컴퓨터 포렌식스 관점에서 파일 지스러기 영역의 활용방법 연구

이석봉[O] 박준형 김민수 노봉남
전남대학교 정보보호협동과정
bernandi@kjssa.co.kr[O] {werther, phoenix}@lsrc.chonnam.ac.kr bongnam@chonnam.ac.kr

A Study of Practical Method Using Slack Space from the Standpoint of Computer Forensics

Sukbong Lee[0] Junhyung Park Minsoo Kim Bongnam Noh
Interdisciplinary Program of Information Security, Chonnam National University

요 약

컴퓨터 범죄가 증가하면서 침입자들의 기법도 갈수록 교묘해 지고 있다. 특히 침입자 자신을 숨기고 자신의 정보를 시스템에 은닉하여 다음에 재사용할 가능성이 제기되고 있다. 컴퓨터 포렌식스 관점에서 이러한 위험성은 간과해서는 안될 중요한 사항이다. 정보를 은닉할 수 있는 곳 중에서 파일 지스러기 영역은 파일 시스템을 통해서 접근이 불가능하기 때문에 무결성 체크 프로그램도 탐지가 불가능하다. 본 논문에서는 파일 지스러기 영역을 활용하여 데이터를 은닉하는 방법을 고찰해 보았다. 그리고 파일 지스러기 영역에 데이터를 은닉하고 복구할 수 있는 프로그램을 구현하여 고찰한 내용이 가능한 것임을 증명하였다.

1. 서론

컴퓨터 범죄가 증가하면서 컴퓨터 침입자들의 기법도 갈수록 교묘해 지고 있다. 특히 침입자들은 피해시스템에 자신의 흔적을 남기지 않기 위해서 많이 노력한다. 또한 침입자들은 자신의 정보를 피해시스템에 숨겨두는 경우가 많다[1]. 침입자들은 대부분 피해 시스템들을 중간 경유지로 이용하기 때문에 백도어(backdoor)를 설치해 두는데 이 백도어가 사용하는 정보나 자신의 정보를 시스템 관리자 몰래 숨겨둘 필요가 있기 때문이다.

컴퓨터 포렌식스(computer forensics) 분야에서 이러한 숨겨진 정보를 찾는 것은 매우 중요한 문제이다. 침입자가 숨겨둔 정보는 침입사실에 대한 직접적인 정보이거나 침입자 자신에 관련된 정보일 확률이 매우 높기 때문이다. 그리고 침입증거를 조사할 때 시스템 전체를 검사하여 침입증거를 찾는 것보다 은닉된 침입증거를 먼저 검사하는 방법이 더 효율적이다. 침입자가 정보를 은닉할 수 있는 가능성이 있는 곳으로 파일시스템의 자유 영역(free space), 파일 지스러기 영역(slack space), 스테가노그라피(steganography) 데이터, 암호화 등으로 알려져 있다[2].

그 중에서 파일 지스러기 영역은 파일을 통해서 접근이 불가능하고 파일의 무결성 검사 방법을 통해서도 탐지가 불가능하기 때문에 중요하게 다루어져야 할 부분이다. 본 논문에서는 컴퓨터 포렌식스 관점에서 파일 지스러기 영역의 활용방법에 대해서 고찰해 보고 실제 파일 지스러기 영역을 활용하는 프로그램을 구현하여 고찰한 내용을 증명하였다.

본 논문의 구성은 다음과 같다. 2장에서는 파일 지스러기 영역에 대해서 분석하고 3장에서는 파일 지스러기 영역의 활용 가능성에 대해서 고찰해 보고 파일 지스러기 영역에 파일을 은닉하고 복구하는 프로그램을 구현하여 실제 파일 지스러기 영역의 활용을 보였다. 4장에서는 결론과 향후 연구방안에 대해서 논하였다.

2. 관련연구

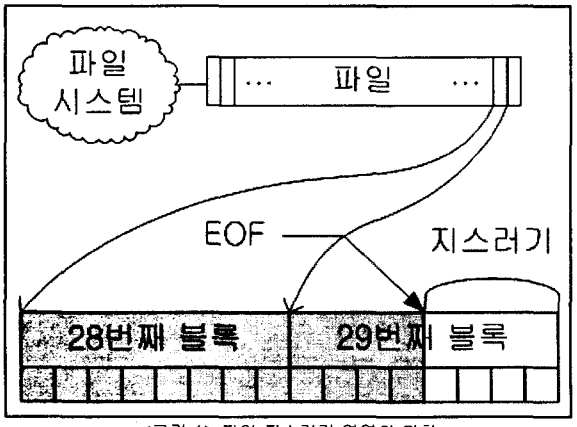
2.1 파일 지스러기 영역

2.1.1 파일 지스러기 영역의 정의

파일 지스러기 영역은 파일이 하드디스크에 할당되는 단위와 실제 파일 크기의 단위가 다르기 때문에 생긴다. 파일시스템에서 파일을 물리적인 디스크에 할당할 때 논리적인 블록단위로 할당한다. 파일시스템이 블록 단위로 파일을 할당하면 버퍼캐시에 저장되는데 버퍼캐시 스케줄링에 따라서 실제 디스크에 저장할 때는 디스크의 기본 저장단위인 클러스터단위로 저장된다. 보통 블록의 크기는 4KB를 사용하고 클러스터는 512Kb 크기를 사용하므로 4KB 크기의 블록에는 8개의 클러스터가 할당된다. 만약 블록 크기가 4KB이고 한 파일의 크기가 114KB라면 29개의 블록이 사용된다. 그림 1을 보면 29번째 블록은 4KB를 모두 사용하는 게 아니라 2(=114 - 28×4)KB 가 사용되고 마지막 2KB는 남게 된다. 그리고 다음 파일을 저장할 때 마지막 2KB도 같이 사용하는 게 아니라 새롭게 블록이 할당된다. 이때 남는 2KB의 공간이 파일 지스러기 영역이다[3]. 어플리케이션이 파일을 읽을 때 파일의 메타정보에 있는 파일 크기를 통해서 읽기 때문에 파일 지스러기 영역의 데이터는 읽을 수 없다. 이는 파일 시스템을 통해서 파일 지스러기 영역에 접근할 수 없음을 말해준다.

2.2.2 파일 지스러기 영역의 중요성

파일 지스러기 영역에는 지워진 파일의 단편들이 남아 있을 수 있다. 파일 시스템의 파일이 지워지면 성능문제 때문에 디스크의 파일을 실제로 지우는 게 아니라 파일시스템이 가진 파일의 메타정보만 지운다. 그렇기 때문에 파일이 지워지더라도 원래 파일의 내용은 자유 영역에 남아있다. 이 블록 들이 새로운 파일에 할당되면 그 파일의 지스러기 영역에 이전 파일의 내용이 남게 되는 것이다[4]. 파일 지스러기 영역의 이러한 데이터들에 자유 영역에서 찾을 수 없었던 정보가 포함되어 있을 수 있기 때문에 컴퓨터 범죄 수사에서 중요하게 다루어야 한다.



<그림 1> 파일 지스러기 영역의 위치

또한 파일 지스러기 영역에 침입자가 의도적으로 정보나 파일을 은닉할 수 있다. 침입자가 파일 지스러기 영역에 접근하려면 관리자 권한을 획득한 후 파일 시스템을 통하지 않고 디스크를 블록수준으로 접근하여 파일의 크기와 블록맵(block map) 정보를 이용하면 파일 지스러기 영역에 접근할 수 있다.

특히 파일 지스러기 영역의 데이터는 파일의 무결성을 검사해 주는 프로그램으로 탐지가 불가능하기 때문에 중요하게 다루어야 한다. 파일의 무결성 검사 프로그램들은 파일의 내용을 기준으로 체크섬(checksum)이나 해쉬(hash)함수를 사용하여 파일의 무결성 검사를 한다. 그러나 파일의 크기를 넘어서는 파일 지스러기 영역의 내용은 검사 대상이 아니다. 이 점이 침입자들에게 정보를 숨기는 곳으로서 파일 지스러기 영역이 매력적으로 보이는 이유이다.

3. 파일 지스러기 영역의 활용 방법

3.1 파일 지스러기 영역에 존재할 수 있는 데이터

침입자는 자신에게 유용한 정보를 파일 지스러기 영역에 남길 필요가 있다. 여기서 유용한 정보란 침입자가 피해 시스템에서 동작시키려고 하는 트로이목마나 백도어 같은 프로그램이 사용하는 정보일 수 있다. 또한 침입자 자신의 IP 주소나 계정정보, 또는 피해 시스템에서 얻어낸 정보와 같이 침입자가 사용하는 정보를 숨겨놓을 수 있다[5]. 컴퓨터 포렌식스 관점에서 보면 이런 정보는 침해사고를 조사하는데 있어서 핵심적인 정보들이다.

침입자는 정보뿐 아니라 자신의 파일을 파일 지스러기 영역에 저장할 수 있다. 이 파일은 침해 시스템에서 동작할 악의적인 프로그램의 실행파일 자체나 소스파일등이 될 수 있다. 침입자는 시스템 관리자에게 의심받지 않기 위해서 파일 시스템에 남기지 않고 파일 지스러기 영역에 숨긴다. 또한 침입자가 매번 시스템에 접속할 때 마다 같은 상황이 아닐 수 있기 때문에 파일을 전송할 수 있는 상황이 되었을 때 파일을 숨겨 놓는다.

3.2 파일 지스러기 영역의 활용 가능성

악의적인 목적의 파일을 파일 지스러기 영역에 숨길 때 파일 크기가 하나의 파일 지스러기 영역의 크기보다 크다면 여러 개로 쪼개서 저장해야 한다. 쪼개져서 숨겨진 조각들을 침입자가 다시 추출해 내려면 파일 시스템의 어느 파일에 얼마만큼의 크기로 은닉되어 있는지 정보가 있어야 한다. 이 정보가 한 파일에 저장되어 있다면 이 파일을 시스템에서 찾아내는 것이 급선무이다. 이 파일은 아래의 정보를 가지고 있어야 한다.

- 전체 조각의 개수
- 은닉된 파일의 크기
- 쪼개진 조각을 가진 파일이름
- 파일 지스러기 영역에 쓴 조각의 크기

은닉된 파일의 쪼개진 정보를 가진 파일을 사용한 경우가 아니라면 쪼개진 조각이 은닉되어 있는 파일 지스러기 영역에 다음 조각을 명시하고 있는 필드가 필요하다. 이 필드는 다음과 같은 값을 사용함에 따라서 서로 다른 활용법이 가능하다.

- 파일이름

  가장 접근하기 쉬운 방법이다. 그러나 파일이름의 문자열 길이가 가변적이기 때문에 다음 파일 지스러기 영역의 정보를 추출하기 위해서는 은닉된 데이터와 구분할 구분자가 필요하다.

- 파일 아이노드(inode)

  파일의 아이노드를 이용하면 고정된 길이의 필드를 사용하기 때문에 은닉된 데이터와의 구분자가 필요 없다. 또한 구현도 간단하다. 그러나 *vi* 편집기처럼 파일의 아이노드를 변경시키는 프로그램을 사용하면 문제가 된다.

- 블록번호(block number)

  파일 지스러기 영역은 항상 파일에 할당된 마지막 블록에 위치하기 때문에 블록번호를 사용하면 다음 파일의 마지막 블록번호 값을 가진다. 그리고 파일 지스러기 영역의 시작점이 마지막 블록 안에서 파일마다 다르기 때문에 마지막 블록의 시작점부터 지스러기 영역의 시작 위치까지 거리를 나타내는 오프셋(offset) 정보를 함께 가지고 있어야 한다.

- 클러스터 번호

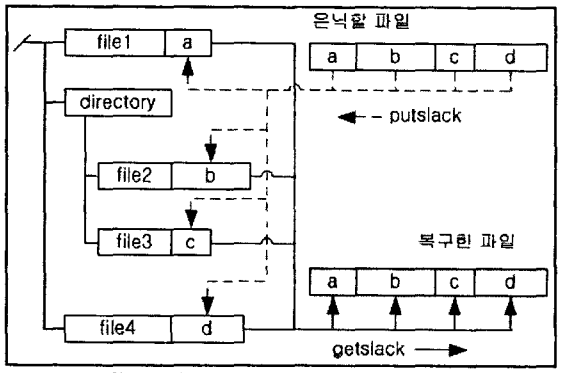
  블록 번호를 사용하는 경우와 비슷한 경우지만 클러스터 번호를 사용하는 경우의 오프셋정보가 다른 값을 가진다. 한 블록에 여러 개(4KB의 블록의 경우 8개)의 클러스터가 존재하므로 여기서 오프셋값은 파일의 끝과 겹치는 클러스터의 시작위치부터 파일 지스러기 영역의 시작점까지의 값을 가진다.

3.3 파일 지스러기 영역 접근 도구

파일 지스러기 영역에 접근할 수 있는 도구가 존재하지만 완전한 컴퓨터 포렌식스 기능을 제공하지는 않는다. Daniel Ridge의 *bmap*는 파일 지스러기 영역의 내용을 보여주고 특정한 정보를 파일 지스러기 영역에 저장하고 파일 지스러기 영역을 청소해(wipe) 주는 기능을 제공한다[6]. 또한 소스코드가 공개되어 있으므로 파일 지스러기 영역관련 프로그램을 구현하는 데 많은 도움이 된다.

파일 지스러기 영역에 접근하는 도구들은 파일 지스러기 영역에 특정 데이터를 저장하는 기능보다는 파일 지스러기 영역을 청소해주는 기능을 위주로 제공한다[6]. 그러므로 컴퓨터 포렌식스 관점에서 파일 지스러기 영역에 접근할 수 있는 도구가 필요하다.

3.4 파일 지스러기 영역의 활용



<그림 2> 파일 지스러기 영역에 파일의 은닉과 복구

파일 지스러기 영역이 악의적인 목적으로 활용될 수 있음을 보이기 위해서 *bmap*의 소스를 이용하여 특정 파일을 파일 지스러기 영역에 쪼개서 저장하고 다시 복구하는 *putslack*과 *getslack* 프로그램을 구현하였다. *putslack* 과 *getslack*이 동작하는 형태가 그림 2에 나타나 있다. 본 논문에서는 파일의 쪼개진 정보를 로그파일 형태로 가지고 있는 경우를 모델로 삼았다.

- putslack

  *putslack*은 은닉하고자 하는 파일을 받아들여서 은닉하고자하는 위치를 디렉터리 이름으로 지정해 주면 하위 파일들의 파일 지스러기 영역에 쪼개서 저장한다. 그리고 어떤 파일의 지스러기 영역에 얼마나 쪼개서 저장했는지 로그정보를 파일로 만들어 준다. 사용법은 아래와 같다.

  #putslack 디렉터리 파일이름

- getslack

  *getslack*은 *putslack*에서 출력 값으로 만들어준 로그파일을 입력으로 받아들여서 로그파일에 기록된 위치에서 조각들을 복구해 준다. 사용법은 아래와 같다.

  #getslack 로그파일

그림 3은 *putslack*이 */usr/bin/telnet* 파일을 */usr/bin* 디렉터리에 있는 파일들의 파일 지스러기 영역에 은닉하는 모습을 보여준다. 보통 실행 파일은 관리자가 지우는 일이 거의 없기 때문에 */usr/bin* 디렉터리를 선택하였다. 그림 4를 보면 *putslack*의 출력값인 로그파일 *written.file*을 이용하여 은닉시킨 파일을 복구한 모습을 보여준다. 그리고 그림 5는 복구된 파일이 제대로 동작하고 있음을 보여준다. 그러므로 파일 지스러기 영역에 침입자가 특정 파일을 은닉하고 다시 복구해서 사용할 수 있음이 증명되었다.

```
[root@ashtarte put_elf]# ./putslack /usr/bin /usr/bin/telnet
/usr/bin/consolehelper
written in /usr/bin/consolehelper, witten size is 2783
/usr/bin/catchsegv
          • • •
```

<그림 3> *putslack*의 실행화면

```
[root@ashtarte get_elf]# ./getslack input/written.file
/usr/bin/consolehelper, 2783
/usr/bin/consolehelper
slack size 2783, read size 2783, witten size is 2783
          • • •
```

<그림 4> *getslack* 의 실행화면

```
[root@ashtarte get_elf]# output/got.file www.kjssa.co.kr
Trying 210.118.73.2...
Connected to www.kjssa.co.kr.
Escape character is '^]'.
Red Hat Linux release 7.3 (Valhalla)
Kernel 2.4.18-3 on an i686
login: bernandi
Password:
Last login: Wed Jul  2 15:32:54 from 210.118.73.11
[bernandi@ns bernandi]$
          • • •
```

<그림 5> *getslack*으로 복구된 파일을 확인

4. 결론 및 향후 연구 방안

본 논문에서는 컴퓨터 포렌식스 분야에서 중요하게 다루어지는 파일 지스러기 영역에 대해서 알아보았다. 그리고 파일 지스러기 영역을 침입자가 자신의 데이터를 은닉하고 복구하는 활용 방법에 대해서 고찰하였다. 또한 실제 파일 지스러기 영역에 파일을 은닉하고 복구하는 프로그램을 개발하여 파일 지스러기 영역이 악의적인 목적으로 활용이 가능함을 증명하였다.

본 연구에서는 파일 지스러기 영역의 위험성과 활용방안을 확인하였으므로 파일 지스러기 영역에 은닉되어 있는 정보의 탐지와 추출에 대한 보다 체계적인 연구를 진행할 예정이다. 그리고 단편화된 정보들이 수거 되었을 경우 원본 데이터로의 복구 방안에 대한 연구도 과제로 남아 있다.

참고문헌

[1] John R. Vacca, *Computer Forensics: Computer Crime Scene Investigation*, Charles River Media, 415-16, June 2002

[2] DFRWS Technical Report, *A Road Map for Digital Forensic Research*, Report From the First Digital Forensic Research Workshop, 23-26, August 2002

[3] Michael A. Caloyannides, *Computer Forensics and Privacy*, Artech House, 33-38, December 2001

[4] Rohas Nagpal, *Recovery of Digital Evidence*, http://www.asianlaws.org/cyberlaw/library/cc/dig_evi.htm, 2002

[5] Ronald L. Mendel, *Computer Crime Investigator's Toolkit*, http://secinf.net/misc/Computer_Crime_Investigators_Toolkit.html, Part IV, October 2002

[6] Anton Chuvakin, *Linux Data Hiding and Recovery*, http://www.linuxsecurity.com/feature_stories/data-hiding-forensics.html, October 2002