情報保護學會論文誌
第15卷 第4號, 2005. 8

저장매체와 프린터를 통한 파일유출 모니터링시스템*

최 주 호,[1†] 류 성 열[2‡]
[1](주)디지탈센스, [2]숭실대학교

Monitoring System of File Outflow through Storage Devices and Printers*

Joo-ho Choi,[1†] Sung-yul Rhew[2‡]
[1]Digitalsense Co.Ltd, [2]Soong Sil University

요 약

통신망과 저장장치의 발달에 따라 내부 사용자에 의한 중요 정보 자산의 외부 유출이 증가하고 있으므로 이에 대한 보안을 강화해야할 필요성이 증대되고 있다. 제안한 파일유출 모니터링시스템은 클라이언트에서 파일이 저장매체에 저장/복사되거나 파일이 종이문서로 인쇄되어 외부로 유출되는 경우에 로그를 발생시켜 서버에서 이를 모니터링한다. 모니터링 방법은 커널 레벨에서 I/O Manager에 의해 발생하는 IRP의 필터링과 Win32 API 후킹 기법을 사용하였다. 특히 파일을 저장매체에 저장하는 경우, 네트워크 공유를 통하여 파일을 저장하는 경우 및 파일의 인쇄를 통하여 유출하는 경우에 로그를 발생시키고 모니터링하는 방법을 구현하였다. 모니터링시스템은 윈도우즈 2000 및 XP 실험환경에서 파일의 복사와 인쇄 시 로그가 100% 발생되고 모니터링 기능이 수행됨을 확인하였다.

ABSTRACT

The files of intellectual property on computer systems have increasingly been exposed to such threats that they can be flowed out by internal users or outer attacks through the network. The File Outflow Monitoring System monitors file outflows at server by making the log when users copy files on client computers into storage devices or print them, The monitoring system filters I/O Request packet by I/O Manager in kernel level if files are flowed out by copying, while it uses Win32 API hooking if printed. As a result, it has exactly made the log and monitored file outflows, which is proved through testing in Windows 2000 and XP.

Keywords : *File Outflow, Monitoring, IRP, Device Driver, Hooking*

Ⅰ. 서 론

보안의 핵심은 정보의 중요자산을 외부로부터 침입/파괴/변조를 방지하는 부분과 외부/내부 사용자에 의해 중요 정보의 유출을 방지하는 부분으로 구분할 수 있다. 지금까지는 외부자의 침입에만 보안정책이 맞추어져 방화벽/IDS를 설치하는 수준이었지만 기업보안 형태가 내부보안의 중요성이 인식되면서부터 점차 사내 보안으로 초점이 맞춰지고 있다. CSI/FBI의 2001년에 실시한 기업의 기밀정보 유출 실태조사에 따르면 내부자에 의한 정보유출의 경우가 외부해커에 의한 경우보다 9배나 높은 것으로 나타나[1] 내부자에 의한 정보유출 정도가 더 심각하다.[2,3] 이러한 실정에서 기업 내 지적 재산에 대해 Firewall, IDS, VPN등으로 내부자에 의해서 발생하는 정보 유출에 대해서는 대응방법이 될

접수일 : 2005년 3월 11일 ; 채택일 : 2005년 8월 5일
* 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음.
† 주저자, admin@digitalsense.co.kr
‡ 교신저자, syrhew@comp.ssu.ac.kr

수 없다.[4] 컴퓨터 내부의 정보는 서버 및 컴퓨터 등이 고도화되고, 저장장치는 소형화와 대용량화되어 디스켓이나 CD-RW, 이동저장매체 등을 통하여 내부정보의 복제, 유출, 인쇄가 용이하게 되었으며, 인터넷과 네트워크는 고속화되어 다량의 데이터를 쉽게 전송할 수 있어 보다 안전한 보호 장치 및 관리시스템이 요구되고 있다.[5]

본 연구에서는 Windows 시스템 환경에서 컴퓨터내의 파일이 내부사용자에 의해 저장장치에 복사되거나 프린터를 통하여 종이문서로 인쇄되어 유출되는 경우 이를 실시간으로 서버에서 모니터링하는 방법을 구현한다. 이를 위하여 운영체제의 후킹(Hooking) 기법과 파일시스템 드라이버에 대한 드라이버 필터링 기법을 적용하였다. 유출 모니터링시스템 설계를 위하여 유출모니터링, 파일시스템과 디바이스 드라이버, 후킹 등에 대해서 관련 논문과 기술서적을 조사하였다. 인터넷을 통한 파일유출 모니터링 솔루션을 참조하여 파일유출에 필요한 로그 항목을 도출하였으며, 설계된 내용을 일부 구현하여 실험을 수행하였다.

이러한 파일유출 모니터링시스템은 조직내의 정보 유출에 대한 취약점을 분석하고 유출된 내용을 추적할 수 있어 내부 보안강화에 크게 기여하며, 컴퓨터 범죄에서 증거확보와 효과적인 분석에 사용할 수 있다.[6]

Ⅱ. 관련 연구

2.1 파일유출모니터링

내부파일이 외부로 유출되는 형태는 ① 인터넷을 통한 유출, ② 저장장치를 통한 유출, ③ 팩스를 이용한 유출, ④ 프린터 출력에 의한 유출, ⑤ 노트북이나 자료 저장매체의 분실에 의한 유출로 구분한다.[7] 통신회선의 고속화로 대용량의 자료를 쉽게 전송할 수 있어 많이 사용되고 있는 인터넷을 통한 유출은 e-mail, Web mail, Web hard, Messenger와 FTP통신을 이용하여 파일을 전송하는 방법 등이 포함된다. 인터넷을 이용하여 정보가 유출되는 것에 대한 보안유지 방법으로 IP와 Port를 제어하는 방법과 네트워크 패킷을 분석하여 유출되는 사항을 모니터링하는 방법이다. IP와 Port의 제어방법은 특정한 IP나 Port를 사용하지 못하도록 통제하는 방법이나 다양한 메일이나 Web hard, Messenger등을 이용하여 내부 정보를 유출시키는 경우 IP가 신규 생성되거나 빈번히 변경되면 통제가 어려운 점이 있어 패킷을 분석하여 이를 모니터링하는 방법을 사용한다. 네트워크를 모니터링하는 방법은 패킷을 캡처한 후 호스트 정보, 프로토콜 정보, 서비스 정보 등을 분석하는 방법과 SNMP(Simple Network Management Protocol)를 사용하여 분석할 수 있으나 SNMP는 링크 계층까지만 분석할 수 있다는 단점이 있다.[8]

휴대가 간편하고 대용량의 자료를 저장할 수 있는 저장장치나 프린터 출력에 의해 내부파일이 유출될 수 있는 위험이 높아 이를 방지할 수 있는 방법이 필요하다. 내부파일의 외부 유출을 방지할 수 있는 방법으로 저장장치를 제어하여 파일의 저장이나 인쇄를 방지하는 제품이 일부 개발되었다.[9] 그러나 업무 환경이 모든 저장매체를 제어하여 유출을 차단할 수 없는 점이 있어 유출을 모니터링할 수 있는 방법이 필요하지만 이에 대한 연구는 매우 적다.

2.2 후킹 방법

컴퓨터 내부파일이 다양한 형태의 저장장치에 저장되거나 인쇄되는 것을 모니터링하기 위해서는 프로세스, 파일시스템, 디바이스 드라이버 작업내용을 후킹하거나 필터링하여 유출 로그(Log)를 생성하고 이를 서버에 전달함으로써 파일이 유출되는 사항을 모니터링한다. 후킹은 특정 프로그램 또는 운영체제 내 실행코드 영역을 가로채어 이 실행코드 영역에서 발생하는 행동 또는 이벤트를 감시하거나 변경하는 것을 말한다. 이러한 후킹에는 User Level Hooking과 Kernel Level Hooking이 있다. User Level Hooking은 윈도우 메시지 후킹, DLL Injection, API후킹으로 나눈다.[10] Kernel Level Hooking은 device driver 프로그래밍을 통해 윈도우 Kernel이 제공하는 Primitive Function들을 후킹하는 기법을 말하며, User Level Hooking과 차이점은 후킹 영역이 윈도우 시스템 전체 영역에 대한 후킹이 가능하다는 것이다.[11]

2.3 디바이스 드라이버 필터링

저장장치의 드라이버를 통하여 발생하는 프로세스를 필터링하기 위해서는 운영체제의 커널모드에서 I/O Manager와 디바이스 드라이버에 대해 이해하

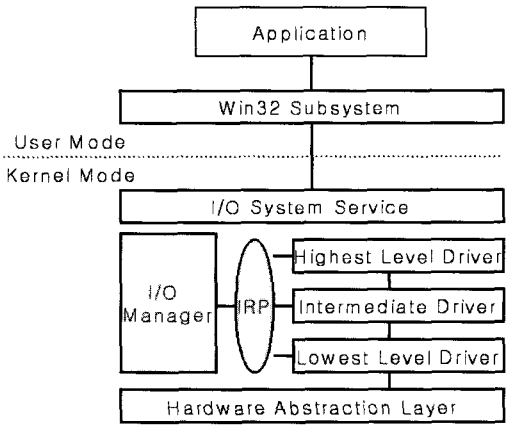

그림 1. 디바이스 드라이버의 호출

여야 한다. I/O Manager는 사용자 프로세스(Process)의 요청을 디바이스 드라이버에게 I/O 요청 패킷(IRP: I/O Request Packet)이라는 형태로 제공한다. 드라이버는 기본적으로 커널모드에서 작동하며 이러한 드라이버는 계층별로는 Lowest-level 디바이스 드라이버, Intermediate 드라이버, Highest-level 드라이버로 구분하며 기능별로는 파일시스템 드라이버, 필터 드라이버, 네트워크 드라이버 등으로 나눈다.[12] 그림 1은 디바이스 드라이버를 호출하여 처리하는 흐름을 나타낸 것으로 응용프로그램은 디바이스 I/O로 Win32를 호출하고 I/O 시스템서비스에 전달되어 IRP를 작성한다.

I/O Manager는 IRP를 1개의 디바이스 드라이버에 전달하고 이 디바이스 드라이버는 하드웨어와 상호작용하며 IRP 처리를 완료한 후 데이터 및 결과를 Win32와 사용자 응용 프로그램에 전달한다.[13] 파일시스템 드라이버는 특정한 파일에 데이터를 쓰는 경우에 응용프로그램에서 요청사항을 드라이버로 전달하여 파일의 Open, Read, Write, Modify, 사용자의 권한, 접근제어 등 파일과 폴더에 관련된 모든 동작을 수행한다.[14]

Ⅲ. 파일유출모니터링 시스템

3.1 개발환경과 요구사항

본 논문에서 제안하는 파일유출 모니터링시스템은 Windows 2000과 XP 환경에서 이동저장매체와 네트워크 공유를 통하여 파일이 복사되어 유출되는 경우와 프린터를 통하여 파일이 인쇄되어 종이문서

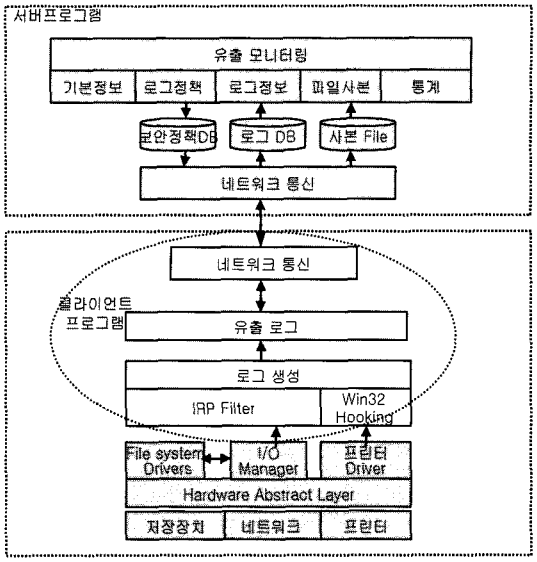

그림 2. 파일유출 모니터링시스템 구성

로 유출되는 경우에 대한 모니터링이다. 시스템의 구성은 그림 2와 같이 클라이언트/서버로 서버 프로그램은 유출모니터링에 대한 보안정책 설정과 로그 발생시 실시간으로 모니터링 할 수 있고, 클라이언트 프로그램은 파일의 저장이나 인쇄 작업시 작업의 내용을 로그로 남기고 이를 서버에 전달하는 Agent 기능을 수행한다.

파일의 유출을 모니터링 하는데 필요한 항목은 When:파일의 복사/인쇄 시간, Who: 복사/인쇄한 컴퓨터 이름, Where(from):드라이브+폴더명, File (from) :파일명, Where(to):드라이브+폴더명, File (to):파일명, Outflow type:파일의 저장/인쇄 작업구분 등이다.

3.2 파일유출 모니터링시스템 구조

파일유출 모니터링시스템의 전체 구조는 그림 3과 같으며 그림의 왼쪽은 운영체제의 내부 처리를 표시한 것이고, 오른쪽은 유출을 모니터링하기 위하여 설계하는 클라이언트와 서버 부분을 표시한 것이다. 파일의 저장사항을 모니터링하기 위해서 Win32 API 후킹방식과 드라이버 필터링 방식이 있다. Win32 API 후킹방식을 이용하면 응용프로그램에서 사용하기는 편리하나 API는 IRP 코드의 하나 또는 다수의 코드가 포함되어 다양한 파일 저장방식에 따른 원본파일의 정보를 정확히 얻기 어려운 점

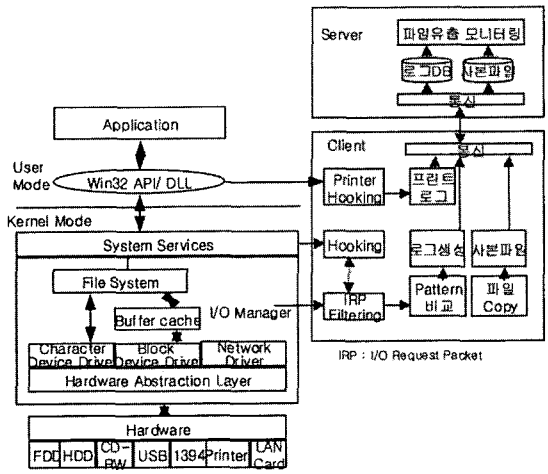

그림 3. 파일유출 모니터링 처리 구조

이 있다.

본 연구에서는 저장장치와 가장 가까운 드라이버에서 수행하는 IRP의 Major code를 필터링하는 방법을 사용한다. 파일을 유출한다는 것은 컴퓨터내의 파일을 외부의 저장매체에 저장하거나 인쇄를 실행하는 것으로, 사용자가 파일을 저장하는 방법에는 Drag & Drop, Copy & Paste, 보내기, Save/ Save as 형태가 있다. 이러한 작업행위를 응용프로그램에서 수행하면 I/O Manager는 이를 시스템 함수에 작업을 수행할 수 있도록 명령을 부여하고 드라이버를 통하여 Hardware에 전달하여 실제 저장작업을 실행한다. 이때 I/O Manager는 명령을 시스템 함수에 IRP 코드 형태로 부여하는데 Major code, Minor code, Fast I/O code로 구분된다. Major code는 작업의 대분류 코드이고, Minor code는 작업의 내용을 세분하여 나타내는 코드이며 Fast I/O code는 Cache Manager가 수행하는 내용으로 파일시스템 드라이버를 거치지 않는 작업을 표시한다. I/O Manager에 의해 생성되는 일반적인 IRP Major code는 표 1과 같고 I/O Request과정에서 필요한 정보를 저장하는 장소로서 드라이버 간에 통신을 위해 사용된다.

IRP의 구조를 살펴보면 그림 4와 같이 고정된 헤더와 하나 또는 여러 개의 스택으로 구성되어 있다.

IO_STATUS_BLOCK에는 I/O Operation에 대한 마지막 상태 정보를 가지고 있으며 I/O_ STACK_ LOCATION은 Function Code나 Parameter등을 저장하기 위한 목적으로 사용된다.

프린트 작업을 모니터링하기 위해서 Win32 API

표 1. 일반적인 IRP Major code

Major code	내 용
IRP_MJ_CREATE	디바이스 파일의 작성 또는 열기
IRP_MJ_CLOSE	핸들 폐쇄
IRP_MJ_READ	읽기
IRP_MJ_WRITE	쓰기
IRP_MJ_CLEANUP	파일 핸들에 대한 미처리 IRP 취소
IRP_MJ_DEVICE_CONTROL	디바이스 I/O 콘트롤
IRP_MJ_POWER	전원관리 요구
IRP_MJ_PNP	플러그 앤 플레이 메시지

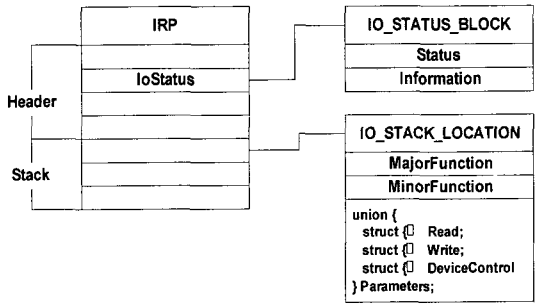

그림 4. IRP 구조

후킹과 Spool을 후킹하는 방법이 있다. 본 연구에서는 Win32 API의 하나인 GDI.dll의 Startdoc 함수를 Injection하여 프린트 시점과 파일의 정보를 후킹하였으며 이 방식은 모든 프린트 작업시 호출하는 함수이므로 안정적인 모니터링이 가능하다. Spool을 후킹하는 방법은 드라이버 계층에서 처리하면 안정적이나 프린트시 파일이 이미지 형태로 생성되어 프린터 버퍼로 전송되는데 이를 후킹하여 사본파일을 생성시키면 파일의 크기가 커져 서버에 전송하는데 시간이 많이 소요되는 문제점을 해결하지 못하였다.

이러한 방법들을 이용하여 안정되고 신뢰성 있는 파일유출 모니터링시스템을 설계하기 위하여 그림 5와 같이 크게 5개 모듈로 구성하였다.

파일시스템을 이용하여 파일을 저장하는 경우에 IRP 필터링을 통하여 유출 모니터링이 가능하나 동기화통신이나 적외선 통신을 통하여 PDA와 같은 장치에 파일을 유출하는 경우에는 통신되고 있는 내용에 대해 IRP 필터링 기법을 적용할 수 없어 모니터링할 수 없다는 단점이 있다.

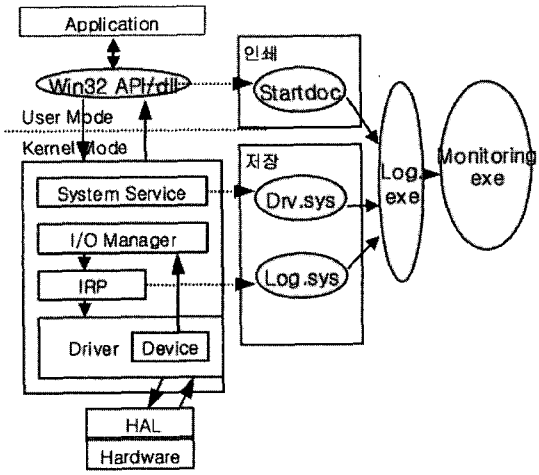

그림 5. 파일유출 모니터링 프로그램 구조

각 모듈의 기능은 표 2와 같다.

표 2. 모듈별 기능

구 분	내 용
Drv.sys	저장장치의 관련 정보를 수집하기 위해 Kernel Mode에서 시스템서비스를 수행하고 있는 프로그램인 Ntkernel.exe를 후킹하며 처리되고 있는 프로세스명과 Open된 파일에 대한 드라이브, 파일명 등의 정보를 수집하여 Log.sys와 Log.exe와 통신
Log.sys	파일이 저장되면 I/O Manager는 처리되는 사항을 IRP 코드 형태로 발생시키고 드라이버 필터링을 통하여 프로세스명, 파일 Path명, 파일명 등의 정보를 추출하여 Log.exe로 전송
Startdoc	프린트 작업이 발생되는 경우 로그를 생성하기 위하여 GDI.dll이 서비스하는 함수를 후킹하여 프린트 시작, 종료, 프린터 파일, 프린터명 등의 정보를 수집하여 로그를 생성
Log.exe	Drv.sys에서 프로세스의 생성과 드라이브에 대한 변동사항을 수신받아 Log.sys를 통하여 파일이 저장되는 정보와 Startdoc를 통하여 인쇄 정보를 받아 로그 정보를 생성한 후 통신 루틴을 통하여 서버 프로그램인 Monitoring.exe에 전송
Monitoring.exe	로그 데이터는 Log.exe를 통하여 수신하여 DB에 입력하고, 사본파일을 폴더에 저장하여 실시간으로 유출사항을 모니터링하며 기본정보, 로그정책, 통계정보 등을 제공

Ⅳ. 유출로그의 생성과 모니터링

4.1 저장매체를 이용한 파일 유출 시 모니터링

파일의 저장 작업을 Drag & Drop, Copy & Paste, 보내기, Save/Save as로 구분하고 이에 대한 작업을 세분하면 표 3과 같이 나눌 수 있다.

표 3. 저장작업의 세부동작

Drag & Drop	Copy & Paste	보내기	Save/as
파일 선택	파일 선택	파일 선택	파일 열기
저장위치로 이동	대상파일 복사	보내기	파일수정(생성)
저장완료	저장위치 선택	저장완료	저장위치 선택
	파일 붙이기		Save/Save as

파일을 저장하면 상기 표 3과 같은 세부동작에 따라 그림 6과 같은 처리과정을 거쳐 로그를 생성한다. I/O Manager는 컴퓨터의 I/O에 관련된 모든 작업을 IRP 코드를 발생시켜 처리하면 Log.sys는 파일의 저장작업에 해당하는 IRP 코드만을 필터링한 후 메모리에 Process ID와 File object를 저장한다. 파일을 저장하는 저장장치 정보는 Drv.sys를 통하여 수집한 후 메모리에 저장된 프로세스와 비교하여 동일 작업이면 파일정보를 이용하여 로그를 생성한다.

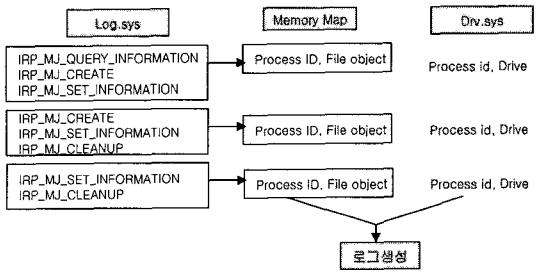

그림 6. 유출로그 생성과정

이에 대한 처리흐름은 그림 7과 같으며 처리 절차는 아래와 같다.

① 파일의 저장 유형별로 처리되는 IRP를 Log.exe의 PatternDefine에 정의한다.

② 파일의 생성이나 열기가 실행되면 Drv.sys를 통하여 해당 Process정보를 추출한다.

③ Drv.sys의 GetProcessName와 Log.sys의 GetProcess를 비교하여 동일 프로세스에 해당하는 MakeMap루틴에서 메모리에 저장한다.

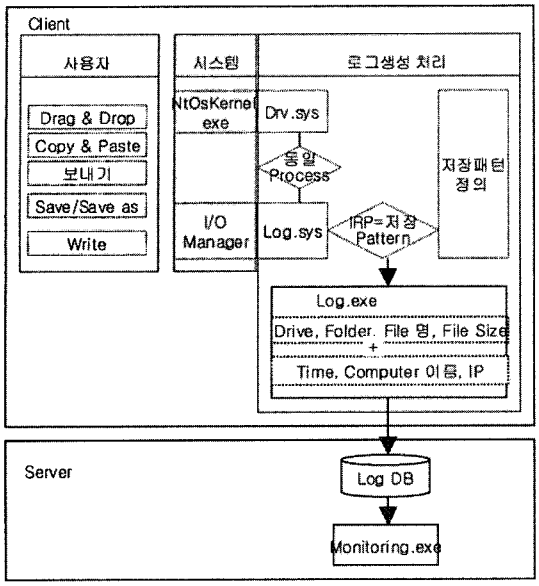

그림 7. 유출로그 생성 처리 흐름

④ Log.exe에서 정의한 Pattern과 Log.sys의 MakeMap에서 생성한 내용을 Log.exe의 PatternCompare 루틴에서 비교한다.

⑤ 비교하여 동일하면 Log.exe의 MakeLog(1)을 통하여 로그를 생성한다

⑥ 유출시간, 작업컴퓨터, IP등 부가정보를 추가하여 MakeLog(2)를 생성한다.

⑦ Log.exe에서 MakeLog(1)과 MakeLog(2)를 합쳐 MakeLog를 생성한다.

⑧ Log.exe의 SendLog에서 서버 프로그램인 Monitoring.exe로 로그를 전송한다.

⑧ HookLanman은 네트워크 공유를 통한 파일 저장시 저장 정보를 수집한다.

⑩ Monitoring.exe는 이를 수신하여 LogMonitoring을 통하여 실시간으로 유출사항을 모니터링하고 로그DB에 저장한다.

4.2 네트워크 공유에서의 유출 모니터링

네트워크 공유상태에서 파일유출의 모니터링은 네트워크 공유방법, 파일유출 형태, 유출작업의 주체, 클라이언트 프로그램의 설치여부 등 다양한 사용 환경에서 유출되는 사항을 모니터링하여야 한다. 네트워크 공유 방법에는 가상드라이브와 폴더공유 방법이 있으며, 파일유출 형태는 사용자의 컴퓨터의 파일을 타 사용자의 컴퓨터에 저장하는 경우와 타 사용자의 컴퓨터 파일을 자신의 컴퓨터에 파일을 복사로 구분한다. 네트워크 공유상태에서 파일의 저장이나 복사의 경우에도 저장매체에 저장하는 경우와 같이 파일시스템의 I/O Manager가 처리하므로 파일의 저장상태 프로세스를 IRP를 통하여 정보를 수집한다. 로그를 생성하는 과정은 HookLanman 함수를 통하여 폴더에 '\\'정보가 추가되어 있어 네트워크 공유상태에서 파일의 복사가 이루어졌음을 식별하고 From, To 정보를 추출한다.

4.3 파일의 인쇄 시 유출 모니터링

파일유출 경로 중 파일을 종이문서로 인쇄하여 유출하는 경우도 많이 발생하고 있어 모니터링시스템은 도면이나 파일이 종이문서로 인쇄된 경우 언제, 누가 어떤 파일을, 어떤 프린터를 통하여 인쇄하였으며, 인쇄한 파일의 내용은 무엇인지를 알 수 있는 방법이다. 그림 8과 같은 과정을 거치며 처리 절차는 아래와 같다.

① 파일을 인쇄하기 위하여 파일을 Open하면 Drv.sys를 통하여 해당 프로세스 정보를 추출한다.

② Log.exe에서 Open된 파일들에 대해 GetProcess와 비교하여 동일 Process에 대해 GetfileInfo를 통하여 파일정보를 메모리에 저장한다.

③ 파일이 인쇄되면 Startdoc를 통하여 인쇄 진행 상태와 파일에 대한 정보를 추출한다.

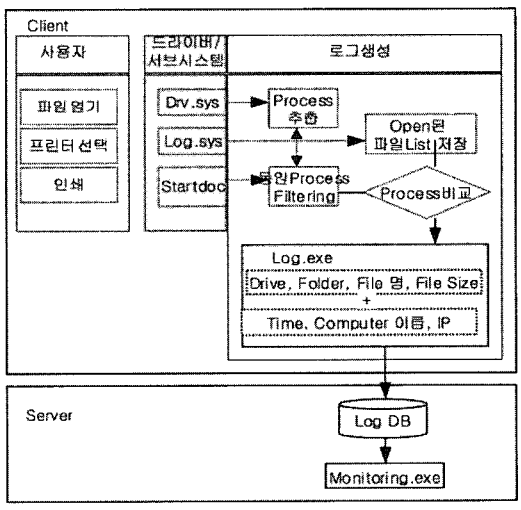

그림 8. 프린트에서 유출로그 생성

④ Log.exe에서 추출한 파일정보와 Startdoc에서 추출한 파일정보를 비교하여 MakeLog(1) 로그를 생성한다
⑤ 인쇄시간, 프린터명, IP등 부가정보를 추가하여 MakeLog(2)를 생성한다.
⑥ Log.exe에서 MakeLog(1)과 MakeLog(2)를 합쳐 MakeLog를 생성한다.
⑦ Log.exe.의 SendLog에서 서버 프로그램인 Monitoring.exe로 로그를 전송한다.
⑧ Monitoring.exe는 이를 수신하여 LogMonitoring을 통하여 실시간으로 유출사항을 모니터링하고 로그DB에 저장한다.

V. 기능 및 성능 분석

Windows환경에서 파일을 저장매체에 복사하는 경우와 네트워크를 통하여 파일을 복사하는 경우, 프린터를 통하여 파일을 인쇄하는 경우에 대하여 실시간으로 유출되는 내용을 모니터링할 수 있도록 구현되어야 하고, 어떤 매체를 통하여 유출되었으며, 유출된 파일의 내용이 무엇인지 서버에서 관리자가 확인할 수 있도록 한다.

5.1 시험 환경

파일의 유출을 모니터링하기 위한 시험 환경은 그림 9와 같이 구성하였으며 '컴퓨터 A'에는 Windows 2000 Server 환경에 서버프로그램과 MS-SQL을 설치하였고, '컴퓨터 B'는 Windows XP Home Edition, '컴퓨터 C'는 Windows 2000 Professional 환경에 클라이언트 프로그램을 설치하고 네트워크로 연결하였다. 이동저장장치는 FDD, 외장형 HDD, USB 2.0 저장장치, 프린터는 기본 프린터와 네트워크 프린터를 설치하였다.

5.2 시험 방법

1) 컴퓨터 A에 서버프로그램을, 컴퓨터 B와 컴퓨터 C에 클라이언트 프로그램을 설치하고 네트워크로 연결한다.
2) 컴퓨터 B에 있는 파일을 컴퓨터 B의 이동저장매체(USB)와 네트워크 공유를 통하여 컴퓨터 C에 Drag & Drop, Copy & Paste, 보내기, Save as로 저장한다.
3) 컴퓨터 B에 있는 파일을 프린터와 Fine Print로 인쇄 작업을 수행한다.

5.3 시험 결과

상기 시험 방법에 따라 파일이 유출되는 경우 모니터링되는 결과는 그림 10과 같다.

1) 시험 방법 1), 2), 3)에서 파일유출에 대한 모니터링 정보의 내용은 그림 10의 ①에 표시된 일련번호, 유출시간, 유출경로, 컴퓨터 이름, 사용자 이름, 폴더명, 파일명, 파일크기이다.
2) 파일유출 경로에 대한 구분은 그림 10의 ②에서 나타난 것과 같이 유출형태는 FDD, Internal HDD, External HDD, Network Share, Removable, Printer로 구분하여 표시함으로써 유출의 경로를 알 수 있다.
3) 파일 저장방법에 따른 로그는 그림 10의 ③에서 나타난 것과 같이 파일 저장방법을 Drag

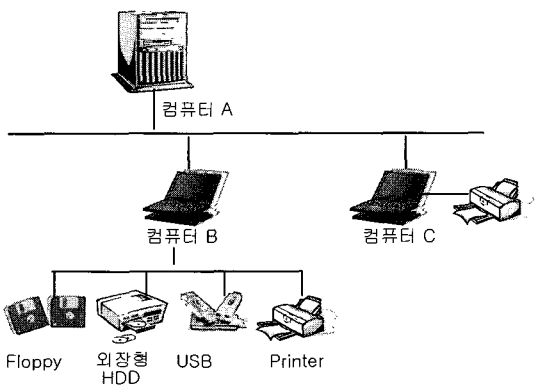

그림 9. 시험 환경의 구성

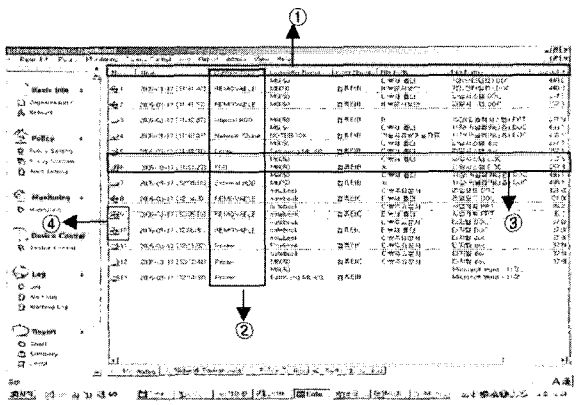

그림 10. 파일유출 모니터링 결과화면

& Drop, Copy & Paste, 보내기로 파일을 저장하는 경우 로그가 생성된다. 파일이 저장되었던 원본파일의 From 정보와 파일이 복사/저장된 To 정보에 대해 로그가 발생하나 Save/Save as인 경우 파일이 Open되어 저장되므로 원본파일에 대한 정보 없이 파일이 저장된 To 정보에 대해 로그가 발생한다.

4) 파일의 인쇄는 기존 파일을 열어 인쇄하는 경우, 파일을 신규 생성하여 인쇄하는 경우, 화면을 그림으로 저장하여 인쇄하는 경우 등에 대해 로그가 생성되었다.

5) 로그 발생의 정확성을 측정하기 위하여 10개의 파일을 Copy & Paste, Drag & Drop, 보내기 방법과 1개의 파일을 Sava as로 USB와 네트워크 공유를 통하여 각각 20회 저장하고 로그가 발생되는 시험의 결과는 표 4와 같다. 상기 시험의 결과에서 나타난 것과 같이 저장매체나 네트워크 공유를 통하여 파일을 저장하는 경우 100% 로그가 발생하고 모니터링 되었다.

6) 파일을 인쇄 작업 시 발생되는 로그의 정확성을 시험하기 위하여 Fine Print와 프린터를 통하여 1개의 파일을 각각 20회 시험을 수행하고 로그가 발생하는 결과는 표 5와 같다. 분석한 결과 프린터로 인쇄시에는 정확히 1회 로그가 발생되었으나 Fine Print로 인쇄시에는 파일로 생성하는 경우와 생성된 파일을 저장하는 경우에 로그가 발생하여 2회 생성되어 파일의 저장작업에는 로그를 발생시키지 않은 처리가 필요하였다. 또한 프린트가 진행된 수량을 확인하거나 프린트 작업이 성공적으로 완료되었는지 또는 중단되었는지의 여부, 중단된 경우에는 몇 페이지가 인쇄되었는지의 등의 프린트 상태정보를 모니터링하기 위해서는 더욱 연구가 필요하다.

7) 본 연구를 통하여 저장매체에 파일이 Drag & Drop, Copy & Paste, 보내기 방법으로 저장되는 경우에 로그가 100% 발생되고 모니터링됨을 확인하였다. 파일이 내장형 HDD에 저장되는 경우는 사용자에 의해 저장하는 경우도 있지만 운영체제나 바이러스 프로그램에 의해 패치되는 경우에도 파일의 저장으로 간주되어 로그가 발생하므로 예외처리가 필요하였다. 로그는 각 2kb 크기를 가지며 서버의 DB에 저장된다. 다수의 사용자가 다량의 데이터를 동시에 복사하는 경우에 대한 서버 전송속도와 트래픽 문제는 서버의 성능과 네트워크 속도에 따라 다르며 추가 시험이 필요하다.

표 4. 파일저장 시 로그 발생건수

구분	시험횟수	USB저장	네트워크 공유
Copy & Paste	1회	10건	10건
	5회	10건	10건
	10회	10건	10건
	20회	10건	10건
Drag & Drop	1회	10건	10건
	5회	10건	10건
	10회	10건	10건
	20회	10건	10건
보내기	1회	10건	10건
	5회	10건	10건
	10회	10건	10건
	20회	10건	10건
Sava as	1회	1건	1건
	5회	1건	1건
	10회	1건	1건
	20회	1건	1건

표 5. 파일인쇄 시 로그발생 건수

구분	시험횟수	프린터로 인쇄	Fine Print
인쇄 작업	1회	1건	2건
	5회	1건	2건
	10회	1건	2건
	20회	1건	2건

Ⅵ. 결론 및 향후과제

본 논문의 파일유출 모니터링시스템은 내부 정보에 대한 유출방지 필요성이 커지고 있는 이 시점에서 컴퓨터내부의 파일, 도면, 컨텐츠, 원시코드 등 중요한 자료들이 내부사용자에 의해 외부로 유출되는 경우에 언제, 어떤 파일이, 무슨 장치를 통하여 유출되었는지 유출사실을 알 수 있어 내부보안을 강화하는데 크게 기여할 것이다.

저장장치나 통신 방법이 다양해짐에 따라 휴대폰, PDA등을 이용하여 컴퓨터내의 파일을 통신케이블을 이용하여 유출하는 방법과 무선 LAN, Bluetooth, 적외선(IrDA)통신 등 무선을 이용하여 파일의 유출하는 방법도 이용될 수 있어 이에 대한 유출모니터링 방법도 연구되어야 한다. 또한 파일의 유출을 근본적으로 방지할 수 있는 저장장치 및 프린

터를 제어할 수 있는 방법에 대한 연구가 필요하다.

참 고 문 헌

[1] 박유나, 기밀문서 유출방지를 위한 전자우편 모니터링 시스템, 석사학위논문, 단국대, p1, 2003

[2] 원종진, 내부자에 의한 기업비밀 유출방지 대책연구, 석사학위본문, 동국대, p1, 2003

[3] 황성국, VxD를 이용한 PC보안시스템 구축에 관한 연구, 석사학위논문, 동국대, p6, 2001

[4] 김종원·최종욱, "기업 정보 유출 방지를 위한 기술", 정보처리학회지, 제10권 제2호, pp. 87-95, Mar. 2003

[5] 차성철, KMS 기반에서의 DRM을 이용한 문서보안 시스템 구현에 관한 연구, 석사학위논문, 성균관대, p.24, 2003

[6] 이형우, 이상진, 임종인, "컴퓨터 포렌식스 기술", 정보보호학회지, 제12권 5호, pp.8-12, Oct. 2002.

[7] 최형규, 정보시스템 보호 및 해킹방지를 위한 보안시스템 구축에 관한 연구, 석사학위논문, 숭실대, p.20-24, 2002.12

[8] 정재홍, 네트워크 모니터링을 이용한 인터넷 환경에서 병렬 분산 처리 시스템의 성능평가, 석사학위논문, 원광대학교, 2003.10

[9] 사파소프트, "waterwall", http ://www.safasoft.co.kr/

[10] Jeffry Richer, Programming Application for Microsoft Windows, Microsoft, pp 751-850, 1999

[11] 이기정, 권태경, 황성운, 윤기송, "신뢰할 수 없는 DRM 클라이언트 시스템 하에서 키 보호를 위한 Secure Storage Device의 연구", 정보보호학회 논문지 제14권 제2호, pp. 3-11, 2004.4

[12] 김대중, 커널 모듈을 이용한 접근제어 설계 및 구현, 석사학위논문, 한서대, p.19, 2002

[13] Chris Cant저, 박햇님 역, 윈도우즈 드라이버 모델, pp26-31, 에이콘, 2002

[14] 이남훈, 유신근,심영철, "Windows 2000 기반의 파일 보호 시스템 설계 및 구현", 정보처리학회논문지 C 제8-C권 제6호, p742, Oct. 2001

APPENDIX

A. 함수의 기능 설명

표 1. Drv.sys

함 수	수행 내용
NtCreateFile	드라이브를 후킹하여 파일 생성시 발생하는 정보를 수집
NtOpenFile	드라이브를 후킹하여 파일열기 시 발생되는 정보를 수집
DispatchIoctl	통신 루틴
GetProcessName	프로세스 정보를 수집
GetfullRegname OrFileName	파일정보를 수집

표 2. Log.sys

함 수	수행 내용
DriveEntry	데이터 초기화
GetProcess	현재 IRP 프로세스 정보 수집
HookDrive	특정 드라이브에 볼륨 필터 드라이버를 로딩
HookLanman	공유 네트워크에 관련된 정보 수집
MakeMap	추출된 데이터를 Memory List 구성
FastIoDeviceControl	Log.exe와 통신

표 3. StartDoc

함 수	수행 내용
CreateDoc	프린트 시작 시점 정보를 추출
Enddoc	프린트 종료시점 정보 추출
GetPrint	프린트 파일에 대한 정보 추출
SendPrint	추출된 정보를 Log.exe에 전송

표 4. Log.exe

함 수	수행 내용
GetProcess	Drv.sys를 통하여 작동되고 있는 Process정보를 수신
GetFileStatus	Drv.sys와 Log.sys를 통하여 파일의 열기, 쓰기 등의 상태정보를 수신
GetFileInfo	Log.sys에서 작업이 진행되고 있는 IRP와 파일의 정보 수신
PatternDefine	파일의 저장형태에 대한 Pattern을 정의
PatternCompare	GetFileInfo와 PatternDefine루틴을 비교하여 로그를 생성
MakeLog(1)	Log를 생성
MakeLog(2)	시간, 컴퓨터이름 등 정보를 추가
MakeLog	MakeLog(1)+MakeLog(2)
MakeCopyFile	로그생성시 파일을 복사하여 사본파일을 생성
InitInstance	Drv.sys와 Log.sys를 로드시키고 변수를 초기화
ThreadActive	쓰레드의 작동여부를 지속적으로 확인
SendLog	생성된 로그데이터와 사본파일을 서버 프로그램인 Monitoring.exe에 전송

표 5. Monitoring.exe

함 수	수행 내용
BasicInfo	네트워크 연결상태 등 기번적인 사항에 대한 정보를 제공
LogPolicy	로그 발생을 위한 정책을 설정
LogMonitoring	수신된 Log 데이터를 DB에 입력하고 실시간 모니터링 처리
CopyFile	본파일을 저장하고 Log데이터와 동기화 처리
LogSearch	발생된 로그를 다양한 검색조건에 맞도록 검색을 지원
LogReport	로그에 대해 유출형태별, 부서별, 일자별, 사용자별 등 다양한 통계와 분석 자료를 제공

〈著者紹介〉

최 주 호 (Joo-Ho Choi) 정회원
1978년 2월: 한양대학교 산업공학과 졸업
1994년 2월: 숭실대학교 정보과학대학원 석사
1997년 5월: 정보처리기술사(정보관리 분야)
1997년12월: 정보시스템 공인감리인
2001년 8월~현재: 숭실대학교 대학원 전자계산학과 박사과정
2002년 2월~현재: (주)디지탈센스
〈관심분야〉 컴퓨터/네트워크 보안, 산업보안, 정보시스템 감리

류 성 열 (Sung-yul Rhew) 정회원
1997년 2월: 아주대학교 컴퓨터학부(공학박사)
1997년 3월~1998년 3월: George Mason University 교환교수
1981년 3월~현재: 숭실대학교 정보과학대학 컴퓨터학부 교수
1998년 3월~2001년 2월: 숭실대학교 정보과학대학원 원장
1998년 3월~2005년 2월: 숭실대학교 전자계산원 원장
〈관심분야〉 소프트웨어 유지보수/재사용, 소프트웨어 재공학/역공학, 정보보호 등