Journal of Knowledge Information Technology and Systems(JKITS), Vol. 9, No. 4, pp. 427~434, August 2014

Journal of Knowledge Information Technology and Systems

ISSN 1975-7700

http://www.kkits.or.kr

Implementation of System Usage Recorder for Personal Computer Security

Oh-Sung Kwon*

Dept. of Computer Education, Gongju National University Of Education

ABSTRACT

In general, data loss accidents is mostly caused by the insiders of organizations than it's outsider. In this paper, we propose a new method which collects the 3 information types of PC(Personal Computer) system to prevent this kind of accidents. The first important type is the screen video made by image capture timer. The other information types are the system's process list and the keyboard contents. we uses messages hooking procedure to catch the system information. To implement a hooking module, we made a hook chain structure, hook procedures, and a message monitoring function. In experiment results, we can successfully collect the screen capture video, the process list and the keyboard contents. In specially the screen capture function was sensitive to the video quality setting. The capacity of high quality videos was about 4.6 times larger than that of low quality.

KEYWORDS: PC Securities, Process Lists, Keyboard Hooking, Hook Procedures, Message Monitoring

ARTICLE INFO: Received 24 June 2014, Revised 14 August 2014, Accepted 14 August 2014.

1. 서 론

최근 들어 금융권 등 대규모 개인 정보 수집 기관의 관리 소홀로 심각한 정보 유출 사고가 자주 발생하고 있다. 이러한 사고는 외부 해킹 등에 의한 것일 수도 있지만, 외부 용역이나 내부 직원에 의한 의도적인 유출 사례도 적지 않게 나타나는 것이 사실이다.

일반적으로 회사나 기관의 정보 인프라 관리자

*Corresponding author is with the Department of Computer Education, Gongju National University of Education, Gongju ChungNam, 314-711, KOREA.
E-mail address: oskwon@gjue.ac.kr

는 관리실 노트북이나 단말 컴퓨터를 통하여 서버에 접속하고 필요한 작업을 진행하게 되는 데, 이 경우 해당 접속 컴퓨터가 주요 유출 경로로 이용된다. 이러한 사고를 예방하기 위해선 관련 내부 인력의 강도 높은 관리 감독도 필요하지만, 유출 가능 컴퓨터의 사용 내역을 감시하는 도구의 설치도 필요하다고 하겠다.

이에 본 논문에서는 이러한 정보 유출 사고 예방을 위한 방안으로 사용자의 컴퓨터의 사용 내역을 실시간 검출하고 기록하는 방안을 제안한다. 제안하는 시스템 사용 기록기는 해당 컴퓨터를 사용 중에는 상시 실시간으로 동작하도록 설계하였다.

본 논문에서 기록하고자 하는 컴퓨터의 사용 내역은 컴퓨터 화면 녹화, 키보드 입력, 프로세스 현황 등 3 가지 유형의 시스템 정보이다. 본 제안 방안은 <그림 1> 의 유출 형태 중에서 내부자에 의한 경우를 위한 방안으로서 사건 발생 후 내용을 파악하기 위한 용도로 사용될 수 있다.

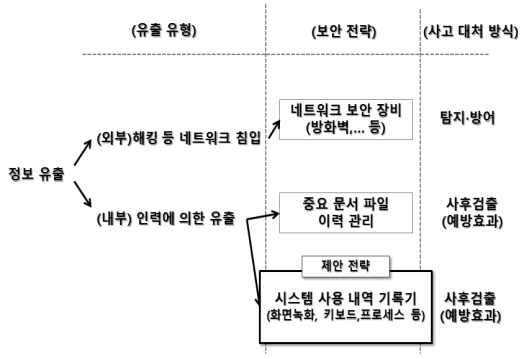


그림 1. 정보 유출 유형에 따른 대처 방식
Figure 1. Coping methods according to information leakage

제안하는 방법을 설명하기 위하여 우선 배경 지식이 되는 사용자 화면의 실시간 캡쳐 기법, 시스템 정보 수집을 위한 시스템 후킹 등의 관련 내용을 살펴보고 수집 시스템의 구성과 구현 내용을 제시한다. 마지막으로 구현 및 실험 결과를 제시하고 결론을 맺는다.

2. 관련 연구

피씨 보안을 목적으로 시스템 사용 내역을 수집하고 기록하는 것을 주제로 한 연구는 현재까지 보고되지 않고 있으나, 이에 필요한 요소 기술에 관련한 연구 논문과 적용 사례는 다수 찾을 수 있었다. 본 논문에서 제안하는 시스템 사용 기록기의 기반 요소 기술은 <표 1> 과 같이 크게 3 가지로 나눌 수 있으며, 그 기반 기술은 동영상 생성, 메시지 후킹, 시스템 정보 검출 등 이다.

표 1. 제안하는 방식의 기반 기술
Table 1. Basic skills of our proposed method

	수집 정보 분류	기반 기술
1	화면 녹화	동영상 생성
2	키보드 입력	메시지 후킹
3	프로세스 수집	시스템 정보 검출

화면 녹화 기술은 일반적으로 동영상 녹화, 교육 자료 및 튜토리얼 작성 등에 주로 이용되며, 대표적인 소프트웨어로는 TechSmith 사의 캠타시아(Camtasia) 등을 들 수 있다[6]. Hwang 의 연구는 캠타시아를 이용하여 의과대학 교육용 튜토리얼 동영상 제작하고 그 학습 효과를 측정하는 내용이었다[3].

메시지 후킹 모듈을 이용한 다양한 응용 연구가 보고되고 있다. Han 의 경우는 후킹 모듈을 이용한 간이형 개인 방화벽 구성 및 운용에 관한 연구로 USB 등 휴대용 저장장치에 적용하였다[2]. Jung 은 후킹 기법을 이용하여 사용자의 다른 응용에서의 행동을 수집하고 인식하여 적응형 응용 프로그램 진행이 가능하도록 하는 연구를 진행하였다[4]. Kim의 연구는 커널 모드에서 동작하는 API 악성코드를 탐지하고 제거하기 위하여 커널 모드에서 동

작 가능한 도구를 개발하였다[5]. Berdajs은 악성 코드 탐지 및 컴퓨터 안전 감시 도구를 위하여 디버깅 DLL 인젝션과 단일 명령 후킹 기법을 적용하는 연구를 수행하였다[1].

본 논문의 주제인 시스템 사용 기록 장치의 실시간 프로세스 현황을 수집하는 기능은 단순히 시스템 이벤트 검출 함수 적용만으로도 구현이 가능하였으며, 이에 관련한 윈도우즈 API 응용들을 다수 찾을 수 있었다[11].

3. 시스템 사용 기록기의 구성

3.1 전체 시스템의 개요

본 논문의 시스템 사용 기록기는 녹화 화면, 키보드 입력, 프로세스 리스트 등 총 3 가지의 정보를 수집하는 기능으로 구성되며, 실시간으로 필요한 정보를 검출하고 확인할 수 있도록 하였다.

기록 장치는 녹화 단위 시간을 정하고 해당 주기별로 위에서 제시한 3 가지 분류에 대하여 각 1 개씩 파일을 저장하도록 구현하였다.

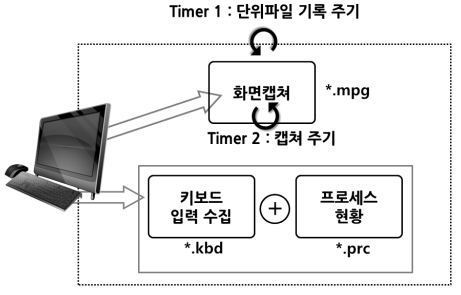


그림 2. 전체 시스템 구성도
Figure 2. Entire system diagram

<그림 2>는 전체 시스템 구성도를 나타내며, 그림에서 Timer 1 은 3 가지 유형 정보를 기록하는 개별 파일의 저장 주기를 지정하며, Timer 2 는 화면 녹화 시에 필요한 화면 캡쳐 주기를 나타낸다. 본 논문에서 Timer 1 은 1~10분 정도의 시간을 사용하였고, Timer 2 는 1~60 Frames/min 정도를 지정하였다. 예를 들어 화면 녹화의 경우 분당 10 프레임 정도 캡쳐하여 비디오를 생성하는 경우 1시간 녹화를 20 초 분량의 비디오(30프레임/초)로 재생하여 확인할 수 있다.

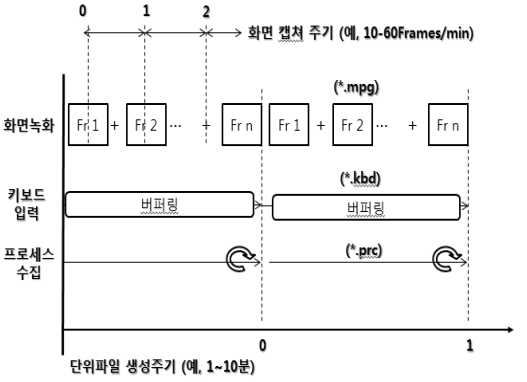


그림 3. 시스템 정보 수집 주기
Figure 3. System information gathering cycle

위 단위 파일들의 생성 주기와 화면 캡쳐 주기는 <그림 3> 과 같은 방식의 상호 관계속에서 운영되도록 구성하였다. 지정된 생성 주기에 맞추어 위에서 제시한 3 가지 내용의 정보들이 수집되어 구별된 별도 파일로 저장되도록 하였다.

위에서 제시한 (1)번 파일은 단위 파일 주기에 1 개씩 mpg 파일을 생성하는 데 동영상을 구성하는 프레임 수 지정을 위한 추가적인 화면 캡쳐 주기의 지정이 필요하다. (2) 번 파일은 지정 주기까지의 키보드 입력 내용을 담게 되고 (3)번 파일은 저장 당시의 프로세스 상태 현황을 조사하여 저장하도록 하였다. 결국, 위에서 제시한 정보 파일을 수집하는 체계는 <그림 3> 과 같이 2개의 타이머 시간 주기에 따른 소프트웨어 구현이 필요하다.

키보드 정보 수집은 후크 체인 및 메시지 유형

별 후크 프로시져에 의해서 진행된다. 사용자가 입력한 키보드 입력의 구체적인 용도는 동반 저장되는 화면 녹화 내용을 보면서 확인할 수 있다. 프로세스 현황은 해당 시간에 메인 메모리에 로드되어 운영체제의 프로세스 관리 대상이 되는 것들을 수집하며 진행되도록 하였다. 프로세스 현황 정보를 통하여 자료 유출이나 해킹을 위한 프로그램의 실행 여부 등을 확인할 수 있다.

3.2 화면 녹화 기능

<그림 3>에서 제시한 3 가지 시스템 정보 중에서 화면 녹화를 위한 코덱(Codec)으로는 GNU 오픈 소스(open source) FFmpeg 을 적용하였다[10]. 동영상 생성과 더불어 생성된 결과물의 변환을 위해선 동일한 오픈 소스인 AviSynth 동영상 변환 도구를 사용하였다[7].

화면 녹화는 <그림 3> 의 화면 캡쳐 주기에 따라 진행되며 주기별로 캡쳐된 화면 이미지는 동영상 생성을 위한 이미지 프레임으로 사용된다. 결국 시간이 진행되며 프레임 수가 증가되고 그에 따라 녹화 동영상이 생성되는 방식이다. 녹화 동영상은 계속 그 크기가 성장하는 경우 파일 유지에 부담을 주기 때문에 <그림 3>의 단위 파일 생성 주기에 맞추어 새 파일로 변경하여 그 크기를 초기화하도록 하였다.

실시간으로 캡쳐되는 화면 이미지는 3가지 시스템 정보 유형 중에서 그 크기가 상대적으로 크기 때문에 시스템에 부담을 줄 수 있고 이를 해소하기 위한 캡쳐 주기, 화질, 프레임 수의 적합한 지정이 선행되어야 한다. 본 논문에서는 이러한 화면 녹화 관련 변수들의 상관 관계를 4장에서 실험하였다.

3.3 키보드 입력 기능

키보드 입력 기능은 후킹과 DLL 인젝션 기능을 이용하여 진행한다. 일반적으로 후킹은 운영 체제나 응용 소프트웨어 등의 각종 컴퓨터 프로그램에서 소프트웨어 구성 요소 간에 발생하는 함수 호출, 메시지, 이벤트 등을 중간에서 바꾸거나 가로채는 명령, 방법, 기술이나 행위를 말한다. 윈도우 메시지 후킹은 다른 응용에서 발생한 메시지나 이벤트를 중간에서 처리하는 과정이다[11]. 이러한 후킹 기법은 외부 해킹 시 메모리 정보, 키보드 입력 등을 취득하기 위한 용도로 사용되기도 하며, 특정 API를 후킹하게 되면 해당 API의 리턴 값 조작도 가능하다[1,2,5,11,12].

본 논문에서는 키보드 입력에 해당하는 <그림 4>와 같은 후크(hook) 프로시져를 작성하여 해당 이벤트를 처리하도록 구현하였다.

```
LRESULT CALLBACK HookProc(int nCode,WPARAM
wParam,LPARAM lParam) {
  // process event
  ...
  return
CallNextHookEx(NULL,nCode,wParam,lParam);
//이벤트를 다음 후크로 전달 }
```

그림 4. 키보드 입력 처리를 위한 후크
Figure 4. Hook for keyboard input processing

후킹을 이용하여 실행 중인 임의의 응용 프로그램에 접근하고자 하는 경우, 해당 응용 소스에 직접 조작이 어렵기 때문에 DLL 인젝션(injection) 기법을 이용하여 API 후킹하는 방법을 사용한다. DLL 인젝션은 실행 중인 프로세스에 특정 DLL 파일을 포함시키는 작업을 의미한다. 곧, 다른 프로세스에게 윈도우즈 LoadLibrary() API를 스스로 호출하도록 명령하여 특정 DLL이 로딩되도록 하는 것이다[8,9].

4. 구현 및 실험 결과

앞서 설명한 시스템 사용 기록기는 윈도우즈 7, 8 상에서 동작하도록 구현하였다. 화면 녹화 모듈은 FFmpeg Mpeg-4 를 사용하였고, 키보드 입력 자료 수집을 위한 후킹 윈도우즈 API 로는 SetWindowsHookEx() 함수를 사용하였다[10].

그림 5. 키보드 및 프로세스 파일 리스트
Figure 5. Keyboard and process file list

화면 녹화 파일, 키보드 입력 내용, 프로세스 현황은 실시간 집계되어 지정한 특정 폴더에 저장되도록 하였다. <그림 5>는 키보드 및 프로세스 집계 파일의 폴더 저장 형태를 보여준다.

<그림 3>의 단위 파일 생성 주기에 따라 프로세스 현황 파일의 예는 <그림 6>과 같다. 구현 프로그램은 분당 프레임 수와 화질을 지정하여 사용할 수 있다. 프레임 수를 늘리면 보다 세밀한 화면 캡쳐 녹화가 가능한 반면 파일 크기가 커지며, 화질은 고화질일수록 용량이 증가하기 때문에 시스템에 부담을 줄 수 있어 성능에 맞는 적합한 설정을 찾는 것이 필요하다. 컴퓨터 사용자가 시스템 사용 기록기가 동작하고 있다는 것을 의식하지 못할 정도로 부담이 되지 않는 범위에서 분당 프레임 수 및 화질을 결정해야 한다.

그림 6. 프로세스 파일 내용
Figure 6. Process file contents

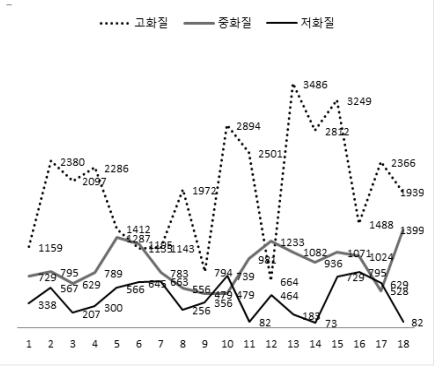


그림 7. 화질 별 녹화 동영상 크기 변이
(단위 : Kilo Bytes)
Figure 7. File capacity variation according to video quality values (Kilo Bytes)

<그림 7>은 화질별 녹화 파일의 크기 변이를 보여준다. 실험에 사용된 캡쳐 화면의 해상도는 가로 1600, 세로 900 화소였다. 실험 결과, 고화질인 경우 파일의 평균 크기는 1987.6 KB, 중화질은 887.5, 저화질의 경우는 426.3의 크기를 보였다. 동일 화질인 경우에도 녹화 파일의 크기가 크게 변화하는

것은 캡쳐 화면 간의 비교에서 변화가 작은 경우와 큰 경우가 있기 때문이다. 예를 들어 화면이 정지되어 있거나 문서 작업처럼 화면 변화가 크지 않은 경우는 상대적으로 적은 녹화 용량을 보이고, 동영상 재생은 용량이 상대적으로 증가할 수 있다. <그림 8>은 화질 별 화면 캡쳐의 예이다. 본 구현의 경우 중화질 정도까지는 화면에 표시된 글씨를 충분히 확인할 수 있는 수준의 선명도임을 알 수 있었다.

(a) 고화질 예시 (b) 중화질 예시 (c)저화질 예시

그림 8. 녹화화면비교
Figure 8. Comparison of capture screens

그림 9. 분당 프레임별 녹화 용량 크기
Figure 9. Recording file capacity according to frame rates

<그림 9>는 분당 녹화되는 비디오의 프레임 수를 10 에서 30 프레임으로 늘려 가면서 저장 파일 크기의 변화를 조사한 것이다. 동일한 비디오를 입력으로 하여 조사하였는데 평균적인 녹화 파일의 평균 크기는 각각 362.0 KB와 384.8 KB 였고 30 프레임으로 늘면서 파일 크기는 6.3 % 증가 된 것을 확인할 수 있었다. <그림 10>은 실험에서 얻은 구체적인 파일 크기를 보여준다. 이미지 해상도는 <그림 7>과 동일한 조건이었다.

	1	2	3	4	5	···	16	17	18	19	20	21	22
····· 10 Frm	258	131	323	380	304	···	433	434	259	413	436	466	395
—— 30 Frm	299	163	340	382	326	···	462	459	273	438	453	510	431

그림 10. 동영상 프레임별 저장 용량의 예
Figure 10. storage capacity example according to video frame rates

녹화 파일의 크기는 분당 프레임 수의 변화 보다는 <그림 7>에서처럼 코덱 화질에 더 크게 영향을 받음을 확인할 수 있었다. 결국, 사용 시스템에 맞는 녹화 화질 정도를 미리 설정하여 컴퓨터 사용에 무리를 주지 않도록 하는 것이 성공적인 시스템 기록장치 운용의 전제 조건임을 알 수 있었다.

5. 결 론

본 논문에서는 기밀 자료 유출 예방을 위한 방안으로 컴퓨터 사용 내역을 실시간 수집하고 및 관리하는 방법과 구현 결과를 제시하였다. 수집 정보는 컴퓨터 사용 화면, 키보드 입력, 프로세스 현황 등 3 가지 유형이다. 실험 결과, 실시간으로 필요한 시스템 정보를 추적하여 저장할 수 있었다.

시스템 자원 관리에 영향을 줄 수 있는 화면 녹화 실험에서 녹화 프레임수보다 코덱의 화질 설정이 더 중요함을 확인할 수 있었다. 실험 결과, 고화질 동영상은 저화질에 비하여 4.66 배의 파일 크기를 기록하였고 프레임 수에 따른 파일 용량 변이는 6.3 % 증가에 그쳤다. 결국, 성공적인 기록장치의 운영은 적절한 화질 설정이 필요함을 확인할 수 있었다.

본 논문의 제안 방법을 사용자에게 사전 고지하는 방식으로 운영하는 경우, 자신의 사용 화면 등

의 주요 정보가 그대로 기록되고 있다는 것을 인지하는 것만으로도 사고 예방의 긍정적 효과를 제공할 것으로 기대된다. 다만, 사용자의 적절한 동의 절차 등이 없이 설치되는 경우 사생활 침해 등의 부작용에 주의할 필요가 있다.

또한, 본 제안 방식의 적용은 실제 사고 발생 시에도 사건 파악에 결정적 정보를 제공하여 기밀 유출 사고를 대비한 블랙박스로서 사용할 수 있을 것이다.

추가적인 연구 사항으로는 시스템 사용 정보를 원격지 NAS에 저장하여 기록 장치를 가동 피씨와 분리하여 관리하는 방안과 저장 자료들을 암호화하여 관리하는 방안을 들 수 있다.

References

[1] J. Berdajs, *Extending applications using an advanced approach to DLL injection and API hooking*, J. of Software: Practice and Experience, Vol. 40 No. 7, pp. 567-584, 2010.

[2] J-G Han, J-C Kim, K-J Ban, C. Kim, and E-K Kim, *Personal firewall operating system using API hooking modules*, Fall Conference Proceedings, The Korean Institute of Maritime Information and Commucation Sciences, pp. 551-553, 2011.

[3] S.-S. Hwang, *A method for creating eaching movie clips using screen recording software*, J. of Korean Radiol Soc, Vol. 56, No. 4, pp. 395-402, 2007.

[4] H. Jung, S. Park, *Combination of an adaptive hypermedia system and an external application using a message hooking mechanism*, The Journal of Korean Association of Computer Education, Vol. 8, No. 4, pp. 107-114, 2005.

[5] Y.-K. Kim, *An API hooking technique based on windows kernel*, Ph. D. dissertation, Hannam University, 2010

[6] Camtasia_Studio, http://en.wikipedia.org/wiki/Camtasia_Studio

[7] *Diagram showing different applications using FFmpeg*, http://en.wikipedia.org/wiki/ffmpeg#Multimedia_frameworks_using_FFmpeg

[8] *DLL Injection*, http://www.imaso.co.kr/ ?doc=bbs/gnuboard.php&bo_table=article&wr_id=35398

[9] *DLL_injection*,http://en.wikipedia.org/ wiki

[10] *FFmpeg license and legal considerations*, http://ffmpeg.org/legal.html

[11] *Hooks overview*, http://msdn.microsoft.com/en-us/library

[12] *Keyloggers*, http://www.securelist.com/en/analysis/204792178/Keyloggers_Implementing_keyloggers_in_Windows_Part_Two

개인용 컴퓨터 보안을 위한 시스템 사용 기록기의 구현

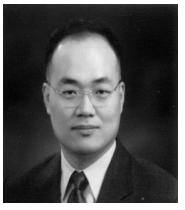
권오성

공주교육대학교 컴퓨터교육과

요 약

일반적으로 기밀 자료의 유출 사고는 외부보다 내부 인력에 의한 고의적인 유형이 많다. 본 논문에서는 이러한 정보 유출 사고를 예방하고 사고 대처 방안으로 사용자 화면, 키보드 입력, 프로세스 현황 등 3 가지 유형의 시스템 사용 정보를 실시간으로 수집하고 저장하는 방법을 제안한다. 첫 번째 정보인 사용자 화면 기록은 주기적으로 진행하는 실시간 화면 캡쳐 이미지 리스트를 동영상으로 변환하는 방식으로 진행하였다. 나머지 프로세스 및 키보드 입력 등은 메시지

후킹 기법을 사용하여 실시간으로 필요한 자료를 수집하여 저장하였다. 시스템 사용 기록기의 성능을 실험한 결과 3 가지 사용 기록 정보의 성공적인 수집과 현장 적용이 가능하였다. 제안하는 방안을 PC(개인용 컴퓨터)에서 상시 운영하는 경우 3 가지 수집 유형 중에서 화면녹화 분야의 시스템 자원 소비가 두드러졌으며, 녹화 프레임 수보다는 화질에 더 민감한 변화를 보여 고화질은 일반 화질로 운영하는 경우보다 4.6 배의 비디오 파일 용량을 보였다.



Oh-Sung Kwon received the Ph.D. degree in the Department of Computer Engineering from Chung-Ang University in 1994. He has been a professor in the Department of Computer Education at Gongju National University of Education since 1995. His current research interests include multimedia data processing and digital image processing.

E-mail address: oskwon@gjue.ac.kr