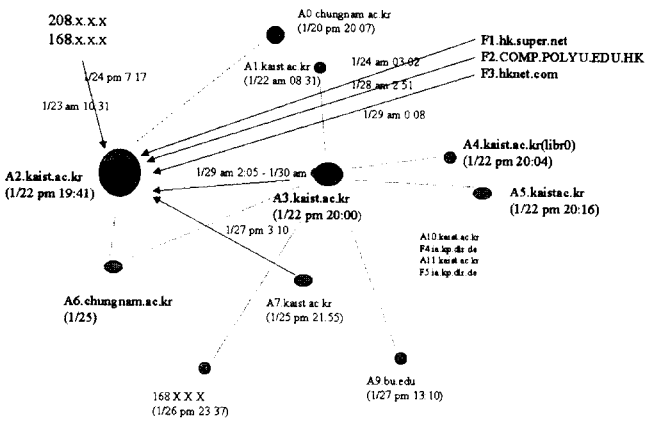

그림 7. 침입 시스템 경유도

Fig. 7. Intruded Systems Route Diagram

5.2. 실제 시스템 침입 추적

99년 1월 24일, 개발된 시스템인 ILVA를 가동하고 있는 시스템인 C2에서 시스템 침입 사실을 실제로 확인할 수 있었다.

침입 방법은 정확히 확인되지 않았는데, 다른 호스트로 부터의 접속이 침입자에 의해 모니터링되어 침입자가 이를 이용해 침입에 성공한 것으로 보여진다. 침입자는 침입후 관리자 권한을 획득하여 자신의 작업 계정을 만들고, 다른 시스템으로의 침입을 시도하고 있는 상태에서 ILVA에서 의해 발견되었다. ILVA를 사용하여 99년 1월 20일 부터 1월 31일 사이에 최소 20여개 이상의 시스템이 침입자에 의해 관리자 권한을 빼앗기는 등의 침입을 받은 사실을 확인하였다. 그 후 일부 시스템에서는 침입자에 의해 하드디스크의 데이터를 손실하는 등의 직접적인 피해가 생겼다.

ILVA는 실제 C2에 설치되어 사용자 작업을 실시간으로 모니터링하고 있어서 침입자가 어떠한 작업을 수행하는지 어느 정도 파악할 수 있었다.

(그림 6)은 ILVA가 생성하는 사용자 작업 로그(UBL)의 일부이다. 이 자료를 통해 kp라는 침입자의 작업 내용과 침입을 시도하려는 시스템과 어떠한 과정으로 침입을 진행해 나갔는지를 확인할 수 있었다. (그림 6)에서는 ftp를 통해 std라는 해킹프로그램을 받은 후 이것을 실행시켜 다른 시스템으로의 침입을 시도한 사실을 알 수 있다. 계속하여 침입자가 시도한 시스템들의 로그들을 분석해 봄으로써 (그림 7)과 같이 침입이 성공한 20여개의 시스템들의 연결관계를 추정할 수 있었다. 특히 독일의 일부 시스템의 경우 미처 침입 사실을 알지 못하다 ILVA의 분석에 의한 결과로 침입 사실을 확인할 수 있었다.

ILVA를 사용한 시스템의 로그 분석은 크게 4가지의 한계를 보여주었다. 첫째, 시스템 침입 과정의 확인에 있어서 ILVA의 자동화 된 로그 분석 기법을 적용하려고 하였으나, 앞 절에서 언급했듯이 로그 자료를 로그 데이터베이스에 저장하는데 소요되는 시간이 상당하여 신속하게 이루어져야할 추적 활동에는 한계가 있었다. 실제로 로그 분석은 수동적으로 일일이 로그 자료를 살펴보는 것으로 이루어졌다. 이를 통해 볼 때 시스템 침입 사건의 추적에 사용될 자동화 로그 분석 도구는 신속한 자료 분석이 필수적이다. 따라서 로그 데이터베이스 생성 속도의 개선이 절대적으로 요구된다. 둘째, ILVA의 사용자 작업 로그(UBL)가 utmp로그를 1분 주기로 저장하는 방법으로 생성되므로, 실제 쉘상태에서 어떤 명령어를 실행시키고 있는지의 감시는 사실상 불가능하였다. (그림 6)에서처럼 사용자 작업 로그로 얻어지는 내용은 사용자들이 실행시킨 프로그램 단위이다. 따라서 좀더 효과적인 작업 모니터링이 이루어져야 한다. 이를 위해서 시스템 커널 수준의 로그 생성 기법의 변경, 쉘 상태에서의 별도의 입력 모니터링 기법 또는 패킷 모니터링 방법이 필요하다. 셋째, 이번 침입 사건에 대한 추적의 가장 큰 어려운 점은 여러 경로를 갖는 침입을 모두 조사해 볼 수 없던 점이다. 실제로 ILVA가 설치되어 있는 C2에서만 침입자의 작업 내용을 어느 정도 파악할 수 있었으나, 다른 시스템에 대해서는 수동적

으로 원시적인 형태의 로그를 일일이 살펴보아야 하는 불편함이 있었다. 그리고, 관리하고 있는 도메인 밖의 시스템에 대해서는 더 이상 추적을 진행할 수 없어 결국 침입자를 찾아내는데 불가능하였다. 이와 같이 여러 경로를 거치는 침입의 경우는 침입자 추적보다는 오히려 침입에 의한 피해 최소화에 더 비중을 두고 대처를 해야하는 한계가 있었다. 이러한 한계점을 해결하려면 각각의 시스템들이 유기적으로 협조 관계를 가지고 한쪽 시스템으로 부터의 침입 사실 확인 요청에 따라 즉시 시스템 로그 분석을 시행할 수 있는 시스템 보안 메커니즘이 갖추어져야 한다. 넷째, ILVA가 생성하는 사용자 호스트 로그(UHL), 사용자 작업 로그(UBL) 에 대해서는 침입에 의한 손상을 고려하지 않았는데, 실제 이번 침입 사건에서는 침입자가 자신이 추적받고 있다는 사실을 알고, 사용자 작업 로그를 찾아 지워 침입자의 작업 내용을 더이상 확인할 수 없었다. 따라서 로그 자료에 대한 기본적인 복구 기법을 고려해야 할 필요가 있다.

VI. 결론 및 향후 과제

본 논문에서는 시스템에서 생성되는 많은 양의 로그를 바탕으로 관리자가 통합적인 로그 분석을 통해 쉽게 시스템의 이상 상태등을 확인할 수 있는 방법을 제안하였다. 그리고, 로그 변환기, 뷰어, 분석기로 이루어진 로그 분석도구를 구현하여 여러 환경에서 적용해 보았다.

ILVA는 소규모 네트워크 환경에서 일부 시스템에 설치되어 있는 많은 로그 자료를 수집, 분석하여 시스템 관리자 입장에서의 유용한 형태의 정보로 가공해 줄 수 있는 자동화된 도구의 역할을 제대로 하였다. 특히 사용자 접속 성향 분석은 침입 탐지에도 효과적으로 적용될 수 있었다. 그리고 사용자의 작업 내용 분석을 통해 실제 침입자를 탐지해 내고 다른 도메인의 침입 시스템까지 확인해 낼 수 있었다. 그러나, ILVA는 실제 발생한 침입 사건의 추적에 활용하는데는 분석 속도, 사용자 작업 로그의 부족함, 여러 시스템의 유기적인 로그 분석 그리고 로그 보존에 한계점을 나타냈다. 따라서 로그 분석에 필요한 전처리 단계의 속도 개선, 보다 세밀한 사용자 작업 모니터링 방법이 개발되어 져야 한다. 특히, 전자상거래 환경이나 전자도서관등의 규모가 큰 네트워크 환경에서는 ILVA에서 제안하고 있는 기본적인 2가지의 형태의 로그 포맷만으로는 사용자의 구체적인 작업 내용등 다양한 시스템 상태를 관리자에게 설명하기 불가능하다.

ILVA와 같은 로그 분석 도구의 개발은 실제 제공되는 정보 형태나 종류에 있어서 부족한 점이 많지만, 초기의 프로토타입 역할을 하였다. 여기에 실제 침입 사건을 통해서 얻어진 한계점을 보완하고 표준화된 로그 포맷을 만들기 위해, 실제 관리자들에게 도움을 줄 수 있는 정보 내용들에 대한 폭넓은 연구와 시스템 분석의 기본이 되는 로그 자료의 안정적인 보존을 위한 방법을 연구 보완한다면 시스템 보안 강화를 위한 훌륭한 로그 분석 도구가 될 것이다. 덧붙여 침입자의 의도적인 추적 회피 방법과 여러 경로를 통한 침입, 분산시스템 환경에서 성능 개선이 요구된다.

참고문헌

[1] Simson Garfinkel and Gene Spafford, Practical UNIX and Internet Security, O'Reilly & Associates, Inc., second edition, 1996

[2] R.G. Bace, NSA, 1995

[3] Computer Operations,Audit, and Security Technology Team, http://www.cs.purdue.edu/coast, 1998

[4] 정보보호센터, 정보보호뉴스 1998년 7월호 (통권 13호)

[5] Abdelaziz Mounzi, 'Languages and Tools for Rule-Based Distributed Intrusion Detection'', Phd Thesis, Facultes Universitaires, Belgium, 1997

[6] Matt Bishop, ''A Standard Audit Trail Format.'' In Proceedings of the 18th National Information Systems Security Conference, pp. 136-145, Oct. 1995

[7] The Open Group, Berkshire, United Kingdom, Preliminary Specification, Distributed Audit Service(XDAS) Base-Draft 8, Feb. 1997

[8] A. Moitra, ''Real-time Audit Log Viewer and Analyzer'', In Proceedings of the 4th Workshop on Computer Security Incident Handling, Aug. 1992

[9] James Hoagland, etc., ''Audit Log Anasis Using

the Visual Audit Browser Toolkit'', UC-Davis TR-CSE-95-11, 1995
[10] 경북대, 전산망 보안상황 모니터 시스템 개발에 대한 연구, 정보보호센터, Dec. 1997
[11] 한국과학기술원, 전산망 보안 평가시스템 연구, 한국통신, Dec. 1997
[12] B.R. Wetmore, ''Paradigms for the Reduction of Audit Trails', Ms Thesis, University of Califonia Davis, 1993
[13] The PostgreSQL Development Team, PostgreSQL programmer's Guide, Feb.,1998

著者紹介----------

조 상 현 (Sang-hyun Cho) **학생회원**

1997년 2월 고려대학교 컴퓨터학과 졸업
1999년 2월, 한국과학기술원 전산학과 석사
1999년 3월 ~ 현재 : 한국과학기술원 전산학과박사과정 재학중

〈관심 분야〉 네크워크 보안, 위험 분석

차 성 덕 (Sung-deok Cha) 종신회원

1983년 University of California at Irvine 학사
1986년 University of California at Irvine 석사
1991년 University of California at Irvine 박사
1990-91 Hughes Aerocraft co. 연구원
1991-94 Aerospace co. 연구원
1994년 9월 ~ 현재 : 한국과학기술원 전산학과 조교수
〈관심 분야〉 소프트웨어공학, 컴퓨터보안, 정형 기법, 소프트웨어 명세 및 검증