情 報 保 護 學 會 誌
第 13 券 第 4 號, 2003. 8

컴퓨터 포렌식스: 시스템 포렌식스 동향과 기술*

황 현 욱**, 김 민 수**, 노 봉 남**, 임 재 명***

요 약

컴퓨터와 인터넷의 발전은 사용자에게 편리함을 가져다 주었으나, 컴퓨터 범죄라는 새로운 역기능을 낳게 되었다. 결국 이는 특정 목적을 가진 범죄자를 낳게 되고 정보화 사회의 발전을 저해하는 커다란 걸림돌로 작용하게 되었으며, 이에 대응하는 정보보호기술은 개인의 사생활 보호와 국가 경쟁력을 판단하는 척도로 자리잡게 되었고, 현대에는 정보보호 기술 자체가 국가간 정보전 형태를 띄면서 그 중요성은 매우 커지고 있다. 이러한 정보보호 기술은 방화벽과 침입탐지 시스템의 꾸준한 개발로 이어졌으나, 아직 컴퓨터 범죄를 다루는 피해 시스템의 증거수집, 복구 및 분석을 하는 컴퓨터 포렌식 기술은 국내에서는 아직 활발히 연구되지는 않고 있다. 본 연구에서는 컴퓨터 포렌식스에서 시스템 포렌식스에 대한 개념과 절차, 국외 기술 동향과 개발된 도구들을 살펴보고, 유닉스/리눅스에서의 백업과 복구 그리고 분석의 관점에서 포렌식 기술에 대해 살펴본다.

Ⅰ. 서 론

정보보호 기술은 기술 자체가 국가의 경쟁력이 되어가고 있다. 외국의 경우는 오래전부터 국가 정책으로 추진되어질 만큼 체계적인 노력을 하고 있다. 그중 사이버테러에 대비하는 정보전 대응체계는 중요한 이슈로 떠오르고 있다. 그중 컴퓨터 해킹 자체는 일반적인 호기심에서부터 특수한 목적을 가진 하나의 범죄형태로 흘러가고 있으며, 해킹의 기술이 발전할수록 보안 기술도 발전해 왔으며 이러한 해킹 행위를 막기 위해 현재까지 수많은 보안 제품이 개발되고 적용되었다. 방화벽 시스템과 침입탐지 시스템은 그 대표적인 보안제품으로 공격자들의 행위를 방어하고 탐지하고 있다. 하지만 알려지지 않은 행위나 새로운 공격기법에 대해서 방어하기에는 매우 역부족이고, 해킹을 당한 대부분의 시스템 관리자들은 자신이 해킹을 당했는지 조차 모르며 또한 체계적인 대응방안을 세우는 데에도 익숙하지 못하다. 공격자의 침입 방법, 침입 후 행위, 침입 흔적, 피해 복구에 대한 전문적인 기술 개발이 체계적인 연구로 이어지지 못하고 있다.

이러한 침해사고 후 복구, 분석을 다루는 컴퓨터 포렌식스(Computer Forensics) 분야는 외국에서는 국가적 전략 기술로 인정받으며 상당한 발전을 거듭해왔지만 국내에는 근래에 들어서야 논의되기 시작하였고, 피해 시스템을 분석하고 복구하는 포렌식 기술의 일반적인 인식 자체가 공익성을 띠고 있다는 생각에 업체에서 제품으로서의 투자를 많이 고려하고 있지 않는 상황이다. 하지만 선진국가에서는 포렌식 기술을 일정한 교육과정과 국가에 필요한 요소라 판단하고 기술개발에 힘을 쏟고 있다.

해킹 피해 시스템의 분석은 해킹 기술이 점점 발전할수록 어려워지고 있다. 전문적인 공격자들은 자신들의 흔적을 남기지 않는 치밀함을 보이고, 피해 시스템이 해킹이 당한 사실 조차 모르도록 완벽한 조치를 취하고 있기 때문에 증거 수집에 있어 매우 어려운 것이 현실이다. 또한 약간의 지식만 가지고도 자동화된 해킹 도구를 사용하여 증거를 삭제하는 것이 일반화 되어가는 것이 추세이다. 따라서 해킹 피해 시스템 분석에 있어서는 전문가의 직관적인 능

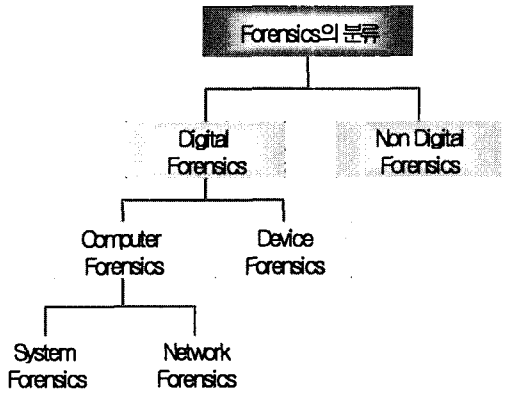
* 본 연구는 한국정보보호진흥원의 기술 용역 연구 결과로 수행되었습니다.
** 전남대학교 일반대학원 정보보호협동과정
*** 한국정보보호진흥원 네트워크모니터링 팀장

력이 매우 중요하며, 이런 전문가 지식이 포함 되어 있는 포렌식 기술이 도구로서 개발되어야 한다.

II. 포렌식스의 개념

1.1 포렌식스의 분류

"포렌식스(Forensics)"의 사전적 의미는 민사 또는 형사 재판에서 과학적이고 기술적인 방법을 사용하여 사건을 조사하고 어떠한 사실을 증명하는 일련의 방법들을 말한다[1]. 용어 자체는 모든 대상에 대하여 광범위한 성격을 띠고 있다. 포렌식스 분야는 [그림 1]과 같이 분류할 수 있다[2,3].



[그림 1] 포렌식스의 분류

포렌식스 분야는 크게 디지털 포렌식스와 비디지털 포렌식스 분야로 나누어진다. 디지털 포렌식스는 계산기부터 데스크탑 컴퓨터에 이르기까지 디지털 기술을 사용한 모든 장치를 포함하며, 이는 또 컴퓨터 포렌식스 분야와 장치 포렌식스 분야로 나뉜다. 장치 포렌식스 분야는 컴퓨터 장치를 제외한, 저장매체를 가지고 있는 디지털 카메라, Palm OS, 개인 저장 장치 등 모든 것을 포함하며, 컴퓨터 포렌식스 분야는 범위에 따라 네트워크 포렌식스와 시스템 포렌식스로 나누어지며 그 특성은 다음과 같다.

일반적으로 네트워크를 대상으로 넓은 범위의 네트워크 포렌식스의 특징과, 시스템 자체를 대상으로 하는 시스템 포렌식스의 특징은 다음과 같다[4].

1.2 네트워크 포렌식스

- 고난도 기술이 필요한 분야로 범위가 매우 넓음
- 광범위하고 복잡한 네트워크 상에 흩어져 있는 디지털 증거를 찾아냄
- 침입탐지 시스템과 같이 모든 정보를 실시간 탐지할 수 있는 시스템이 필요
- 데이터를 저장하기 위해 매우 큰 저장장치 필요
- 이러한 증거는 매우 일시적(transient)일 수도 있으며, 저장매체에 저장되지 않기도 함
- 상용 도구간의 연동은 현재의 네트워크, 분산 환경에서는 적합하지 않으며 표준화된 형식이 필요

1.2 시스템 포렌식스

- 일반적인 컴퓨터 포렌식스 분야라고 함
- 컴퓨터 범죄 순간에 포착된 데이터로부터 증거를 모으는 일련의 작업
- 저장매체 조사, 지워진 파일 복구, 지스러기, 자유영역(slack and free sapce) 검색, 법적대응을 위한 수집된 자료 보존
- 일부 기능을 지닌 여러 가지 도구들이 존재

2. 컴퓨터 포렌식스 특징과 절차

"컴퓨터 포렌식스" 분야는 컴퓨터 데이터에 대한 보존, 증명, 검출, 문서화, 해석방법에 대한 내용들을 다룬다. 범죄학적인 측면에서 증거를 찾아내고 보존하는 일은 아주 중요하다. 만일 피해 시스템에 대한 조사를 목적으로 마구 헤집고 다닌다고 했을 때, 그것은 절대 법적인 증거자료로서 보장을 받지 못하게 된다. 조사자가 '무엇을', '왜' 하게 되었는지는 모든 것을 철저하게 기록해야 한다.

기본이 되는 세 가지 원칙은 다음과 같다.

o 원시데이터에 대한 변경이나 훼손 없이 증거자료의 획득
o 획득한 증거가 원시데이터의 일부라는 것을 인증
o 원시데이터의 변경 없이 분석

위의 원칙을 바탕으로 다음과 같은 컴퓨터 포렌식스를 절차가 행해진다. 컴퓨터 포렌식스 절차는 다음과 같다.

① 준비(Preparation) - 도구나 기술들을 정의하고 준비하는 단계이다. 이 부분에서 포렌식스 절차를 본격적으로 시행하기 앞서 행해야 할 모든 것을 준비한다.
② 접근 전략(Approach strategy) - 포렌식스 절차를 어떤 방식으로 접근해야 할 것 인지를 정

의하고 관련 기술들에 대해 숙지한다. 전략의 목적은 피해 시스템의 손상을 최소하고 관련 증거의 수집을 최대화 할 수 있도록 한다.

③ 보존(Preservation) - 증거물을 고립하고 안전하게 보존하는 과정이다. 보존 절차를 명백히 하기 위해 증거물 보관 사항에 대해 모두 문서화 한다.

④ 수집(Collection) - 표준화되고 법정에서 인정될 수 있는 절차들에 따라 디스크의 물리적인 이미지를 백업하고, 관련 증거를 기록한다.

⑤ 조사(Examination) - 범죄로 의심되는 것에 관련된 모든 증거를 체계적인 탐색을 통해 깊이 있게 분석한다. 이 단계에서는 잠재적으로 숨겨진 증거를 밝혀내는 것에 집중한다. 분석에 관해 자세하게 문서화를 함께 병행해야 한다.

⑥ 분석(Analysis) - 발견된 증거를 토대로 데이터의 조각과 사건의 흐름을 재구성해본다. 조사 단계와 분석 단계에서 많은 시간이 소요된다. 분석 후 전체적인 사건의 흐름을 밝혀낼 수 있다.

⑦ 보고(Presentation & report) - 결론에 대한 설명과 사건에 대한 핵심을 제시한다. 피해자나 법정에 관련 자료를 제출한다.

포렌식스 절차에 있어 또 하나 생각해야 할 부분은 증거물에 대한 인증이다. 증거물의 원본이 손상되지 않았음을 증명하는 것은 법적인 관점[5]에서나 사건에 대한 설득력을 주기 위해 매우 중요한 과정이다. 일반적으로 각각의 파일 또는 하드디스크의 무결성과 시간소인을 증명하는데 있어 전자적 입증 방법들이 사용되는데 대표적인 방법으로는 소프트웨어적으로 계산되는 해쉬 값을 사용하는 방법이 있다. 반드시 데이터를 조사하기에 앞서 반드시 해쉬 값을 생성시켜야 한다. 흔히 MD5 또는 SHA 알고리즘이 많이 사용되며, 데이터에 해쉬 값을 부여하는 'Tripwire' 등의 응용 프로그램도 사용되어 진다. 기술된 컴퓨터 포렌식스 과정은 시스템 포렌식스 과정에 그대로 적용되어질 수 있다.

Ⅲ. 시스템 포렌식 도구

포렌식 도구는 미국과 영국을 중심으로 발전되어 있는 상황이다. 도구들은 하나의 기능만을 가지는 도구도 있으며, 여러 기능을 통합적으로 가지고 있는 도구도 있다. 이들 도구 중 통합적인 기능을 가진 포렌식 도구는 Encase, TCT, TCT를 발전시킨 TCTUtils, Sleuth kit과 GUI 형식을 지원하는 Autospy가 대표적인 도구이다. 각각의 기능들을 살펴보면 다음과 같다.

1. EnCase[6]

EnCase는 현재 가장 강력한 기능을 가진 컴퓨터 포렌식 도구로 알려져 있다. 1980년대부터 개발되어 현재는 버전 4가 개발되어 널리 쓰이고 있다. EnCase는 포렌식스 소프트웨어가 갖추어야 할 증거 보존 및 분석 기능을 모두 갖추고 있으며, 미 연방 법원의 EnCase를 통해 얻은 결과물을 법적인 증거로 채택한 판례로 인해 더욱 성능을 인정받고 있는 도구 이다. EnCase의 주요 기능은 다음과 같다.

- o 증거 자료의 무결성 보장(digital finger printing)
- o 유연한 이미지 추출 방법 제공
- o 사용자 정의 스크립트 작성을 통한 자동화 작업 가능
- o 파일의 정확한 시간대 추적
- o 삭제된 파일, 폴더 및 비할당 클러스터 영역 검색 및 복구
- o 뛰어난 보고 기능

① 디지털 무결성 보장

증거 자료로서의 완벽한 무결성을 보장하기 위해서 증거 파일을 생성하기 위한 최초 단계에서 피해 시스템의 하드 디스크에 대하여 MD5 해쉬 알고리즘을 사용하여 무결성을 보장한다. MD5 외에도 다양한 해쉬 알고리즘을 정의하여 사용할 수 있다.

② 유연한 이미지 추출 방법 제공

- · Fast Bloc을 이용한 비트 단위의 드라이브 간 이미지 생성
- · 병렬 포트를 이용한 이미지 전송
- · 활성화된 네트워크 카드를 통한 네트워크 전송 지원

또한 EnCase는 NTFS, FAT12/16/32 외에도 리눅스의 ext2, 유닉스의 UFS, 그리고 MacOS 파일시스템까지 분석이 가능하다.

③ 사용자 정의 스크립트 작성을 통한 자동화 작업 가능

미리 정의된 스크립트 또는 사용자가 직접 작성한 스크립트를 사용하여 확장된 검색이 가능하다. EScript

라고 하는 스크립트 검색 도구를 사용하여 숨겨진 e-mail, NT Security event log, Ingernet History 등을 수월하게 찾아낼 수 있다. 그리고 윈도우즈 시스템의 경우 각각의 파일 특성은 확장자를 통해 인식하게 되는데 이러한 스크립트를 사용하여 확장자가 잘못된 파일들을 찾아낼 수 있게 된다. 또한 알려진 파일 증거에 대한 정보를 바탕으로 수상한 파일들을 찾아낼 수 있다.

④ 파일의 정확한 시간대 추적

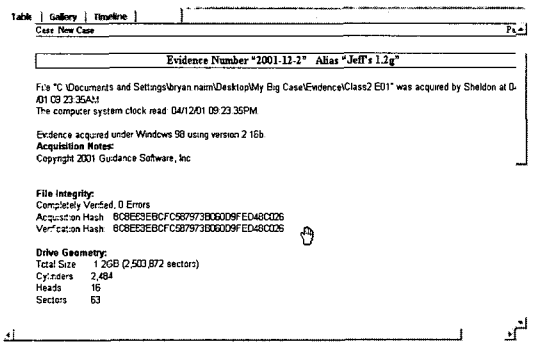
하드 디스크에 남겨진 흔적을 통해 특정 파일에 대해 시간대별로 어떤 작업이 수행되었는지 추적할 수 있다.

⑤ 삭제된 파일, 폴더 및 비할당 클러스터 영역 검색 및 복구

- 유닉스의 grep 과 유사한 기능의 키워드 검색 기능이 있으며, 찾아낸 파일 복구
- 비할당 클러스터 영역 검색 및 복구

⑥ 뛰어난 Report 기능

앞서 언급한 모듈화 된 기능들의 조사결과를 보고서로 작성하는 기능이다. 리포트 형식은 일반 텍스트와 HTML 형식을 지원한다. [그림 2]는 단순히 증거 파일이 생성될 때의 시간 정보와 무결성을 확인하는 해쉬 값 그리고 하드 디스크 지오메트리, 파티션, 매개변수 등의 정보만 기록되어 있지만 수행된 작업에 따라 보고 형식은 순서나 첨부 내용은 달라진다.



[그림 2] EnCase의 Report 화면1

2. TCT(The Coroner's Toolkit)[7]

TCT는 UNIX계열 시스템에서 수행되는 강력한 컴퓨터 포렌식 도구이다. TCT는 침해 사고 발생 당시의 이벤트에 대한 분석을 보다 정확하고 수월하게 수행하기 위한 기능을 제공한다. 피해 시스템의 상태 정보에 대한 스냅샷(snapshot)을 생성하며, 백업된 디스크 이미지 분석 및 파일 복구에도 아주 유용하게 사용할 수 있는 공개용 포렌식 도구이다. 아래의 [표 1]에서는 TCT에 포함된 프로그램에 대한 주요기능을 간략하게 소개하였다.

[표 1] TCT 구성 프로그램의 주요 기능 요약

프로그램	설명
grave -robber	forensic 데이터를 수집하는 프로그램으로, 특히 피해 시스템의 휘발성 정보를 캡쳐 하는데 아주 강력한 기능을 제공한다.
pcat	메모리에 로드 된 프로세스를 표준 출력으로 복사한다.
ils	지정된 디바이스의 모든 아이노드 정보를 출력한다. 기본적으로 지워진 파일에 대한 아이노드 정보를 제공하며, 다양한 옵션을 사용하여 아이노드에 대한 정보를 얻을 수 있다.
icat	주어진 아이노드 번호에 해당하는 파일을 표준출력으로 복사한다.
file	파일 형태를 출력한다.
unrm	데이터 블록을 복사한다. 기본적으로 디스크에서 할당이 해제된 데이터 블록을 복구한다.
lazarus	복구된 데이터 블록을 구조화시킨다.
mactime	파일의 접근, 수정 시간을 조사하고 timeline을 구성하여 출력한다.

TCT는 다양한 OS 환경에서 동작하며, 여러 파일 시스템을 지원한다. 또한 데이터 복구 기능을 지원한다. unrm과 lazarus 라는 도구를 사용하여 데이터 블록 단위의 복구가 가능하다.

① 먼저 지워진 데이터가 위치한 디바이스를 데이터 블록 단위로 복구 및 복사하여 하나의 이미지를 만들어 낸다. 기본적으로는 할당이 해제된 부분에 대한 블록 단위의 복구를 수행한다.

```
$ bin/unrm /dev/hda7 > unrm_output
```

② 그 다음 lazarus를 사용하여 unrm으로 복구된 구조화되지 않은 데이터로부터 지워진 파일 또는 데이터 블록을 복구한다. 덤프 된 큰 파일을 파일단위로 쪼갠다(파일의 크기에 따라 시간이 많이 소요된다.).

```
$ bin/lazarus -h unrm_output
```

lazarus를 사용한 복구 작업이 끝나면 미리 설정

된 디렉토리에 복구된 데이터들이 [block number].[data type].[txt]의 형태로 생성된다.

3. TCTUtils[8]

TCTUtils는 TCT의 기능에 확장된 기능을 제공하는 포렌식 도구이다. TCT에 비해 가장 두드러지는 추가 기능은 기존의 TCT가 파일 아이노드 정보만을 처리하는 반면에 TCTUtils를 사용하여 디렉토리 아이노드에 대한 정보를 수집할 수 있으며, 디스크 이미지에서 파일 구조 정보를 수집할 수 있다는 점들을 들 수 있다. 아래의 표 2는 TCTUtils에 포함된 프로그램에 대한 주요기능을 간략하게 소개하였다.

[표 2] TCTUtils 구성 프로그램의 주요 기능 요약

프로그램	설명
bcat	디스크 블록의 내용을 출력
blockcals	unrm으로 생성된 이미지의 블록넘버를 원래의 디스크 이미지에서의 블록넘버로 변환
fls	directory inode 정보의 파일과 디렉토리 정보를 출력한다. 지워진 파일의 이름도 추출할 수 있다.
find_file	주어진 inode 값을 사용하는 파일 또는 디렉토리를 찾는다.
find_inode	주어진 이미지의 블록 number에 할당된 inode를 찾는다.
istat	지정된 이미지의 inode에 대한 상세 정보를 출력한다.

4. Sleuth Kit[9]

Sleuth Kit은 TCT, TCTUtils 프로그램과 유사한 기능을 갖추고 있지만, 각각의 각 프로그램에 분석을 더욱 용이하게 하는 옵션들이 추가된 공개용 포렌식 도구이다. 몇 가지 프로그램들을 제외하고 Sleuth Kit은 TCT와 TCTUtils의 프로그램들 기능을 그대로 따르고 있다. Sleuth Kit에서는 보다 정확한 시간대 생성을 위하여 피해 시스템과 분석 시스템 간의 시간 차이를 초 단위로 입력이 가능 하고, GMT, EST 등 시간영역 동기화 기능을 지원한다. TCT나 TCTUtils와 비교하여 새로이 제공되는 프로그램과 그 기능은 아래의 [표 3]과 같다.

Sleuth Kit의 가장 두드러지는 기능은 다양한 파일 시스템을 지원한다는 점이다. 기존의 TCT, TCTUtils로 분석 가능한 파일 시스템은 FFS(OpenBSD, Solaris) 또는 EXT2FS(Linux) 뿐이었지만, Sleuth Kit에서는 윈도우즈 계열의 파일시스템 그리고 리눅스의 EXT3 파일시스템도 분석이 가능하다.

[표 3] Sleuth Kit의 주요 프로그램

프로그램	설명
dstat	데이터 블록의 상세정보를 출력
fsstat	주어진 이미지의 파일 시스템 정보를 상세하게 출력
hfind	해쉬데이터 베이스에서 해쉬값 검색
sorter	이미지내의 모든 파일에 대한 파일의 종류를 카테고리 별로 분 류하여 상세하게 출력한다.

5. Autopsy Forensic Browser[10]

Autopsy는 TCT, TCTUtils, Sleuth Kit을 사용하여 피해 시스템 분석을 진행하는데 있어서 작업을 용이하게 하기 위해 개발된 웹 브라우저 기반의 그래픽 기반 프로그램이다. 분석자가 피해 시스템의 디스크 이미지를 파일, 블록 또는 아이노드 수준에서 분석하고 조사할 수 있다. 또한 이미지 내에서 문자열 검색을 위한 편리한 인터페이스를 제공하고 있다.

6. 기타 포렌식 도구

국외에서 연구 개발 된 포렌식 도구를 분석하면 여러 가지 기능의 도구들이 존재한다. 해킹이 발생했는지 여부를 판단하는 도구, 파일을 복구해주는 도구, 침입자를 추적하는 도구, 여러 가지 분석 방법론을 제시하는 도구 등 다양한 모습을 보여준다. 대표적인 다음 도구들은 다음과 같은 특징을 지니고 있다.

○ 컴퓨터 포렌식 기술 현황 분석 : [표 4] 참조
- 공개여부 : Com(상용), Free(공개용)
- 이미지 생성 및 검사 : Disk(디스크 이미지), OS(운영체제 이미지), Traffic(IP 트래픽 이미지)
- 무결성 검사 : Hard(하드웨어 변동 검사), File(파일 무결성 검사), Finger(전자 지문 검사)
- 저수준 복구 : Raw(저수준 파일 편집), Delete(삭제 파일 복구), Key(암호키 복구)
- 기타 지원 기능 : Boot(긴급 부팅 지원), Plug(플러그인이나 프로그래밍 지원), Report(자동 보고 지원)

Ⅳ. 시스템 포렌식 기술

시스템 포렌식 기술에는 크게 백업, 복구, 분석 단계가 있다. Encase나 TCT 계열 등의 기존 포렌식 도구들이 백업이나 복구에 있어서는 많은 기능들이 있으나 분석에 있어서는 전문가의 직관적인 능력을 많이 요구하고 있다. 따라서 포렌식 기술의 발전에 있어 통합적인 분석을 지원하는 추론 기능이 필수 요소로 자리 잡을 것이다. 본 연구에서는 유닉스/리눅스 기반의 포렌식 기술을 중심으로 살펴본다.

1. 피해 시스템 백업

백업은 침입을 인지하고 분석 및 대응을 하기 위한 첫 번째 단계이다. 침해사고 분석 시 얻어지는 모든 정보가 법적인 증거로서의 효력을 갖기 위해서는 우선, 실제 피해 시스템에서 분석을 시작한 시점 이후로 데이터가 손상되지 않았다는 전제가 바탕이 되어야 하므로 피해 시스템의 데이터 획득 및 분석 과정에서 조작상의 문제, 시스템 자체의 오류 등으로 인해 시스템의 정보가 수정이 되어버리는 경우, 그리고 분석을 시작할 당시의 프로세스와 메모리 정보, 시스템 자원 사용 정보 등의 휘발성 정보에 대한 백업이 이루어지지 않은 경우 침해사고 분석 결과에 치명적인 영향을 줄 수 있을 것이다. 특히 서비스의 지속성이 필요하여 온라인으로 분석해야 하는 경우 피해 시스템을 분석하거나 모니터링 한다는 것을 공격자가 알게 되면 시스템 전체에 대한 데이터 삭제

〔표 4〕 포렌식 도구 일람

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 3.2	Windows	Com	Disk, OS	Hard, File, Finger	Raw, Delete	Plug, Report
Byte Back III	Windows	Com	Disk, OS, Traffic	Hard, File	Raw, Delete	
Detective	Windows	Com	Disk			
Computer Incident Response Suite	Windows	Com	Disk	Hard, File		
Windows NT Forensic Utility Suite	Windows	Com				
dtsearch	Windows	Com	Disk		Raw	
CD-R Diagnostic	Windows	Com	Disk		Raw, Delete	
CD-R Inspector						
FormatSecure	Windows	Com	Disk		Raw	
Hasher	Windows	Free		File		
HEX Workshop	Windows	Com			Raw	
ImageCast	Windows	Com	Disk, OS			
LOSTPASSWORD Recovery	Windows	Com			Key	
NT Password Recovery Bootable CD	Windows	Free			Key	Boot
Offline NT Password & Registry Editor, Bootdisk	Windows	Free			Raw, Key	Boot
SearchIT	Windows	Free	Disk			
WINGREP	Windows	Com	Disk			
sandersonforensics.co.uk softwares	Windows	Com	Disk			

를 하는 경우가 있으므로 백업은 피해 시스템 분석에 앞서 수행해야 할 가장 필수적인 조치이다. 백업 시 고려해야 할 시스템의 정보는 크게 다음과 같다.

o 휘발성 정보 백업
o 시스템 상태 정보 백업
o 전체 백업 및 네트워크를 통한 전송

1.1 휘발성 정보 백업

침입에 대한 정보를 분석하기 위해 대부분의 경우 일단 시스템을 종료시키거나 네트워크에서 격리시키는 등의 조치를 하게 된다. 이러한 과정에서 공격자의 로그인 상태 정보, 네트워크 연결 정보, 커널 모듈 정보 등 중요한 정보가 손상되게 되므로 피해 시스템을 격리하기 전 현재 시스템의 상태를 백업해야 한다. 이러한 과정을 "Freezing the scene" 이라고 한다. 이러한 휘발성 정보의 백업은 메모리 상에서 프로세스 정보나 네트워크 정보들을 가져와야 한다. 하지만 시스템 명령(ps, netstat 등)을 사용하여 그 결과를 백업하는 방법은 침입자가 피해 시스템에 설치한 루트킷과 같은 악성 프로그램과 프로그래밍에 의해 변조된 정보를 가져오게 될 가능성이 크다. 따라서 변조되기 전의 정보인 커널에서 직접 원천적인 데이터를 추출하는 방법은 변조되지 않는 정보를 얻을 수 있게 된다. 휘발성 정보에서는 다음과 같은 정보를 얻어야 한다.

o 프로세스 상태 정보
o 네트워크 세션 정보(arp 정보, netstat 정보, route 정보,open port 정보 등)
o 현재 적재 된 커널 모듈(LKM) 정보
o 네트워크 장치 카드의 상태 정보(Promisc 검사)
o proc 정보

1.2 시스템 상태 정보 백업

전체 백업을 하는 데는 시간이 많이 소요되기 때문에 중요 데이터의 보고가 우선인 경우 필수적으로 백업이 필요한 디렉터리나 파일들을 먼저 백업한다. 일반적으로 중요 데이터는 아래와 같이 시스템 정보 및 환경 설정 내용, 로그 파일, 데이터 디렉터리, 기타 응용 프로그램 관련 파일 등이다.

o OS 정보
o 네트워크 정보
o 주요 시스템 프로그램을 실행한 결과 값
o /var/log 디렉토리의 시스템 로그
o /etc 시스템 설정 파일 정보
o 주요 서비스 데몬의 설정 및 데이터 파일(Apache, DB, FTP, Proxy, Mail, etc)
o 각 사용자의 /home 디렉토리의 .*sh_history, .ssh_known_hosts 등의 파일
o 네트워크 장치 카드의 상태 정보(스니핑 유무)
o 열려 있는 포트 정보

1.3 디스크 이미지를 이용한 전체 백업

피해 시스템의 증거를 훼손하지 않고 복사된 정보를 분석하기 위하여 피해 시스템의 디스크전체에 대하여 파티션별로 분석 시스템으로 복사한다. 이때 유용하게 사용되는 도구는 "dd"라는 시스템 명령이다. 백업 단계에서 "dd"를 이용하여 각각의 파티션을 비트 단위로 이미지 백업을 수행하도록 한다. 전체 백업의 수행 절차는 아래와 같다.

o 백업 시스템에서 "netcat" 도구를 이용하여 백업 준비를 한다.

```
sh-2.04#  nc -l -p 10000 > /dir/to/victim.hda1.dd
```

이와 같은 명령어로 네트워크 연결을 통해 비트 단위로 저장할 수 있게 된다.

o 피해 시스템에서는 "dd" 명령어를 사용하여 시스템 이미지를 저장한다.

```
sh-2.04#  dd bs=1024 < /dev/hda1 | nc [백업 시스템 IP] [Port] -w 3
```

이와 같은 명령어를 통해 파일 시스템에서 1024 바이트 단위로 이미지를 복사한 후 "netcat"을 사용하여 백업 시스템에 전송한다.

1.4 네트워크를 통한 백업

필요한 경우 백업된 파일을 네트워크를 통해 전송하기도 한다. 격리된 로컬 네트워크 내에서의 전송의 경우라면 일반적인 파일 전송 프로토콜(FTP) 또는 netcat 등의 도구를 이용하여 백업된 파일을 전송해도 안전하다 할 수 있지만, 외부 네트워크를 통해 전송하는데 있어서 기밀성과 무결성을 보장받기 위해서는 반드시 안전한 채널을 통하여 기밀성과 전송해야 한다. 에이전트에 탑재해야 할 데이터 전

송기능 모듈을 구현하는데 있어서 가장 참고할 만한 기법은 데이터의 안전한 전송을 위해 SSH(Secure Shell)을 사용하는 scp를 이용하여 암호화된 전송을 하는 것이다. scp 명령은 네트워크에서 호스트 간에 파일을 복사하는데 사용된다. 이때는 데이터 전송시 SSH를 사용하며 SSH와 동일한 보안을 제공한다.

2. 파일 시스템 복구

2.1 파일 시스템 복구

포렌식스 과정에 있어 파일 시스템의 복구는 반드시 필수적인 요소이다. 조사와 분석 단계 이전에 모든 지워진 데이터를 복구함으로써 사건에 대한 재구성을 할 수 있으며, 파일 지스러기(slack) 영역의 데이터를 복구함으로써 부분적으로 남아있을 수 있는 각종 데이터의 조각들을 얻을 수 있으며 증거 자료로 사용할 수 있게 된다. 복구 영역을 정의하면 다음과 같다.

o rm 명령어를 통해 지워진 경우
- 일반적으로 파일을 지우면 파일시스템이 가지고 있는 링크만 사라질 뿐 물리적인 데이터는 다른 파일이 덮어쓰기 전까지 그대로 남아있다. 파일 이름은 복구가 되지 않으나 데이터는 복구가 가능하다.

o 지워진 블록위에 새로운 정보가 덮어 써진 경우
- 한 블록 전체를 덮어 쓰지 않았다면 지스러기 영역의 복구가 가능하다.

o 파일의 EOF 이후에 쓰레기 공간에 정보를 기록해 놓은 경우
- EOF 이후에도 공간을 검사하여 정보를 검색한다. 지스러기 영역의 검사를 통해 데이터를 확인할 수 있다.

해킹 사고의 패턴을 분석해보면 공격자는 시스템의 권한을 얻은 뒤 시스템에 루트킷을 비롯한 해킹 도구를 설치한다. 설치 후 흔적을 지우기 위해 도구를 삭제하며, 시스템에 남아있을 로그 파일을 삭제하게 된다. 따라서 파일 시스템의 복구는 공격자의 증거를 파악하는데 매우 중요한 행위라고 할 수 있다. 복구된 로그 파일이나 해킹 도구들은 행위를 유추하는데 매우 중요한 단서가 되며 조사의 시발점이 될 수 있다.

리눅스 시스템을 예로 들면, 리눅스의 대표적인 파일 시스템인 ext2fs는 지워진 아이노드 테이블 정보가 유지됨으로써 그 정보를 토대로 복구가 가능하다.

o 로그 기록 확보
o 공격자가 의도적으로 은폐하기 위해 시스템에 남을 수 있는 로그를 변조 및 삭제하는 등의 행위후의 로그 기록 복원
o 공격자 신원 파악
o 피해 산출

2.2 파일 지스러기(slack) 영역의 은닉데이터 탐지

리눅스 시스템의 경우, 블록 사이즈는 보통 4KB이다. 파일의 내용의 크기가 4KB단위로 정확히 나누어지지는 않기 때문에 맨 끝에는 낭비되는 공간이 존재하게 된다. 이 공간을 지스러기 공간(Slack space)이라고 한다. 지스러기 공간을 복구하고 검사하는 것은 크게 다음과 같은 2가지 이유를 들 수 있다.

o 지스러기 영역의 데이터 복구
- 지워진 파일 후에 새로운 블록이 할당 되었더라도 전부 쓰지 않았다면 지스러기 영역의 정보를 복구할 수 있다.

o 지스러기 영역의 악의적인 파일 탐지
- 공격자는 지스러기 영역을 이용하여 자신의 비밀정보나 바이너리 파일을 쪼개서 넣을 수 있다. 지스러기 영역에 저장된 데이터는 무결성 검사와 같은 일반적인 검사에는 탐지가 불가능하다. 지스러기 영역의 정보를 수집하는 도구로 bmap과 같은 도구가 있다.

3. 피해 시스템 분석

피해 시스템을 분석하는데 있어서 분석 방법은 상황에 따라 매우 다양하다. 지속적인 서비스를 제공해야 하는 경우가 있으며, 원격에서 피해 시스템을 분석해야 하는 경우, 피해 증거가 발견되지 않아 장기간의 분석이 필요한 경우, 반대로 빠른 분석 후 복구를 결정해야 하는 경우 등 여러 상황이 존재한다. 피해 시스템 분석 방법은 다음과 같이 크게 나눌 수 있다.

o 격리 분석
- 대체 백업 시스템이 있어 정상적인 서비스에 지

장이 없을 경우, 또는 분석할 동안 서비스를 하지 않아도 될 경우, 정확한 증거보존이 필요한 경우, 그리고 분석 시스템을 이용하여 아주 철저한 분석을 원할 경우 가능하다. 단, 격리 이후에는 공격 프로그램 또는 침입자를 모니터링하기 어렵게 된다.

o 온라인 분석
- 대체 백업 시스템이 없어, 해당 시스템이 없으면 정상적인 서비스를 하지 못할 경우 가능하다. 피해 시스템에 온라인으로 로그인해서 분석하게 되며, 주로 원격지의 시스템을 빨리 분석해야 할 경우에 적합하다. 공격 프로그램이나, 공격자의 활동 등을 지속적으로 모니터링 할 수 있다. 단, 분석 도중에 침입 흔적이 파괴되거나 손상될 수 있어 정확한 분석이 힘들다. 이것은 최소한 자원으로 최소한의 분석만을 원할 경우의 분석 방법이다.

o 분석 시스템을 이용한 분석
- 피해 시스템의 디스크의 이미지를 복사해서 분석 시스템을 이용하여 분석하는 방법으로 "컴퓨터 포렌식스" 증거를 훼손하지 않기 위한 분석 방법이다. 피해 시스템의 자원을 이용하지 않고 분석 시스템의 자원을 이용하기 때문에 보다 정확한 분석이 가능하다. 단, 분석 시스템 준비, 디스크 복사 등 피해 시스템 분석에 앞서 준비할 사항이 많으며 시간이 오래 걸린다.

기본적으로 온라인 시스템에서 공격자의 위협적인 행동에 영향을 받지 못하도록 빠른 시간 내에 최대한 시스템의 휘발성 정보를 기록하고, 네트워크를 차단 후 분석시스템으로의 이미지 복사 후 분석시스템을 이용한 격리 분석 형태를 가진다.

3.1 시스템 상황 분석

시스템 분석을 시작하게 되면 가장 먼저 해야 할 일은 시작한 처음 시점의 시스템 상황을 기록하는 것이다. 디스크 상의 모든 파티션 정보와 디렉토리, 파일 구성 정보, 현재 구동 프로세스, 주요설정 파일구성 정보를 파일 또는 프린터로 출력한다. 시스템 분석을 하면서 IP파악, 범행 시간 추정, 변조된 파일, 해킹 출처확인, 손상된 시스템 등을 조사하고 조사된 모든 정보에 대해서는 반드시 기록하고 문서화한다. 분석된 결과물에 대해서는 DB로 정리해서 보관한다. 다음 [표 5]는 온라인 상황과 디스크 이미지 복사 후 오프라인 분석에서 기본적으로 기록되어야 할 상황을 유닉스/리눅스의 기본 명령어 형태로 나타내었다.

[표 5] 시스템상황 기록

	command	설명
온라인 상황	ps -elf	프로세스 상에서 프로세스의 모든 정보
	who	로그인명, 터미널 라인, 로그인 시간, 원격 호스트 또는 X 디스플레이
	w	누가 로그인 하였는지, 마지막 무엇을 하였는지에 대하여 표시
	netstat -an	네트워크에 연결된 모든 활성 상태를 표시
	nmap -sT(sU) -p1-65535 IP	외부에서 네트워크에 열린 모든 포트를 검사
	lsof -i	열려진 소켓 정보 출력
오프라인 상황	ls -alR	모든 파일의 리스트를 표시
	df -k fdisk -l	시스템 디스크 파티션의 정보
	find	find를 이용하여 setuid, setgid 등을 가진 파일 정보 유지, /dev 아래에 정규파일 등의 기록

3.2 휘발성 정보 분석

휘발성 정보란 프로세스나 네트워크 정보처럼 시스템이 온라인 상태에서 살아있는 시스템이 가지는 일시적인 정보를 말한다. 피해 시스템을 분석하는데 있어 초기 휘발성 정보의 수집은 매우 중요한 의미를 가진다. 파일시스템의 이미지만을 조사하는 오프라인 성격의 포렌식 도구는 현재 동작중인 프로세스나, 외부에서 연결된 네트워크 정보를 가지고 있지 않으므로, 공격자의 연결 위치나 공격자가 동작시켜 놓은 프로세스 정보를 파악할 수 없다. 따라서 휘발성 정보를 수집하는데 있어 살아있는 피해 시스템에서 휘발성 정보를 저장하는 작업을 빠른 시간에 수행하고 네트워크를 차단하는 절차에 들어간다. 하지만 최근 해킹에 있어서 공격자 자신의 흔적을 대부분 숨기기 위해 rootkit을 사용하고 있으며, 피해 시스템의 유틸을 이용하여 정보를 수집하는 것은 오용의 범위가 매우 크다. 현재 kernel rootkit의 기능은 커널과 통신하는 시스템 콜 자체를 변조시키므로 시스템에서 사용되는 관리 유틸(/bin/ps, /bin/netstat 등)들을 이용하여 시스템 콜을 호출하는 방식 역시 올바른 휘발성 정보를 가져오지 못한다.

따라서 분석자는 메모리의 변조되지 않은 정보를 가져와야 하며, 이는 시스템의 커널에서 직접 원천적인 데이터를 추출하는 방법을 통해 변조되지 않는 정보를 얻을 수 있다.

3.3 루트킷 탐지

근래 공격자들의 침입 후 루트킷의 사용은 일반화되었으며, 이를 탐지하고 분석하는 것은 매우 중요하다. 루트킷은 크게 traditional rootkit과 kernel rootkit, LKM을 사용하지 않고 직접 kmem을 변조하는 advanced rootkit으로 나눌 수 있다[11].

3.3.1 traditional rootkit[12]

루트킷을 탐지하는데 있어 t0rnkit과 같은 일반적인 시스템의 어플리케이션을 변조하는 traditional rootkit을 검출하기 위해서 백업시스템에서 탐지 모듈을 동작시킨다. traditional rootkit은 2가지 측면에서 검출될 수 있다.

첫째는, 루트킷이 가지는 독특한 특징을 중심으로 검출하였다. 예를 들어 t0rnkit과 같은 루트킷은 설치시 패스워드를 설정하기 위해 /etc/ttyhash 값에 암호화된 값을 저장한다. 이러한 특징을 이용하여 일반적인 검출을 한다. 하지만 특징을 이용하여 검출할 경우 공격자가 루트킷의 설치 파일을 새롭게 변조한다면 찾기가 힘드나, 탐지되었을 때 쉽게 피해 시스템에 설치된 루트킷을 파악 할 수 있다는 장점도 있다.

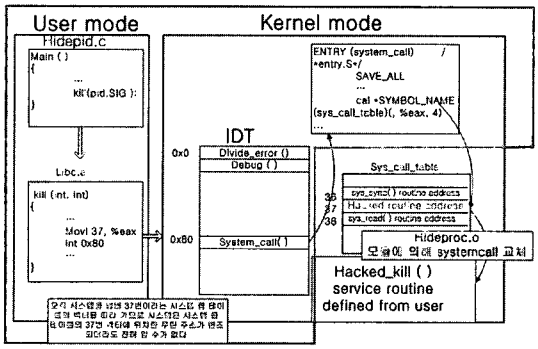
둘째는 rootkit 자체가 변조해놓은 시스템 바이너리 파일의 변조여부를 보고 rootkit이 설치된 사실을 파악 할 수 있다. 이는 직접적으로 rootkit의 정확한 내용을 파악 할 수는 없지만 시스템이 루트킷에 의한 변조를 확인 할 수 있다.

3.3.2 kernel rootkit[13]

공격기술이 발전하면서 루트킷 역시 기존의 일반적인 백도어에서 커널 기반의 백도어로 발전하게 되었다. 커널 기반의 백도어는 LKM(Loadable Kernel Module) 형태로 적재되어서 시스템 함수의 정상적인 실행을 바꾸는 방법을 사용한다. 따라서 응용 수준에서 사용되는 프로그램으로 탐지하기에는 어려움이 따르게 된다.

커널 기반의 루트킷은 기본적으로 다음과 같은 기능을 지니고 있다.

- o 루트 권한 얻기
- o 파일 감추기
- o 모듈 감추기
- o 실행되고 있는 프로세스 감추기
- o 프로세스 종료하기



〔그림 3〕 커널 루트킷의 동작 원리

〔그림 3〕은 시스템 콜인 kill 루틴을 가로채는 모습으로 커널 루트킷의 기본적으로 동작하는 원리를 그림으로 표현한 부분이며 시스템 호출의 변경을 통해 악의적인 시스템 호출을 수행하게 한다.

커널 루트킷을 탐지하는 방안은 다양한 기법이 존재하며 〔표 6〕과 같은 기법을 통해 탐지한다.

〔표 6〕 커널 루트킷 탐지 방법론

탐지 방법론	설명과 탐지여부
숨겨진 프로세스 검출	readdir()과 chdir() 함수를 이용한 무차별 프로세스 점검으로 비교한다. 비교적 커널 루트킷에서는 readdir이 사용하는 getdents를 사용하나 변조가능성이 있고, chdir()에서는도 잠재적 변조가능성 존재한다.
	task structure의 linked list 검사하며 탐지 가능성이 높다.
system.map 파일과 비교	system.map 파일과 현재 상태 비교하며, 탐지 가능성 있으나 system.map 파일의 변조 가능성이 존재한다.
커널 내부에 백도어 방지모듈 설치	LKM 백도어가 커널에 적재되면, 백도어 탐지가 매우 어렵다는 것을 인식하여 미리 커널내부에 백도어를 방지할 수 있는 모듈을 설치하는 방법이나, 포렌식스의 사후 조사라는 특수성 때문에 기법 도입이 힘들다.
LKM의 linked list추적	LKM이 존재한 실제 주소값을 검출하여 비교하며, 탐지 가능성이 높다.

3.4 시간정보를 이용한 분석

해킹 피해 시스템을 조사하는데는 시간적인 요소를 고려하는 것은 매우 중요하다. 시간대를 통해 파일의 삭제 유무나 해킹이 일어난 시간, 공격자가 그 시간 동안 생성한 파일 등을 판별할 수 있다. 유닉스/리눅스에서 제공하는 시간은 mtime, atime, ctime 이렇게 크게 3가지가 있다. 여러 가지 명령어에 따른 파일의 시간대 변경 유무를 [표 7]에서 알 수 있다.

o mtime : 파일의 최근 수정시간
o atime : 파일의 최근 접근시간
o ctime : 파일의 최근 속성정보 수정시간

일반적으로 공격자들은 mtime이나 atime은 바꾸어도 ctime은 바꾸지 못하는데, 뛰어난 공격자들은 저수준 디스크 장치에 쓰거나 ctime까지 변경할 수 있다. 특히 ext2파일 시스템의 리눅스에서는 debugfs를 이용하여 아이노드 내용을 바꿀 수 있다. 피해 시스템을 분석하는데 있어 시간적인 분석은 매우 중요한 요소를 지니게 된다. 특히 부분 삭제된 경우의 파일은 atime과 mtime의 값은 변하나 ctime의 값은 변하지 않아 세 가지 시간을 비교하였을 때 atime과 mtime은 일치하나 ctime이 변화 되었을 때 부분 삭제된 파일로 간주 할 수 있다. 또한 공격자의 증거를 발견하였을 때 시간적 순서를 중심으로 공격자의 행위 흐름을 나타낼 수 있으며 이때 삭제된 파일을 추적함으로써 복구의 중요한 실마리로 작용할 수 있으며 또한 추적의 실마리로 접근할 수 있다. 공격자가 활동한 시간대에 생성된 파일을 의심하는 것은 분석 관점에서 매우 중요한 행위이다.

로그 파일의 경우 기록된 시간의 간격을 통해 삭제여부를 판단하는 방법도 유용한 기법이다.

[표 7] 파일관련 이벤트 시간

명령어		atime (접근시간)	mtime (파일수정시간)	ctime (inode 수정시간)
		ls -alu	ls -alt	ls -alc
ls		X	X	X
vi	:q	O	X	X
	:wq!	O	O	O
find ./ -type f		X	X	X
file		O	X	X
cat	내용보기	O	X	X
	내용추가	X	O	O
more, head, tail		O	X	X
파일 실행		O	X	X
cp		O	O	O
mv		X	X	O
chown, chgrp, chmod		X	X	O
touch	touch file	O	O	O
	touch -t 04050900 file	O	O	X

3.5 무결성 검사를 통한 바이너리 파일의 변조 유무 분석

포렌식 기술에 있어 무결성 검사는 시스템의 바이너리 파일의 무결성을 검사하는 중요한 단서가 된다. 포렌식 기술에 있어 무결성 검사는 기존의 무결성 검사와는 성격이 다르다. 그 이유는 포렌식스 분야 자체가 어떤 공격 행위에 대해 적극적인 대처를 하는게 아니라, 침해사고가 발생한 후 상태에 조사하므로 피해 시스템에 tripwire와 같은 무결성 검사 도구를 설치하지 않았으면 무결성에 대한 검사를 하기가 쉽지 않다.

하지만 솔라리스의 시스템의 경우에는 바이너리 패키지에 대해 md5 책섬 값을 제공하고 있다. 따라서 공격자가 시스템에 바이너리 파일들을 변조 할 경우 분석자는 이 값을 비교함으로써 바이너리 파일에 대한 무결성 검사를 할 수 있다. 반면 리눅스의 경우에는 다양한 배포판과 그 배포판에 속해있는 바이너리 파일들의 버전 정보가 일정하지 않아 무결성 검사를 하는데 있어 다소 어려운 것이 사실이다. 따라서 솔라리스 시스템의 경우와 같이 각각의 버전별 바이너리 파일들의 무결성 정보를 DB화 해놓아 해결 할 수 있다.

3.6 로그 분석

기본적으로 로그 분석은 리눅스 시스템에 있어 /var/log 하단의 messages, secure, xferlog, utmp, wtmp 등 다양한 로그를 얻을 수 있다. 또한 로그 정보에 있어서 응용프로그램들의 로그 정보인 viminfo나 ftp 로그, apache 로그 등은 다양한 정보를 제공해 준다. 이들을 분석해보면 특정 응용프로그램을 사용한 결과에 대한 정보를 얻을 수 있게 된다. 예로 공격자들은 서버에서 파일을 자신의 시스템으로

전송하거나 받아올 때 ftp나 ssh 또는 http 프로토콜을 이용할 수 있는데 이를 분석하면 공격자가 사용한 파일의 정보를 얻을 수 있다.

또한, 공격자들이 특정 서버에 권한을 획득하기 위해 원격에서 특정 서비스의 취약점을 통해 임의의 권한을 획득하는데 이러한 취약점을 통해 들어온 공격들은 시스템에 흔적을 남기게 되고 이를 통해 서버에 발생한 외부로부터의 공격방법을 유추할 수 있다. 이러한 흔적을 조사하게 되면 공격 방법에 대한 단서를 얻을 수 있다. [표 8]은 lpd 원격 포맷스트링 공격을 통해 루트 권한을 획득했을 때 /var/log/messages에 남는 로그 형태이다.

[표 8] lpd remote format string attack 의 로그

```
Feb 7 17:25:59 chsdhs SERVER[850]: Dispatch_input:
bad request line 'BB?ㅎ웝ㅎㅎ윛ㅎㅎ윙ㅎㅎ쮦
XXXXXXXXXXXXXXXX00000000000000000000000000
000000000000000000000000000000..........0000000000
0000000000000000000003221224720럾럾럾럾럾럾럾럾럾럾
럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾
럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾럾
럾럾럾럾럾럾럾럾1??육F?뒮1冷f뒶1??????뎲?M澈
1?E?f????O'뎲?E?E變E?P뒶뒟澈 뒶CC?뒶
```

3.7 공격자 추적

침입자를 추적하기 위해서는 침입 시간 추적과 침입자의 근거지 추적의 두가지 형태로 파악 될 수 있다. 이는 피해 시스템 분석과 복구된 모든 정보에서 침입 경로, 침입 시간, 침입 흔적 등을 종합하여 판단하여야 한다. 침입 시간 추적에 있어서는 앞에서 언급했던 것처럼 해킹 행위가 발생했던 당시의 시간의 검증과 침입자에 의해 생성된 파일의 MAC(mtime, atime, ctime) 정보에 의해 시간대 분석이 이루어져야 한다. 침입자 추적에 있어서는 로그파일에 남은 침입자의 기록이 중요한 판단의 기준이 되기는 하나 이는 침입자들이 자신의 침입흔적을 지우는 경우가 많으므로 실제 침입자의 IP를 찾는 작업은 그리 쉬운 일은 아니다. 따라서 공격이 일어난 시점에 지워진 로그 파일의 복구와 여러 가지 응용 로그, 그리고 존재한다면 IDS 로그형태까지 종합하여 침입자의 IP 주소를 파악해야 한다. 단 스푸핑(spoofing) 되거나, 경유지를 통해 들어온 경우 침입자에 대해 명백히 결론을 내리는 것은 어리석은 일이다.

침입자의 IP를 밝혀낸 후에는 whois 서비스와 traceroute 정보 등 관련 정보를 통해 침입자의 위치를 파악하는 것이 중요하며, 국내 IP의 경우 국내 ISP 업체와 긴밀한 협조 하에 침입자의 MAC 주소나 기타 지역 등에 대한 정보를 파악 할 수 있다.

다음은 IP주소의 whois 정보를 제공하는 사이트이다.

o http://nic.or.kr
 - 한국인터넷 정보센터로 국내 도메인 현황과 등록 정보가 있다.

o http://www.ipindex.de/
 - ipv4의 할당된 IP의 국가별 현황이 나와 있다.

3.8 보고 기능

포렌식스 과정에서 도구가 지녀야 할 가장 중요한 기능은 보고 기능이라고 할 수 있다. 보고 기능에서는 포렌식 도구에서 분석된 결과를 일목요연하게 볼 수 있어야 하며 기본적으로 육하원칙에 의거하여 기술하는 것이 좋다. [표 9]는 보고 기능에서 제시할 수 있는 결과를 나타내었다.

[표 9] 분석 결과에 따른 보고 내용

보고 영역	분석 결과
시스템 정보	OS 정보, 하드웨어정보, 네트워크 정보, 프로세스정보 등
파일 복구 정보	로그파일, 일반파일, 숨겨진 파일, 중요정보 등
공격방법	시스템 취약점, 공격시 나타난 로그정보 등
시스템 피해상황	해킹 도구(rootkit, 백도어) 설치 여부, 로그파일 변조유무, 정보유출 유무
공격자 추적	의심가는 공격자 위치, IP 정보, MAC 정보, whois 정보
사후처리 정보	보고기능, 복원 절차

V. 결 론

본 연구에서는 포렌식스 분야의 영역과 그리고 컴퓨터 포렌식스의 절차에 대해 기술하고, 국외에서 개발된 포렌식 기능을 지닌 여러 도구에 대해 고찰해보았다. 또한 시스템 포렌식 도구가 지녀야 할 백업과 복구 그리고 분석 기능에 있어 기술들을 논의함으로써 실제 개발에 있어 필요한 지침이 될 수 있도록 기술하였다. 현재의 포렌식 도구는 백업과 복구의 기능의 기능은 발달해 있는 반면 피해 시스템을 분석하고 사고 자체의 흐름을 조합하는데 있어서는 분석자에게 상당한 지식을 요구하고 있다. 따라서 포렌식 도구 개발에 있어서 전문가의 도움 없이 도

구 자체에서 다양한 분석 기법을 통해 관리자에게 쉽게 전달될 수 있는 기능이 추가되어야 할 것이다. 그러기 위해서는 인공지능 기법을 통한 다양한 추론 모듈의 개발이 필수 요소가 될 것이다.

여기에서 자세히 언급하지 않은 시간 동기화(time synchronization) 문제와 파일의 암호화 문제는 또 다른 과제이다. 시스템 포렌식스 분야는 네트워크 영역으로 확장되면 네트워크 포렌식스 분야와 결합되어 더욱 다양한 증거 자료에 대한 분석이 필요하다.

참 고 문 헌

[1] Warren G.Kruse II, Jay G.Heiser, "COMPUTER FORENSICS : Incident Response Essentials", Addison Wesley

[2] Gary Palmer. A Road Map for Digital Forensic Reserch. Technical Report DTR-T001-01, DFRWS, November 2001. Report From the First Digital Forensic Research Workshop(DFRWS).

[3] Mark Reith, Clint Carr, Gregg Gunsch, "An Examination of Digital Forensic Model", International Journal of Digital Evidence, 1(3), Fall 2002.

[4] John R. Vacca, "COMPUTER FORENSICS : Computer Crime Scene Investigation", Charles River media

[5] Brian Carrier, "Open Source Digital Forensics Tools: The Legal Argument", @STAKE Inc. (2002)

[6] Lee Garber. EnCase: A Case Study in LP Computer-Forensic Technology. IEEE Computer Magazine January 2001. Available at:http://www.computer.org/computer/homepage/January/TechNews/technews2.htm

[7] D. Farmer and W. Venema, The Coroners Toolkit(TCT) v1.06, available at: http://www.porcupine.com/tct

[8] B. Carrier, TCTUTILs v1.01, available at: http://www.cerias.purdue.edu/homes/carrier/forensics.html

[9] B. Carrier, Sleuth Kit v1.62, available at: http://www.sleuthkit.org/index.php

[10] B. Carrier, Autospy v1.73, available at: http://www.sleuthkit.org/autopsy/index.php

[11] Ed Skoudis, "COUNTER HACK", Prentice Hall PTR, (2002) 253-319

[12] "Root Kits" and hiding files/directories/processes after a break-in, Available at: http://staff.washington.edu/dittrich/misc/faqs/rootkits.faq

[13] J. K. Rutkowski, "Execution path analysis: finding kernel rootkits", Phrack Magazine Vol-ume 10, Issue 59 (2002)

〈著 者 紹 介〉

황 현 욱 (Hwang Hyun-Uk)

2000년 : 조선대학교 정보통신공학과 졸업(학사)
2002년 : 조선대학교 대학원 전자공학과 졸업(공학석사)
2002년~현재 : 전남대학교 대학원 정보보호협동 박사과정 재학
관심분야 : 시스템 보안, 네트워크 보안, 정보보안, 포렌식스, 해킹 등

김 민 수 (Kim Min-Soo)

1993년 : 전남대학교 전산통계학과 졸업(학사)
1995년 : 전남대학교 대학원 전산통계학과(이학석사)
2000년 : 전남대학교 대학원 전산통계학과(이학박사)
2000년~2001년 : 한국정보보호진흥원 선임연구원
2001년~현재 : 전남대학교 리눅스시스템보안연구센터 객원교수
관심분야 : 시스템 보안, 네트워크 보안, 정보보안, 신경망 등

노 봉 남 (Noh Bong-Nam)

1978년 : 전남대학교 수학교육과 졸업
1982년 : KAIST 대학원 전산학과(이학석사)

1994년 : 전북대학교 대학원 전산통계학과(이학박사)
1983년~현재 : 전남대학교 컴퓨터정보학부 교수
〈관심분야〉 컴퓨터와 네트워크 보안, 해킹과 사이버 보안, 사이버 사회와 윤리

임 재 명 (Lim Jae-Myung)

1981년 : 한양대학교 전자공학과 졸업
1983년 : 한양대학교 공과대학원 전자공학과 졸업
1983년~1990년 : (주)동양정밀 컴퓨터 개발과장
1996년~2000년 : ICU 부설 정보통신교육원 참사
2000년~현재 : 한국정보보호진흥원 네트워크모니터링 팀장
관심분야 : 통신망관리, 정보보안, 시스템 및 네트워크 보안 등