Secure OS 기반에서 상호연관 기법을 통한 효과적 상세 로그 감사

구하성* · 박태규**

Efficient Fine-grained Log Auditing using Correlation Method based on Secure OS

Ha-Sung Koo* · Tae-Kyou Park**

요 약

본 논문은 Secure OS 기반으로 운용되는 중요 임무 서버에서 효율적인 상세 보안 감사 방법을 다룬다. 이를 위해서 임무 서버들 내에서 프로세스, 객체, 사용자 명령, DB 쿼리 수준에서의 상세한 보안 로그가 3종의 로그 수집 모듈에 의하여 수집된다. 로그 수집 모듈은 자체 개발한 것으로서, 보안 시스템의 한 구성 요소로 포함되어 있다. Secure OS의 모듈은 프로세스와 객체 단위의 시스템 보안 로그를, BackTracker의 모듈은 사용자 수행명령 세션 로그를, SQLTracker의 모듈은 데이터베이스 쿼리를 상세한 수준으로 수집한다. 특정 사용자 혹은 객체에 대한 사용 행위를 감사하고 추적하고자 할 때, 본 보안 로그 간의 상호연관 기법은 상세 감사 및 모니터링 업무를 효과적으로 지원할 수 있다.

ABSTRACT

This paper presents the effective and detailed secure monitoring method being used based on Secure OS. For this, the detailed secure log of process, object, user's command and database query in task server are collected by 3 kinds of log collecting module. The log collecting modules are developed by ourselves and contained as constituents of security system. Secure OS module collects process and system secure log of objective unit, Backtracker module collects user's command session log, SQLtracker module collects database query in details. When a system auditor monitors and traces the behaviour of specified user or individual user, the mutual connection method between the secure logs can support detailed auditing and monitering effectively.

키워드

로그, 감사, 상호연관, Secure OS, SOX

Key word

Log, Auditing, Correlation Method, Secure OS, SOX

* 정회원 : 한서대학교 컴퓨터정보공학과 교수
** 정회원 : 한서대학교 컴퓨터정보공학과 교수(교신저자, tkpark@hanseo.ac.kr)

접수일자 : 2011. 08. 31
심사완료일자 : 2011. 10. 14

Ⅰ. 서 론

2002년 미국에서 재무회계 개혁법인 사베인-옥스리 법안(Sarbanes-Oxley Act)(이하 SOX 법안)이 제정되었다. 이에 따라 미국 내 상장기업은 재무정보의 정확성과 건전성을 반드시 입증해야 한다. 즉 모든 상장기업은 재무보고 시 내부통제가 이루어지고 있고, 이 과정이 이행되었음을 감사 보고서에 경영자가 서명하여 책임을 지도록 하고 있다. 일본의 경우에도 J-SOX 법안이 발효됨에 따라 2008년부터 의무적으로 상장기업은 물론 약 5만여 중소기업도 내부통제 제도와 기업회계의 투명성을 보증하도록 의무화하고 있다. 우리나라에서도 향후 글로벌 표준으로서의 SOX 법안을 채택하지 않을 수 없을 것이며, 이를 위한 내부통제 기술이 요구될 것이다. 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 말한다. 이와 같은 내부통제의 기술적 목표를 달성하기 위해서는, 중요 서버에 적용되는 보안 커널 기반의 접근통제 시스템, 즉 Secure OS(보안 운영체제)의 활용이 매우 효과적인 방법으로 제시되고 있다[1]. 본 논문에서는 SOX 법안을 준수하기 위해서 기업에서 수행하는 내부통제를 위한 기업정보시스템에서의 감사 및 모니터링을 위한 로그의 상호연관(Correlation) 방법을 다룬다.

Ⅱ. 로그 수집 시스템

시스템 관리자에게 기업정보시스템의 감사 및 모니터링 수행 중 가장 큰 문제는 응용 서버 로그, 웹 서버 로그, 데이터베이스 로그 등 응용프로그램 로그와 시스템 자체의 로그 등 많은 로그들을 수집, 유지, 관리하는 것이다. 시스템에 그 외의 응용 프로그램이 설치되어 있다면 상호연관을 해야 할 로그는 더욱 다양해질 수밖에 없다. 더욱이 시스템 로그 파일의 양이 증가하면서 보안 관리자가 수작업으로 불법적 침입 흔적이나 시스템 사용자의 작업 이력을 추적하는 것은 매우 힘든 일이다.

보안 관리, 보안 감사, 시스템 모니터링, 더 나아가 포렌식(Forensic)에 대응하는 방법은 기존 보안 시스템인 방화벽, 침입탐지시스템 등에서 발생되는 보안 로그와 중요 임무 서버(Mission Critical Server)들에서 발생되는 각종 시스템 로그 등을 활용하여, 각 시스템의 로그 간의 상호연관 기법을 이벤트 로그 분석기를 통하여 수행하는 것이 일반적이며, 이를 위한 기존 도구들이 다수 존재한다[2,3,4]. 그러나 이 경우에는 두 가지 중요한 문제점이 발생된다. 첫째는 프로세스 및 객체(파일, 디렉터리 등), 각종 사용자 ID 수준의 상세한 로그 정보 획득이 불가능하다. 이는 SOX 법안 준수나 포렌식에서 요구하는 수준의 엄격한 내부 통제, 시스템 침해 대응 및 침해 사건 추적의 수요에 대처하기가 어렵다. 둘째, 각기 기능이 상이한 시스템 그리고 각기 다른 제조사로 인하여 로그의 표준화를 따르지 않아 로그 정보의 정규화와 로그 간의 상호연관 시에 한계를 가질 수밖에 없다. 따라서 상세 수준의 로그 감사가 어렵게 된다. 한편 본 논문에서 활용하는 Secure OS에서 제공하는 수준의 상세한 로그 정보 감사 시스템은 아직 존재하지 않는다.

최근 들어 운영체제에서 제공하는 임의적 접근제어(Discretionary Access Control)에 부가하여 추가적인 강제적 접근 제어(Mandatory Access Control)를 운영체제의 커널에서 참조 모니터(Reference Monitor)로 구현하여, 주체(프로세스)와 객체(파일, 디렉터리 등)를 세밀하게 통제하는 Secure OS가 내부 통제용으로 주목을 받고 있다[5]. Secure OS에서는 프로세스 및 파일 단위에서 상세한 사용 정보 획득이 가능하게 된다. 즉, 서버에서 운용되는 각 응용에 의존하지 않고 모든 응용에서 발생하는 로그를 커널 수준에서 참조 모니터를 통하여 프로세스와 객체 단위로 상세한 서버 정보를 수집하는 Secure OS를 활용하면 상호연관을 통한 감사 및 모니터링이 매우 효과적이다[5]. 본 논문에서는 로그 수집을 위한 3개의 모듈을 자체적으로 개발하였으며, 각 보안 시스템에 포함되어 있다. Secure OS의 모듈은 프로세스와 객체 단위의 시스템 보안 로그를, BackTracker의 모듈은 사용자 수행명령 세션 로그를, SQLTracker의 모듈은 데이터베이스 쿼리를 상세한 수준으로 수집한다.

본 논문에서 다루는 감사 시스템은 기업의 중요 서버 사용자의 작업 이력 중 파일의 편집이나 수정 시 변경된 내용까지도 모니터링하기 위해서 사용자의 작업

이력을 상세하게 수집하는 BackTracker 모듈의 로그, 데이터베이스 작업 이력을 수집하는 SQLTracker 모듈의 로그를 통합 수집하여, 이 로그들을 Secure OS의 로그를 중심으로 상호연관을 시켜 서버 시스템에서의 사용자 행위 감사 및 모니터링, 불법행위에 대한 역추적에 사용된다.

2.1 수집 감사로그의 종류

중요 서버 사용자의 활동 내역, 예외 처리 내역 등을 기록한 감사로그를 수집하기 위해서 수집하는 로그의 종류로 본 논문에서는 시스템 보안로그, 사용자 수행 명령 세션로그, 데이터베이스 쿼리 로그로 한정한다. 시스템 보안로그는 그림 1의 Secure OS 모듈을 통해서 수집되는데, Secure OS 모듈은 커널 레벨에서 주체(프로세스)와 객체(파일)간, 네트워크(IP, Port), 명령어 등의 접근제어를 제공하며, 해킹 등의 불법 접근 행위 시도를 차단하는 기능도 수행한다.

이때 커널에서 실행되는 참조 모니터 기능을 통해서 모든 프로세스의 작업이력 및 접근제어(임의적, 강제적)를 로그로 남기게 된다. 일반적으로 운영체제에서 기본적으로 제공하는 사용자 세션로그는 각 응용 프로그램이 생성한 로그를 수집해서 분석해야 얻을 수 있다. 각 응용이 생성한 로그는 통일성이 없기 때문에 분석에 어려움이 많고, 사용자 세션로그로는 부족한 면이 많다. 따라서 본 시스템에서는 그림 1의 BackTracker 모듈을 사용해서 얻은 사용자 수행 명령 세션로그를 사용하는데, 이 모듈은 사용자 쉘을 후킹하여 터미널 서비스를 받는 모든 사용자의 작업이력을 남기며, 사후에 사용자 행위에 대한 재연(replay)도 가능하다. 데이터베이스 서버의 경우에는 사용자 및 응용 프로그램을 통한 데이터베이스 쿼리 로그를 수집하기 위해서 그림 1의 SQLTracker 모듈를 사용한다.

데이터베이스 쿼리 로그의 경우 DBMS 자체적으로 상세 내용을 기록하기 때문에 시스템 보안로그 수집 모듈이나 사용자 명령 세션로그 수집 모듈로는 수집 정보에 한계가 있기 때문에 별도의 모듈이 필요하다. 그림 1에서와 같이 각 모듈에서 수집된 로그는 데몬을 이용하여 로그 서버 DBMS에 저장된다. 이때 전송되는 로그를 중간에서 공격자가 가로채어 알 수 없도록 암호화하여 전송한다.

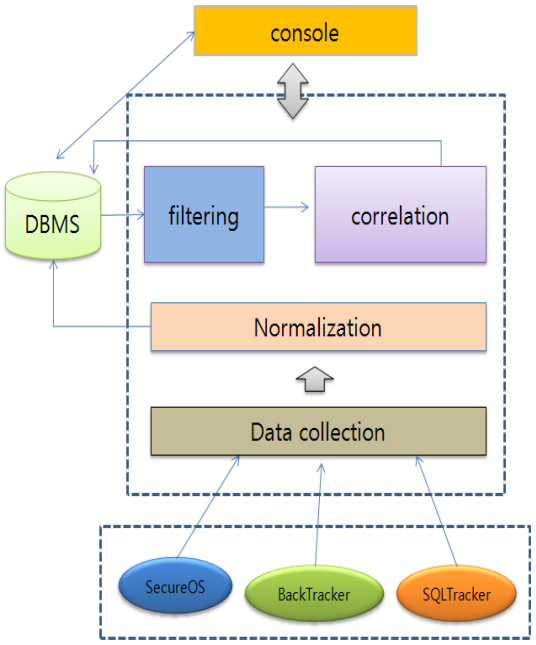

그림 1. 로그 수집 시스템의 구성도
Fig. 1 Block Diagram of log collection system

2.2 감사로그의 정규화

중요 서버 내에서 독립적인 각 로그 수집 모듈에서 발생한 시스템 보안로그, 세션로그, 쿼리 로그를 상호 연관시키기 위해서는 먼저 정규화(Normalization) 과정이 필요하다[6]. 정규화 과정은 각 로그 수집 모듈에서 발생하는 다양한 형식의 개별 로그와 이벤트의 필드를 동일한 포맷으로 변환하는 절차로서, 향후 필터링과 상호연관 단계에서 효율성을 높이기 위해서 필요하다. 표 1은 Secure OS 모듈의 시스템 보안로그로 정규화 될 필드를 나타낸다.

임의적 혹은 강제적 접근제어 규칙에 위배, 또는 불법적인 해킹 행위 등에 대한 다양한 프로세스별, 사용자 ID에 따른 상세한(fine-grained) 로그 필드 정보가 수집된다. 표 2는 BackTracker 모듈의 사용자 수행명령 세션로그로 정규화 될 필드를 나타내며, 로그인 한 사용자가 실행한 명령 등 상세한 행위 로그 필드 정보를 수집함을 보여준다.

표 1. Secure OS 로그 정규화
Table. 1 Log Normalization of Secure OS

필 드	의 미
TIME	로그발생 시간
EVCLS	이벤트 메시지 타입
EVMSG	이벤트 메시지
SXIP	Source IP 주소
SPNO	Source PORT 주소
DXIP	Dest IP 주소
DPNO	Dest PORT 주소
TTY	터미널명
MPID	프로세스ID
MPNM	프로세스명
PPID	부모프로세스ID
PPNM	부모프로세스명
EUID	Effective USER ID
RUID	Real USER ID
OUID	Original USER ID
ONAME	Object 명
EVWHY	접근제어 rule
DMHCK	해킹 감지 여부

표 2. BackTracker 로그 정규화
Table. 2 Log Normalization of BackTracker

필 드	의 미
PPID	부모프로세스 ID
PPNM	부모프로세스명
PID	프로세스 ID
RUID	로그인 사용자 ID
RUNM	로그인 사용자명
SIP	서버 IP
CIP	클라이언트 IP
LTIME	로그 발생 시간
OTIME	로그 종료 시간
LTYPE	로그타입
CUID	현 사용자 ID
CUNM	현 사용자명
CPID	현 프로세스 ID
CPNM	현 프로세스명
LTIME	명령실행 시간
TTY	터미널명
IDATA	입력명령

표 3은 SQLTracker 모듈의 데이터베이스에 대한 쿼리 로그로 정규화 될 필드를 나타내며, 누가 어떤 Query를 실행했는지 여부를 알 수 있는 수준의 상세한 필드 정보가 수집된다. 각 모듈에서 수집된 로그는 수집되는 위치나 시점이 다를 수가 있기 때문에 상호 관계를 연결해 줄 수 있는 상호연관 요소 변수가 존재해야 한다.

표 3. SQLTracker 로그 정규화
Table. 3 Log Normalization of SQLTracker

필 드	의 미
PID	프로세스ID
db_user	DB 사용자
db_addr	DB IP 주소
db_port	DB PORT 번호
clent_user	접속 사용자
clent_addr	접속 IP 주소
client_port	접속 PORT 번호
T_Login	Login 시간
Types	Log type (kernel)
db_type	ORACLE/MSSQL/DB2)
db_name	SID
db_version	DB 버전
protocol	프로토콜
client_os	접속 OS 명
Machine	접속 컴퓨터명
Program	접속 프로그램명
Terminal	접속 터미널명
action	접속허용 여부(Tracker)
status	접속허용 여부 (DB자체)
Error	접속 거부 메시지
Why	접속 거부 원인
T_Logout	Logout 시간
Querys	실행한 query/results
Select_list	테이블 필드 리스트
Ref_list	테이블 리스트
Join_list	Join 리스트
Where_list	Where 절
Group_list	Group 절
Having_list	Having 절
Order_list	Order 절
Hacking	SQL Injection 공격 여부

표 4는 수집되는 로그를 6하 원칙에 따라 분류한 것이며, 분류된 로그 데이터에서 상호연관 관계를 만들 수 있는 공통적인 요소 3가지를 유추할 수 있다. 첫 번째는 언제(when)에 해당하는 시간(TIME)이다. 각 모듈에서 수집된 로그의 시간은 수집 및 측정 위치가 다르기 때문에 하나의 행위에 대하여 시간 차이가 발생할 수 있다.

표 4. 6하 원칙에 따른 분류
Table. 4 Classification according to 5W1H

6하 원칙	모듈명		
	SecureOS	Back Tracker	SQL Tracker
언제 (when)	TIME	LTIME	T_Login
		OTIME	T_Logout
누가 (who)	MPID	CPID	PID
	MPNM	CPNM	
	PPID	PPID	
	PPNM	PPNM	
	EUID	CUID	
	RUID	RUID	
	OUID		
		CUNM	
		RUNM	
			db_user
			client_user
			client_os
			machine
			program
어디서-어디로부터 (where)	SXIP	SIP	db_addr
	SPNO		db_port
	DXIP	CIP	client_addr
	DPNO		client_port
	TTY	TTY	Terninal
무엇을 (what)		IDATA	Querys
	ONAME		Select_list
			Ref_list
			Join_list
			Where_list
			Group_list
			Having_list
			Order_list
			Hacking_list
			db_name
			db_version
			db_type
어떻게 (how)	EYCLS	LTYPE	
	EYMSG		
왜(why)	EYWHY		Why
	DMHCK		Hacking

이 시간의 차이는 상호연관 알고리즘에 따라 상호 연결한다. 따라서 침해 사고 발생 시 시간 흐름에 따라 분석을 정확하게 하기 위해서는 로그 데이터를 정규화 하기 전에 로그 시간이 동기화되어 있어야만 한다. 본 시스템은 로그 시간 기록에 있어 동기화는 NTP(Network Time Protocol)를 사용하여 해결하였다. 즉 로그 서버를 NTP 서버로 설정하여 모듈이 설치된 각 서버는 NTP 서버로부터 시간을 맞추도록 하여 동기화 문제를 해결하였다.

두 번째는 누구(who)에 해당하는 PID이다. 한 서버 내에서는 PID는 유일한 것이기 때문에 각 모듈에서 발생한 로그를 상호 연관 시킬 수 있다. 셋 번째는 어디서(where)에 해당하는 IP와 Port 이다. 이는 실질적인 행위의 주체가 누구인지를 파악할 수 있다. PID는 한 서버 내에서만 식별자로 유효하지만, 여러 서버가 존재하는 네트워크상에는 IP/Port 정보가 행위자를 나타내는 유일한 식별자라 할 수 있다. 각 모듈의 서로 다른 로그의 형식에서 필요한 부분과 공통부분을 추출하여 정규화 과정을 거친 뒤 로그 서버 DB에 저장한다. 이후 단계로 필터링이 필요한데, 이 과정은 특이한 보안 위협을 나타내지 않는 일상적인 로그를 제거하는 것으로서, 보안 로그의 항목을 이용하여 일상 유형의 보안 경보들을 초기에 제거함으로써 로그 분석 대상을 줄이기 위한 작업이다[7,8].

Ⅲ. 프로세스 기반의 감사로그 상호연관

각 모듈로부터 수집되어 DBMS에 저장된 로그는 시점은 동일하나 수집된 위치가 다르기 때문에 서로 다른 정보로 나타난다. 따라서 다른 위치에서 수집된 로그의 상호연관[9,10,11,12,13]은 결국 더 정확하고 효과적인 감사나 추적을 위해 사용된다. 상호연관은 두 가지 방법으로 수행된다. 첫 째는 단일 시스템(Single-system) 내의 각각 다른 모듈에서 수집된 로그를 하나의 통합 레코드(Unification Record)로 구성하는 방법이며, 둘째는 서로

다른 시스템 내에서 수집된 로그를 상호연관을 시켜 하나의 시스템 내에서 발생한 로그만으로는 파악할 수 없는 공격 여부나 감사 데이터를 작성하는데 사용된다. 동일한 시스템에서는 PID 정보가 상호연관에 중요한 변수 값으로 사용되며, 서로 다른 시스템의 경우에는 Port와 Time이 상호연관에 중요한 변수 값으로 사용되며 각 모듈에서 수집된 정규화된 로그는 사전에 정의된 규칙에 따라 분석된다.

3.1 단일 시스템 로그 상호연관 알고리즘

Secure OS 모듈, BackTracker 모듈, SQLTracker 모듈에서 수집된 로그는 각각 시간(Time), PID (process ID), IP 주소 등을 공통 키(key) 값으로 가지고 있다. 즉, 사용자가 시스템에 접속하여 작업을 수행하게 되면 동일 시점에 커널 수준, 사용자 수준, DBMS 내부(DB 조작 시)에서 각기 로그가 수집된다. 이 키 값을 마스터 키(Master Key)로 가지는 하나의 레코드는 한 번의 조회만으로도 모든 정보를 획득할 수 있다. 그림 2는 각 모듈의 정규화된 데이터에서 상호 공통 요소인 IP, PID를 이용해서 6하 원칙에 따라 분류 및 연결하여 단일 계층의 레코드(Single-layered Record) 로그를 생성하는 알고리즘이다. 상호연관 동기화 Rule Data는 상호연관 시 로그 연결의 기준점, 로그 타입, 우선순의 등의 규칙을 정의한 것이다.

여기서 1차 상호연관 로그 데이터는 추후 사용을 위해서 DBMS에 저장되며 또한 상호연관 동기화 Rule data와 정규화 과정을 거친 로그 데이터는 상호연관 기법의 개념을 바탕으로 상호연관 결과 로그를 생성한다. 예를 들어, 어떤 사용자가 파일을 수정한 뒤 해당 파일을 DBMS에 저장하게 되면, Secure OS 로그에서는 어떤 IP에서 접속한 사용자의 어떤 프로세스를 실행하여 어떤 파일들을 접근했는지에 대한 내용을 확인할 수 있으며. BackTracker 로그에서는 어떤 IP에서 접속한 사용자가 어느 프로세스를 이용하여 파일의 내용을 어떻게 수정했는지에 대한 내용 확인이 가능하며, 이를 DBMS에 저장했을 경우 어떤 IP에서 접속한 사용자가 어떤 DML을 이용하여 어떤 테이블에 접근했는지에 대한 이력을 확인할 수 있다. 이때, 각 로그에서 키 값으로 사용되는 TIME, PID, IP를 이용하여 하나의 상호연관 기법을 적용함으로써 하나의 레코드에서 사용자의 이력에 대한 모든 로그를 확인할 수 있다.

하나의 레코드 정보를 얻기 위해서는 일단 각 모듈에서 발생한 로그가 정규화 과정을 거친 정규화된 로그데이터가 존재해야 한다.

```
/* NetworkInfoFile: 네트워크망 정보 표시: 서버
접근가능 IP범위, 접근 순서 정의 */
read(NetworkInfoFile)
for(Normalized_logdatas){
    valid_ip(ipInfo_value);      /* IP범위 검증 */
    if (S.IP == ip || B.IP == ip || Q.IP == ip){
        search_logdata(pid);  /* 로그검색 */
    }
    if (log_branch())      /* 로그 분류: 6하 원칙 */
        save_record()    /* record로 저장 */
}
for(correlation_record){
   checkNetworkProcedure() /* network순서검사 */
   if (TimeRange_Check){
      if (data_correlation_diff())
         record.NO = indexNo;  /* 연결 설정 */
 }
}
```

그림 2. 단일 시스템의 상호연관 알고리즘
Fig. 2 Correlation algorithm of single system

표 5. 단일 시스템에서의 상호연관 결과 로그
Table. 5 Correlated log of single system

6하원칙	single-correlation_record
언제(when)	LTIME
	OTIME
누가(who)	PID
	PNAME
	PPID
	PPNAME
	CUID
	CUNM
	RUID
	RUNM
	OUID
	db_user
	client_user
	client_os
	machine
	program

어디서-어디로부터 (where)	SIP
	SPORT
	DIP
	DPORT
	TTY
무엇을(what)	IDATA
	ONAME
	Querys
	Select_list
	Ref_list
	Join_list
	Where_list
	Group_list
	Having_list
	Order_list
	Hacking_list
	db_name
	db_version
	db_type
어떻게(how)	LTYPE
	EYCLS
	EYMSG
왜(why)	EYWHY
	DMHCK
	Why
	Hacking

3.2 다중 시스템 로그 상호연관 알고리즘

사용자가 직접 DB 서버에 접속하는 경우는 관리자를 제외하고는 극히 드문 일이다. 보통 사용자의 경우 Web을 통하거나 APP(응용) 서버를 통하여 DB 서버와 통신하게 된다. 이 경우 DB 서버에 접속한 Client IP는 APP 서버나 WEB 서버로서 최초 접속 IP와는 다르다. 또한 사용자의 경우에도 최초 WEB이나 APP 서버에 접속한 사용자와 DB 서버에 접속하는 사용자가 다를 수 있기 때문에 한 시스템 내에서 수집된 로그만을 상호연관을 해서는 정확한 감사 자료의 작성이나, 침입의 탐지가 불가능하다. 따라서 네트워크 구성에 맞게 여러 시스템의 로그를 상호연관 하는 것이 로그 분석의 정확성 확보와 추적에 필요하다. 그림 3은 여러 시스템에서 수집된 로그를 상호연관 시키는 흐름도이다. 1차 상호연관된 로그 데이터에서 서로 연결된 레코드를 찾아 상호 연결시키기 위해서 일정시간 범위 내에 source port와 dest port 가 일치하는 로그데이터를 찾아 레코드에 표시를 한다. 후에 이 표시에 따라 레코드 간 연결된 정보를 출력하게 된다.

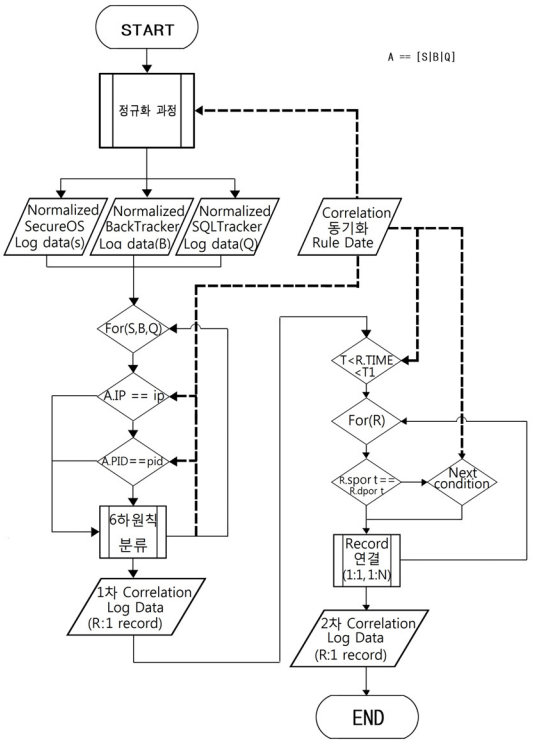

그림 3. 다중 시스템 상호연관 흐름도
Fig. 3 Correlation flow of multi-system

Ⅳ. 시험 시나리오 및 결과

4.1 시험 시스템 구성

시험 시스템 구성은 WEB 서버와 APP 서버, DB 서버로 구성되며 WEB 서버와 APP 서버에는 각각 Secure OS 모듈과 BackTracker 모듈이 설치되고, DB 서버에는 각각 Secure OS 모듈, BackTracker 모듈, SQLTracker 모듈이 설치된다. 각 서버에서 수집된 로그는 데몬을 이용하여 Log Machine의 DBMS (PostgreSQL)에 저장되며 Log Machine의 로그는 임의의 사용자가 위・변조 할 수 없도록 Secure OS를 설치하여 로그 파일의 접근을 강제적으로 통제한다.

4.2 감사 시험 시나리오

로그 상호연관 작업에 대한 분석시간 단축과 정확성을 시험하기 위해 실제 SOX 등에서 요구하는 내부통제 감사 시나리오를 구성하고, 감사인이 요구하는 최종 결과가 나올 때까지의 검색회수를 측정하였다. 이 때 감사인이 원하는 최종 결과의 내용은 여러 가지 방법으로 검색이 가능하기 때문에, 첫 번째 시험 시나리오에서는 서로 다른 3가지 검색 절차를 제시하였으며, 두 번째 시험 시나리오에서는 서로 다른 두 가지 검색 절차로 수행하였다. 상호연관이 안된 로그의 경우에 소요된 검색 횟수와 상호연관이 된 로그에서 소요된 검색 횟수를 비교함으로써 효율성을 결과로 제시하였다. 표 6과 표 7은 두 가지 시나리오에 따른 상호연관로그와 비상호연관 로그의 검색 절차와 검색 횟수를 비교하여 시험 결과로 나타내었다.

표 6. 시나리오 I에 따른 상호연관과 비상호 연관의 검색 횟수 비교
Table. 6 Comparison of number of query between correlation and non-correlation in scenario I

시나리오 1		
지난 한 달간 DB 시스템에 접속한 모든 사용자의 활동에 대한 정기 보고서를 제출하라.		
상호연관로그	검색절차	9월달 발생한 로그 검색 -> 사용자, 프로세스명, itime, otime, SIP 추출
	검색횟수	2회
비상호연관로그	검색절차	9월 DB시스템에서 발생한 secure OS 로그 검색 -> 사용자, 프로세스명, TIME 추출 -> 9월 DB시스템에서 발생한 BackTracker 로그 검색- > OTIME 추출 -> SIP를 기준으로 APP 서버 Secure OS 로그 검색 -> SIP 를 기준으로 WEB 서버 Secure OS 로그 검색 -> SIP 추출
	검색회수	7회
상호연관로그	검색절차	9월달 발생한 로그 검색 -> EVCLS = 'login denied' 인 사용자, SIP 추출
	검색횟수	2회
비상호연관로그	검색절차	9월달 발생한 Secure OS 로그 검색 -> EVCLS = 'login denied' 인 사용자, SIP 추출 -> SIP를 기준으로 APP에서 Secure OS로그 검색 -> SIP를 기준으로 WEB 서버 Secure OS 로그 검색 -> SIP 추출
	검색회수	5회
상호연관로그	검색절차	9월달 발생한 오전 9:00 부터 18:00 까지 발생한 로그 검색 -> 사용자, object, ref_list 추출
	검색횟수	2회
비상호연관로그	검색절차	9월달 발생한 오전 9:00 부터 18:00 까지 발생한 Secure OS로그검색 -> 9월달 발생한 오전 9:00 부터 18:00 까지 발생한 BackTracker 로그 검색 -> 9월달 발생한 오전 9:00 부터 18:00 까지 발생한 SQLTracker 로그 검색 -> 사용자, objects, ref_list 추출
	검색회수	4회

표 7. 시나리오 II에 따른 상호연관과 비상호연관의 검색회수의 비교
Table. 7 Comparison of number of query between correlation and non-correlation in scenario II

시나리오 II		
2008. 6. 30일 작성하여 DB에 저장된 '회계감사' 파일이 수정되었다. 누가, 언제 '회계감사'파일을 수정하였는지 추적하라.		
상호연관로그	검색절차	object = '회계감사' 이거나 IDATA 에 '회계감사' 를 포함하거나 query 문에 '회계감사'를 포함하는 로그 검색 -> IDATA 에 'vi' 나 query 문에 insert/update 포함하는 로그 검색 -> 사용자, SIP 추출
	검색횟수	3회
비상호연관로그	검색절차	object = '회계감사'인 Secure OS 검색 -> IDATA 에 '회계감사'를 포함하는 Back Tracker 로그 검색 -> query 문에 '회계감사'를 포함하는 SQLTracker 로그 검색 -> pname = vi 인 Secure OS 로그 검색 -> IDATA 에 'vi' 를 포함하는 BackTracker 로그 검색 -> query 문에 insert/update 포함하는 SQLTracker 로그 검색 -> SIP 를 기준으로 APP 서버 Secure OS로그 검색 -> SIP를 기준으로 WEB 서버 Secure OS 로그 검색 -> 사용자, SIP추출
	검색회수	8회
상호연관로그	검색절차	object = '회계감사' 이거나 IDATA 에 '회계감사' 를 포함하거나 query 문에 '회계감사'를 포함하는 로그 검색 -> IDATA 에 'vi' 나 query 문에 insert/update 포함하는 로그 검색 -> IDATA에 '회계감사'를 포함하는 로그 검색 -> IDATA추출
	검색횟수	3회
비상호연관로그	검색절차	IDATA 에 '회계감사'를 포함하는 BackTracker 로그 검색 -> IDATA 에 'vi' 포함하는 로그 검색 -> IDATA 추출
	검색회수	3회

그 결과, 시나리오 I에서의 검색 횟수비율은 비상호연관로그 대비 37.5% (즉, $37.5\% = \frac{2+2+2}{7+5+4} \times 100$)로 나타났다. 이는 비상호연관로그에서 평균적으로 5.3회 검색으로 원하는 결과를 찾는 반면, 상호연관로그에서는 평균 2회 만에 감사인이 원하는 결과를 찾는다는 것을 의미한다. 시나리오 II에서는 검색 횟수비율은 비상호연관로그 대비 54.5% (즉, $54.5\% = \frac{3+3}{8+3} \times 100$)로 나타났다. 이는 비상호연관로그에서 평균적으로 5.5회 검색으로 원하는 결과를 찾는 반면, 상호연관로그에서는 평균 3회 만에 감사인이 원하는 결과를 찾는다는 것을 뜻한다.

V. 결 론

본 논문에서 효율적인 내부통제를 하기위한 기법을 제시하였다. 기존의 시스템 로그와 응용 프로그램 로그 정보만으로는 기업의 중요한 업무의 보안 감사 및 모니터링을 위해서 불충분하며, SOX 법안과 같은 기업의 중요한 회계, 재무 업무 처리 등의 엄격한 내부통제를 위한 기업정보시스템에서의 감사 및 모니터링은 구조적·시간적·인적 한계를 가진다. 따라서 시스템에 설치되는 응용 프로그램에 의존하지 않고, 모든 응용 프로그램에서 발생하는 로그를 Secure OS의 커널 수준에서 수집되는 시스템 보안로그, 사용자의 명령어 실행 이력을 수집한 사용자 수행 명령 세션로그, 데이터베이스 쿼리 로그를 통합적으로 이용하여 중요 업무의 보안 감사 및 모니터링, 더 나아가 컴퓨터 포렌식 차원의 불법적 침입에 대한 추적을 지원하기 위한 상호연관 기법과 시험 결과를 기술하였다. 본 제안 기법은 두 시험 시나리오를 통하여 검색횟수가 비상호연관기법 대비 각각 37.5%와 54.5%로 감소함을 보였다.

이를 활용하여 중요 업무 감사 시 요구되는 사건 정보의 정확성과 추적 시간을 향상시킬 수 있었다. 향후 이 방법은 네트워크 정보 등 다양한 로그 데이터를 추가하여 컴퓨터 및 네트워크 포렌식 업무를 수행하기 위한 도구로 활용될 수 있을 것이다.

참고문헌

[1] 지식경제부, '보안 OS기반 SOX 대응 내부통제 시스템 개발', 티에스온넷(주), 2009.

[2] Splunk Inc., White paper: Splunk for Security, 2011.

[3] ArcSight Inc., White paper: Using Advanced Event Correlation to improve Enterprise Security, Compliance and Business Posture, 2011.

[4] Wipro Technologies, White paper: Understanding Event Correlation and the Need for Security Information management, 2011.

[5] 박태규, 임연호, "커널 기반의 보안 리눅스 운영체제 구현", 정보보호학회, 2001.

[6] Definition of Normalization. Web site http://www.dmreview.com/glossary/n.html

[7] 김성락, "상호연관성 분석을 이용한 웹서버 보안관리 시스템", 한국컴퓨터정보학회 논문지, 2004.

[8] 황현욱, 김민수, 노봉남, "감사로그 상관관계를 통한 호스트기반의 침입탐지시스템", 정보보호학회 논문지, 2003.6.

[9] Definition of correlation. Web site http://www.ojp.usdoj.gov/BJA/evaluation/glossary/glossary_c.htm

[10] netIQ John Q, W.2001. White Paper. Security event correlation: Where are we now? Electronic version found at Development," Communications of the ACM, 40, pp. 71-79, May 1997.

[11] Robert Rinnan, "Benefits of Centralized Log file Correlation", Gjovik University College, 2005.

[12] Forte, DV.2004. The "art of log correlation". http://www.infosecsa.co. za /proceedings2004/006.pdf

[13] Cristina Abad et al, "Log correlation for intrusion detection: A proof of concept", 19th Annual Computer Security Applications Conference, Las Vegas, 2003.

저자소개

구하성(Ha-Sung Koo)

1989년 광운대학교
전자통신공학과 학사
1991년 광운대학교
전자통신공학과 석사
1995년 광운대학교 전자통신공학과 박사
1997-현재 한서대학교 컴퓨터정보공학과 교수
※관심분야 : 영상처리, 패턴인식, 디지털 포렌식

박태규(Tae-Kyou Park)

1980년 경북대학교 전자공학사
1989년 충남대학교 전산학석사
1996년 성균관대학교
정보공학박사
1992년 ~ 현재 : 한서대학교 컴퓨터정보공학과 교수
※관심분야 : 정보보호, 실시간 운영체제 등