라이브 시스템의 패스워드 크래킹[1)]

*이석희 *김권엽 *이상진 *임종인

*고려대학교 정보경영공학전문대학원

*gosky7@korea.ac.kr,

Password Cracking of Live System

*Seokhee Lee *Kwonyeop Kim *Sangjin Lee *Jongin Lim

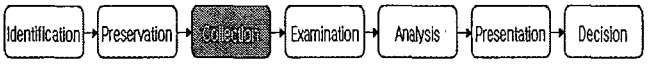
*Graduate School of Information Management and Security, Korea University

요약

본 논문에서는 라이브 시스템으로부터 패스워드를 획득하는 방법에 대해서 논하며, 이를 바탕으로 컴퓨터 관련범죄 발생 시 초기대응 과정 중에 사용할 수 있는 페이지파일 수집기를 구현하였다. 페이지파일 수집기를 이용하여 실 사용자들의 페이지파일을 수집하고 분석방법을 제시하였다. 또한 페이지파일로부터 어느 정도의 패스워드가 검출되는 가를 확인하였다. 이는 포렌식 수사에는 큰 도움을 줄 수 있지만, 포렌식 수사도구가 해킹을 위한 도구로 사용되었을 때에는 심각한 개인정보 유출을 야기할 수도 있다.

1. 서론

DFRWS에서 보여진 디지털 포렌식 수사과정은 Identification, Preservation, Collection, Examination, Analysis, Presentation, Decision으로 구분할 수 있다[1]. 이전 연구[4]에서 우리는



〈그림 1〉 디지털 포렌식 수사과정

Collection 단계에서의 증거 수집 절차의 표준인 RFC3227[2]를 언급하였다. RFC32227에서 제시하는 수집절차는 너무나 일반적인 내용을 담고 있어 실제 수사에 적용하기가 힘들다. 따라서 좀 더 세부적인 절차로 분류하였고, 그 중에서 메모리 정보를 수집하고 분석하는 방법에 대해서 중점적으로 제안하였다. 이러한 절차와 분석방법을 바탕으로 디지털 증거 수집도구를 구현하였다.

이전 연구[4]에서는 윈도우가 가상 메모리의 일부인 페이지파일을 점유하고 있어서 활성 시스템에서는 페이지파일 획득이 불가능 하였다. 따라서 Forensic CD나 혹은 일반 Live Linux CD[6]로 재부팅하여 파일을 수집하였으나[8], 후속연구의 일환으로 개발한 페이지파일 수집기는 윈도우가 실행되고 있는 활성 시스템에서도 페이지파일을 수집할 수 있도록 하였다. 이는 페이지파일을 수집하는데 소요되는 시간을 비약적으로 단축 시켜, 디지털 범죄 초동 수사에서 요구되는 신속성을 만족시키고 있다. LIVE Linux CD를 이용한 수집 방법은 대상 시스템의 종료, Linux CD 부팅, 파일 시스템 마운트(NTFS나 FAT 파일 시스템 사용을 위해), 외부 저장매체 마운트, 복사라는 과정으로 이루어진다. 이러한 절차는 약 1기가의 페이지파일을 획득할 경우, 평균적으로 15~20분 정도가 소요된다. 하지만 페이지파일 수집기는 활성 시스템에서 USB 외부 저장 장치를 연결하여 곧 바로 페이지파일을 획득할 수 있어 평균적으로 3~4분 정도 밖에 소요되지 않는다. 이는 컴퓨터의 사용자가 잠깐 자리를 비운 사이에 프로그램을 실행시켜 페이지파일을 추출해 가는 "Lunch Time Attack"을 가능하게 한다[3].

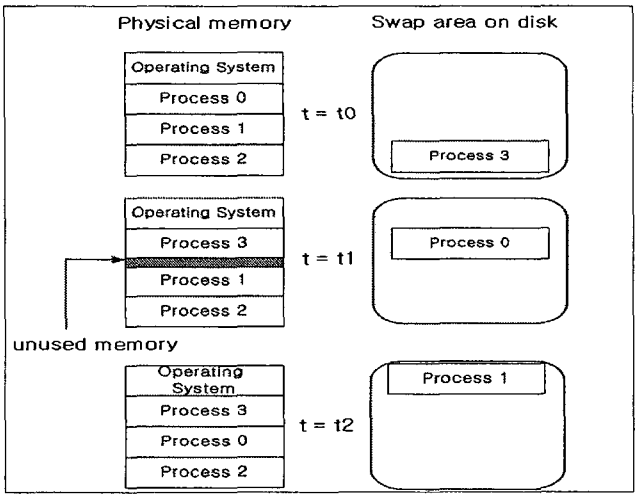
이러한 페이지파일 수집기는 포렌식 수사관 관점에서 볼 때는 매우 유용한 디지털 증거 수집 도구이지만, 해킹 도구로써 사용될 경우에는 심각한 프라이버시 정보가 유출될 수 있다.

2. 가상 메모리 시스템

현재 사용되고 있는 대부분의 컴퓨터 시스템은 가상 메모리 기법을 이용한다. 가상 메모리(Virtual Memory)는 컴퓨터와 운영체계에 의해 구현되는 개념으로서, 프로그래머에게 큰 용량의 메모리나 데이터 저장 공간을 사용할 수 있도록 허용하는 것을 말한다. 보조기억 장치를 마치 주기억 장치인 것처럼 이용하는 방법으로, 즉 실제 존재하지 않는 메모리를 있는 것처럼 사용하는 방법이다. 이때 주기억 장치와 보조 기억 장치 사이의 데이터

1) 본 연구는 과학재단 디지털 정보 획득 기반기술 연구(M10640010005-06N4001-00500)의 지원으로 수행되었습니다.

교환을 스왑핑(swapping)이라고 한다. 그리고 보조 기억 장치, 즉 하드 디스크에서 주기억 장치로 들어오는 것을 swap-in, 주기억 장치에서 하드 디스크로 물러나는 것을 swap-out이라고 한다[5].

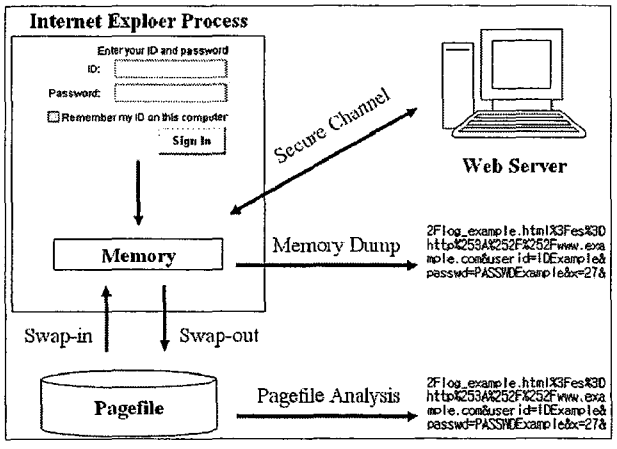


〈그림 2〉 Swap 과정

그림 1의 스왑과정은 다음과 같다.

1. 프로그램 실행
2. 메모리에서 중요하지 않다고 판단되는 프로세스 선택
3. 하드디스크의 스왑 영역으로 이동
4. 메모리에 빈 공간 확보
5. 프로그램 로딩

윈도우 NT/2000 시스템에서 사용하는 스왑 영역은 통상 실제 메모리 용량의 1.5배 크기로 C:\pagefile.sys 파일로 만들어지고 윈도우 9x에서는 C:\windows\win386.swp 파일로 만들어 진다.



〈그림 3〉 메모리 정보와 Pagefile 정보

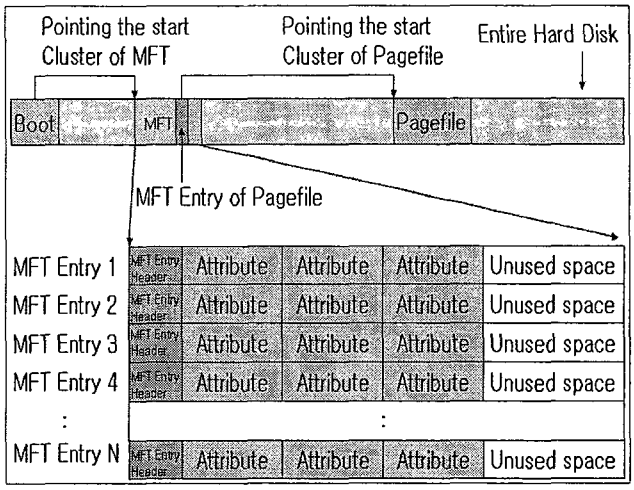
그림5를 보면 웹서버에 보안접속 과정을 거쳐 로그인을 하더라도 ID와 패스워드는 반드시 메모리에 적재되었다가 암호화되어서 통신을 하기 때문에 메모리에 암호화되기 전의 평문 상태가 그대로 남아있게 된다. 이러한 평문은 메모리 스왑 과정을 통해 하드디스크의 페이지파일에 저장될 수도 있다.

3. 페이지파일 수집 및 분석

가. 페이지파일 수집

기본적으로 시스템이 동작하고 있는 활성 상태에서는 운영체제가 페이지파일을 점유하고 있어서 일반적인 입출력을 사용한 방법으로는 파일 복사가 불가능하다. 하지만 본 연구를 통해 NTFS 파일 시스템을 분석하고 하드디스크로부터 직접 페이지파일을 읽어 들이는 페이지파일 수집기를 구현하였다. 이를 이용하여 공공장소에 존재하는 PC(인터넷 검색, 도서관의 도서 검색, 인터넷 카페)들을 대상으로 페이지파일을 수집하였다.

페이지파일 수집기를 구현하기 위해서는 먼저 NTFS 파일 시스템의 이해가 필요하다. 그중에서도 NTFS의 핵심이라고 할수 있는 $MFT파일의 구조를 분석해야 할 필요가 있다. 다음 그림4는 $MFT 파일의 구조를 도식화 한 것이다.



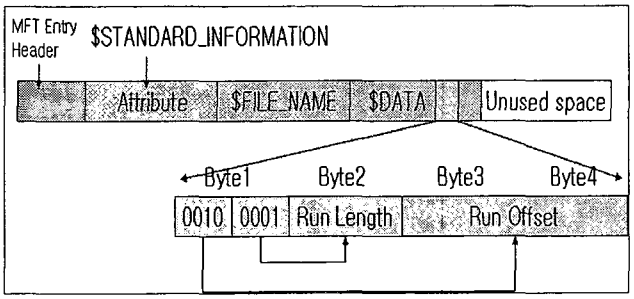
〈그림 4〉 $MFT 구조

하드디스크에 존재하는 모든 파일은 $MFT안에 MFT Entry를 반드시 가지고 있다. MFT Entry는 1KB의 크기의 매우 간단한 구조로 이루어져 있으며, 파일이름, 파일 크기, 파일의 물리적인 저장 위치 등의 정보를 저장하고 있다.

Boot Sector는 $MFT 파일의 주소를 가지고 있다. 앞에서 언급한 것과 같이 $MFT는 하드디스크의 모든 파일에 대한 Entry를 가지고 있다. 따라서 수집하고자 하는 페이지파일의 Entry 역시 $MFT안에 존재한다. $MFT 속에서 페이지파일 Entry를 찾아내고 하드디스크에 저장된 물리적인 위치를 찾아낸다면 직접 페이

지파일을 획득할 수 있다[7].

다음 그림5는 MFT Entry의 구조이다. Entry는 여러개의 Attribute로 이루어져 있는데, $DATA attribute에 파일의 물리적인 하드디스크 클러스터 위치와 크기가 저장되어 있다. 그림5와 같이 $DATA의 특정 부분중에 Run Offset는 하드디스크의 처음부터 해당 파일이 존재하는 시작 클러스트의 위치를 나타내고 있다. 그리고 Run Length는 시작클러스터로부터 파일이 할당되어 있는 길이를 나타내고 있다. Run Length와 Run Offset을 합하여 Runlist라고 부르며, Runlist의 정보에 따라서 페이지파일을 하드디스크로부터 직접 획득할 수 있다[7].



〈그림 5〉 $Data의 RunList

나. 페이지파일 분석

페이지파일은 텍스트 문자열, 바이너리 코드 값 등이 섞여 있어 사용자가 쉽게 인식할 수 없는 형태이다. 따라서 페이지파일을 수집하였을 때에는 사람이 읽을 수 있는 형식으로 변환하는 작업이 필요하다. 본 논문에서는 가상 메모리 파일로부터 주요 키워드를 추출하는 필터링 프로그램을 작성하였다. 여러 시스템으로부터 획득한 페이지파일은 필터링 프로그램을 통해 가독성 있는 형태로 변경한 다음에 분석하였다. 가상 메모리 시스템을 사용하기 때문에 페이지파일에는 swap-out된 정보가 하드 디스크에 그대로 남아 있게 된다. 그래서 페이지파일로부터 다음과 같은 정보들을 획득할 수 있었다[4].

[표 1] 스왑파일(pagefile) 내의 정보

1. 문서의 글자
2. ID, 패스워드
3. 사용자가 보았었던 내용(URL, 메신저 내용)
4. 타이핑했던 정보
5. 기타 정보

여러 가지 정보 중에서 특히 패스워드를 중심으로 조사를 하였는데 결과는 다음과 같다.

[표 2] Pagefille 패스워드 검사 결과

전체 검사 수	57개
패스워드 발견 시스템	26개

[표 3] Pagefile의 크기 별 패스워드 존재 확률

Pagefile 크기	메모리 크기	검사 파일 수	패스워드 존재 수
256M 미만	192M미만	4개	0개
384M	256M	7개	5개
768M	512M	27개	14개
1G 이상	768M이상	15개	9개

표 2를 보면 약 페이지파일의 절반 정도에 패스워드가 존재하는 것을 확인할 수 있으며, 표 3을 보면 Pagefile의 크기가 768M 바이트 이상일 경우 패스워드의 존재 확률이 약 43% 이상임을 알 수 있다. 요즘 PC 사양은 메모리 크기가 대부분 512M 바이트 이상임을 고려할 때 프라이버시 정보가 더욱 많이 존재할 것으로 예상되고, 페이지파일이 유출된다면 다음과 같은 문제점이 발생할 수 있다[4].

첫째, 패스워드, 주민등록번호 등 중요 정보가 유출될 수 있다. 특히 대부분의 사용자는 ID/패스워드를 여러 가지 사용하지 않고, 동일한 ID와 패스워드를 사용한다. 유출된 ID와 패스워드를 사용해서 인터넷 사이트 등에 로그인을 한다면, 인종 및 민족, 사상 및 신조, 출신 및 본적지, 정치적 성향 및 범죄기록, 건강상태 등 사용자의 기본적 인권을 현저하게 침해할 우려가 있는 정보가 추가적으로 유출될 수 있다.

둘째, 사용자의 과거 시스템 사용기록 및 성향을 유추가능하게 한다. 페이지파일에는 과거 메모리 사용 내역이 기록되어 있다. 따라서 이를 분석하면 시스템을 사용한 내역을 유추하여, 프라이버시 정보가 침해당할 가능성이 있다.

셋째, 사용자가 문서를 열람 후, Anti-Forensic[9] 기법으로 완벽히 삭제한다고 하더라도 페이지파일에 남아 있을 가능성이 있어 중요한 정보가 노출 될 수 있다.

4. 결론

본 논문에서는 윈도우 시스템의 페이지파일을 수집하고 분석해서 프라이버시 정보, 특히 패스워드 정보가 약 절반 정도로 검출되는 것을 확인할 수 있었다. 따라서 페이지파일이 해킹의 표적이 될 가능성이 매우 높은 것으로 판단된다. 비단 패스워드뿐만 아니라 이름, 아이디(ID), 주민등록번호, 주소, 전자메일주소, 휴대폰번호, 자택 전화번호에 이르는 개인의 사적인 정보가 무방비 상태로 완전 노출될 가능성이 있었고 일부는 노출되어 있었기 때

문이다. 페이지파일 수집기는 수사관의 입장에서는 매우 유용한 수사도구 이지만, 해킹의 도구로서 사용될 경우 심각한 피해를 일으킬 수 있다. 페이지파일은 매우 손쉽게 획득이 가능하기 때문에 해킹의 표적이 될 수 있다. 따라서 반드시 보호해야할 필요성이 있다. 향후 연구로는 프로그램 개발단계에서부터 프라이버시를 정보의 유출을 방지하는 프로그래밍 기법과 시스템 커널단계에서 제공하는 모델을 제안하고자 한다. 또한 제안된 프로그래밍 기법에 따라서 실제 응용프로그램을 구현하여 테스트 결과를 도출할 예정이며, 시스템 모델에 대한 좀 더 세부적인 사항을 디자인해보고자 한다.

참 고 문 헌

[1] G. Palmer, "A Road Map for Digital Forensic Research", Utica, New York, technical report DTR-T001-0, 2001.

[2] RFC3227, "Guidelines for Evidence Collection and Archiving", http://www.faqs.org/rfcs/rfc3227.html, 2002.

[3] Adi Shamir and Nicko van Someren, "Playing hide and seek with stored key", September 22, 1998, Lecture Notes in Computer Science

[4] 이석희, 김현상, 이상진, 임종인, "윈도우 시스템에서 디지털 포렌식 관점의 메모리 정보 수집 및 분석 방법에 관한 고찰", 한국정보보호학회 논문집, 제16권 1호, 2006년 2월

[5] A. Silberschatz, P. Galvin, "Operating System Concepts", fifth edition, 1998.

[6] Kyle Rankin, "KNOPPIX HACKS", O'RELLY, pp. 256-263

[7] Brian Carrier, "FILE SYSTEM FORENSIC ANALYSIS", Addison Wesley, pp. 273-396

[8] Douglas Schweitzer, "Incident Response:Computer Forensics Toolkit", Wiley Publishing Inc. 2003

[9] S. Rekhis, N. Boudriga, "Formal Forensic Investigation Eluding Disk-based Anti-Forensic Attacks,", Workshop on Information Security Applications, Jeju Island, Korea, August 2005.