디지털 포렌식 관점에서의 $LOGFILE 분석 및 활용방안*

이동은, 변근덕, 이상진
고려대학교 정보보호대학원

Analysis and Practical Use of a $LOGFILE in Digital Forensic

Dong Eun Lee, Byun Geun Deok, Lee Sang Jin
Center for Information Security Technologies, Korea University

요 약

본 논문에서는 NTFS 파일시스템의 $LOGFILE을 활용하여 범인의 행동 흐름 파악 메커니즘을 제시한다. 기존에 Carrier, Brian에 의해 제시된 NTFS파일 시스템의 분석은 주로 할당되지 않은 $MFT의 분석을 통해서 이루어졌다.[1] 증거물을 포착하기 위해서 사용되었던 $MFT 분석 방식을 벗어나 $LOGFILE을 통해 범죄자의 파일에 대한 조작 행위의 순서를 파악한다.

I. 서론

최근 PC의 보급과 인터넷 사용인구의 증가에 따라 컴퓨터는 생활필수품이 되었다. 이에 따라 오프라인의 활동이 온라인 환경으로 이동하고 있으며, 오프라인에서 이루어지던 범죄 또한 온라인에서 발생하고 있다. 또한 오프라인의 범죄에서도 직간접적으로 컴퓨터를 사용하고 있다.

컴퓨터 포렌식은 컴퓨터를 이용한 범죄에 대해 조사 및 수사를 지원하고 추후 수집된 증거가 법적 효력을 갖도록 절차와 방법을 연구하는 학문이다. 따라서 컴퓨터를 이용한 범죄의 증가에 따라 컴퓨터 포렌식의 중요성이 대두되고 있다.

컴퓨터 포렌식은 사고 대응 준비, 증거 수집, 증거 분석, 보고서 작성으로 이루어져 있으며, 본 논문에서는 증거 분석 절차 중 $LOGFILE에 대해 고찰해 보고자 한다.

기존의 연구에서 분석의 대상으로. 하드디스크의 슬랙 공간, 페이지 파일[2], 레지스트리 정보 등이 있으나 $LOGFILE의 경우 그 구조가 많이 알려져 있지 않고, 동작 구조만이 알려져 있을 뿐이다. 본 논문에서는 이러한 $LOGFILE의 구조에 대해 자세히 분석하고, 기록되어 있는 정보에 대해 포렌식 관점에서 어떻게 활용될 수 있는지 알아본다.

II. $LOGFILE의 개요

NTFS는 시스템 오류가 생기거나 갑작스런 전원 공급 차단이 일어났을 때 작업 중이던 파일을 복구하기 위해 $LOGFILE을 사용한다. NTFS는 $LOGFILE에 "레코드" 단위로 로그를 남기고 "레코드"에 저장된 정보를 조합하여 복구를 시작한다. 하지만 $LOGFILE이 모든 파일의 복구를 책임지는 것은 아니다. $LOGFILE은

* 본 연구는 정보통신부 대학 IT 연구센터 육성·지원 사업의 연구 결과로 수행되었습니다.

디스크 총용량의 12%를 사용하기 때문에 최근에 작업한 파일에 한하여 복구를 실행한다. 본 논문은 복구 목적의 $LOGFILE을 디지털 포렌식 수사에 활용하는 방안을 제시한다.

III. $LOGFILE의 데이터 구조

$LOGFILE은 두 영역으로 나누어져 있으며 그 두 영역은 "재시작 영역"과 "로깅 영역"이다. 그림 1과 같이 로그 파일은 특정 개수의 페이지를 보유하고 있으며 한 페이지 안에는 여러 개의 레코드가 포함되어 있다. 여러 개의 페이지 중 앞 단의 두 페이지를 재시작 영역이라 하고 그 나머지 페이지를 로깅 영역이라 한다.

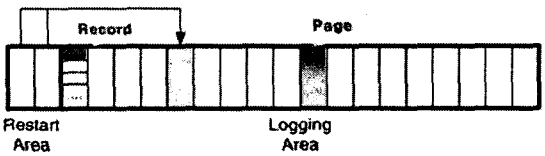

그림 1 $LOGFILE의 구조

2.1 재시작 영역

"재시작 영역"은 현재 사용되고 있는 "로깅영역"을 LSN($LOGFILE Sequence Number)을 통해 가리킨다. LSN은 고유한 일련 번호로 각 레코드에 0x10씩 증가하면서 매겨진다. "로깅 영역"의 각 레코드는 이 값을 "This LSN(그림 5)"항목에 저장한다.

그림 3에서 볼 수 있듯이 "재시작 영역"은 "Current LSN"항목을 33바이트 오프셋에 저장한다[3]. 이것은 현재 사용 중인 "로깅 영역"의 "Last End LSN"과 동일한 값을 갖는다. "Last End LSN"은 현 페이지에 존재하는 레코드 중 가장 마지막 레코드의 LSN을 뜻하고 그림 4와 같이 33바이트오프셋에 위치한다. 그림 2에서 이를 확인해 보면 "Current LSN"과 "Last End LSN"이 "FC 92 5D 24 00 00 00 00"으로 동일함을 알 수 있다. 결국 "Current LSN"은 "Last End LSN"을 가리키는 지시자 역할을 한다.

이를 바탕으로 수사관은 "재시작 영역"을 분석하여 디스크 획득 당시 범인이 작업 중이던

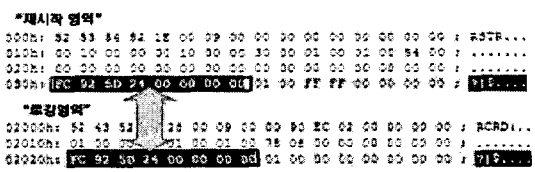

그림 2 재시작영역"과 "로깅영역"의 LSN

파일의 페이지를 수집 할 수 있다. 페이지의 각 레코드는 구체적으로 범인이 작업을 수행한 시간 정보 및 작업 종류를 포함하고 있기 때문에 수사관은 범인의 마지막 행동에 대한 정보를 얻을 수 있게 된다.

2.2 로깅 영역

"로깅영역"의 페이지는 페이지 헤더와 여러 개의 레코드로 구성되어 있다. 페이지 헤더는 페이지의 메타 정보를 포함하고 있고 레코드는 레코드의 메타정보와 복구에 필요한 정보를 포함한다. 레코드의 데이터 구조는 그림 5와 같으며 본 절은 디지털 포렌식 관점에서 중요 정보를 차례로 분석한다.

"This LSN"과 "Previous LSN"은 현재 레코드의 LSN과 이전 레코드의 LSN을 말한다. "This LSN"과 "Previous LSN"으로 하나의 리스트를 구성할 수 있으며 이것으로 레코드 생성 순서를 알 수 있다.

"TransactionID"는 이벤트의 종류를 의미한다. 각 파일에 사용자나 시스템이 "이벤트"를 발생할 때마다 레코드가 생성되며 그 이벤트의 종류는 "TransactionID"에 기록된다. 이벤트는 "쓰기", "읽기", "수정"등 사용자의 파일에 대한 조작을 가리킨다.

"Creation Time", "Modified Time", "MFT Modified Time", "Access Time"은 생성시간, 수정시간, MFT수정시간, 접근 시간을 뜻한다.[4]

이 세 가지 정보를 결합하면 사용자가 이벤트를 발생한 파일의 "Creation Time", "Modified Time", "MFT Modified Time", "Accessed Time"을 레코드의 생성 순서대로 인지할 수 있다.

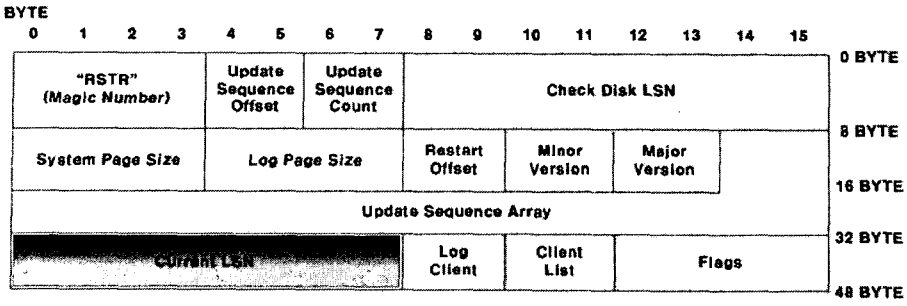

그림 3 "재시작 영역"의 데이터 구조

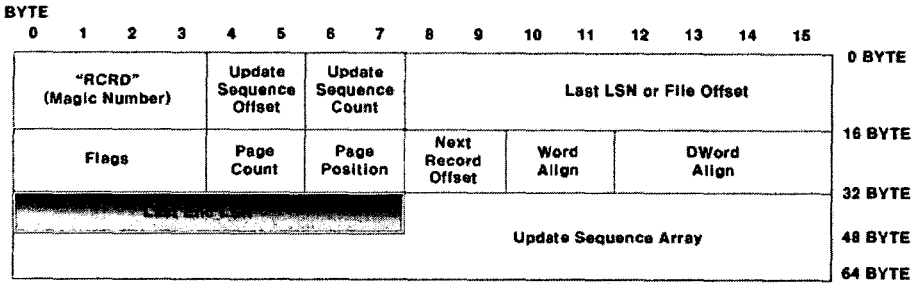

그림 4 "로깅 영역 헤더" 데이터 구조

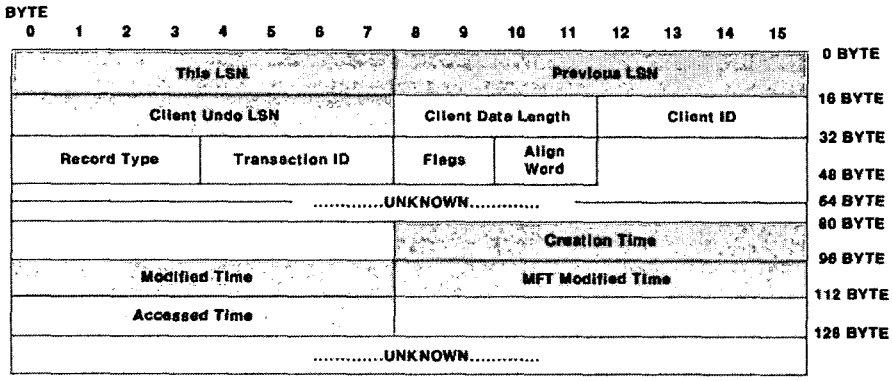

그림 5 "로깅 영역"에 존재하는 레코드의 데이터 구조

수사관은 이것을 이용하여 범죄자가 실행한 이벤트의 종류와 그 시간정보를 실행순서대로 알 수 있다.

IV. 파일 검색 메커니즘

$LOGFILE의 레코드는 메타 정보 및 복구에 필요한 정보를 담고 있지만 파일의 이름이나 위치에 대한 정보는 없다.

파일의 이름이나 위치 정보를 알아내기 위해서는 레코드의 "This LSN"과 MFT 엔트리의 "LSN"이 필요하다. 레코드의 LSN과 MFT엔트리의 LSN이 동일할 때 MFT엔트리에서 파일 이름과 위치의 정보를 획득할 수 있다. MFT엔트리의 LSN은 시작지점에서 9바이트 떨어진 위치에 존재한다.[1]

V.결론

로그 파일은 두 가지 영역으로 나누어 지고 각 영역은 고유의 데이터 구조를 지니고 있다.

그 두 가지 영역은 재시작 영역과 로깅 영역이며 재시작 영역은 현재 발생한 이벤트를 기록중인 로깅영역의 로그 페이지를 가리키고 있다. 또한 로깅 영역은 실제 수사관이 얻을 수 있는 정보들의 집합체이다. 따라서 수사관이 주목할 로그파일의 부분은 "로깅 영역"이다.

"로깅 영역"의 중요 정보는 LSN과 TransactionID와 시간정보이다. LSN은 각 레코드마다 증가하는 구분자의 역할을 하며 TransactionID는 이벤트의 종류를 나타낸다. 마지막으로 시간정보는 파일에 이벤트가 행하여진 시간을 말한다. 각 레코드는 또한 세 가지의 LSN을 보유하는데 그 세 가지 LSN은 This LSN, Previous LSN, Next Undo LSN이다. 각 레코드의 This LSN과 Previous LSN을 연결하면 하나의 리스트로 구성할 수 있으며 그 리스트는 수사관이 범인의 행동 흐름을 파악할 수 있게 해준다.

따라서 수사관이 $LOGFILE을 분석하여 범죄가 일어난 시각과 특정 시각에 PC에서의 범인의 행적을 로그파일로부터 추적해야 한다.

[참고문헌]

[1] Carrier, Brian, "File System Forensic Analysis", Addison-Wesley, 2005, p273~p396

[2] 이석희, 김현상, 이상진, 임종인, "윈도우 시스템에서 디지털 포렌식 관점의 메모리 정보 수집 및 분석 방법에 관한 고찰", 정보보호학회 논문지, 2006년 1월

[3] Richard Russon, Yuval Fledel, http://www.linux-ntfs.org/content/view/104/43/

[4] 안정수, 김권엽, 이상진, 임종인, "윈도우 시스템에서 시간 조작 탐지 방법", KoreaCrypt, 2005년 11일