디지털 증거 확보를 위한 안전한 로깅 방식 연구

신 원*

A Secure Logging for Collection of Digital Evidences

Shin, Weon*

요 약

컴퓨터에서 발생하는 다양한 위협을 조기에 발견하고 대응 방안을 제시할 수 있는 근거가 될 수 있는 것이 바로 로그이다. 또한, 로그를 이용하여 각종 위협에 대한 증거자료를 확보하도록 하고 사이버 수사에 도움을 주어 법적인 효력을 가지도록 할 수도 있다. 본 논문은 컴퓨터 시스템에서 안전한 로깅 개발을 목표로 한다. 이를 위하여 새로운 로깅 방안을 제안하고 다양한 환경을 고려하여 제안 방안을 수정한다. 제안 방안은 컴퓨터 시스템의 디지털 증거를 수집하기 위한 안전한 로깅 방안을 개발하는데 도움이 될 것이다.

ABSTRACT

By analyzing logs in computer systems, we can find early various threats and discover answers to respond against them. Also logs can help to collect digital evidences for cybercrimes and to be legal effects for investigation. In this paper, we intend to develop a secure logging in computer systems. Therefore we propose a new secure logging scheme and improve it to apply various environments. The proposed scheme will helps to develop a better logging of collection for digital evidences in computer systems.

키워드

로깅, 디지털 증거

Key word

Logging, Digital evidence

* 동명대학교 정보보호학과 (shinweon@tu.ac.kr)

접수일자 : 2010. 02. 17
심사완료일자 : 2010. 03. 12

Ⅰ. 서 론

정보통신 기술의 발전으로 개인용 컴퓨터가 보편화 되었으며 매년 관련 산업이 폭발적인 성장을 거듭하고 있다. 최근에는 일반 개인용 컴퓨터 보다는 노트북 컴퓨터를 기반으로 한 넷북, UMPC(Ultra Mobile Personal Computer), MID(Mobile Internet Device)가 등장하고 있으며, 간단한 업무 처리는 물론 게임, DMB(Digital Multimedia Broadcasting)와 무선 인터넷 접속 등과 같은 다양한 작업들을 수행할 수 있다. 이를 통하여 누구나 손쉽게 장소에 구애받지 않고 어디에서나 개인용 컴퓨터를 사용할 수 있게 되었다. 그러나 사용자의 폭발적 증가와 새로운 기능의 경쟁적 도입에 따라 이를 이용한 역기능 또한 함께 증가하고 있는 추세이다. 해킹을 통한 정보 변조 및 유출, 악성코드를 통한 피해, 분산 서비스거부공격(DDoS)을 위한 좀비 역할, 다른 공격을 위한 경유지 등의 사례가 증가하는 것으로 보고되고 있다[1].

컴퓨터 시스템에서 발생하는 다양한 침해를 분석하기 위해서는 무형의 정보를 디지털 증거물로 수집하여 법적 효력을 갖는 증거물로 제시하는 절차인 컴퓨터 포렌식이 필요하다. 즉, 사이버범죄 등에 대하여 컴퓨터 등 정보기기 장치에 내장된 디지털 자료를 근거로 그 장치를 매개체로 발생한 행위의 사실 관계를 규정·증명하는 기술 또는 일련의 절차가 바로 컴퓨터 포렌식이다[2]. 컴퓨터 포렌식은 보안 서비스의 한 분야로 검찰, 경찰 등의 국가 수사 기관에서 범죄 수사에 활용하고 있으며 일반 기업체 및 금융 회사 등의 민간 분야에서도 포렌식 서비스의 필요성이 증가하고 있는 추세이다. 예를 들면, 포렌식 기술은 보험사기 및 인터넷 뱅킹 피해 보상에 대한 법적 증거 자료 수집 및 내부 정보 유출 방지, 회계 감사 등의 내부 보안 강화에 활용 가능하다. 이러한 컴퓨터 포렌식을 수행하기 위해서는 컴퓨터에 저장된 디지털 증거 분석이 필수적인데, 기본이 되는 자료가 바로 로그이다. 즉, 로그를 어떻게 확보하느냐에 따라 디지털 범죄에 대한 수사의 향방이 결정되므로 로그의 수집은 컴퓨터 포렌식에서 필수 작업이다. 반대로 범죄자 입장에서는 자신의 범죄 사실을 숨기기 위해서 범죄 후 로그를 반드시 삭제해야 하는 필연성을 가진다.

본 논문에서는 컴퓨터 시스템에서 안전한 로깅 방안을 제안하고자 한다. 먼저 2장에서는 기존 연구와 적용 가능한 로깅에 대하여 살펴보고, 3장에서는 안전한 로깅 방안을 제안한다. 4장에서는 제안 방안의 안전성 효율성을 분석하고 응용 분야를 살펴보고, 마지막 5장에서 결론을 맺는다.

Ⅱ. 로깅 개요

2.1 로깅

컴퓨터 시스템에서 로깅(Logging)은 프로그램 개발이나 시스템 운영 시 발생하는 문제점을 추적하거나 운영상태를 모니터링하기 위한 텍스트 형식의 데이터를 남기는 것을 말한다[3]. 네트워크 문제로 인한 데이터베이스 연결 실패, 데이터베이스 내부 처리에서 발생한 문제, 서버 시작/종료 관련 메시지, 기타 서버 프로그램 동작 중 발생한 문제 등에서 발생하는 내용에 대한 "시스템 로깅"과 사용자 로그인 히스토리, 특정 페이지 요청 기록, 사용자 권한 관련, 특정 메서드 호출 및 파라미터 정보, 처리 데이터 확인용 메시지 등 관련된 내용에 대한 "어플리케이션 로깅"이 있다. 그림 1과 그림 2는 윈도우 시스템의 이벤트 뷰어와 리눅스 시스템 접속 로그로 각각 이벤트와 접속에 대한 로깅 예이다.

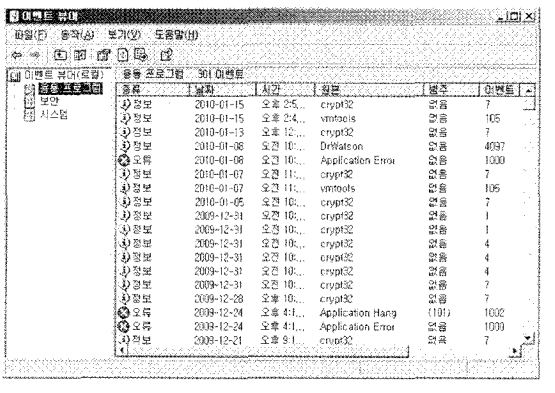

그림 1. 윈도우의 이벤트 뷰어 예제
Fig. 1 Windows event viewer

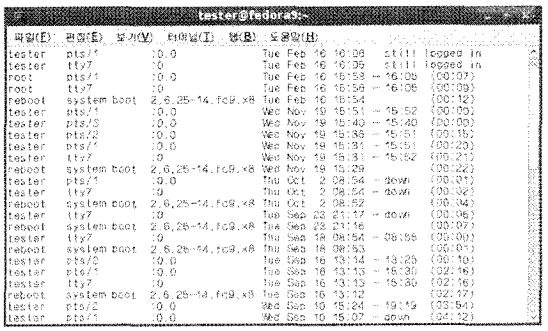

그림 2. 리눅스의 접속 로그 예제
Fig. 2 Linux wtmp

로깅은 시스템동작 중 발생하는 여러 가지 이벤트를 저장하는 기록 과정이므로 이를 통해 발생하는 로그는 관리적 관점에서는 시스템의 동작을 모니터링하고 문제점을 파악하여 해결하는 실마리를 제공할 뿐만 아니라 보안 관점에서는 내외부 공격 여부를 판단하게 해주고 공격자의 동작을 증거 자료로써 기록하여 법적인 증거 효력의 의미를 가지기도 한다.

2.2 기존연구

일반적으로 로깅에 대한 연구는 기존 로깅 과정을 수정하여 암호학적으로 안전한 부가 정보를 추가하여 로그 데이터의 수정을 검출 또는 방지하는 쪽으로 초점을 맞추고 있다. 대표적인 연구로는 syslog-sign[4], Schneier and Kelsey 방안[5], Ma and Tsudik 방안[6] 등이 있다. 본 논문에서 사용되는 표기법과 가정은 다음과 같다.

L_i : i번째 로그
S_i : 각 방안을 적용한 i번째 로그
$H(m)$: 메시지 m에 대한 해쉬값
$MAC_K(m)$: K를 이용한 m에 대한 메시지 인증값
$\{m\}_K$: K를 이용하여 메시지 m을 암호화
R_A : A가 생성한 임의의 난수값

<가정>

1. 안전한 해쉬 함수를 사용하므로 공격자가 이를 공격하는 것은 계산적으로 불가능하다.
2. 안전한 MAC(Message Authentication Code)을 사용하므로 공격자가 이를 공격하는 것은 계산적으로 불가능하다. 단, 키를 아는 경우는 쉽게 공격할 수 있다.
3. 암호화에 사용하는 알고리즘은 안전하므로 공격자가 이를 공격하는 것은 계산적으로 불가능하다. 단, 키를 아는 경우는 쉽게 복호화할 수 있다.
4. 임의의 난수값을 정확히 추측하는 것은 계산적으로 불가능하다.

syslog-sign[4]은 로그 데이터 각각을 해쉬시킨 다음 이들을 순서대로 모아서 만든 해쉬값을 이용하여 서명을 수행하는 방식으로 이루어진다. 그림 3은 syslog-sign의 동작방식을 보여준다.

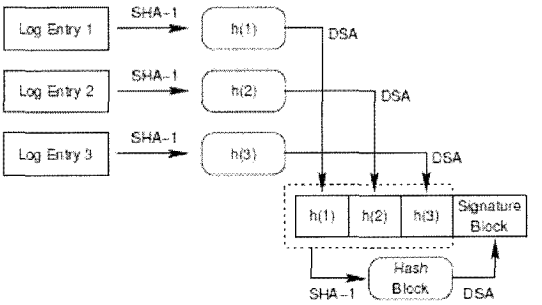

그림 3. syslog-sign의 동작방식
Fig. 3 syslog-sign scheme

Schneier and Kelsey 방안[5]은 로그 데이터를 특정 키를 계속 해쉬하고 이를 이용하여 암호화하고, 이전 로그 데이터에 의한 해쉬값을 현재 로그에도 반영하고 있다. 즉, 해쉬 체인을 이용하여 삭제 및 삽입과 같은 수정 공격을 검출할 수 있다. 그림 4는 Schneier and Kelsey 방안에 의한 로깅을 보여 준다.

$$L_i := \boxed{W_i \mid \{E_i\}_{K_i} \mid Y_i \mid Z_i}$$

그림 4. Schneier and Kelsey 방안
Fig. 4 Schneier and Kelsey scheme

Ma and Tsudik 방안[6]은 로그에 메시지 인증을 위한 MAC 체인으로 삭제 및 삽입과 같은 수정 공격을 검

출할 수 있다. 또한, 검증자에 의한 부정 방지까지 포함한다.

앞에서 설명한 기존 연구를 분석하면 다음과 같다. syslog-sign은 전자서명을 통하여 로그에 대한 인증성을 보장하지만 특정 로그가 불법 수정된 경우, 로그가 수정되었다는 사실은 확인할 수 있으나 어떤 로그가 수정되었는지 찾기가 힘든 문제점을 가지고 있다. Schneier and Kelsey 방안은 로그 데이터에 대한 암호화를 제공할 뿐만 아니라 해쉬 체인에 비밀값 $A_i = H(A_{i-1})$을 이용하여 안전성을 보장하는데, 공격자에게 A_x가 노출되면 A_{x+1}을 계산할 수 있으므로 안전성에 문제가 있다. Ma and Tsudik 방안은 단순하여 효율성이 높고 MAC 체인을 이용하여 안전성을 보장해준다. 그러나, MAC의 비밀키를 $A_i = H(A_{i-1})$로 계산하므로 공격자에게 A_x가 노출되면 A_{x+1}을 계산할 수 있으므로 안전성에 문제가 있다.

Ⅲ. 새로운 로깅 방안의 제안

본 장에서는 로깅에 대한 기존 연구의 문제점을 해결하고 보다 효율적이며 안전한 방안을 제안한다.

3.1 안전한 로깅의 요구사항

안전한 로깅을 위해서는 로그 데이터 수정 방지, 작은 오버헤드, 전방향 무결성을 만족하여야 한다. 이와 같은 요구사항의 구체적인 내용은 다음과 같다.

○ 로그 데이터 수정 방지

로그 데이터는 시스템에 대한 기록을 유지함으로써 특정 사건 발생시 당시에 어떠한 작업을 수행했는지 알 수 있도록 근거를 제공하는 중요한 자료가 된다. 따라서, 시스템 공격자의 입장에서는 자신의 범행에 대한 증거가 될 수 있는 로그 데이터 수정에 대한 동기를 가진다. 반대로 시스템 관리자 입장에서는 이를 방지하기 위하여 로그 데이터가 수정/삽입/삭제 공격에 대하여 안전하도록 운영하여야 한다.

○ 작은 오버헤드

로그 데이터의 무결성과 안전성을 보장하기 위해서는 로그 데이터 이외에 부가 정보가 필수적이다. 이 부가정보들은 계산량이 충분히 작아서 시스템에 부담을 주지않고 효율적으로 생성할 수 있어야 한다. 또한, 크기도 작아서 시스템의 저장 공간을 작게 차지하여 추가적인 공간없이도 쉽게 저장할 수 있어야 한다.

○ 전방향 무결성(Forward Integrity)

이전의 모든 키가 노출되어 공격자가 그것을 알고 있다 하여도 과거의 데이터를 위조할 수 없는 성질[7]로, 이미 한번 작성된 데이터에 대해서는 설사 본인이라 하더라도 그 데이터를 수정할 수 없도록 하는 특성이 전방향 무결성이다. 즉, 각 로그 데이터 간의 특정한 상관관계를 설정하여 수정/삽입/삭제 되었다 하더라도 이를 검출할 수 있어야 한다.

3.2 안전한 로깅 방안의 제안

앞에서 설명한 안전한 로깅의 요구사항을 만족하는 새로운 로깅 방안을 아래와 같이 제안한다.

$$
\begin{aligned}
&S = \{S_1, S_2, \cdots, S_n\} \\
&S_i = (L_i, M_i),\ 1 \le i \le n \\
&M_i = MAC_{K_i}(L_i \| M_{i-1}),\ M_0 = MAC_{K_0}(SYSTEM) \\
&K_i = H(R_{USER} \| M_{i-1}),\ K_0 = H(R_{SYSTEM})
\end{aligned}
$$

제안 로깅 방안은 $S = \{S_1, S_2, \cdots, S_n\}$로 이루어 지고, 각 로그는 $S_i = (L_i, M_i)$로 구성된다. 여기서, L_i는 로그 데이터이고, M_i는 로그 데이터의 안전성을 보장해주기 위한 부가 정보로 다음과 같이 생성된다. 여기서, M_0는 시스템 이름을 MAC한 값을 사용한다.

$$
\begin{aligned}
M_0 &= MAC_{K_0}(SYSTEM) \\
M_i &= MAC_{K_i}(L_i \| M_{i-1}) \\
&= MAC_{K_i}(L_i \| MAC_{K_{i-1}}(L_{i-1} \| M_{i-2})) \\
&= MAC_{K_i}(L_i \| L_{i-1} \| \cdots \| MAC_{K_1}(L_1 \| M_0) \cdots)
\end{aligned}
$$

MAC에서 사용하는 비밀값은 다음과 같이 구성된다. 여기서, K_0는 시스템이 생성한 임의의 난수값으로 안전하게 보관한다.

$K_0 = H(R_{SYSTEM})$

$K_i = H(R_{USER} \| M_{i-1})$

K_i는 사용자가 자신의 난수값을 이용하여 이전 M_{i-1}과 함께 자체적으로 생성한다.

특히, R_{SYSTEM}과 R_{USER}은 각각 시스템과 사용자가 생성한 임의의 난수값으로 안전하게 보관한다. 결국, S_i는 로그 데이터 L_i와 이를 해쉬한 후 하나의 키를 사용하여 메시지 인증한 M_i로 구성된다. 여기서, 키 K_x가 노출된다 하더라도 R_{USER}를 알 수 없으면 K_{x+1} 계산이 불가능하다.

VI. 제안 로깅 방안의 분석

4.1 제안 방안의 분석

제안 로깅 방안과 기존의 방안에 대한 성능 분석 자료는 표 1과 같다. Ma and Tsudik 방안과 제안 방안은 거의 비슷한 성능을 가지고 있는데, 기존 방안과 비교하여 제안 방안이 근소한 차이로 가장 효율적으로 동작함을 확인할 수 있다. 그림 5는 성능에 대한 평균 데이터를 보여준다.

제안 방안에서 해쉬 함수를 이용한 MAC를 사용한다면, 각 로그 당 160비트(20바이트)가 추가되어야 하는데, 로그 크기가 512바이트인 경우 4%, 1,024바이트인 경우 2%의 오버헤드를 가지므로 일반적인 컴퓨터 시스템에서 충분히 수용가능한 크기이다. 제안 방안의 $S_i = (L_i, M_i)$에서 M_i은 $M_i = MAC_{K_i}(L_i \| M_{i-1})$를 통하여 계산할 수 있으므로, MAC가 안전하고 K_0가 비밀로 유지된다면 공격자는 위조 로그 $S_x = (L_x, M_x)$를 만들 수 없다. 또한 공격자가 로그 S_i를 임의로 삭제한 경우, 로그 $S_{i-1} = (L_{i-1}, M_{i-1})$와 로그 $S_{i+1} = (L_{i+1}, M_{i+1})$와의 관계에서 $M_{i+1} \neq MAC_{K_{i+1}}(L_{i+1} \| M_{i-1})$가 성립되지 않으므로 공격자가 삭제한 로그를 찾아낼 수 있게 된다. 또한, 공격자가 $S_i = (L_i, M_i)$를 알고 있고 심지어 K_i를 알아냈다고 하더라도, MAC가 안전하고 K_0가 비밀로 유지된다면 공격자는 로그 $S_{i+1} = (L_{i+1}, M_{i+1})$를 만들 수 없으므로 전방향 무결성이 보장된다.

따라서, 제안 방안은 작은 오버헤드를 가지고 있으며 로그 수정 방지는 물론 전방향 무결성의 성질도 함께 가진다.

4.2 제안 방안의 수정

제안 로깅 방안을 기본으로 안전성을 더 강화하거나 성능을 향상시키기 위하여 다음과 같이 수정하여 적용할 수도 있다.

표 1. 각 로깅 방안의 성능 비교 (세부)
Table 1. Comparison of schemes (Detailed)

		생성시간 (ms)											
방안		Schneier and Kelsey 방안				Ma and Tsudik 방안				제안방안			
회수		1회	2회	3회	평균	1회	2회	3회	평균	1회	2회	3회	평균
생성 로그	100개	181	164	157	**167**	124	109	83	**105**	81	98	76	**85**
	200개	357	360	333	**350**	200	214	191	**202**	168	195	174	**179**
	400개	740	744	677	**720**	355	428	373	**385**	306	307	306	**306**
	800개	1,341	1,467	1,311	**1,373**	820	756	785	**787**	678	626	609	**638**
	1,600개	2,899	2,724	2,692	**2,772**	1,391	1,494	1,363	**1,416**	1,323	1,291	1,265	**1,293**
	3,200개	5,460	5,003	5,476	**5,313**	2,678	2,484	2,959	**2,707**	2,492	2,409	2,428	**2,443**
	6,400개	9,922	10,843	10,922	**10,562**	5,312	5,900	5,567	**5,593**	4,958	5,438	5,052	**5,149**
	12,800개	21,600	21,079	21,542	**21,407**	11,145	10,672	10,630	**10,816**	10,956	10,546	10,662	**10,721**
	25,600개	39,824	42,516	43,820	**42,053**	20,354	21,747	23,417	**21,839**	20,355	21,267	21,845	**21,005**
	51,200개	79,404	76,503	78,917	**78,275**	42,615	41,668	42,083	**42,122**	41,480	39,281	42,122	**40,961**

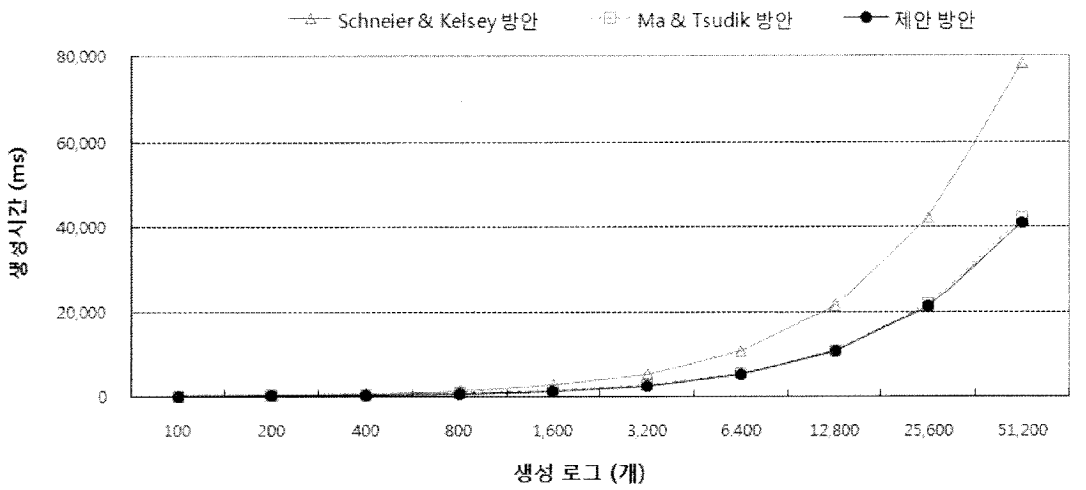

그림 5. 로깅 방안의 성능 비교 (평균)
Fig. 5 Comparison of schemes (Average)

첫째, 제안 방안은 로그 데이터 자체를 암호화하지 않는데, 로그에 대한 비밀성이 요구된다면 다음과 같이 암호화를 통하여 이를 보장해줄 수 있다.

$$S = \{S_1, S_2, \cdots, S_n\}$$
$$S_i = (\{L_i\}_{K_0}, M_i),\ 1 \le i \le n$$

여기서, K_0는 시스템만이 알고 있으므로 로그 데이터를 복호화하여 열람할 수 있는 대상은 시스템뿐이다.

둘째, 제안 방안을 수정하여 중앙집중적인 로그 생성과 관리가 가능하게 할 수 있다. 기본 방안에서는 MAC에서 사용하는 키를 최초에 시스템에서 생성하고 이를 이용하여 사용자가 생성하도록 다음과 같이 구성되어 있다.

$$K_0 = H(R_{SYSTEM})$$
$$K_i = H(R_{USER} \| M_{i-1})$$

그러나, 아래와 같이 수정하면 R_{SYSTEM}를 알고 있는 시스템만이 MAC을 생성할 수 있게 된다.

$$K_0 = R_{SYSTEM}$$
$$K_i = H(R_{SYSTEM} \| M_{i-1})$$

이를 통하여 로그에 대한 부가정보 M_i를 사용자 자율이 아니라 시스템만이 생성하도록 하여, 중앙집중적인 로그 생성과 관리가 가능하게 된다.

셋째, 제안 방안을 웹서버와 같은 시스템에 적용하는 경우 성능을 향상시키기 위하여 수정하여 사용할 수 있다. 대용량의 로그가 발생하는 시스템에서 각 로그 데이터마다 해쉬와 MAC을 적용한다는 것은 상당한 오버헤드를 야기할 수 있으므로 이를 위하여 다음과 같이 수정하여 사용한다.

$$S = \{S_1, S_2, \cdots, S_n\}$$
$$S_i = (\{L_{i,1}, L_{i,2}, \cdots, L_{i,m}\}, M_i),\ 1 \le i \le n$$
$$M_i = MAC_{K_i}(\{L_{i,1}, L_{i,2}, \cdots, L_{i,m}\} \| M_{i-1})$$
$$K_i = H(R_{USER} \| M_{i-1})$$

즉, 로그 데이터를 1개 단위가 아니라 m개 단위로 묶은 후 MAC을 적용함으로써 성능을 향상시키는 것이다. 여기서, m은 단위 개수 또는 단위 시간에 따라 시스템에 맞도록 적절하게 선택한다.

V. 결론

누구나 어디에서나 컴퓨터와 인터넷을 통하여 원하는 정보를 얻을 뿐 아니라 각종 거래도 수행하게 되었고, 매일 수천만 대의 컴퓨터가 인터넷을 통하여 대량의 정보를 전송하고 있다. 그러나 컴퓨터와 인터넷에서 발생하는 해킹, 악성코드, 컴퓨터 범죄 등은 새로운 위협으로 매년 그 피해도 폭발적으로 증가하고 있는 상태이다. 컴퓨터에서 발생하는 다양한 위협을 조기에 발견하고 대응 방안을 제시할 수 있는 근거가 될 수 있는 것이 바로 로그이다. 또한, 로그를 이용하여 각종 위협에 대한 증거자료를 확보하도록 하고 사이버 수사에 도움을 주어 법적인 효력을 가지도록 할 수도 있다. 이를 위하여 로깅 과정에서 로그 자체를 불법적으로 수정할 수 없도록 하여야 하고, 전방향 무결성을 제공하여야 하며 오버헤드가 작아서 컴퓨터 시스템에 무리없이 동작할 수 있어야 한다.

본 논문에서는 안전한 로깅의 요구사항을 만족하는 새로운 로깅 방안을 제안하였다. 제안 방안은 기존 방안에 비하여 효율적으로 동작할 뿐만 아니라 안전성에서도 뛰어난 장점을 가지고 있다. 이로 인하여 제안 방안은 컴퓨터를 이용한 각종 위협에 대한 디지털 증거자료 확보에 직접적인 도움을 줄 수 있을 것으로 판단한다.

참고문헌

[1] "인터넷 침해사고 동향 및 분석 월보 200912", 한국인터넷진흥원 인터넷침해대응센터, 2009.

[2] 김소이, "전자금융사고 발생유형 및 대응현황", 금융결제원 금융결제연구소 지급결제와 정보기술 제38호, 2009

[3] Computer data logging, http://en.wikipedia.org/wiki/Computer_data_logging

[4] J. Kelsey and J. Callas, "Signed syslog messages," IETF Internet Draft, 2005, http://www.ietf.org/internet-drafts/draft-ietf-syslog-sign-16.txt

[5] B. Schneier and J. Kelsey, "Security audit logs to support computer forensics," ACM TISSEC, vol. 2, no. 2, pp. 159 - 176, 1999

[6] D. Ma and G. Tsudik, "A new approach to secure logging," ACM Trans. on Storage, vol. 5, no. 1, pp. 1 - 21, 2009.

[7] M. Bellare and B. Yee, "Forward integrity for secure audit logs," University of California, San Diego, Dept. of Computer Science & Engineering, Tech. Rep., 1997

저자소개

신원(Shin, Weon)

2005.3~현재 동명대학교 정보보호학과 조교수

2002.3~2005.1 (주)안철수연구소 선임연구원

※관심분야 : 악성코드 확산, 디지털 포렌식, 소프트웨어 보안, 암호 프로토콜 응용