情報保護學會論文誌
第 21 卷 第 4 號, 2011. 8

# 윈도우 운영체제 상의 사용자 행위 추적에 관한 연구

정 창 성†, 김 영 찬‡
한밭대학교 정보통신전문대학원 컴퓨터공학과

# A Study on System Tracing User Activities in the Windows Operating System

Chang-sung Jung†, Young-chan Kim‡
Dept. of Computer Engineering, Graduate School of Information and Communications, Hanbat National University

## 요 약

산업스파이 및 악의적인 내부 이용자에 의한 기업 내부 핵심 정보유출 사건이 증대되고 있는 현실을 반영하듯, 감사 및 로그 보안기술의 중요성이 한층 부각되고 있는 것이 현실이다. 본 논문에서는 기업의 중요 정보를 처리하는 운영체제에 접속한 사용자에 대한 행위를 면밀히 감시 및 분석하여, SOX 법안에서 요구하는 수준의 기업 내부통제를 위한 세션로깅시스템을 제안한다. 본 논문에서 제안하는 시스템은 운영체제에 접속한 사용자의 불법적인 행위를 모니터링하고, 그 행위 과정의 세부 로그를 통해 명확한 증거를 제시할 수 있도록 하고 있다. 이를 위해 논문에서 제안한 별도의 여러 서비스를 추가하여 운영체제를 변경하는데, 이러한 서비스는 기존의 운영체제가 제공하는 인터페이스를 이용하면서 접근을 제어하고 로그를 획득하기 위한 기능을 추가적으로 제공한다. 서버에 접속한 관리자 및 사용자의 행위에 대한 세션 로그를 중앙 집중적으로 저장, 관리하고 해당 로그에 대한 검색 및 조회 기능을 제공할 수 있도록 하여, SOX 법안에서 요구하는 수준의 기업 내부통제를 위한 컴퓨터 포렌식 시스템 및 로그 기술을 제안한다.

## ABSTRACT

As it seems like critical information leakages have been increasing due to industrial espionage and malicious internal users, the importance of introducing audit and log security technology is growing every now and then. In this paper, we suggest the session logging system for the company's internal control to meet the SOX legislation level, by monitoring and analyzing users behaviors connecting to the business-critical Operating System. The system proposed in this paper aims to monitor the user's illegal activities in the Operating System, and to present the clear evidence of purpose of those activities by detailed logs. For this purpose, we modified Operating System by adding multiple services suggested in this paper. These services utilize interfaces provided by the existing Operating System and add functions to control access and get logs. The system saves and manages session logs of users or administrators connected to the server with centralized log storage. And the system supports session log searching and lookup features required by SOX legislation for the company's internal controls with the level of computer forensics and logging technology.

**Keywords**: Sarbanes-Oxley Act, Computer Forensics, User Activities

접수일(2011년 1월 12일), 수정일(2011년 4월 21일), 게재확정일(2011년 5월 30일)

† 주저자, csjung@hanbat.ac.kr
‡ 교신저자, yckim@hanbat.ac.kr

## I. 서 론

내부통제는 기업정보시스템의 사용자 관리, 정보의 엄격한 내부접근통제, 정확하고 신뢰성 있는 정보자원의 유지, 기업정보시스템의 감사 및 모니터링 등과 같은 목표를 달성하기 위하여 기업이 지속적으로 이행해야 하는 일련의 내부 절차를 의미한다[1]. 기업의 부정이나 잘못을 방지하려면 기업회계 프로세스를 투명하게 유지하는 체계가 반드시 필요하다.

미국의 SOX 법안(Sarbanes-Oxley Act)[1][2]은 이러한 흐름을 반영하여 처음으로 법제화한 것이다. 미국의 뒤를 이어 일본판 SOX가 법제화 되었으며, 이외에도 영국, 캐나다, 프랑스, 중국, 한국 등이 SOX 법의 영향을 받아 기업의 재무보고 부정을 방지하는 법 규제를 실시하고 있거나, 유사 법을 적용하고 있다[3].

로그는 일반적으로 문제점을 해결하기 위해 이용되어 왔지만 최근에는 시스템이나 네트워크의 효율을 최적화하거나 사용자의 행동을 기록하거나 또는 악의적인 행위들을 조사하기 위한 유용한 데이터로 사용되고 있다. 주기적인 로그 검토 및 분석을 통해 보안 사고가 발생하였을 때 추적할 수 있는 증거 자료를 확보할 수 있다. 로그파일을 분석함으로써 침입 경로나 침입 후 실행한 작업에 대한 정보를 획득할 수 있기 때문이다. 또한 이 로그파일은 수사기관에서 요청했을 때 증거 자료로써 활용할 수 있는 매우 중요한 자료가 된다[4]. 최근 사이버나 일상적인 전산침해현황 수사에서 기본적으로 분석하는 것이 로그파일이다.

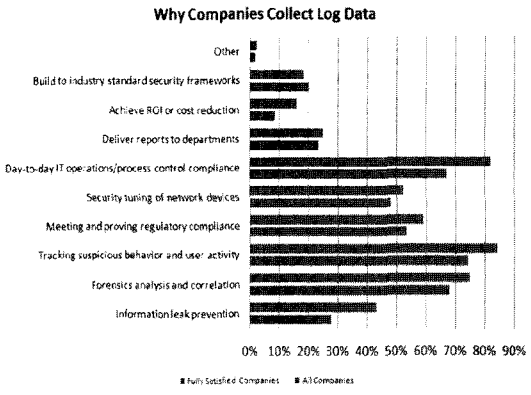
시스템에 존재하는 로그파일을 분석하여 증거자료로 사용하기 위한 다양한 방안이 연구되어 왔다. 하지만 로그의 양이 방대한 것은 물론 사용자 세션에 의해 분리되어 있지 않기 때문에 사용자의 작업 내역을 분석하거나 사건의 연관성을 찾는데 어려움이 많고, 로그파일에서 사건 당시의 증거가 사라지거나 혹은 로그파일이 권한 있는 시스템 관리자에 의해 쉽게 변경되기 때문에 무결성을 보장하기 힘들다는 등의 문제점을 가지고 있다.

본 논문에서는 보안 사고의 증거자료로 활용하기 위해 운영체제에 접속한 사용자의 다양한 행위를 정의하고, 이를 기반으로 사용자가 운영체제에 로그온하여 로그오프할 때까지 수행된 작업 내역에 대한 로그를 획득하여 사용자의 행위를 추적하기 위한 방안을 제안한다. 이를 위해 사용자가 운영체제에 접속하여 수행한 모든 작업 내역을 추적하기 위한 다수의 모듈을 제안한다. 구체적으로는 사용자의 로그온과 로그오프 정보 획득, 입력 장치에 의한 작업 내역, 네트워크 서비스 접속 이후 작업 내역 및 파일시스템 변경 내역 등을 추적할 수 있는 모듈을 제안한다.

본 논문은 2장에서 본 연구의 배경과 기존 시스템 로그의 문제점 및 로그 관리에 대한 요구사항에 대하여 기술하고, 3장에서는 운영체제에 접속한 사용자의 작업 내역에 대한 로그를 획득하기 위한 세부적인 추적 기술에 대해 설명하고, 4장에서는 본 논문에서 제안한 시스템에 의해서 수집된 로그 데이터의 다양한 활용 방안을 제시한다.

## II. 로그 관련 연구

다수의 시스템 로그파일이 보안 측면에서 생성되고 있으며 실제로도 보안 측면에서 로그파일을 수집하고 활용하고 있는 것으로 보고되고 있다[5]. 2009년 San Diego Log Management Summit에서의 SANS 팀의 보고에 의하면 다음 [그림 1]과 같이 로그파일의 수집 목적은 의심스러운 행동의 추적과 사용자 활동의 모니터링, 포렌식 분석과 연관분석 등으로 나타나고 있는데, 정보 유출 방지가 새로운 항목으로 나타난 것이 특징이다. 미국 국방부의 보고서에 의하면 최근의 시스템에 대한 사이버공격의 영향을 최소화하거나 예방하기 위한 방법으로 로그 관리가 중요한 관리항목으로 추천되고 있다.



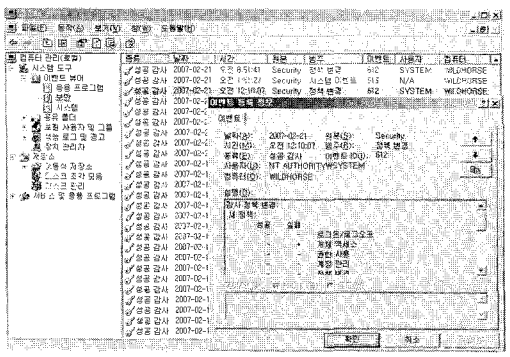

[그림 1] 로그파일의 수집 목적

### 2.1 기존 연구

시스템 로그는 운영체제에 접속한 사용자들의 행위에 대한 결과를 저장한 기록이다[6]. 시스템 로그는

시스템 운영체제의 로그, 보안장비의 로그, 웹이나 데이터베이스와 같은 소프트웨어의 로그 그리고 서비스를 제공하는 애플리케이션 로그 등 수많은 종류가 있다. 대다수 통신장비와 보안장비에서는 syslog를 로그 표준으로 하고 있지만, 윈도우 운영체제의 로그는 syslog를 기본으로 제공하지 않고 시스템, 보안, 응용프로그램 및 서비스 로그 등의 독자적인 형식을 채택하고 있다. 시스템 로그파일들은 독립적으로 분석하는 것이 아니라 서로 유기적인 관계를 갖고 분석해야 한다[7]. 즉, 하나의 로그파일에서 침입 흔적이 발견되면 다른 로그파일들에서 해당 사용자와 관련된 로그를 분석해야 하는 어려움이 있으며, 여러 가지 도구를 사용하여 쉽게 변경하거나 삭제할 수 있다는 관리상의 문제점도 있다. 시스템 로그는 syslog와 W3C 등 소수의 표준화된 형식이 있지만, 다양한 형태로 존재하고 모두 서로 상이하여 통합하기 힘들고 데이터가 대용량이라는 특성 또한 존재한다.

기존의 윈도우 운영체제에 대한 이벤트 추적은 [그림 2]와 같이 정적으로 기록된 사건 로그파일을 참조하는 별도의 감사 추적 프로그램을 이용하여 가능하였다. 이 경우 시스템 관리자 등에 의한 실시간 추적이 불가능할 뿐만 아니라 운영체제 내부에서 감지되는 다양한 사건 정보의 획득이 원활하지 못하다. 또한 사건의 연관 관계를 명확하게 분석하기 힘들고, 로그파일의 크기 제한 등의 문제로 인하여 장기간 관리하는데 어려움이 많다. 특히 다중 시스템 환경에서는 개별적인 파일 로그를 통합하여 관리하는 데에도 문제점이 많다.



[그림 2] 정적 이벤트 로그

## 2.2 로그 관리에 대한 기술적 요구사항

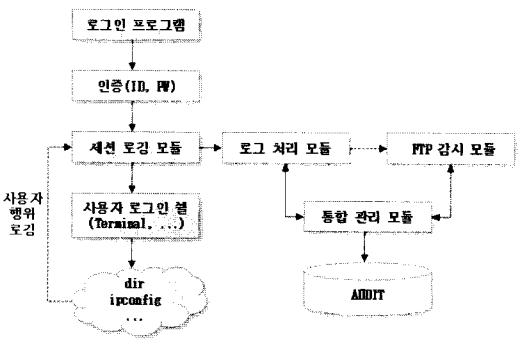
보안 로그 관리 및 분석은 일관성이 없는 로그의 항목과 형식 그리고 발생 시간 기록과 로그 소스들의 다양성과 발생빈도 등의 문제점 때문에 초기의 로그 생성과 관련하여 잠재적인 문제점이 있다[8]. 로그의 항목은 초기 단계에서부터 차후 확장성을 고려하여 설계해야 하며, 서로 다른 항목과 형식으로 된 로그들을 하나의 형식과 고정된 데이터 필드로 변환시키는 작업이 필요하다[9][10]. 또한 다수의 시스템을 관리하는 경우에는 중앙 집중식의 원격 로그서버를 구축하여 운영하는 방안이 효율적이며, 주기적으로 로그를 백업하여 시스템 오류 및 장애로 인한 로그의 손실을 최대한 줄여야 한다.

시스템 관리자가 기업의 주요 정보시스템을 감시하는데 있어서 가장 큰 문제는 웹 로그, 데이터베이스 로그, 메일 로그, 파일 로그 등과 같은 다양한 응용프로그램 로그와 시스템 로그, 보안 로그 등의 수많은 로그를 수집하고 유지 관리하는 것이다. 추가적으로 다른 응용프로그램이 설치되어 있다면 서로 연관하여 분석해야 하는 로그는 더욱 다양해질 수밖에 없다. 더욱이 시스템 로그 파일의 크기가 증가함에 따라 시스템 관리자가 수작업으로 운영체제에 접속한 사용자의 작업 내역이나 불법적인 침입 흔적을 추적하는 것은 매우 힘든 일이다. 본 논문에서는 다수의 추적 모듈이 실행되는 환경에서 동일한 형식으로 로그를 생성하고, 각각의 사용자 세션 단위로 로그 데이터를 처리하도록 하였다. 각각의 추적 모듈에서 생성하는 로그는 구현 방법이나 동작 방식 등의 원인으로 인하여 서로 다른 형태로 정보가 획득될 수 있다. 서로 다른 모듈에서 다른 형태로 로그를 생성하면 로그의 데이터를 서로 참조하여 조회할 수 없고, 결과적으로 사건의 내용을 분석하는데 복잡도와 모호성이 증가하기 때문에 로그의 데이터를 동일한 형태로 변환하여 데이터베이스에 저장하도록 한다. 차후에 사용자의 행위를 면밀히 추적하고자 할 경우, 각 추적 모듈에서 생성되는 로그에서 동일하게 검색할 수 있는 세션 정보와 사용자 정보 등을 기준으로 상호 연관하여 분석하면 중요 업무 감사 시에 요구되는 사건 정보의 정확성을 향상시킬 수 있다. 사건 분석에 대한 오해 소지와 각 시스템에서 발생하는 사건간의 시간 간격의 계산 오류를 방지하기 위해 모든 로그 발생 서버에 대한 시간을 일정하게 동기화하여 관리할 필요성이 있는데, 이를 위해 시스템 시간을 인터넷상의 시간 서버와 동기화시켜주는 서비스를 활용할 수 있다. 이와 같은 방안을 적용하여 차후 사건의 연관 관계를 편리하고 명확하게 분석하여 보고할 수 있다.

로그를 관리하는데 있어서 유의할 점은 생성된 로그의 기밀성, 무결성 그리고 유효성이 부주의로 혹은 의도적으로 파손될 수 있다는 것이다. 로그 관리는 로그의 유용성을 보장하는 것과 함께 로그의 기밀성과 무결성을 파손의 위험으로부터 보호하는 것도 포함한다. 그렇기 때문에 보안 로그 자체에 대한 보안에도 관심을 기울여야 한다. 로그파일에 대한 접근을 제한해야 하며, 각 로그 항목을 생성하는 프로세스도 안전하게 보호해야 한다. 또한 로그가 저장되어 있는 파일이나 데이터베이스에 대해서는 강제적 접근제어 정책을 적용하여 권한이 없는 사용자의 접근을 원천적으로 차단해야 한다.

## III. 사용자의 행위 추적 시스템

기존 시스템 로그의 문제점을 보완하기 위해 본 논문에서는 로그 수집과 관련하여 다음과 같은 별도의 모듈을 제안한다. 먼저, 수정된 GINA(Graphical Identification and Authentication) 모듈이 필요하다. GINA는 사용자 이름과 비밀번호 또는 스마트카드 개인 식별 번호를 획득하기 위해 실행하는 DLL(Dynamic Linking Library) 파일로, 윈도우 운영체제에서 로그온할 때 사용되고 있으며 윈도우 시스템 디렉터리에 존재하는 MSGINA.DLL 파일을 의미한다. 수정된 GINA를 이용하여 사용자 계정의 추가, 삭제 및 변경 등에 대한 관리는 물론 계정 정책에 따라 로그온을 허용하거나 차단하여 세션을 제어하거나 해당 이벤트에 대한 로그를 획득한다. 시스템에 접속한 사용자는 입력 장치를 사용하여 작업을 수행하는데 이러한 이벤트를 추적하기 위한 모듈을 제안하여 로그를 수집하도록 한다. FTP나 TELNET 등의 네트워크 서비스에 접속한 사용자의 작업 내역과 파일시스템 변경 내역 추적 모듈을 이용하면 시스템에 접속한 사용자에 의한 주요 파일의 조작 내역을 세부적으로 확인할 수 있다.




〔그림 3〕 사용자 행위 추적 모듈 구성도

〔그림 3〕은 사용자의 행위를 추적하기 위한 모듈의 구성을 보이고 있다.

본 논문에서 제안하는 사용자 행위 추적 시스템은 사용자가 시스템에 최초 로그온하여 필요한 작업을 수행한 후에 로그오프하는 전체 과정에 대한 로그 데이터를 획득하도록 구성하였다. 즉, 사용자가 시스템에

〔표 1〕 로그 형식

| 번호 | 원인 | 항 목 | 내 용 | 예 |
|---|---|---|---|---|
| 1 | | SERIAL | 일련번호 | 000999 |
| 2 | | FORMAT -LOCALE | 로케일 ID | CP949 |
| 3 | | AGENT MAC | 에이전트 MAC Address | 00:1F:30:3F:43:5A |
| 4 | | AGENT IP | 에이전트 IP Address | 255.255.255.255 |
| 5 | 언제 | ONTIME | 로그온 및 명령 실행 Date Time | 2011-03-26 11:11:11.111 |
| 6 | | OFFTIME | 로그오프 Date Time | |
| 7 | | SID | 세션 ID | W48b66244065d1172 |
| 8 | | PPID | 부모프로세스 ID | 2339 |
| 9 | | PPNM | 부모프로세스 이름 | explorer.exe |
| 10 | | PID | 프로세스 ID | 234 |
| | | PNM | 프로세스 이름 | cmd.exe |
| 11 | | RUID | 로그온사용자 ID | 505 |
| 12 | 누가 어디서 | RUNM | 로그온사용자 이름 | tom |
| 13 | | SIP | 서버 IP | 255.255.255.255 |
| 14 | | CIP | 클라이언트 IP | 255.255.255.255 |
| 15 | | TTY | 터미널 이름 윈도우 타이틀 | RDP#01 명령 프롬프트 |
| 16 | | SUNM | 다른 계정으로 실행한 사람 이름 FTP 서버 정보 파일 연산 | tom, jane FTP Create, Delete, etc |
| 17 | | LTYPE | 로그온 타입 | Console, Remote |
| 18 | 무엇을 어떻게 | IDATA | 입력한 명령 | dir, ipconfig |
| 19 | | RESERVED | 예약 영역 | |

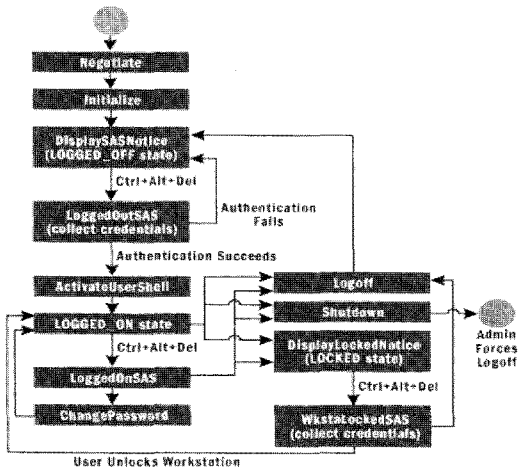
접속하여 수행하는 모든 작업 내역을 추적하기 위한 별도의 로그 획득 모듈을 제안하고 있다. 본 논문에서 제안하고 있는 각 모듈에서 생성되는 로그를 데이터베이스에 통합하여 저장하기 위하여 [표 1]과 같이 로그의 형식을 제안한다.

로그 데이터베이스의 각 레코드는 SERIAL 항목을 기준으로 구분할 수 있으며, 시스템 로케일에 맞도록 로케일 ID를 획득하여 운영체제에서 발생한 로그에서 멀티 바이트 문자를 저장할 때는 코드 변환 처리를 하여 다국어 환경에 대응할 수 있다. 사용자가 언제 로그온, 로그오프하였는지 날짜와 시간을 기록하여 차후에 사용자의 작업 범위를 결정할 수 있다. 어떤 사용자에 의한 행위인지 로그온사용자와 프로세스 정보를 획득하여 기록하는데, 세션 ID(Identifier) 정보는 운영체제에 접속한 각 사용자의 세션을 구분할 수 있는 유용한 정보로 활용할 수 있다. 또한 사용자가 어디에서 접속하였는지 클라이언트 IP, 터미널 이름과 로그온 타입 등을 획득하여 저장하고, 마지막으로 사용자가 실제 실행하고 입력한 명령을 로그에 기록한다. 차후 확장성을 고려하여 예약 영역을 추가하였다. 이를 활용하여 기업의 중요 데이터에 누가, 언제, 어디서, 무엇을, 어떻게 접근하는지를 일목요연하게 감시할 수 있도록 하였다.

사용자의 행위를 추적하기 위해 본 논문에서 제안하는 각 모듈의 세부적인 처리 방안은 다음과 같다.

### 3.1 사용자 로그온 및 로그오프 추적

운영체제에는 사용자의 접속 처리를 담당하는 로그온 모듈이 존재하는데 사용자가 접속을 시도하면 로그온 모듈이 구동되어 로그온 과정에서 [그림 4]와 같은 상태를 거치게 된다. 수정된 GINA 모듈을 적용하려면 먼저 수정하여 구현한 GINA.DLL 파일을 '%SystemRoot%\System32' 디렉터리로 복사한다. 그 다음 레지스트리 키 'HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon' 하위에 수정된 GINA 모듈의 이름으로 REG_SZ 형의 레지스트리 값을 생성하고, 수정된 GINA.DLL 파일의 전체 경로를 문자열로 데이터 항목에 입력한 다음 운영체제를 다시 시작하면 된다.




[그림 4] 사용자 로그온 처리 상태

사용자가 최초 접속을 시도하면 로그온 모듈을 초기화하고 사용자에게 로그오프 상태임을 알려준다. 로그오프 상태에서 사용자가 [Ctrl]+[Alt]+[Del] 키보드를 동시에 누르면 로그온 요청상태가 되고, 사용자에게 인증 요청 화면을 출력한다. 단, 원격에서 접속 시에는 로그오프 상태를 알리지 않고 바로 로그온 요청 상태가 되어 인증 요청 화면이 출력된다.

본 논문에서는 기존의 GINA 모듈에서 LoggedOutSAS와 Logoff 함수에 로그온을 제어하고 로그를 획득하기 위한 기능을 추가하도록 수정하였다. 로그온 절차인 [그림 4]의 LoggedOutSAS 상태에서는, 사용자가 사용자 이름과 비밀번호를 입력하여 로그온을 시도하면 현재 시스템의 시간과 로그온을 시도한 사용자의 이름 그리고 원격 세션 정보를 이용하여 로그온을 시도하는 클라이언트 IP 주소 등의 로그온 관련 정보를 획득한다. 이러한 정보를 이용하여 로그온 제어 정책과 비교하여 로그온을 허용하거나 거부할 수 있다. 로그온을 허용할 경우 생성될 세션 ID 정보를 획득하여 로그에 기록한 후에 다음 상태로 진입할 수 있도록 하면 ActivateUserShell 상태를 거쳐 정상적으로 로그온이 성공하고, 로그온을 거부하면 DisplaySASNotice 상태로 진입하여 로그온 대기 상태가 된다. 운영체제에 로그온한 사용자가 작업을 수행한 다음 로그오프 시에도 로그를 획득해야 한다. 이를 위해 로그온 상태인 LOGGED_ON 상태에서 사용자가 로그오프를 시도할 때 호출되는 [그림 4]의 Logoff 함수에서 현재 시스템의 시간과 로그온 당시에 획득한 사용자 이름, 클라이언트 IP 주소 그리고 세션 ID 정보 등을 이용하여 사용자 로그오프에 대한 로그를 기록한다. 로그오프한 사용자는 DisplaySASNotice 상태로 돌아가서 로그온 대기 상태가 된다.

〔표 2〕 GINA DLL 함수 목록

| 함수 | 설명 |
|---|---|
| ActivateUserShell | 사용자 쉘 프로그램을 활성화시킨다 |
| DisplayLockedNotice | GINA DLL이 잠금 정보를 출력하도록 허용한다 |
| DisplaySASNotice | 사용자가 로그온되지 않았을 때 Winlogon이 본 함수를 호출한다 |
| DisplayStatusMessage | Winlogon이 출력할 상태 메시지와 함께 본 함수를 호출한다 |
| GetConsoleSwitchCredentials | 현재 로그온한 사용자의 신원 증명을 있는 그대로 목적 세션에 전송하기 위해서 읽을 때 Winlogon이 본 함수를 호출한다 |
| GetStatusMessage | Winlogon이 현재 상태 메시지를 얻기 위해서 본 함수를 호출한다 |
| Initialize | 특정 윈도우 스테이션을 위한 GINA DLL을 초기화한다 |
| IsLockOK | 워크스테이션 잠금이 승인되었는지 검증한다 |
| IsLogoffOK | 로그오프가 승인되었는지 검증한다 |
| LoggedOnSAS | 사용자가 로그온되어 있고 워크스테이션이 잠겨있지 않은 동안 SAS(Secure Attention Sequence) 이벤트를 받을 때 Winlogon이 본 함수를 호출한다 |
| LoggedOutSAS | 사용자가 로그온하지 않은 동안 SAS 이벤트를 받을 때 Winlogon이 본 함수를 호출한다 |
| Logoff | 로그오프 연산이 요청되었음을 GINA DLL에 공지한다 |
| Negotiate | 현재 버전의 Winlogon이 GINA DLL과 사용되지 있는지 여부를 알려준다 |
| NetworkProvidedLoad | Winlogon이 유효한 신분 증명 정보를 수집하기 위해 네트워크 제공자를 로드할 때 본 함수를 호출한다 |
| RemoveStatusMessage | GINA DLL가 상태 메시지 출력을 중단하도록 하기 위해 Winlogon이 본 함수를 호출한다 |
| ScreensaverNotify | GINA가 스크린 세이버 연산과 상호작용하도록 허용한다 |
| Shutdown | 시스템 종료 바로 전에 Winlogon이 본 함수를 호출하는데, GINA가 판독기로부터 스마트카드를 빼내는 등의 종료 작업을 수행하도록 허용한다 |
| StartApplication | 시스템이 사용자의 컨텍스트에서 실행된 애플리케이션이 필요할 때 Winlogon이 본 함수를 호출한다 |
| WkstaLockedSAS | 워크스테이션이 잠겨있는 동안 SAS를 받을 경우 Winlogon이 본 함수를 호출한다 |

〔표 2〕는 수정된 GINA DLL이 반드시 제공해야 하는 함수의 목록을 보이고 있다.

서버의 로그온 모듈은 사용자의 계정명과 비밀번호 이외에 계정 잠금 상태에 따라서 사용자의 접속 여부를 결정한다. 이와 같은 모듈을 이용하여 언제 어디서 어떤 사용자가 로그온 및 로그오프 하였는지에 대한 정보를 획득할 수 있으며, 이 정보는 로컬 혹은 원격 접속 사용자의 세션을 구분하기 위한 정보로 활용된다. 〔표 1〕의 로그 형식에서 세션 ID, 서버 IP, 로그온사용자, 로그온 Date Time, 로그오프 Date Time, 클라이언트 IP, 로그온 타입 항목에 대한 정보를 획득하여 로그를 생성한다. 다음 〔표 3〕은 Charles 사용자가 원격에서 터미널서비스를 이용하여 운영체제에 로그온하여 작업한 다음 로그오프할 경우 생성되는 로그의 형식을 〔표 1〕에서 추출하여 보이고 있다. 본 논문에서 제안하는 로그온 추적 모듈은 서버의 로그온 기본 조건을 확장하여 날짜, 시간, 요일, 접속을 시도하는 클라이언트의 IP 주소 등의 다양한 조건에 따라 좀 더 세부적으로 서버에 접속하는 사용자 로그온에 대한 제어 및 로그 획득이 가능하기 때문에 로그온 시스템의 보안을 강화할 수 있다.

〔표 3〕 사용자 로그온 로그오프 로그 형식

| 세션 ID | 서버 IP | 로그온 사용자 | 로그온 Date Time |
|---|---|---|---|
| W48b66244065d1172 | 192.168.0.151 | Charles | 2011-03-26 11:11:11.111 |
| W48b66244065d1172 | 192.168.0.151 | Charles | |
| 로그오프 Date Time | 클라이언트 IP | 로그온 타입 | |
| | 192.168.0.157 | Remote | |
| 2011-03-26 12:11:11.111 | 192.168.0.157 | Remote | |

### 3.2 사용자 입력에 의한 작업 내역 추적

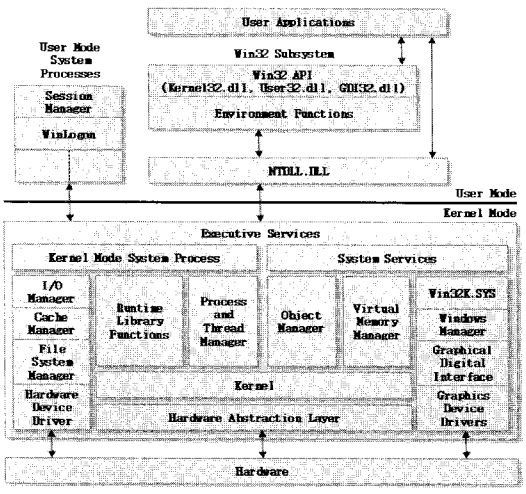
사용자 입력에 의한 작업 내역 추적 모듈은 사용자 세션에 대한 감시 모듈로, 사용자의 로그온 이후에 실행되는 사용자의 쉘을 감시하여 사용자의 입력 행위에 대한 모든 이벤트를 로그로 남긴다. 이를 위해 사용자의 쉘이 실행되는 서비스(Terminal Service, Console 등)의 세션에 대한 로그를 획득한다. 구체적인 추적 방안은 다음과 같다.

먼저, 사용자의 행위를 추적하는 모듈을 시스템에

등록하여 사용자가 로그온 할 때 자동으로 실행되도록 한다. 모듈이 실행되면서 현재 로그온한 사용자가 추적 대상 사용자인지 보안 정책과 비교한 다음, 추적 대상 사용자일 경우에는 사용자의 로그온 시간과 사용자 계정명, 세션 ID, 세션 이름 그리고 원격지에서 접속한 사용자라면 클라이언트의 IP 주소와 해당 서버의 IP 정보를 획득한다. 유일한 세션 정보를 생성하기 위해 시간, 프로세스 ID 및 세션 ID 등의 정보를 이용하여 다른 세션과 구분되는 값을 생성하기 위해 해쉬 함수를 활용하여 세션 구분자를 생성한다.

입력 장치에 의한 사용자 이벤트 후킹은 글로벌 후킹으로 동적 적재 라이브러리 형태의 후킹 프로시저를 이용한다. 후킹은 코드를 실행하는 특정 섹션을 가로채는 기술을 의미한다[11]. 특정 목적으로 파일의 생성이나 뮤텍스의 생성 혹은 레지스트리 접근 등의 이벤트를 추적하고자 할 때 특정 이벤트 관련 API (Application Programming Interface)나 시스템 서비스 콜을 후킹할 수 있다. 후킹은 사용자의 필요에 따라 운영체제의 동작을 수정하기 위한 유용한 방법을 제공할 수 있으며 시스템 동작에 대한 연구 및 시스템 디버깅에도 사용되는 기술이다. 또한 특정 작업의 성능 측정 및 통계를 산출하는 데에도 유용하게 사용되는 기술이다.

후킹은 [그림 5]와 같이 유저 레벨 후킹과 커널 레벨 후킹의 두 가지 형태로 분리할 수 있다[12]. 유저 레벨 후킹은 유저 모드 DLL에 의해 제공되는 함수들을 후킹하는 방법으로 이러한 함수들에 대한 문서화가 잘 되어 있기 때문에 인자들에 대한 정보를 획득하기 쉽고 결과적으로 후킹 함수를 작성하기 쉽다는 장점이 있지만 후킹의 동작이 유저 모드로 제한되고 커널 모드까지 확장되지 않는다는 단점이 있다. 이에 반해 커널 레벨 후킹은 디바이스 드라이버를 작성하여 후킹하는 방법으로 유저 모드 호출이나 커널 모드 호출에 의한 이벤트를 모두 모니터링 할 수 있다. 하지만 시스템 서비스들이 대부분의 경우 문서화되어 있지 않기 때문에 요청에 대한 인자를 해독해야 할 경우가 발생




[그림 5] 유저 레벨과 커널 레벨

[표 4] SetWindowsHookEx 함수의 idHook 인자 목록

| 인자 | 설명 |
|---|---|
| WH_CALLWNDPROC | 시스템이 목적지 윈도우 프로시저에 메시지를 전달하기 전에 메시지를 감시하는 Hook 프로시저를 설치한다 |
| WH_CALLWNPROCRET | 목적지 윈도우 프로시저에 의해 처리된 후 메시지를 감시하는 Hook 프로시저를 설치한다 |
| WH_CBT | CBT 애플리케이션에게 유용한 통지를 받는 Hook 프로시저를 설치한다 |
| WH_DEBUG | 다른 Hook 프로시저를 디버깅하는데 유용한 Hook 프로시저를 설치한다 |
| WH_FOREGROUNDIDLE | 애플리케이션의 Foreground 쓰레드가 유휴 상태로 될 시점에 호출되어질 Hook 프로시저를 설치한다 |
| WH_GETMESSAGE | 메시지 큐에 붙여진 메시지를 감시하기 위한 Hook 프로시저를 설치한다 |
| WH_JOURNALPLAYBACK | WH_JOURNALRECORD에 의해 이전에 기록된 메시지를 붙이는 Hook 프로시저를 설치한다 |
| WH_JOURNALRECORD | 시스템 메시지 큐에 붙여진 입력 메시지를 기록하는 Hook 프로시저를 설치하는데, 이 Hook은 매크로를 기록하는데 유용하다 |
| WH_KEYBOARD | 키스트로크 메시지를 감시하는 Hook 프로시저를 설치한다 |
| WH_KEYBOARD_LL | 저수준의 키보드 입력 이벤트를 감시하는 Hook 프로시저를 설치한다 |
| WH_MOUSE | 마우스 메시지를 감시하는 Hook 프로시저를 설치한다 |
| WH_MOUSE_LL | 저수준의 마우스 입력 이벤트를 감시하는 Hook 프로시저를 설치한다 |
| WH_MSGFILTER | 다이얼로그 박스, 메시지 박스, 메뉴나 스크롤바의 입력 이벤트에 의한 결과로 생성된 메시지를 감시하는 Hook 프로시저를 설치한다 |
| WH_SHELL | 쉘 애플리케이션에 유용한 통지를 받는 Hook 프로시저를 설치한다 |
| WH_SYSMSGFILTER | WH_MSGFILTER와 동일하게 동작하면서 모든 프로그램에서 동작하는 Hook 프로시저를 설치한다 |

하며 커널 모드에 전송되어 온 데이터가 유저 모드의 데이터와 상이할 수 있기 때문에 이에 대한 처리가 필요할 수 있다[12]. 통상적으로 커널 모드 후킹이 구현하기 더 힘들지만 더 좋은 결과를 얻을 수 있다는 장점이 있다.

운영체제가 다양한 인터페이스를 지원하면서 그에 따른 보안 인터페이스의 필요성도 제기되었다. 특히 윈도우 시스템의 경우 멀티유저 환경에서의 사용이 많은 관계로, 보안 관리자는 각 사용자가 올바른 인터페이스를 통해 작업을 수행하고 있는지 확인할 수 있는 보안 인터페이스가 필수적이다. 본 논문에서 제안하는 후킹 함수는 SetWindowsHookEx이며, WH_ MOUSE, WH_KEYBOARD, WH_GETMESSAGE 등을 인자로 전달하여 후킹을 시도하여 입력 장치에 의한 사용자의 작업 내역을 추적할 수 있다.

HHOOK WINAPI SetWindowsHookEx(_in int idHook, _in HOOKPROC lpfn, _in HINSTANCE hMod, _in DWORD dwThreadId);

[표 4]는 SetWindowsHookEx 함수의 idHook 인자에 전달 가능한 값의 목록을 보이고 있다.

키보드와 마우스 등의 입력 장치를 이용한 사용자의 작업 이벤트 중 감시하고자 하는 이벤트가 발생하면 현재 프로세스 ID, 프로세스 이름, 윈도우 타이틀 제목 등을 획득하여 어떠한 프로그램이 실행 중인지 확인할 수 있다. 획득한 데이터는 내부적으로 메시지를 이용한 통신으로 정보를 교환하며, 획득한 데이터를 로그에 기록할 것인지 결정하기 위해 프로세스 이름을 통한 예외 정책, 입력 장치의 선택 여부 등을 비교한다. [표 1]에서 사용자 입력에 의한 작업 내역을 추적하기 위한 항목들에 대한 정보를 획득하여 로그를 생성한다. 다음 [표 5]는 키보드의 엔터키가 감시 대상 이벤트로 설정되어 있는 상태에서, Charles 사용자가 명령 프롬프트 상에서 키보드를 이용하여 ipconfig 명령을 입력하고 엔터키로 실행했을 때 생

[표 5] 사용자 입력에 의한 작업 내역 로그 형식

| 세션 ID | 로그온 사용자 | 명령 실행 Date Time | 프로세스 ID |
|---|---|---|---|
| W48b66244065d1172 | Charles | 2011-03-26 11:11:11.111 | 379 |
| 프로세스 이름 | 입력한 명령 | 윈도우 타이틀 | |
| cmd.exe | ipconfig | 명령 프롬프트 | |

성되는 로그의 형식을 [표 1]에서 추출하여 보이고 있다. 표에는 생략되어 있지만 부모프로세스에 대한 정보도 기록하여 사건 추적 시에 연관관계 등의 유용한 정보로 활용할 수 있다.

이와 같은 절차를 통해 추적 대상 사용자가 시스템에 접속하여 입력 장치를 이용하여 작업을 수행한 내역에 대한 로그를 획득할 수 있으며, 사용자 로그온 및 로그오프 추적 모듈에서 획득한 세션 정보와 상호 연관하여 분석하면 특정 사용자 세션에서 관리 대상 파일의 변경과 같은 사용자의 행위를 세부적으로 확인할 수 있다.

### 3.3 네트워크 서비스 접속 후 작업 내역 추적

FTP 및 TELNET 등의 네트워크 서비스에 접속한 사용자의 작업 내역을 추적하기 위해서는 별도의 서비스 형태로 동작하는 감시 모듈이 필요하다. 본 논문에서는 FTP 접속 사용자의 작업 내역을 추적하기 위한 방안 위주로 기술한다. FTP 서비스 감시 모듈은 FTP 서비스가 사용하는 포트를 감시하여 FTP 명령어 대한 로그를 획득한다. FTP 서비스 포트는 변경될 수 있으므로 추적 모듈을 기동하기 전에 서비스 포트를 설정해야 한다. 이 모듈은 내부적으로 PCAP 라이브러리를 이용해서 획득한 패킷을 분석하여 사용자가 접속한 클라이언트의 IP 주소별로 세션 로그를 기록한다.

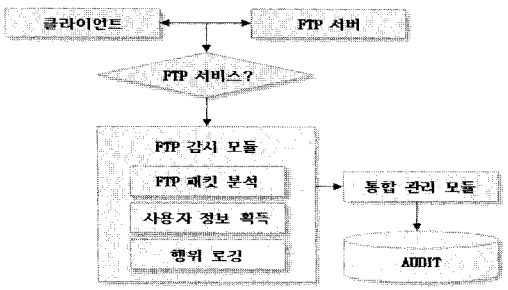
이 모듈에서는 콜백 함수를 사용하는데, 콜백 함수는 운영체제에 의해 호출되는 응용프로그램의 함수라고 할 수 있다[11]. 호출되는 방향이 정상적인 호출과는 달리 반대된다는 의미에서 콜백이라고 하며, 프로그램이 실행되는 동안 지속적으로 수행해야 하는 작업이 있을 때 사용된다. FTP 서비스를 추적하기 위해 설정된 콜백 함수로 패킷이 넘어오게 되면 패킷의 데이터와 FTP 통신 규약(FTP COMMANDS)을 활용하여 사용자의 행위에 대한 로그를 획득할 수 있다. 예를 들어 USER가 붙으면 사용자 이름, LIST가 붙으면 파일의 리스트, DELETE가 붙으면 파일 삭제, 230이면 시작, 530이면 종료와 같은 방법으로 구분할 수 있는데, 그 결과 FTP 서비스에 접속한 사용자가 악성코드를 업로드하거나 혹은 기업의 중요자료를 불법적으로 다운로드하는 등의 작업 내역을 구체적으로 확인할 수 있다. [표 1]의 로그 형식에서 세션 ID, 명령 실행 Date Time, FTP 서버 정보, 입력한 명령에 대한 정보를 획득하여 로그를 생성한다. 다음 [표

6]은 원격에서 FTP 서비스를 이용하여 서버에 로그온하여 작업한 다음 로그오프할 때 생성되는 로그의 형식을 [표 1]에서 추출하여 보이고 있다.

[표 6] FTP 서비스 접속 후 작업 내역 로그 형식

| 세션 ID | 명령 실행 Date Time | 서버 정보 | 입력한 명령 |
|---|---|---|---|
| W48b66244065d1172 | 2011-03-26 11:11:11.111 | FTP | ls |
| W48b66244065d1172 | 2011-03-26 11:11:11.222 | FTP | Directory Send OK. |
| W48b66244065d1172 | 2011-03-26 11:11:21.111 | FTP | put ftp.txt |
| W48b66244065d1172 | 2011-03-26 11:11:21.222 | FTP | File Send OK. |
| W48b66244065d1172 | 2011-03-26 11:11:31.111 | FTP | QUIT |

다음 [그림 6]은 FTP 서비스에 대한 감시 모듈의 구성을 보이고 있다.




[그림 6] FTP 서비스 감시 모듈

TELNET 등과 같은 다른 네트워크 서비스에 접속한 사용자의 작업 내역도 유사한 방식으로 획득할 수 있다.

### 3.4 파일시스템 변경 내역 추적

파일시스템에 대한 변경 내역을 추적하기 위해서 먼저 시스템에 연결되어 있는 고정 디스크를 구해 각 디스크 별로 핸들을 얻은 다음 파일시스템 변경 통지 ReadDirectoryChangesW 함수를 등록하여 파일 및 디렉터리의 변경 내역을 구조체로 받는다.

```
BOOL WINAPI ReadDirectoryChangesW (_in HANDLE hDirectory, _out LPVOID lpBuffer, _in DWORD nBufferLength, _in BOOL bWatchSubtree, _in DWORD dwNotifyFilter, _out_opt LPDWORD lpBytes Returned, _inout_opt LPOVERLAPPED lpOverlapped, _in _opt LPOVERLAPPED_COMPLETION_ROUTINE lpCompletionRoutine);
```

파일시스템 변경 내역을 추적하기 위한 모듈은 서비스 형태로 구성되며, 구체적으로 다음의 절차에 따라 동작한다. 먼저, 'HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services' 레지스트리 키의 하위에 서비스 이름과 동일한 이름으로 레지스트리 키를 등록하여 운영체제가 시작될 때 서비스가 자동으로 시작되도록 설정한다. 서비스가 시작되면, 윈도우 시스템의 논리적인 하드디스크 정보를 수집하여 논리적인 하드디스크 하나당 파일시스템의 이벤트를 받는 쓰레드를 생성한다. 각 쓰레드는 ReadDirectoryChangesW 함수를 사용하여 파일시스템의 변경 이벤트를 등록한다. 발생한 이벤트에 대한 처리를 위해 하나의 쓰레드가 이벤트를 대기하며, 이벤트 발생 시 발생한 이벤트에 대한 정보인 이벤트 발생 시간, 파일 이름, 연산 정보 등을 획득하고, 발생한 이벤트에 대한 정보와 로그가 발생한 서버의 IP 주소, 로케일 ID 정보 등을 취합하여 로그에 기록한다. 로그 처리의 최적화를 위해 큐를 이용하고, 큐에 쌓인 로그를 처리하기 위해 별도의 쓰레드를 이용한다. 파일시스템의 변경 이벤트를 처리하는 쓰레드와 교착 상태를 피하기 위해 동기화가 필요하다.

[표 7]은 ReadDirectoryChangesW 함수의 dwNotifyFilter 인자에 전달 가능한 값의 목록을 보이고 있다. 넘겨오는 구조체에는 읽기(마지막 접근 시간), 생성, 삭제, 재 명명, 수정(보안속성 변경, 크기 변경, 생성시간 변경, 마지막 접근시간 변경, 마지막 쓰기시간 변경, 속성 변경) 등의 정보가 기술되어 있다. 이 정보를 입력 장치에 의한 사용자의 작업 내역 로그와 조합하여 통합된 이벤트를 확인할 수 있다.

[표 1]에서 파일시스템 변경 내역을 추적하기 위한 항목에 대한 정보를 획득하여 로그를 생성한다. [표 8]은 Charles 사용자가 메모장을 이용하여 'C:\Data\integrity.hwp' 파일을 변조했을 때 생성되는 로그의 형식을 [표 1]에서 추출하여 보이고 있다. 표에는 생략되어 있지만 부모프로세스에 대한 정보도 기록하여 사건 추적 시에 연관관계 등의 유용한 정보로 활용할 수 있다.

〔표 7〕 ReadDirectoryChangesW 함수의 dwNotifyFilter 인자 목록

| 인자 | 설명 |
|---|---|
| FILE_NOTIFY_CHANGE_FILE_NAME | 대상 디렉터리 또는 하위 트리 내의 파일 이름 변경에 대한 변경 통지 대기 연산이 응답하도록 하는데, 파일 이름 변경, 생성과 삭제를 포함한다 |
| FILE_NOTIFY_CHANGE_DIR_NAME | 대상 디렉터리 또는 하위 트리 내의 디렉터리 이름 변경에 대한 변경 통지 대기 연산이 응답하도록 하는데, 디렉터리 생성과 삭제를 포함한다 |
| FILE_NOTIFY_CHANGE_ATTRIBUTES | 대상 디렉터리 또는 하위 트리 내의 속성 변경에 대한 변경 통지 대기 연산이 응답하도록 한다 |
| FILE_NOTIFY_CHANGE_SIZE | 대상 디렉터리 또는 하위 트리 내의 파일 크기 변경에 대한 변경 통지 대기 연산이 응답하도록 하는데, 운영체제는 파일이 디스크에 저장되었을 때에만 파일 크기의 변경을 감지한다 |
| FILE_NOTIFY_CHANGE_LAST_WRITE | 대상 디렉터리 또는 하위 트리 내의 파일 마지막 수정한 시간 변경에 대한 변경 통지 대기 연산이 응답하도록 하는데, 운영체제는 파일이 디스크에 저장되었을 때에만 파일 마지막 수정한 시간의 변경을 감지한다 |
| FILE_NOTIFY_CHANGE_LAST_ACCESS | 대상 디렉터리 또는 하위 트리 내의 파일 마지막 액세스한 시간 변경에 대한 변경 통지 대기 연산이 응답하도록 한다 |
| FILE_NOTIFY_CHANGE_CREATION | 대상 디렉터리 또는 하위 트리 내의 파일 생성 시간 변경에 대한 변경 통지 대기 연산이 응답하도록 한다 |
| FILE_NOTIFY_CHANGE_SECURITY | 대상 디렉터리 또는 하위 트리 내의 보안 기술자 변경에 대한 변경 통지 대기 연산이 응답하도록 한다 |

〔표 8〕 파일시스템 변경 내역 로그 형식

| 세션 ID | 로그온 사용자 | 명령 실행 Date Time | 프로세스 ID |
|---|---|---|---|
| W48b66244065d1172 | Charles | 2011-03-26 11:11:11.111 | 379 |
| **프로세스 이름** | **파일 연산** | **입력한 명령** | |
| notepad.exe | Modify | C:\Data\integrity.hwp | |

파일시스템 변경 통지 ReadDirectoryChangesW 함수를 OVERLAPPED_COMPLETION_ROUTINE과 함께 사용하기 위해서는 WaitForSingleObjectEx 등의 Ex 계열 Wait 함수를 사용해야 하며, Ex 계열 함수에 공통적으로 존재하는 Alertable 인자를 TRUE로 설정해야 한다. 또한 ReadDirectoryChangesW 함수와 Ex 계열 Wait 함수는 동일한 쓰레드에서 호출되어야 한다.

## IV. 로그의 활용 방안

시스템 로그파일은 소유자나 시스템 관리자에 의해 쉽게 변경될 수 있기 때문에 무결성을 유지하기 어렵고, 로그의 양이 방대하기 때문에 법적 증거자료로 분석하는데 많은 시간이 소요되는 것은 물론 증거가 사라지는 문제가 발생하여 사이버 범죄가 발생했는지 확인할 수 없는 경우도 있다. 또한 권한 있는 내부 사용자의 행위에 대한 로그 획득이 어려울 수도 있다.

본 논문에서 제안한 시스템을 활용하면 서버에 접속하는 사용자의 작업 행위에 대한 로그에 기록한 다음 차후에 추적할 수 있도록 하여 서버 자원에 대한 보안을 유지할 수 있다. 시스템 로그와는 달리 권한 있는 내부 사용자 및 관리자의 행위에 대한 로그도 획득할 수 있으며, 소유자나 시스템 관리자에 의해 쉽게 변경될 수 없도록 별도의 데이터베이스에 로그를 저장하여 무결성을 유지할 수 있다. 다수의 분리된 파일에 저장되던 다량의 로그를 하나로 통합하고 이를 사용자 세션으로 검색하도록 하여 보안사고 발생 시에 원인 분석의 어려움과 많은 시간이 소요되던 문제에 좋은 방안이 될 수 있다. 다수의 추적 모듈에서 획득한 로그는 생성되는 계층이나 시간이 다를 수 있기 때문에 서로 다른 추적 로그 간의 관계를 연결해 줄 수 있는 참조 요소가 필요하다. 본 논문에서는 로그의 세션 ID를 기준으로 누가, 언제, 어디서, 무엇을, 어떻게 접근하였는지 조회하도록 하였으며, 세션 ID 단위로 분류된 로그의 내부에서 상호 연관 관계를 형성할 때 공통 요소인 로그온 사용자와 시간에 대한 정보를 이용할 수 있다. 사용자 행위 추적 시스템은 본 논문에서 제안하는 보안 모델의 핵심 기능이며, 향후 컴퓨터 포렌식의 근거 자료로 활용할 수 있다[13]. 사용자 행위 추적 시스템에서 획득한 사용자 로그온 및 로그오프 로그, 사용자 입력에 의한 작업 내역 로그, 네트워크 서비스 접속 후 작업 내역 로그와 파일시스템 변경 내역 로그 등을 서로 연관시키기 위해서는 정규화 과정이 필요하다. 다수의 로그를 상호 연관할 수 있도록 로그를 구성하는 데이터의 형식을 통일하기 위한 정규화는 물론 보안 사고가 발생했을 경우 시간의 흐름에 따라 분석을 정확하게 하기 위해서 로그 생성 시간을 동기화하기 위한 정규화를 수행해야 한다. 즉, 서로

다른 로그의 형식으로부터 필요한 부분과 공통부분을 추출하여 정규화 과정을 수행한 다음 데이터베이스에 저장한다. 사건 분석에 대한 오해 소지와 각 시스템에서 발생하는 사건간의 시간 간격의 계산 오류를 방지하기 위해 윈도우 운영체제는 시스템 시간을 인터넷 상의 시간 서버와 동기화시켜주는 윈도우 타임 서비스를 제공한다. 추가적으로 로그의 항목을 이용하여 특정한 유형의 로그와 중복되는 정보들을 초기에 제거하여 분석 대상을 줄이면 향후 분석 단계에서의 효율을 높일 수 있다.

로그 관리의 고유한 이점을 제외하더라도 몇몇 법률과 규정들은 기관들로 하여금 특정 로그들을 보존하고 검토하도록 요구하고 있다. 다음은 기관들이 로그 관리에 대한 필요성을 명확히 하는데 도움이 되는 미국 내 주요 규정들과 표준 그리고 지침들의 목록이다.

- □ 연방정부의 정보 보안관리에 대한 결의(FISMA) : FISMA는 각각의 정부기관들이 그들의 사업과 자산을 뒷받침하는 정보시스템에 대한 정보보안을 제공하는 프로그램을 기관 전체적으로 개발하고 문서화하며 수행해야 할 필요성에 대해 강조하고 있다.
- □ Gramm-Leach-Bliley 결의(GLBA) : GLBA는 금융기관들이 보안상의 위협으로부터 그들 고객의 정보를 보호하도록 요구하고 있다.
- □ 미국 의료 데이터 보관기준(HIPAA) : HIPAA는 특정한 의료정보에 대한 보안 기준을 제시하고 있다.
- □ Sarbanes-Oxley 법안(SOX) : 2001년 말 미국의 Fortune지의 10대 기업에 속했던 엔론社, 웰드컴社와 유수 기업들이 회계 법인과 공모로 분식회계를 통해 주주에게 막대한 재무적 손실을 발생시킨 사건으로 인하여 폴 샤베인 상원의원과 마이클 옥슬리 하원의원의 발의에 의하여 2002년 획기적인 회계개혁법인 SOX 법안이 미국에서 제정되었다[14]. 이에 미국 내 모든 상장기업은 자신들이 발표한 재무정보가 정확하고 건전한 것임을 반드시 입증해야 한다. 즉, 모든 상장기업은 년차 보고서에 자사의 재무보고에 대해 적절한 내부통제가 이루어지고 있고, 이러한 내부통제가 유효하게 이행되었음을 최종 감사 보고서에 경영자가 서명하게 함으로써 책임을 지도록 하고 있다. 2006년에는 미국 내 모든 내·외국 상장기업에 준수를 요구하였으며 2007년에는 증권거래위원회(SEC : Securities and Exchange Commission)에 등록된 모든 기업과 비상장 기업에 준수를 요구하고 있다.

SOX 법안이 기본적으로 재무나 회계업무에 적용되고 있지만, 이들 업무를 지원하는 정보기술의 역할까지도 그 영향력 하에 있다. 보안 침해의 흔적을 조사하기 위해 규칙적으로 로그를 점검하거나 감사자들이 기록들을 검토할 수 있도록 로그와 로그 점검사항들의 기록을 보존하여 SOX 법안의 준수를 지원할 수 있다.

SOX 법안 404조항에는 기업 정보시스템의 사용자 관리, 정보의 엄격한 내부 접근통제, 정확하고 신뢰성 있는 정보 자원의 유지, 기업 정보시스템의 감사 및 모니터링 등과 같은 목표를 달성하기 위하여 기업이 지속적으로 이행해야 하는 일련의 내부 프로세스를 정의하고 있다[2]. 내부통제의 네 가지 요구 조건으로는 다음 [표 9]와 같이 인가된 사용자로 엄격 제한, 자동화 프로세스 및 승인 절차에 의한 처리, 정의된 절차에 의해서만 접근 허용, 시스템 및 보안 장애 시 즉시 복구 및 대응 체제 지원을 언급하는데 대부분 IT 보안의 기술적, 운영적인 측면에 전적으로 의존하고 있다. 본 논문에서 제안한 사용자 로그온 로그오프 추적 모듈을 이용하여 사용자의 접근에 대한 인증과 통제를 강화하여 허용되지 않은 접근을 제한하는 것은 물론 언제 어디서 어떤 사용자가 로그온 및 로그오프 하였는지에 대한 로그를 획득하여 사용자의 접근 내역에 대한 검증을 지원할 수 있다. 사용자 입력에 의한 작업 내역 추적과 네트워크 서비스 접속 후 작업 내역 추적 그리고 파일시스템 변경 내역 추적 모듈은 중요한 회계나 개인정보가 인적인 접근이 허용되지 않은 자동화된 프로세스에 의해 정의된 절차를 통해서만 접근되는지와 같은 중요 업무에 대한 투명성과 비밀성에 대한 감시와 로그 기록을 지원할 수 있다. 예를 들어, 'C:\Data\integrity.hwp' 파일은 기업의 기밀 정보가 저장되어 있는 문서로 반드시 조직에서 내부적으로 합의된 문서 편집 프로그램을 이용하여 관리되어야 하는데, 어느 시점인가에 이 문서가 변조되는 사건이 발생하였다. 이를 추적하기 위해 먼저 이 문서가 변조된 시간을 기준으로 어떤 사용자가 운영체제에 접속했었는지 [표 3]의 사용자 로그온 및 로그오프 로그를 조회하였고, 그 결과 Charles 사용자가 사건의 시간 범위 내에서 원격 터미널을 이용하여 로그온 및 로그오프를 수행한 내역을 확인한 다음 해당 세션에 대한 세션 ID 정보를 획득하였다. 이 세션 ID를 기준으로 다

시 〔표 5〕의 사용자 입력에 의한 작업 내역 로그를 조회하여 Charles 사용자가 윈도우 메모장으로 'C:\Data\integrity.hwp' 파일을 열어서 키보드와 마우스 등의 입력 장치를 이용하여 문서를 수정한 내역을 문자 단위의 편집 내역까지 면밀하게 확인하였다. 'C:\Data\integrity.hwp' 파일을 주요 관리 대상 문서로 등록하여 파일시스템 변경 내역 추적 모듈에서 감시하도록 하면, 〔표 8〕의 파일시스템 변경 내역 로그에서도 어떤 사용자가 어느 세션에서 해당 파일에 어떠한 작업을 수행하였는지 그 내역을 확인할 수 있다.

〔표 9〕 SOX 404조 준수를 위한 내부통제 요구조건

| 요구조건 | 내용 |
|---|---|
| 사용자 인증 | 회계 및 업무 관련 정보 시스템은 허용된 사용자만으로 엄격히 제한되어야 하며, 허용되지 않은 접근을 제한할 수 있는 검증된 기술적, 절차적 통제조치가 구비되어야 한다 |
| 투명성 | 입력된 회계 및 개인정보는 인적인 접근이 허용되지 않은 자동화된 프로세스와 승인 절차에 의해 처리되어야 한다 |
| 비밀성 | 처리되고 저장되는 회계 및 개인정보는 정의된 절차를 통한 접근만을 허용하며, 도청 및 도난 등의 접근으로부터 노출되지 않도록 검증된 보안통제가 적용되어야 한다 |
| 업무연속성 계획 | 회계 및 업무 관련 정보시스템은 시스템 및 보안장애 시에 감내할 수 있는 시간 내에 복구 및 대응을 통해서 서비스가 제공되어야 한다 |

일본, 영국, 캐나다, 프랑스, 중국, 대한민국 등이 미국 SOX 법 성립의 영향을 받아 기업의 재무보고 부정을 방지하는 법규제를 실시하고 있다. 법규제에 의해 기업에게 내부통제를 요구하는 것은 국제적인 흐름이다. 일본판 SOX법으로 불리는 금융상품거래법에서는 유가증권보고서 등의 재무보고와 관련된 서류의 기재사항에 잘못된 부분이나 부정이 없는지를 보증하는 '관리체제'의 확립을 요구하고 있다. 관리체제란 내부통제가 유효하게 기능하는 체제를 말하며, 긴급성이 높은 조항에서부터 순차적, 단계적으로 시행되고 있다. 내부통제보고서의 제출 및 감사와 관련해서는 부칙 제 15조에 '2008년 4월 1일 이후에 개시하는 사업년도부터 적용'이라고 정하고 있으며, 2009년 3월기 결산부터 상장기업 약 3,850사와 그 연결자회사 및 아웃소싱회사를 대상으로 적용하고 있다. 내부통제란 기본적으로 업무의 유효성 및 효율성, 재무 보고의 신뢰성, 사업 활동에 관련된 법령 등의 준수 및 자산 보전의 4개 목적의 달성이라는 합리적인 보증을 얻기 위해서, 업무에 포함되어 조직 내의 모든 사람에 의해 수행되는 프로세스를 말하며, 통제 환경, 위험의 평가와 대응, 통제 활동, 정보와 전달, 감시 활동 및 IT에의 대응에 관한 6개의 기본적 요소로 구성된다. 일본판 SOX법에서는 내부통제의 구성요소로서 'IT에의 대응'을 추가하고 있는 점이 특징이다. 'IT에의 대응'이란 '조직의 목적을 달성하기 위해서 사전에 적절한 방법과 절차를 정하고, 이를 바탕으로, 업무를 수행하는데 있어 조직 내외의 IT에 적절히 대응하는 것'을 의미한다. 바꾸어 말하면, 정보 시스템이 내부통제에서 중요한 위치를 차지하고 있다. 기업의 부정이나 잘못을 방지하려면 기업회계를 투명화하기위한 체계가 불가결하다. 서버에 접속한 사용자의 작업 내역 추적 로그에 관련한 것은 내부통제의 기본적 요소 가운데 감시 활동 및 IT에의 대응이며, IT에의 대응 중에서는 IT의 통제가 해당된다. IT의 통제란 IT를 도입한 정보 시스템에 관한 통제이며, 자동화된 통제를 중심으로 하지만 수작업에 의한 통제도 포함된다. IT에 대한 통제 활동은 전반 통제와 업무 처리 통제로 구성된다. IT와 관련된 전반 통제란 업무 처리 통제가 유효하게 기능하는 환경을 보증하기 위한 통제 활동을 의미하며, 통상적으로는 복수의 업무 처리 통제에 관계된 방침과 수속을 말한다. IT와 관련된 전반 통제의 구체적인 예로는 시스템의 개발 및 보수와 관련된 관리, 시스템의 운용 및 관리, 내외로부터의 접근 관리 등과 같은 시스템의 안전성의 확보, 외부 위탁에 관한 계약의 관리 등이다. IT와 관련된 업무 처리 통제란 업무를 관리하는 시스템에 대하여 승인된 업무가 모두 정확하게 처리되고 기록되는 것을 확보하기 위해 업무 프로세스에 포함된 IT와 관련된 내부통제이다. IT와 관련된 업무 처리 통제의 구체적인 예로는 입력 정보의 완전성, 정확성, 정당성 등을 확보하는 통제, 예외처리의 수정과 재처리, 마스터 데이터의 유지 관리, 시스템의 이용에 관한 인증, 조작 범위의 한정 등 접근의 관리이다. 이상의 통제 활동에 있어서 필요하게 되는 로그는, 관리를 올바르게 실시하고 있다는 증거를 나타내고 정보가 정당한 것임을 나타내는 것이다. 그러한 올바름을 완전하게 증명하기 위해서는 본 논문에서 제안하는 바와 같이 운영체제에 접속한 사용자의 행위에 대한 모든 로그를 획득할 필요가 있다. 현재의 내부통제에서는 그 정도 수준까지의 보증은 요구하지 않는 실정이지만, 이러한 내부통제가 유효하게 이행되었음을 최종 감사 보고서에 경영자가 서명하게 함으로

써 책임을 지도록 하고 있기 때문에 앞으로 요구하는 수준은 지속적으로 상승할 것이다.

대한민국에서도 2007년부터 전자금융거래법 제정안의 시행으로 금융권과 공공시장의 정보보호에 대한 책임, 의무가 강화됨에 따라 보안 로그 분석에 대한 수요가 계속적으로 확산될 것이다.

공격에 대한 근원지를 찾아낼 수 있다면 공격으로 인한 자산의 손실에 대해서 공격자에게 법적 책임을 묻기 위해 로그를 증거로 제출할 수 있을 것이다. 이렇듯 보안관리를 위해서는 시스템에서 발생할 수 있는 취약점 점검 및 제거와 같은 예방활동도 중요하지만 만일의 사고에 대비하여 로그를 관리하고 주기적으로 분석하는 작업 또한 중요한 것이다.

## V. 결 론

기존의 시스템 로그와 응용프로그램 로그 정보만으로는 기업의 중요한 업무의 보안 감사 및 모니터링을 위해서 불충분하다. 특히 SOX 법안에서와 같이 기업의 중요한 회계, 재무 업무 처리 등의 내부통제를 위한 기업정보시스템에서의 감사 및 모니터링은 구조적·시간적·인적 한계를 가진다. 따라서 시스템에 설치되는 응용프로그램에 의존하지 않고, 시스템에 접속한 모든 사용자의 작업 내역을 수집하고 이를 통합적으로 이용하여 중요 업무의 보안 감사 및 모니터링, 더 나아가 불법적 침입에 대한 추적을 지원하기 위한 상호 연관 기법을 활용하여 중요 업무 감사 시 요구되는 사건 정보의 정확성을 향상시키고 추적하는데 걸리는 시간을 현저하게 줄일 수 있다.

본 논문에서는 기업의 중요 정보를 처리하는 서버의 운영체제에 접속한 사용자에 대한 행위를 면밀히 감시, 분석함으로써, SOX 법안에서 요구하는 수준의 기업 내부통제를 위한 세션로깅시스템을 제안하였다. 그 결과, 기업의 중요 정보에 대하여 누가, 언제, 어디서, 무엇을, 어떻게 접근하는지에 대한 정보를 획득하여 운영체제에 접속한 내·외부 사용자 세션 단위로 작업 내역에 대한 로그를 명확하게 획득할 수 있다. 이를 활용하여 기업 내의 핵심 기술 유출 등 내부침해 발생 시에 책임 규명을 분명히 하고 법적인 대응 수단을 제공할 수 있도록 중요 프로그램 및 데이터에 대한 상세 접근 내역 관리 및 역추적 기능을 제공하여 컴퓨터 포렌식 기술을 지원할 수 있다. 통합된 로그 데이터를 이용하면 사건 정보에 대한 정확성을 향상시키는 것은 물론 사건을 추적하는데 걸리는 시간을 현저히 줄일 수 있다. 현재 SOX법안에서 요구하는 내부통제를 위한 기본적 요소 가운데 감시 활동 및 IT에의 대응 등에 활용될 수 있다.

본 연구에서 제안한 시스템은 통합보안시스템의 핵심 요소기술로도 활용될 수 있다. 특히 내부통제에 따른 감사 로그의 수집 및 관리, 컴퓨터 포렌식 기술 등은 미래 기업통합보안 솔루션 개발에 반드시 필요한 중요한 요소기술로서 그 가치가 매우 크다고 할 수 있다.

기업에 가장 민감하고 중요한 핵심 기술정보의 관리와 통제를 외국 기술에 전적으로 의존하게 된다면, 이로 인하여 많은 보안 문제가 발생할 것으로 예상된다. 따라서 본 논문에서 제안하는 관리자 및 사용자의 세션 로그 개발 기술 외에도 추가적으로 핵심적인 컴퓨터 포렌식 대응 기술의 확보가 필요하다. 향후 이러한 로그 관리와 컴퓨터 포렌식 기술은 국가적인 차원에서도 중요한 인프라 기술이 될 것이며, 그 응용분야와 수요는 크게 증가할 것으로 예측된다.

내부통제 모니터링 시스템에서 수집되는 데이터는 법정에서 채택될 수 있는 증거 수준의 자격이 있어야 하고, 관련이 있어야 하며, 인증되어야 한다. 또한 SOX 법안의 요구에 맞게 로그 서버에서 통합 관리하는 과정이 필요하다. 이 때 막대한 양의 로그 및 증거 데이터를 효과적으로 저장하고 관리할 수 있는 통합된 형태의 로그관리 서버 구축 기술에 대한 연구가 지속적으로 필요하다.

## 참고문헌


[1] M. Souppaya and K. Kent, "Guide to Computer Security Log Management," National Institute of Standards and Technology, Apr. 2006.

[2] D.C. Brewer, "Security Controls for Sarbanes-Oxley Section 404 IT," Wiley, May. 2006.

[3] 내부회계관리제도운영위원회, "내부 회계 관리제도 모범규준," 금융감독원, 2005년 6월.

[4] 이도영, 김일곤, "법적 증거능력 및 증명력을 위한 컴퓨터 포렌식에 관한 연구," 한국정보처리학회 춘계학술발표대회 논문집, 11(1), pp. 1149-1152, 2004년 5월.

[5] J. Shenk, "SANS Annual 2009 Log Management Survey," SANS, Apr. 2009.

[6] 김완집, 염홍열, "이기종 로그에 대한 통합관리와 IT 컴플라이언스 준수," 정보보호학회지, 20(5), pp. 65-73, 2010년 10월.
[7] 고은주, 오세민, 장은겸, 이종섭, 최용락, "컴퓨터 포렌식스 지원을 위한 시스템 로그 및 휘발성 정보수집에 관한 연구," 한국정보기술전략혁신학회, 10(4), pp. 41-56, 2007년 12월.
[8] J. Babbin, D. Kleiman and E.F. Carter, Security Log Management, SYNGRESS, pp. 244-251, Jan. 2006.
[9] R. Rinnan, "Benefits of Centralized Log file Correlation," Master's Thesis, Gjovik University College, Jan. 2005.
[10] J.Q. Walker, "Security Event Correlation: Where Are We Now?," NetIQ Corporation, Nov. 2001.
[11] I. Ivanov, "API hooking revealed," available at http://www.codeproject.com/KB/system/hooksys.aspx, 2002.
[12] P. Dabak, S. Phadke and M. Borate, Undocumented Windows NT, M&T Books, Oct. 1999.
[13] M. Pollitt, "Computer Forensics: An Approach to Evidence in Cyberspace," Proceedings of the National Information Systems Security Conference, Vol. II, pp. 487-491, Oct. 1995.
[14] 김경호, "회계투명성 확보는 기업의 장기 성장전략: 엔론 사태의 교훈," 회계기준위원회, 2002년 3월.

## 〈著者紹介〉

정 창 성 (Chang-sung Jung) 종신회원
2000년 2월: 한밭대학교 컴퓨터공학과 학사
2002년 2월: 충남대학교 컴퓨터과학과 석사
2009년 3월~현재: 한밭대학교 컴퓨터공학과 박사과정
2001년 8월~2002년 2월: 한국전자통신연구원 위촉연구원
2002년 3월~현재: 티에스온넷(주) 정보보호연구소 책임연구원
〈관심분야〉 컴퓨터 포렌식, 보안 운영체제, 시스템 보안, 네트워크 보안

김 영 찬 (Young-chan Kim) 정회원
1985년 2월: 아주대학교 전자공학과 학사
1987년 2월: KAIST 전기 및 전자공학과 석사
1995년 2월: KAIST 전기 및 전자공학과 박사
1985년 2월~1990년 8월: 삼성전자 종합연구소 연구원
1995년 9월~1996년 2월: 한국과학기술원 위촉연구원
1996년 3월~1997년 2월: University of Arizona Visiting Scholar
1997년 5월~1998년 9월: 한국전자통신연구원 선임연구원
2005년 2월~2006년 1월: Oklahoma State University Visiting Scholar
1998년 10월~현재: 한밭대학교 교수
〈관심분야〉 임베디드시스템 소프트웨어, 데이터베이스 시스템, 시스템 보안, 시스템 검증