디지털 포렌식 기술 및 동향

Technologies and Trends of Digital Forensics

u-IT839의 정보보호 이슈 특집

정익래 (I.R. Jeong) 암호기술연구팀 선임연구원
홍도원 (D.W. Hong) 암호기술연구팀 팀장
정교일 (K.I. Chung) 정보보호기반그룹 그룹장

목 차

디지털 포렌식은 정보기기에 내장된 디지털 자료를 근거로 삼아 그 정보기기를 매개체로 하여 발생한 어떤 행위의 사실 관계를 규명하고 증명하는 신규 보안서비스 분야이다. 본 고에서는 디지털 포렌식 수행절차인 증거 수집, 증거 분석, 증거 제출 과정에서 필요한 기술들과 문제점들에 대해서 살펴본다. 또한 현재까지 개발된 포렌식 툴들의 현황과 활용 분야 및 발전 방향에 대해서도 살펴본다.

I. 디지털 포렌식

IT 기술의 발전 및 급격한 정보화 사회로의 변화는 정보의 디지털화를 가속시켜서 컴퓨터 관련 범죄뿐만 아니라 일반 범죄에서도 중요 증거 또는 단서가 컴퓨터를 포함한 디지털 정보기기 내에 보관하는 경우가 증가함에 따라, 증거 수집 및 분석을 위한 전문적인 디지털 포렌식 기술이 요구된다.

디지털 포렌식은 정보기기에 내장된 디지털 자료를 근거로 삼아 그 정보기기를 매개체로 하여 발생한 어떤 행위의 사실 관계를 규명하고 증명하는 신규 보안서비스 분야이다.

디지털 포렌식은 검찰, 경찰 등의 국가 수사기관에서 범죄 수사에 활용되며, 일반 기업체 및 금융회사 등의 민간분야에서도 디지털 포렌식 기술의 필요성이 증가하고 있다. 예로써, 포렌식 기술은 보험사기 및 인터넷 뱅킹 피해보상에 대한 법적 증거 자료 수집 및 내부 정보 유출 방지, 회계 감사 등의 내부 보안 강화에 활용 가능하다.

디지털 포렌식은 크게 증거 수집, 증거 분석, 증거 제출과 같은 절차로 이루어진다[1],[2].

- 증거 수집: 손상되기 쉽고, 사라지기 쉬운 디지털 증거가 저장된 저장매체(컴퓨터 메모리, 하드디스크, USB 등)에서 데이터의 무결성을 보장하면서 데이터를 읽어 내야 한다. 이 때 무결성이란 원 저장매체에 대한 데이터 변조가 일어나지 않음을 의미한다. 증거 수집에서 유용한 기술로는 무결성을 보장하는 이미징 기술 등이 있다.
- 증거 분석: 증거 수집으로 얻은 데이터로부터 유용한 정보를 이끌어 내야 한다. 유용한 정보는 보통 저장 매체에 존재하는 파일 시스템의 내부나 외부에 존재할 수 있다. 예를 들면, 범죄자는 저장매체에 존재하는 NTFS와 같은 파일 시스템 내부나, NTFS에서 사용하지 않는 저장매체 구역에 중요 정보를 숨길 수 있다. 증거 분석에서 유용한 기술로는 삭제된 파일 복구 기술이나 암호화된 파일 해독 및 문자열 검색 기술 등을 들 수 있다.
- 증거 제출: 입수된 디지털 증거가 법적 증거로 채택되기 위해서는 증거자료의 신뢰성이 확보되어야 한다. 이를 위해 법률적으로 디지털 포렌식에 대한 표준 절차뿐만 아니라 포렌식 툴에 대한 검증 절차 또한 이루어져야 한다.

디지털 포렌식의 종류는 다음과 같이 나누어 볼 수 있다.

- 컴퓨터 포렌식: Windows나 Unix와 같은 운영체제를 탑재한 범용 컴퓨터를 대상으로 하는 디지털 포렌식을 말한다.
- 임베디드(모바일) 포렌식: 핸드폰과 같은 모바일 기기나 디지털 카메라, 캠코더, PDA와 같은 다양한 디바이스에 대한 디지털 포렌식을 말한다.
- 네트워크 포렌식: 컴퓨터나 핸드폰과 같은 통신 디바이스를 사용해서 통신이 이루어지는 경우에, 이런 통신 디바이스에서 네트워크 정보, 사용자 로그, 인터넷 사용 기록 등과 같은 정보를 수집 및 분석하는 포렌식을 말한다.

● 용 어 해 설 ●

디지털 포렌식: 정보기기에 내장된 디지털 자료를 근거로 삼아 그 정보기기를 매개체로 하여 발생한 어떤 행위의 사실 관계를 규명하고 증명하는 것을 말한다.

무결성: 디지털 증거의 무결성이란 증거 자료의 신뢰성을 확보하기 위해서 수집된 디지털 데이터가 변조 및 손상되지 않았음을 보장하는 것을 말한다.

Ⅱ. 증거 수집

보통의 디지털 기기는 운영체제가 탑재되어 있으며, 운영체제는 휘발성 저장매체와 비휘발성 저장매체를 사용한다. 휘발성 저장매체란 DRAM과 같이 컴퓨터 운영체제가 종료되면 더 이상 데이터를 복구할 수 없는 저장매체를 말하며, 비휘발성 저장매체란 EEPROM, 플래시메모리, 하드디스크와 같이 운영체제가 종료된 이후에도 데이터를 복구할 수 있는 저장매체를 말한다. 디지털 포렌식에서 증거 수집은

대상 매체의 운영체제 종료 여부에 따라서 다음과 같이 나눌 수 있다.

- 데드 시스템상에서의 증거 수집: 운영체제가 종료된 컴퓨터나 핸드폰 같은 기기에 대한 증거 수집을 말하며, 주로 하드디스크나 플래시 메모리로부터 데이터를 얻는 것으로 이루어진다.
- 라이브 시스템상에서의 증거수집: 운영체제가 종료되지 않은 컴퓨터나 핸드폰 같은 기기에 대한 증거 수집을 말한다. 하드디스크와 같은 비휘발성 매체뿐만 아니라 컴퓨터 메모리와 같은 휘발성 저장매체로부터 데이터를 얻는 것으로 이루어진다.

라이브 시스템상에서의 디지털 포렌식을 수행하기 위해서는 운영체제가 사용중인 휘발성 메모리와 하드디스크를 접근할 필요가 있다. 하지만 Windows와 같은 운영체제에서는 중요한 메모리 영역이나 하드디스크상의 파일에 대한 사용자 프로그램의 접근을 막고 있다. 따라서 라이브 시스템상에서의 포렌식을 위해서는 운영체제의 보호기능을 우회할 수 있는 기술이 필요하다.

1. 데드 시스템상에서의 증거 수집

데드 시스템상에서의 증거 수집 기술은 포렌식 대상 기기에 따라 달라지게 된다. 핸드폰 기기와 같이 비휘발성 저장매체를 분리 및 접근하기가 용이하지 않은 기기는 상대적으로 컴퓨터 하드디스크와 같이 쉽게 저장매체를 분리 및 접근할 수 있는 기기보다 데이터 획득이 어렵다. 데이터를 쉽게 획득할 수 있는 컴퓨터 하드디스크에서도 원본 데이터의 이미지를 만들게 되는데, 이는 나중에 증거 분석을 할 경우에 원본데이터가 변경되는 것을 막기 위해서이다. 따라서 본 저장매체에 있는 데이터의 무결성을 보장할 수 있는 이미징 기술이 필요하다.

2. 라이브 시스템상에서의 증거 수집

운영체제가 종료되지 않은 시스템에서의 데이터 획득 순서는 휘발성 저장매체에 있는 데이터들을 먼저 획득한 후에, 비휘발성 저장매체에 있는 데이터들을 획득하는 순서로 이루어진다. 포렌식 대상이 되는 라이브 시스템에서 휘발성 저장매체나 비휘발성 저장 매체에서 데이터를 획득하기 위해서는 라이브 시스템 운영체제에 있는 명령어들을 사용하기 보다는 포렌식 툴을 사용해서 데이터를 획득해야 하는데, 그 이유는 다음과 같다.

첫째, 대상 시스템의 운영체제 명령들이 공격자에 의해서 이미 바뀌어 있어서 그 명령을 사용할 경우 사건 증거들을 삭제할 가능성이 있기 때문이다.

둘째, 운영체제 명령어들이 바뀌지 않았다 하더라도, 정상적인 운영체제 명령의 실행이 시스템 정보를 변경할 가능성이 있기 때문이다. 예를 들어 보면, Windows 운영체제에서 단순히 탐색기 창을 여는 것만으로 여러 파일들의 마지막 접근 시간(accessed time)이 변경되게 된다. 파일과 관련해서 여러 시간 정보가 존재하는데 마지막 접근시간 이외에 마지막 수정시간(modified time), 생성시간(created time)이 존재한다. 이런 MAC(Modified, Accessed, Created) 시간은 사건을 조사하는 데 있어서 중요한 요소이므로 절대 변경되어서는 안된다.

셋째, 운영체제는 시스템 보호를 위해서 일부 데이터나 파일들에 대해 사용자들의 접근을 막고 있다. 즉, 운영체제에서 제공하는 명령어들에 의해서는 접근할 수 없는 데이터나 파일들이 존재한다. 따라서 운영체제의 보호 메커니즘을 우회할 수 있는 포렌식 툴의 사용이 필요하다.

라이브 시스템에서 휘발성 데이터의 획득과 비휘발성 데이터를 획득하는 데는 다양한 어려움들이 존재하며, 이는 라이브 시스템 운영체제에 따라 달라지게 된다. 다음 두 하위 절에서는 Windows나 Unix 운영체제를 사용하는 라이브 시스템에서 디지털 포렌식이 이루어지는 방식과 문제점에 대해서 설명한다.

가. 라이브 시스템 메모리 덤프

Windows나 Unix 운영체제를 사용하는 시스템

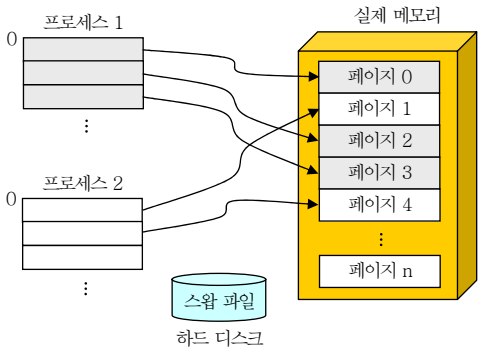
에서는 애플리케이션에 따라서 사용자의 ID나 패스워드가 휘발성 저장매체인 컴퓨터 메모리에 올라와 있을 수 있다. 때문에 메모리상의 데이터를 모두 얻을 수 있다면 이런 메모리 데이터로부터 중요한 정보를 획득할 수가 있다.

Windows나 Unix에서는 물리적 메모리의 한계를 극복하기 위하여 (그림 1)과 같은 가상 메모리(virtual memory)를 사용하고 있다[3],[4].

예를 들어서, 컴퓨터의 실제 물리적 메모리는 256M 바이트라도 가상 메모리를 사용하면 각각의 프로세스는 혼자서 4G 바이트의 메모리를 사용한다고 느낄 수 있다. 이런 가상 메모리 관리를 해주는 모듈을 가상 메모리 매니저(virtual memory manager)라고 한다. 가상 메모리 매니저는 여러 프로세스의 수행을 위해서 물리적 메모리를 페이지라는 단위로 나누고 프로세스 실행시 필요한 프로세스들의 일부분을 물리적 메모리로 올려서 실행시키며, 나머지 부분은 하드디스크에 존재하는 스왑파일(swap file)에 저장시킨다.

가상메모리를 사용하는 시스템에서 하나의 사용자 프로세스가 사용하는 가상 메모리를 모두 덤프하는 것은 가능하며, 덤프하기 위해서는 물리적 메모리에 있는 프로세스에 할당된 페이지뿐만 아니라 하드디스크에 있는 스왑파일 내용 일부까지 덤프해야 함을 알 수 있다.

물리적 메모리의 일부분은 운영체제에 의해서 보호되고 있기 때문에 컴퓨터 부팅 전에 특별한 셋팅을 미리 해놓지 않는 한 물리적 메모리의 전부를 덤프하는 방법은 현재까지 알려져 있지 않다.



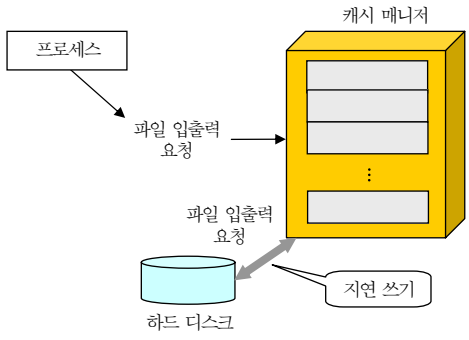
(그림 1) 가상 메모리 구조

나. 라이브 시스템 하드디스크 이미징

애플리케이션이나 운영체제는 중요한 데이터를 저장하기 위해서 임시 파일을 하드디스크에 만들어 사용하다가 시스템 종료 시에 삭제하는 경우가 있다. 따라서 컴퓨터를 끄기 전에 하드디스크를 이미징한다면 이런 중요한 데이터를 얻을 수 있을 것이다. 하지만 만약 운영체제가 캐시(cache)를 사용한다면, 하드디스크만을 이미징하는 것은 데이터의 일관성(consistency)에 문제가 생길 수 있다.

파일에 대한 입출력을 좀 더 효율적으로 하기 위해서 운영체제는 (그림 2)와 같은 캐시를 사용한다[3],[4]. 예를 들어, 프로세스가 어떤 파일의 한 바이트를 읽어오라는 명령을 하면, 운영체제는 연속된 256K 바이트를 한꺼번에 읽어와서 캐시 메모리에 저장하고 프로세스에게는 한 바이트만을 돌려준다. 프로세스가 읽어들인 한 바이트 옆의 또 한 바이트를 읽어오라고 명령하면, 운영체제는 이번에는 하드디스크에 접근할 필요 없이 캐시 메모리에 저장된 한 바이트를 돌려주면 된다. 하드디스크 접근 시간보다 캐시 메모리 접근 시간이 훨씬 빠르기 때문에 캐시를 사용해서 하드디스크 접근을 줄이면 시스템의 효율성이 더욱 증가하게 된다. 캐시 메모리 관리를 하는 모듈을 캐시 매니저(cache manager)라고 한다.

파일을 읽어들이는 경우에는 일관성에 아무 문제가 발생하지 않는다. 일관성에 문제가 발생하는 경



(그림 2) 캐시 구조

우는 캐시 매니저가 시스템 효율성을 높이기 위해서 지연쓰기(delayed write)를 하는 경우이다. 지연쓰기란 프로세스가 읽어들인 바이트를 수정해서 다시 파일로 쓰도록 명령을 내릴 경우에 캐시 매니저가 캐시 메모리 데이터를 먼저 수정하고 시간이 흐른 후에 하드디스크 파일에 수정된 데이터를 쓰는 것을 말한다. 지연쓰기 역시 하드디스크 접근을 줄이고 시스템 효율성을 높이기 위해서 사용된다. 지연쓰기를 하는 시스템에서 하드디스크 이미징을 하는 경우에 아직 완전히 수정이 안된 파일이 존재할 수 있으므로 일관성에 문제가 발생할 수 있다. 그러므로 하드디스크 이미징을 하기 전에 캐시 메모리에 존재하는 데이터 중에 아직 하드디스크에 기록이 안된 데이터를 하드디스크에 쓰도록 하는 기법이 필요하다.

Ⅲ. 증거 분석

증거 수집에서 얻어진 데이터들로부터 유용한 정보를 얻는 것을 증거 분석이라고 한다. 유용한 정보는 사건에 따라 다르겠지만 일반적으로 다음과 같은 증거 분석 기술들이 사용될 수 있다.

1. 덤프 메모리 분석

프로세스가 사용중인 가상 메모리의 덤프를 획득했을 경우에 사용자 ID나 패스워드와 같은 유용한 정보가 가상 메모리에 남아 있을 수 있다. 프로세스를 위한 가상 메모리는 보통 코드 영역, 데이터 영역, 스택 영역 등으로 나뉘어지며, 데이터 영역이나 스택 영역이 프로세스에서 필요한 여러 정보를 저장하고 있으므로 포렌식 툴은 프로세스가 가상 메모리를 어떻게 사용하는지를 분석할 수 있어야 한다.

2. Windows 레지스트리 분석

Windows는 레지스트리(registry)에 프로그램이나 시스템에 관한 다양한 정보를 저장하고 있으므로 포렌식 툴은 이를 분석할 수 있어야 한다. 레지스트리 Hive 파일들은 [SystemRoot]₩System32₩Config 폴더에 위치하며, regedit와 같은 명령으로 살펴볼 수 있다. 레지스트리 Hive 파일들 중 SAM 파일은 패스워드들의 해시 정보를 가지고 있으며, 운영체제에 의해서 암호화되어 보호되고 있다. 포렌식 툴은 SAM 파일의 패스워드들을 복구할 수 있어야 한다.

3. Timeline 분석

파일 시스템들은 각각의 파일들이 만들어진 시간 정보와 마지막으로 접근된 시간 정보 그리고 마지막으로 수정된 시간 정보들을 가지고 있다. 포렌식 툴이 이런 시간 정보를 가지고서 시간의 흐름에 따라 어떤 파일들이 생성되고 접근되었는지를 알기 쉽게 보여줄 수 있다면 증거 분석을 좀 더 수월하게 할 수 있다. 또한 NTFS 파일 시스템에서는 $LogFile과 $UsrJrnl이라는 시스템 파일이 존재하며, 파일 시스템에 대한 사용 로그를 남기고 있으므로 이로부터 좀 더 많은 정보를 얻을 수 있다.

4. 삭제된 파일 복구

하나의 파일은 여러 클러스터들의 리스트로 이루어져 있으며, 이런 리스트 정보가 파일 시스템에 들어 있다. 일반적으로 하나의 파일을 삭제할 경우에 파일시스템은 클러스터들에 들어 있는 파일 내용을 지우는 것이 아니라 파일에 할당된 클러스터들을 프리시키는 것으로 파일을 지운다. 따라서 프리된 클러스터들이 다른 파일에 할당되지 않는 한 삭제된 파일을 복구할 가능성이 있다.

비록 삭제된 파일을 복구할 수 없을지라도 파일이 존재했다라는 사실이 사건에 중요한 단서가 될 수 있다. 어떤 파일이 존재했었는지의 존재 여부는 다양한 방법으로 이루어질 수 있다. 예를 들어서, Thumbs.db라는 숨김 속성 파일은 디렉토리에 이미지 파일이 있을 경우에 Windows에서 이미지 파일의 정보 값을 파악하기 용이하도록 자동으로 생성하는 데이터 베이스 파일이며, 이미지들을 모두 삭제해도

이 데이터베이스 파일이 남아 있을 수 있으므로 삭제된 이미지 파일의 존재여부를 증명해 줄 수 있다.

5. 비정상적인 파일 찾기

사용자가 중요한 데이터를 숨길 경우에 Windows에서 파일을 숨김 속성으로 놓거나 파일 확장자를 바꾸어서 데이터를 숨기려 할 수 있다. 따라서 포렌식 툴이 숨김 속성을 가진 파일들이나 파일 확장자가 바뀐 파일들을 따로 찾아줄 수 있다면 분석에 많은 도움이 될 수 있다.

보통 하나의 파일 형식은 하나의 파일 확장자를 가지며 또한 하나의 식별자(identifier)라 불리는 유일한 값을 가진다. 이 식별자는 파일 생성시 헤더에 자동으로 저장된다. 따라서 확장자를 바꿀 경우에는 파일 확장자와 이 식별자가 맞지 않으므로 확장자가 바뀐 파일들을 찾을 수 있다.

6. 이메일 분석

파일 시스템에서 삭제된 파일을 복구하는 것과 비슷하게 삭제된 이메일을 복구할 수 있다. 하나의 이메일을 삭제할 경우에 이메일 프로그램은 메일박스에 있는 이메일의 내용을 지우는 것이 아니라 이메일의 헤더 값을 바꾸어서 이메일을 삭제하게 된다. 따라서 삭제된 이메일을 복구할 가능성이 있다.

7. 로그 분석

어떤 장치나 응용 프로그램을 사용하게 되면, 운영체제나 응용 프로그램이 로그를 남기는 경우가 있으며 이런 로그는 사건 분석에 중요한 정보가 될 수 있다. 중요한 로그들로는 다음과 같은 것들이 있다.

- 파일 시스템 로그: NTFS 파일 시스템의 경우 $LogFile, $UsrJrnl에 파일 생성, 접근 등에 관한 로그가 남아 있다[5].
- USB 사용 로그: USB 포트에 연결했던 USB들의 사용로그가 레지스트리에 남아 있다.
- 인터넷 사용: 임시파일, 쿠키, 즐겨찾기, ActiveX 등으로부터 인터넷 사용 행적을 조사할 수 있다.

8. 슬랙 공간 분석

파일 시스템은 하나의 큰 파일을 저장할 때 여러 클러스터들로 나누어 저장하게 된다. 이 때 가장 마지막 클러스터에는 파일의 가장 뒷부분을 저장한 다음 남게 되는 공간이 생길 수 있는데 이런 공간을 파일 슬랙 공간(slack space)이라고 한다. 예를 들면, 클러스터의 크기가 8K이고 파일의 크기가 1K이면, 7K의 사용하지 않는 파일 슬랙 공간이 생기게 된다. 이외에도 하드디스크에는 할당되지 않은 공간들과 볼륨 슬랙 공간, 파티션 슬랙 공간 등이 있다. 사용자들이 이런 슬랙 공간에 데이터를 숨겨 놓을 수 있기 때문에 포렌식 툴은 이런 슬랙 공간의 데이터를 분석할 수 있는 기능을 가져야 한다.

9. 스트링 서치

디지털 증거 분석시 수사에 필요한 정보가 어떤 파일에 어떤 형태로 저장되어 있는지 모르는 경우가 많기 때문에 모든 파일들을 대상으로 키워드를 가지고 검색을 반복해야 하는 경우가 많다. 이러한 검색은 대용량의 저장 매체일 경우 상당한 시간이 소요되므로 검색범위를 축소하는 기술이 필요하게 된다. 조사 대상의 검색범위를 축소하기 위해서는 잘 알려진 파일은 검색 대상에서 제외하고, 주목해서 검색할 대상을 선정하여 검색 범위를 축소하고, 조사우선순위를 부여해야 한다. 이러한 기능을 제공하는 검색 기술 중 하나가 해시 검색(hashed search)으로써, 준비된 참조 데이터 셋(RDS)을 사용해서 널리 알려진 파일은 조사 분석 대상에서 제외시킨다. 미국에서는 이러한 해시 검색 기술을 활성화하고, 일반 수사관들도 쉽게 사용할 수 있게 하기 위해서 잘 알려진 파일들의 표준 해시 DB를 NIST에서 제작하여 무상으로 배포하는 NSRL 프로젝트를 실시하고 있다[6]. NSRL 프로젝트 목적은 “범죄에 사용되는 컴퓨터 파일의 식별 자동화”, “증거에 포함된 파일 조사를 효율적으로 지원”이며, 이를 위해 약 수 년

간 각종 소프트웨어 및 알려진 파일을 수집하고 이에 대한 정보와 해시 값을 DB화하여 RDS를 구축했다. 국가 주도의 해시 검색 기술의 인프라를 구축하자, 미국 내의 컴퓨터 포렌식 관련 산업계가 이를 적극 활용하고 현장에 적용하고 있다.

Ⅳ. 증거 제출

1. 무결성

디지털 증거 무결성 확보기술은 증거 자료의 신뢰성을 확보하기 위해서, 수집된 데이터가 변조 및 손상되지 않았음을 해시 및 오류 검증 알고리듬을 이용하여 증명하는 기술이다. 이는 입수된 디지털 증거가 법적 증거로 채택되기 위해서 반드시 필요하며, 이를 위해 법적으로 제정된 디지털 포렌식 표준 절차 및 기술이 필요하다.

2. 포렌식 툴 검증

디지털 포렌식에 사용되는 포렌식 툴에 대한 인증이 없이는 포렌식 툴에 의해서 얻어진 분석 결과를 믿을 수 없는 것은 자명하다. 디지털 포렌식 툴의 검증을 위해서 미국에서는 미국 국립표준기술연구소(NIST)에서 디지털 포렌식 툴 검증(CFTT)을 시행하고 있다[7]. CFTT에서는 디지털 포렌식 툴의 검증 및 평가 방안을 제시하고, 평가 결과 보고서는 미국의 국가 법무연구소(NIS)와 함께 공동으로 발간하여 일반인들도 쉽게 열람할 수 있도록 하고 있다. 컴퓨터 범죄 수사관들은 이 보고서를 참조하여 디지털 포렌식 툴의 선정 기준을 확립하며, 변호사와 검사들은 디지털 증거의 객관성을 증명하기 위한 자료로 활용하고 있다.

Ⅴ. 포렌식 시스템 및 툴 현황

현재 상용화되어 있는 컴퓨터 포렌식 증거 수집 및 분석 소프트웨어는 Guidance Software사의 EnCase와 AccessData사의 ForensicToolkit이 가장 널리 사용되고 있다. Paraben사는 모바일 기기에 대한 전문 분석가들을 위해서 Cell Siezure, PDA Siezure 등의 소프트웨어와 각종 휴대용 기기와의 연결을 지원하는 툴박스 형태의 상용 제품을 제공하고 있으며, 메모리를 직접 분석할 수 있는 소프트웨어도 개발하여 제공한다. 포렌식 툴의 현황은 <표 1>과 같다.

〈표 1〉 컴퓨터 포렌식 기술 연구 현황

도구 이름	지원 운영체제	공개 여부	이미지 생성 및 검사	무결성 검사	저수준 복구	기타 지원 기능
ForensicX	Unix/Linux	Com	Disk, OS, Traffic	Hard, File, Finger	Delete	Plug, Report
MaresWare	Windows	Com	Disk	Hard, File		
	Linux	Com	Disk	File		
The Coroner's Toolkit	Unix/Linux	Free	Disk	Hard	Delete, Key	
Tom's Rootboot	Linux	Free	Disk, OS			Boot
EnCase	Windows	Com	Disk, OS	Hard, File, Finger	Raw, Delete	Plug, Report
Byte Back III	Windows	Com	Disk, OS, Traffic	Hard, File	Raw, Delete	
ForensicToolkit	Windows	Com	Disk	Hard, File	Raw, Delete	Report

주 1) 공개여부: Com(상용), Free(공개용)
2) 이미지 생성 및 검사: Disk(디스크 이미지), OS(운영체제 이미지), Traffic(IP 트래픽 이미지)
3) 무결성 검사: Hard(하드웨어 변동 검사), File(파일 무결성 검사), Finger(전자 지문 검사)
4) 저수준 복구: Raw(저수준 파일 편집), Delete(삭제 파일 복구), Key(암호키 복구)
5) 기타 지원 기능: Boot(긴급 부팅 지원), Plug(플러그인 지원), Report(자동보고 지원)

Ⅵ. 포렌식 활용 분야

1. 수사기관

검찰, 경찰, 국정원, 기무사 등에서는 스파이, 기술 유출, 공갈, 사기, 위조, 해킹, 사이버 테러와 같은 컴퓨터 범죄 수사 분야에 활용하고 있다.

2. 기업체

회사 정보 및 기술 유출은 모든 종류의 기업체에서 발생하고 있으며, 이로 인하여 측량하기 어려운 손해가 발생하고 있다. 따라서 증권, 보험, 은행 등의 금융회사를 포함한 일반회사에서도 금융사고, 회계감사 및 정보유출 등의 보안사고 발생시 민·형사상 책임소재를 가리기 위한 증거자료 확보를 위해 컴퓨터 및 모바일 포렌식 기술을 활용할 수 있다.

미국에서는 2007년 말부터 증거공개명령제도를 시행할 예정이며, 미국에서 경영활동을 하는 글로벌 기업들은 국내법처럼 준수해야 한다.

3. e-discovery

미국에서는 디지털 증거에 대한 제출을 정당화하는 e-discovery 제도가 시행되어 2006년 12월에 통과되었다.

따라서 민·형사 분쟁 발생시 방대한 양의 디지털 자료로부터 분쟁에 필요한 자료를 효율적으로 추출하는 포렌식 툴이 필요하다.

Ⅶ. 포렌식 발전 방향

디지털 포렌식이 유용하게 사용되기 위해서 해결되어야 하는 문제점들은 다음과 같다.

첫째, 포렌식에 의해 얻어진 디지털 증거를 법적 증거로 채택하기 위해서는 형사소송법상의 증거문제 해결이 필요하며, 포렌식 절차에 관한 표준화가 필요하다. 또한 여러 포렌식 툴간의 호환성을 위해서는 포렌식 이미지의 포맷에 대한 표준화가 필요하다.

둘째, 포렌식 대상이 되는 디지털 데이터는 점점 대용량화되어 가고 있으므로 포렌식 이미징을 만드는 작업이나 검색작업 등이 고속화되어야 하며 이에 대한 연구가 필요하다.

셋째, 외산 포렌식 장비를 사용할 경우 국산 디지털 파일들을 분석하는 기능이 부족할 수 있으며, 국내 사법 제도 등 국내 환경에 대한 특성이 반영되지 않아 수사상 어려움이 존재할 수 있으므로 국내 실정에 맞는 포렌식 장비의 개발이 필요하다.

넷째, PDA, 핸드폰, 디지털 카메라, 캠코더 등 다양한 형태의 디바이스가 새로 나오고 사용되고 있으므로 다양한 형태의 디바이스에 대한 포렌식 툴 개발이 함께 진행되어야 한다.

위에서 언급한 여러 문제점들이 해결되면 디지털 포렌식은 현재 활용되고 있는 분야 이외에도, 디지털 증거의 인증 서비스와 같은 새로운 보안 서비스 시장을 창출하거나 활성화 시킬 수 있으리라 예측된다.

약 어 정 리

CFTT	Computer Forensic Tool Testing
e-Discovery	Electronic Discovery
NIST	National Institute of Standards and Technology
NSRL	National Software Reference Library
RDS	Reference Data Set

참 고 문 헌

[1] 이형우, 이상진, 임종인, “컴퓨터 포렌식스 기술,” 정보보호학회지, 2002. 10.

[2] 김종섭, 김귀남, “국내 Computer Forensics의 연구동향과 발전방향,” 정보보증논문지, 2003. 3.

[3] 정덕영, “Windows 구조와 원리,” 한빛미디어, 2006. 3.

[4] David A. Solomon and Mark E. Russinovich, “Inside Windows 2000,” Microsoft Press, 2000. 9.

[5] Brian Carrier, “File System Forensic Analysis,” Addison-Wesley, 2005. 3.

[6] National Software Reference Library (NSRL), http://www.nsrl.nist.gov.

[7] Computer Forensics Tool Testing (CFTT) Project, http://www.cftt.nist.gov.