MFT 분석기술을 이용한 Alternate Data Stream 탐지 기법

김요식* · 류재철** · 박상서*

요 약

NTFS의 ADS는 매킨토시의 계층적 파일 시스템과의 호환을 위해 개발되었으나 최근에는 악의적 사용자들에 의해 악성코드 또는 안티 포렌식 목적의 데이터 은닉 용도로 활용되고 있다. 은닉된 ADS의 존재여부를 파악하고 정보를 추출하는 것은 컴퓨터 포렌식 분야에서 중요한 요소이다. 본 논문에서는 NTFS의 MFT정보를 이용하여 ADS를 탐지하기 위한 방법을 제안하였다. 이 방법을 구현하여 비교·실험한 결과, 기존의 방법에 비해 검색속도와 탐지건수 면에서 우수함을 확인하였다. 이 방법을 이용하면 운영체제에서 사용중인 파일도 검사할 수 있으며, 라이브 시스템뿐 아니라 이미지에 대해서도 탐지가 가능해 포렌식 목적에 부합된다.

Alternate Data Stream Detection Method Using MFT Analysis Module on NTFS

Yosik Kim* · Jaecheol Ryou** · Sangseo Park*

ABSTRACT

Alternate Data Streams (ADS) in NTFS originally has developed to provide compatibility with Macintosh Hierarchical File System. However, it is being used by the malware writers in order to support hiding malwares or data for the purpose of anti-forensics. Therefore identifying if hidden ADSs exist and extracting them became one of the most important component in computer forensics. This paper proposes a method to detect ADSs using MFT information. Experiment reveals that proposed method is better in performance and detection rate then others. This method supports not only identification of ADSs which are being used by the operating systems but also investigation of both live systems and evidence images. Therefore it is appropriate for using forensic purpose.

Key words : Alternate Data Stream, NTFS, Master File Table

* ETRI 부설연구소
** 충남대학교 정보통신공학부 교수

1. 서 론

디지털 포렌식에서는 은닉되어 있는 증거 데이터를 찾아내는 것이 중요한 요소 중 하나이다. 이를 방해하기 위한 목적으로 다양한 은닉 프로그램들이 제작되고 있는데, NTFS의 Alternate Data Stream이 이 용도로 일부 활용되고 있다.

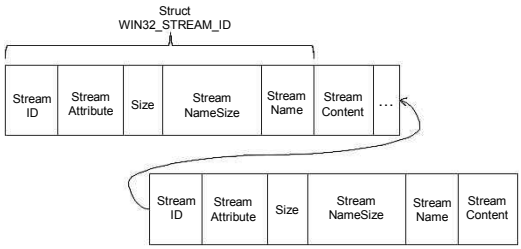
매킨토시의 HFS(Hierarchical File System)계층적 파일 시스템과의 호환을 위해 개발된 NTFS의 Alternate Data Stream[1]이 보안 위협요소로 활용될 수 있다는 내용이 보고되고 있으며 실제로 악성코드와 다양한 종류의 파일 및 메타 데이터 정보가 NTFS를 사용하는 Windows 2000, XP 등에서 사용되어 지고 있다. NTFS Alternate Data Stream의 원래 목적은 Data Fork, Resource Fork의 두 가지 파트로 구성되는 매킨토시 HFS용 파일 시스템의 호환성을 지원하기 위함이다. Data Fork가 윈도우의 Main Unnamed Stream 즉, 메인 파일을 의미하며, Resource Fork가 Alternate Data Stream과 같은 역할을 수행하고 있는 구조이다. 하지만, Alternate Data Stream은 악성코드와 같은 실행파일을 은닉 시킬 수 있다는 측면에서 포렌식에 위협요소로 작용될 수 있으며 수사를 회피하기 위한 안티 포렌식 목적으로 대량의 데이터를 은닉시키기 충분하다[2, 3].

본 논문에서는 Alternate Data Stream를 검색·탐지 하기위한 새로운 방법을 제안하고 구현하여 실험한다. 본 논문의 구성은 다음과 같다. 먼저, 제 2장에서는 Alternate Data Stream의 특징에 대해 간략히 살펴보고, 제 3장에서는 Alternate Data Stream을 위한 제안된 기법을 기술하고, 제 4장에서는 제안된 기법을 구현하여 시험·분석하며, 제 5장에서는 본 연구의 결론 제시한다.

2. Alternate Data Stream

NTFS의 MFT(Master File Table) 엔트리내에 저장되는 여러 속성 중 $DATA 속성은 파일의 내용을 담고 있는 속성이다. 하나의 파일이나 디렉토리는 여러 개의 $DATA속성을 가질 수 있다. $DATA 속성 외에 추가적으로 존재하는 $DATA 속성을 Alternate Data Stream 속성이라 하며 이러한 집합체를 Multiple File Stream이라고 한다[4, 5].

Alternate Data Stream이 사용되는 예는 다양하다. 윈도우 탐색기를 통하여 확인할 수 있는 파일 등록정보의 요약부분인 메타 데이터와 인터넷 익스플로러를 이용하여 인터넷으로부터 파일을 다운로드했을 경우에 다운로드한 파일에 26바이트 크기를 가지는 Zone.Identifier라는 Stream이 그 예라 할 수 있다. 그리고 임의 폴더에 JPG, GIF 등의 이미지 파일이 있을 경우 윈도우 탐색기의 미리보기 옵션을 활성화시키면 해당 폴더에 Thumbs.db 파일이 생성되면서 동시에 0바이트 크기를 가지는 encryptable이라는 이름의 ADS 속성이 추가된다. Alternate Data Stream은 연접해서 계속 추가될 수 있는데 (그림 1)은 Main Unnamed Stream 뒤에 연접해서 구성된 Alternate Data Stream의 구조를 나타낸다.



(그림 1) Alternate Data Stream 구조

Alternate Data Stream은 별도의 도구를 이용해야만 존재여부를 확인할 수 있을 뿐 윈도우 탐색기나 dir 명령을 통해서는 확인할 수 없다. 또한, Alternate Data Stream은 파일이기 때문에 실행이 가능하다는 특성을 갖는다. 즉, 메인 파일이외에도

보이지 않는 여러 개의 실행 가능한 Stream을 가질 수 있기 때문에 악성코드를 은닉시키는데 악용되기도 한다[6-8].

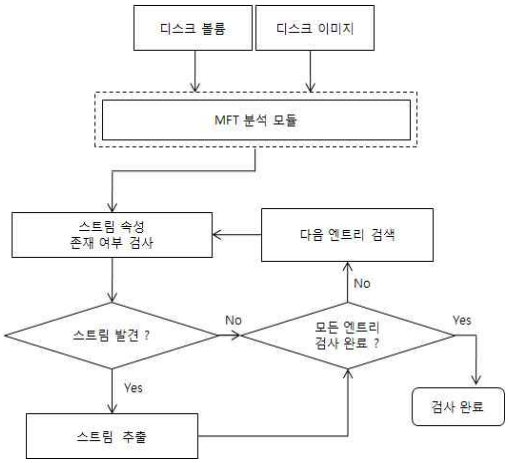
LADS, ADSView에서는 Alternate Data Stream을 식별하기 위하여 마이크로소프트에서 제공하는 File Management 함수군[9]을 사용하여 볼륨 전체의 디렉토리와 파일을 반복 검색하고 Backup 함수군[10]을 사용하여 Stream 정보를 추출하는 것으로 파악되었다. 이 방법은 각 디렉토리와 파일의 수가 많아지게 되면 디렉토리와 파일을 검색하는 API 호출 횟수가 많아져 오버헤드가 발생하게 된다. 뿐만 아니라, 각 파일을 직접 액세스하여 Backup 함수군을 이용하여 Stream 정보를 읽고 헤더 정보를 추출한 뒤 Stream 파일의 이름 크기를 검사하는 루틴이 동작해야 하므로 Alternate Data Stream의 검색・식별에 많은 시간을 필요로 하게 된다. 또한, 라이브 시스템에 대해서는 운영체제가 사용중인 파일에 대해서는 Alternate Data Stream의 존재여부를 분석할 수 없는 단점도 있다. 더구나, 포렌식 목적으로 획득된 이미지에 대해서는 전혀 검출이 불가능하다.

3. MFT 분석을 통한 검색・탐지

3.1 Alternate Data Stream 탐지 방법

본 논문에서는 Alternate Data Stream을 탐지하기 위해 볼륨에 존재하는 모든 파일과 디렉토리에 대한 정보를 담고 있는 MFT 엔트리 전체를 분석하는 방법을 제안한다. MFT는 볼륨에 존재하는 모든 파일과 디렉토리의 정보를 담고 있는 테이블이다. 그러므로 이 테이블을 분석하면 볼륨에 있는 모든 파일과 디렉토리에 대한 정보를 알아낼 수 있다. 즉, MFT 엔트리 정보를 이용하면 증거 디스크 이미지 분석을 통하여 Alternate Data Stream이 존재하는지 식별이 가능하게 된다. 이 방법을 이용하면 라이브 시스템 뿐만 아니라 포렌식 목적으로 획득된 디스크 이미지에 대해서도 Alternate Data Stream을 탐지할 수 있다.

(그림 2)에서 보는 바와 같이 MFT분석 모듈을 이용하여 검색 대상이 되는 디스크 볼륨과 디스크 이미지 모두를 대상으로 Stream을 검색・식별한다. 먼저, Stream 속성이 존재하는지를 검사하여 Stream이 존재할 경우 Stream ID, Stream 속성, Stream 이름, Stream 크기, Stream 내용 등을 추출한다. 모든 MFT엔트리에 대해 검사가 완료될 때까지 반복하여 검색한다.



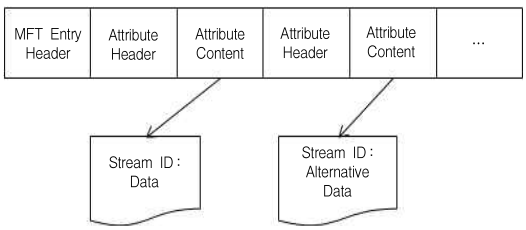
(그림 2) Alternate Data Stream 탐지방법

3.2 MFT 엔트리 분석 방법

디스크 이미지로부터 Alternate Data Stream을 탐지하기 위해서는 윈도우 시스템에서 적용된 File Management, Backup와 같은 Win32 API를 직접적으로 사용하기 어렵다. 따라서 디스크 이미지를 마운트하여 가상의 디스크로 윈도우 시스템에서 인식시킨뒤 Win32 API를 사용하거나 직접 디스크 이미지가 가지는 NTFS 파일 시스템의 MFT 엔트리를 분석하여야 한다. 본 논문에서는 후자의 방법을 이용한다.

MFT 엔트리는 MFT 엔트리 헤더와 그 외의 빈공간으로 이루어져 있으며 그 빈 공간에 속성이 저장되어 있는 형태이다. 속성으로는 디렉토리나 파일의 최근 접근시간, 생성시간, 파일의 접근 제어와 보안속성, 파일시스템의 버전 등이 저장된다. Alternate Data Stream이 존재할 경우에는 여러 개의 $DATA 속성이 존재하며 원본 즉, Visible한 파일에는 파일명이 존재하지 않지만, Stream 데이터의 경우에는 Alternate Data Stream에 대한 파일 이름이 존재한다. $DATA 속성을 검사하고 $DATA가 2개 이상일 경우 이름의 크기를 검사하여 0보다 크면 Alternate Data Stream이 존재하는 것으로 판단할수 있다.

따라서, 라이브 시스템 뿐 아니라 이미지에 대해서도 MFT의 엔트리를 분석하고, 파싱함으로써 Alternate Data Stream이 존재하는지 검사하고, 숨겨진 내용을 복원할 수 있다. (그림 3)은 MFT 엔트리내의 속성 정보내에 Alternate Data Stream이 존재할 경우를 나타낸다. 첫 번째 속성의 Stream ID는 BACKUP_DATA를 가지며 두 번째 Stream의 경우에는 Stream ID가 BACKUP_ALT ERNATE_DATA가 된다.



(그림 3) MFT 엔트리 속성 정보

4. 시험 및 분석

본 논문에서는 라이브 시스템과 디스크 이미지로부터 Alternate Data Stream을 탐지하기 위해서 NTFS의 MFT 엔트리 분석 기능, NTFS의 MFT 엔트리 정보 파싱 기능, 그리고 식별된 Alternate Data Stream 정보 추출 기능을 구현하였다. 또한, 비교를 위하여 디스크 볼륨분석 모듈도 구현하였다. 시험은 Windows XP의 서비스팩 2가 설치된 라이브 시스템 환경에서 수행되었다. 시험에 사용한 하드 디스크는 <표 1>과 같으며, 각각의 볼륨내의 디렉토리 개수, 파일 개수 및 MFT 엔트리 개수는 <표 2>와 같다.

<표 1> 디스크 정보

구분	크기	RPM	파일시스템	사용공간
HDD1	80G	7500	NTFS	78GB
HDD2	250GB	7500	NTFS	95GB
HDD3	250GB	7500	NTFS	119GB

<표 2> 디렉토리, 파일, MFT 엔트리 수

구분	디렉토리수	파일수	MFT 엔트리수
HDD1	1,053	15,903	42,966
HDD2	6,972	82,958	83,185
HDD3	11,532	123,297	157,153

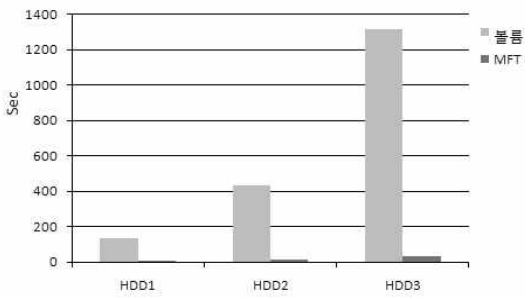
볼륨 분석 모듈을 사용하여 라이브 시스템상에서 Alternate Data Stream을 검색 · 탐지하기 위해서는 File Management 함수군을 사용하여 디렉토리와 파일을 핸들링해야 하기 때문에 계속 하드 디스크에 접근하게 된다. 따라서 실험결과 디렉토리 수와 파일 수에 비례하여 처리 속도는 더 오래 걸리는 현상을 확인하였다. <표 3>과 같이 디스크 볼륨분석 모듈이 파일 한 개당 Alternate Data Stream을 탐색하는데 걸리는 시간은 0.016~1.23초 가량 소요되었다. 반면 MFT 분석 모듈은 HDD1의 250GB의 볼륨에 대해 약 150MB 크기의 MFT 정보를 메모리에 적재하는데 필요한 초기지연시간이 12.281초의 시간이 소요되었으며 MFT 엔트리 한 개당 Alternate Data Stream을 탐색하는데 걸리는 시간은 0.001~0.423초 가량 소요되었다.

<표 3> 검색·탐지 시험 결과

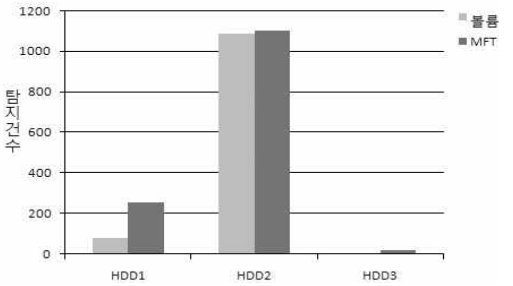
구분	디스크 볼륨분석기법			MFT 분석기법		
	HDD1 (80GB)	HDD2 (250GB)	HDD3 (250GB)	HDD1 (80GB)	HDD2 (250GB)	HDD3 (250GB)
디렉토리/파일수	16,956	89,930	134,829	-	-	-
MFT 엔트리/삭제파일수	-	-	-	42,966/23,784	83,185/5	157,153/16,568
초기지연시간(Sec)	0.34	1.95	4.43	3.45	5.01	12.281
전체 검색·탐지 시간(Sec)	134.12	427.89	1310.20	8.81	11.36	32.68
ADS 탐지건수	82	1,087	7	255	1,094	22
ADS 보유파일 건수	82	1,041	6	208	1,049	20

파일이나 디렉토리의 수가 많아질수록 MFT의 크기가 더 커지게 되며, 파일이나 디렉토리를 삭제하여도 한번 늘어난 MFT의 크기는 줄어들지 않는다. 즉, 오래 사용되고 대량의 파일이 존재하는 경우 MFT의 엔트리수가 많아져 실제 존재하지 않는 파일에 대해서도 엔트리를 분석해야 하는 오버헤드가 발생한다. 하지만, 실험결과 (그림 4)에서 보는 바와 같이 디스크 볼륨분석 모듈을 이용한 방법보다 MFT 분석 모듈을 이용한 방법의 검색·탐지 처리속도가 빠르다는 것을 확인하였다. 그리고 (그림 5)와 같이 MFT 분석 모듈을 이용한 탐지 방법이 더 많은 Alternate Data Stream 건수를 탐지하였다. 그 이유는 File Management 함수군으로 접근하지 못하는 디렉토리나 파일 중에는 MFT 엔트리로만 접근이 가능한 것이 존재하기 때문이다. 또한, 디스크 볼륨분석 모듈은 운영체제가 사용하는 일부 파일에 대해서는 접근이 불가능하여 처리하지 못하였으나, MFT 분석 모듈은 오류없이 정상적으로 처리하였다.

두 번째로 동일한 하드디스크를 대상으로 AFF 포맷으로 디스크 이미지를 생성한 후, 실험한 결과 MFT 분석 모듈이 오류없이 정상적으로 처리함을 확인하였다.



(그림 4) 검색·탐지 소요 시간



(그림 5) Alternate Data Stream 탐지건수

5. 결 론

본 논문에서는 Alternate Data Stream을 검색·탐지하기 위해 MFT 정보를 이용한 방법을 제안하고 시험하여 기존의 방법보다 우수함을 보였다. MFT 분석을 이용한 탐지 방법은 File Management와 Backup 함수군을 사용하는 LADS와 ADS Viewer보다 처리속도와 탐지건수에서 월등히 우수함을 알 수 있었고, 일부에서는 운영체제가 사용중인 파일의 경우도 검사 대상에 포함시킬 수 있으로 효과적 방법임을 확인하였다. 또한 본 방법을 이용하면 기존 방법에서는 불가능했던 이미지

대상의 탐지가 가능하여 디지털 포렌식을 위한 적절한 방안이라 할 수 있다.

참 고 문 헌

[1] Microsoft, “A Programmer’s Perspective on NTFS 2000 Part 1 : Stream and Hard Link”, http://msdn2.microsoft.com/en-us/library/ms810604.aspx.

[2] Daniel Bachfeld, “Dangers from the Twilight Zone”, http://www.heise-security.co.uk/articles/74892.

[3] Windowsecurity, “Hidden Threat : Alternate Data Streams”, http://www.windowsecurity.com.

[4] Irongeek.com, “Practical Guide to Alternative Data Streams in NTFS”, http://www.irongeek.com/i.php?page = security/altds.

[5] Derek Bem and Ewa Z. Huebner, “Alternate Data Streams in Forensic Investigation of File Systems Backups”, May 2006.

[6] globalknowledge.com, “Alternate Data Streams What’s Hiding in Your Windows NTFS?”, http://www.globalknowledge.com.

[7] Diamondcs.com, “Hidden NTFS Alternate Data Streams (ADS) Explained”, http://www.diamon dcs.com.au/web/streams/streams.htm.

[8] Damon Martin, “Windows, NTFS and Alternate Data Streams”, http://www.giac.org/certified_professionals/practicals/gsec/0715.php.

[9] Microsoft, “File Management Functions”, http://msdn2.microsoft.com/en-us/library/aa364232.aspx.

[10] Microsoft, “Backup Functions”, http://msdn2.microsoft.com/en-us/library/aa362512.aspx.

김 요 식

1997년～1999년 (주)지란지교 소프트 연구원
2000년～2004년 (주)케이사인 선임연구원
2005년 공주대학교 바이오정보 학과(석사)
2004년～현재 ETRI 부설연구소 연구원
2007년～현재 충남대학교 컴퓨터공학과 박사과정

류 재 철

1985년 한양대학교 산업공학과(학사)
1988년 Iowa State University (전산학 석사)
1990년 Northwestern University(전산학 박사)
1991년～현재 충남대학교 정보통신공학부 교수

박 상 서

1991년 중앙대학교 전자계산 학과(공학사)
1993년 중앙대학교대학원 전자계산학과(공학석사)
1996년 중앙대학교대학원 컴퓨터공학과(공학박사)
1996년～1998년 국방정보체계 연구소 선임연구원
1999년～2000년 국방과학연구소 선임연구원
2000년～현재 ETRI 부설연구소 선임연구원