https://doi.org/10.33778/kcsa.2023.23.1.097

안티 포렌식 동향 분석 및 대응 방안 연구

한 현 동*, 조 영 준*, 조 재 연*, 김 세 온*, 한 완 섭*, 최 용 준*, 이 정 훈*, 김 민 수**

요 약

4차 산업혁명 시대에 디지털기기 보급의 대중화와 함께 이를 대상으로 한 사이버 범죄가 증가함에 따라 디지털 데이터 증거 확보의 중요성이 대두되고 있다. 하지만 디지털 데이터 증거 확보의 어려움은 디지털 데이터를 조작, 삭제, 난독화 등 분석 시간을 증가시키거나 불가능하게 만드는 안티 포렌식 기법을 활용하기 때문이다. 이러한 안티 포렌식은 디지털 포렌식 측면에서 증거물의 회손 및 차단하는 일련의 행위로 정의하며, 안티 포렌식 기법으로 데이터 파괴, 데이터 암호화, 데이터 은닉, 데이터 변조 등으로 분류된다. 따라서 본 연구에서는 안티포렌식 기법을 데이터 은닉 및 삭제 (난독화 및 암호화)로 분류하여 최근 연구동향을 조사 및 분석하고 향후 안티포렌식의 연구 방향을 제안하고자 한다.

Analysis of anti-forensic trends and research on countermeasuresucation

Han Hyundong*, Cho Young Jun*, Cho Jae Yeon*, Kim Se On*, Han Wan Seop*, Choi Yong Jun*, Lee Jeong Hun*, Kim Min Su**

ABSTRACT

With the popularization of digital devices in the era of the 4th industrial revolution and the increase in cyber crimes targeting them, the importance of securing digital data evidence is emerging. However, the difficulty in securing digital data evidence is due to the use of anti-forensic techniques that increase analysis time or make it impossible, such as manipulation, deletion, and obfuscation of digital data. Such anti-forensic is defined as a series of actions to damage and block evidence in terms of digital forensics, and is classified into data destruction, data encryption, data concealment, and data tampering as anti-forensic techniques. Therefore, in this study, anti-forensic techniques are categorized into data concealment and deletion (obfuscation and encryption), investigate and analyze recent research trends, and suggest future anti-forensic research directions.

Keyword: Digital Forensic, Anti-Forensic, Digital Evidence, Data Hiding and Deletion, Data Obfuscation and Encryption

접수일(2023년 02월 28일), 수정일(2023년 03월 30일), 게재확정일(2023년 03월 31일)

* 중부대학교 정보보호학과 학부생
** 중부대학교 정보보호학과 교수(교신저자)

1. 서 론

4차 산업혁명 시대에 디지털기기 보급의 대중화와 인터넷 기술의 발전과 함께 이를 대상으로 한 사이버 범죄가 증가함에 따라 디지털 데이터 증거 확보의 중요성이 대두되고 있다.

디지털 증거의 확보 및 입증을 위한 디지털 포렌식 기술 분야의 발전에 반하여 증거 확보 및 분석에 대하여 디지털 데이터에 대한 분석 시간을 증가시키거나 불가능하게 만드는 안티 포렌식 기법의 활용도 발전 및 증가하고 있는 실정이다.

이러한 안티 포렌식은 디지털 데이터를 조작, 삭제 또는 난독화하여 조사를 어렵게 만들거나 시간이 많이 소요되도록 하거나, 아니면 거의 불가능하게 만드는 것으로 정의된다[1].

데이터 삭제의 경우 소프트웨어 및 하드웨어를 기반으로 데이터를 일부 혹은 완전 삭제 기법이 적용되고, 데이터의 위·변조의 경우 원본 데이터를 보존하면서 읽거나 분석하기 어렵게 만드는 난독화 기법을 적용한다. 데이터 은닉은 논리적 파티션과 물리적 파티션 사이에 발생하는 공간에 데이터를 은닉하는 기법을 활용하게 된다.

즉, 안티 포렌식 기법은 디지털 증거 삭제, 암호화 및 변조, 은닉 등으로 분류할 수 있으며, 본 연구에서는 안티 포렌식 기법의 최근 연구 동향을 조사 및 분석하고 향후 안티 포렌식에 대한 대응방안 및 연구 방향을 제안하고자 한다.

2. 관련 연구

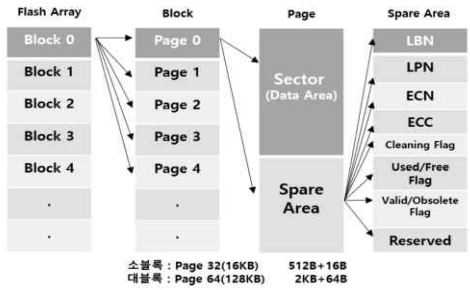
2.1 데이터 삭제기법

데이터 삭제 기법에는 소프트웨어 및 하드웨어를 기반으로 데이터를 완전히 삭제하는 와이핑(Wiping) 기법과 소프트웨어 및 하드웨어를 완전히 삭제하는 도구인 Eraser, SDelete, CyberShredde, Drive Cleanser, Final Eraser, BlackMagic 등을 이용하는 방법, 역추적을 회피하기 위한 로그 삭제, SSD의 특징을 이용한 Trim이 있다 영준[1][2][3].

위 기법 중 최근 가장 널리 사용되는 메모리인 낸드 플래시 메모리(Flash Memory)에 대한 안티 포렌식 기법 중 소프트웨어를 사용한 대표적인 기법인 데이터 와이핑(Data Wiping) 기법과 로그삭제, SSD Trim 대해 기술한다.

2.1.1 낸드플래시메모리의 특성 및 데이터 와이핑 기법

낸드형 플래시 메모리는 블록(Block), 페이지(Page), 섹터(Sector), 여유 영역(Spare Area)으로 구분된다. 장치는 여러 개의 블록으로 이루어져 있고, 하나의 블록에는 여러 개의 페이지로 구성되어 있다. 한 페이지 안에는 데이터가 기록되는 섹터와 페이지의 메타정보가 저장되는 여유 영역으로 구성되어 있다. 여유 영역에는 LBN(Logical Block Number), LSN(Logical Sector Number), ECN(Erase Count Number), ECC(Error Correction Code), 블록 및 페이지 상태 등 여러 가지 정보가 저장된다[4].



(그림 1) Flash Memory 구조

낸드 플래시 메모리의 제자리 덮어쓰기가 되지 않는 물리적 제약으로, 메모리와 파일 시스템 사이에 FTL (Flash Translation Layer)을 사용한다 [5]. 플래시 메모리는 FTL의 특성에 따라 무효블록(Invalid Block)을 수집하고 관리하는 가비지 컬렉션(Garbage Collection) 기법을 이용한다[6].

파일 삭제는 파일 시스템의 메타데이터에만 수정이 발생하여 데이터가 잔존하게 되며, 이를 해결하기 위해 데이터 와이핑 기법을 사용한다. 데이터 와이핑 기법은 데이터를 영구 삭제하기 위해서, 데이터가 기록된 동일한 논리주소 영역에 0 또는 1, 특정 패턴데이터, 난수 데이터를 여러 번 덮어쓰는 기법이다[7].

2.1.2 데이터 삭제 파일 복구 방지 기법

최근까지 연구된 데이터 삭제 파일 복구 방지에 관련된 연구로는 미리 파일 시스템 전체의 빈 공간을 쓰레기 파일로 채워 놓고 파일 쓰기 연산이 수행될 때 파일 시스템에 채워진 쓰레기 파일 일부분을 삭제하여 삭제된 공간에 쓰기 연산을 수행하여 파일 복구를 방지하는 안드로이드 어플리케이션을 이용한 복구 방지 기법[8]이 있다.

그리고 플래시 메모리의 특성을 이용하여 삭제 대상이 되는 파일과 관련이 있는 페이지들의 데이터와 스페어 영역의 비트를 1에서 0으로 쓰기 연산을 수행하는 덮어쓰기 효과를 통해 데이터를 소멸시켜 복구를 방지하는 최소 오버헤드 선택 복구 방지 기법[9]이 있다.

또한 파일 저장 시 블록의 페이지 순서를 치환하여 임의적인 페이지 순서를 생성하는 블록 치환에 기반하여 데이터가 기록된 페이지의 순서를 알지 못하도록 하여 삭제 후 원본 데이터가 존재함에도 불구하고 데이터를 복구하지 못하도록 하는 블록 치환을 이용한 복구 방지 기법[10] 등이 있다.

2.1.3 로그 삭제

시스템의 서비스나 이벤트 기록이 담긴 로그 파일은 취약점을 발견할 수 있는 증거가 된다. 대표적으로 윈도우 이벤트 로그 등이 있으며 공격자 경우 흔적을 숨기기 위하여 로그 파일을 위조하거나 삭제하기 때문에 로그 파일은 포렌식 분석의 중요 데이터가 된다.

자신의 공격 또는 침입 사실을 숨기기 위하여 행하는 공격들 중 한 종류가 로그 파일 자체의 항목들을 삭제하거나, 로그 파일 자체를 삭제하는 공격이다(Total Deletion Attack)[2].

2.1.4 SSD Trim

SSD는 기존의 데이터를 삭제한 다음 새 정보를 쓸 수 있다. SSD의 플래시 메모리는 쓰기 속도에 비해 데이터를 지우는 소거속도가 많이 느리기 때문에 이러한 처리 과정은 비효율적이고 느릴 수밖에 없다. 때문에 유효시간에 소거될 데이터를 미리 소거할 수 있도록 도와주는 명령어가 Trim 이다. Trim 기능을 지원하는 SSD는 그렇지 않은 SSD보다 성능이 뛰어나기 때문에 사용자 입장에서는 Trim 기능을 쓰지 않을 이유가 없다. 하지만 이러한 Trim 기능은 디지털포렌식 입장에서는 상당한 골치거리다. Trim 기능으로 인해 삭제된 데이터는 복구 가능성이 낮고 복구율이 제한적이다[3].

2.2 암호화 및 변조(난독화)

암호화 기법은 데이터를 외부 유출로부터 보호하기 위한 기법으로 중요 데이터를 보호하는데 사용될 수 있다. 하지만 암호화 된 비밀 데이터를 인터넷이나 무선 네트워크로 전달하면서 직접적으로 통제가 불가능하게 되었기 때문에 데이터 암호화 프로그램들은 악의적인 용도로 사용되어 안티포렌식에 사용되는 경우가 많아졌다. 그 중 쉽게 다운받아 사용할 수 있어 안티포렌식에 사용되고 있는 BitLocker, LockMyPix에 대하여 기술한다.

난독화 기법은 프로그램 변환 기법의 일종으로 프로그램의 원래 의미를 보존하면서 읽거나 분석하기 어렵게 만드는 프로그램 보호 기법이다. 하지만 난독화 기법을 사용하여 악의적인 목적을 가진 사람이 악성코드를 만들고, 그 악성코드를 난독화 하여 포렌식을 할 때 분석 시간을 지연시킬 수도 있다.

2.2.1 BitLocker

기존의 EFS나 RMS같은 보안 매커니즘은 디스크의 물리적인 접근을 막지는 못했다. 따라서 컴퓨터를 도난, 분실할 경우 저장되어 있는 데이터가 누출 될 수 있는데 이러한 보안상의 취약점을 보완하기 위해서 사용한다.

BitLocker는 Windows Vista에서 처음 사용되는 디스크 자체에 대한 암호화 방법으로 기존의 Windows 시스템에서 사용하던 RMS나 EFS와 같은 데이터를 보호 하기 위한 매커니즘을 보완하기 위해서 사용된다.

BitLocker의 사용으로 물리적으로 디스크에 접근하는 등의 문제가 되었던 부분을 보완할 수 있고, TPM을 사용해서 보안 매커니즘을 강화하여 구축하는 것이 가능하다[11].

2.2.2 LockMyPix

사진 및 동영상 데이터는 디지털 포렌식 관점에서 분석을 통해 다양하게 활용될 수 있다. 그러나 일부 사용자들은 데이터를 은닉 및 암호화하는 기능을 악용하

여 포렌식 수사에 어려움을 주고 있는데, 이 중 Lock MyPix라는 어플리케이션은 사진 및 동영상 데이터를 암호화하여 숨김 처리된 특정 디렉토리에 복사하여 저장한다. 이때, 원본 데이터는 삭제되지 않으나 사용자의 기기 갤러리 애플리케이션에서는 보이지 않게 된다.

PIN번호 또는 패스워드는 초기 설정 시 지정하며, LockMyPix 애플리케이션을 이용하여 암호화된 데이터에 접근 시 입력하여 설정된 값과 같아야만 데이터에 접근할 수 있다[12].

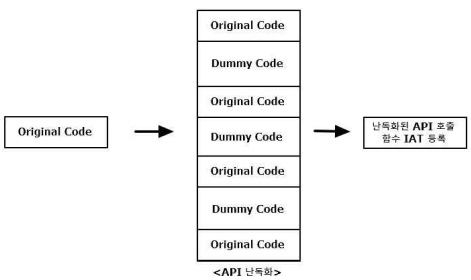
2.2.3 난독화 기법 분류

난독화 기법은 Colberg가 처음으로 분류하였고, 제어 난독화, 레이아웃 난독화, 방지 난독화, 데이터 난독화로 분류할 수 있다[13].

제어 난독화는 소스 코드의 실행 흐름을 난독화하여 해석과 분석을 어렵게 만드는 소프트웨어 보안 기술로 프로그램의 제어 흐름에 대한 논리를 바꾸거나, 불필요한 코드를 추가하거나, 중간 단계의 결과를 처리하는 부분에 특별한 명령어를 삽입하여 소스 코드의 실행 경로를 난독화한다. 데이터 난독화는 데이터의 기밀성을 유지하기 위해 데이터를 의도적으로 왜곡하거나 암호화하는 기술이다.

레이아웃 난독화는 소스 코드의 레이아웃을 변경하여 코드를 읽고 이해하는 것을 어렵게 만드는 소프트웨어 보안 기술로 소스 코드의 들여쓰기, 공백, 줄바꿈 등의 레이아웃을 변경하여 코드를 난독화한다. 방지 난독화는 소스 코드를 난독화하여 코드 분석과 해석을 어렵게 만드는 것을 방지하거나 우회하는 기술이다.

2.2.4 Themida



(그림 2) Themida API 난독화 방법

패커는 패킹을 해주는 프로그램으로 역공학을 지연시키는 패커는 컴프레서, 크립터, 프로텍터, 번들러 4가지로 분류할 수 있다[14].

그 중 Themida는 프로텍터로 PE 파일을 리버싱으로부터 보호하기 위해 다양한 분석방해 기술을 사용한다. 프로텍터는 자신의 PE 파일의 코드 및 리소스를 보호하고 싶은 개발자가 적용하기도 한다. 하지만 분석 방해 기술을 사용하여 분석 시간을 지연시키는 특성을 악용하여 악성코드에 프로텍터를 적용하기도 한다.

2.3 데이터 은닉

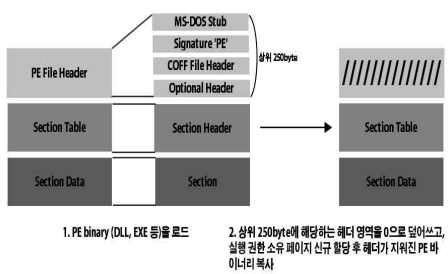
데이터의 은닉은 대표적으로 논리적 파티션과 물리적 파티션 사이에 발생하는 공간에 데이터를 은닉함으로써 디지털 포렌식을 방해한다.

데이터를 은닉할 시, 포렌식 과정에서 표면적으로 드러나는 데이터는 불필요한 데이터이기에 다양한 방법으로 은닉된 데이터를 탐지하는 과정에서 포렌식 과정이 지체되거나, 필요한 데이터를 찾지 못하는 문제가 야기된다[1].

2.3.1 PE파일 헤더 제거를 통한 은닉

Portable Executable(PE) 형식은 윈도우 계열 운영체제에서 사용되는 Object 파일 형식으로 PE 파일은 File Header, Section Table, Section Data로 이루어진다. 일반적인 메모리 포렌식 도구는 PE 파일을 분석하기 용이하게 하기 위해 PE파일 헤더를 참조하여 메타데이터, PE binary 등을 검사한다.

이를 활용하여 PE 이미지의 File Header를 0으로 덮어씌워 헤더를 제거하거나, 태그값을 수정하여 포렌식 도구의 메모리 덤프 획득 과정을 방해한다[15].



(그림 3) PE 구조체와 헤더 제거

2.3.2 Bootice 활용 은닉

파티션 관리 및 편집 도구인 Bootice를 사용하여 논리적인 파티션을 나누어 해당영역에 데이터를 은닉하는 기법이다. Deep hidden 모드를 활용, 데이터 파티션만 엔트리에 기록하고, 부팅 파티션에 기록하지 않으며, 일반적인 파티션 상의 은닉 기법에서 MBR의 파티션 엔트리를 제거함으로써 주요 포렌식 도구에 탐지되지 않게 응용한 기법으로써 실제 주요 포렌식 도구에 탐지되지 않기에 효과적인 안티포렌식 기술이다.[16]

<표 1> 각 Tool의 Bootice 은닉 영역 탐지 결과 [16]

	Windows		Digital Forensics			
Tools	Disk management	Windows Explore	EnCase	FTK Imager	Autopsy	X-Ways Forensics
Detect	x	x	x	o	x	x

2.3.3 플래시 메모리 기반 은닉

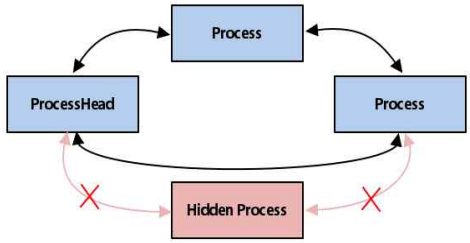
하드디스크에서의 은닉은 물리적 데이터를 수집을 통해 안티포렌식 대응이 충분히 가능했으나, SSD나 USB와 같은 플래시 메모리를 컨트롤러 조작으로 데이터 은닉 시 기존의 이미징 방식으로는 데이터 탐지조차 불가하다.

진행된 연구는 하드웨어 방식을 사용하는 보안 USB를 사용하여 PC에 설치된 접근 제어 프로그램 상에서 인증하는 방식으로 저장공간의 일부를 CD영역으로 만들어 은닉한다. 때문에 프로그램을 사용하기 전까지는 용량이 조금 작아진 일반 USB로 인식되기에 더욱 탐지가 어렵다.[17]

2.3.4 DKOM 기반 루트킷의 클라우드까지의 발전, 현황

보안 프로그램으로부터 악성코드를 보호하고 은닉하기 위한 목적인 루트킷의 프로세스와 드라이버를 은닉하기 위해 DKOM 기반 루트킷은 자신의 커널 오브젝트로부터 이중 연결리스트를 끊는다. 커널 단에서 이뤄지기에 많은 데이터의 취득도 탐지도 어렵다. 하지만 이중 연결리스트 무결성 검사와 메모리 스캐닝, 레지스트리 검색을 적용함으로 해당 안티 포렌식 기법에 대응책이 존재한다[18][19].

하지만 이 DKOM 기법 또한 클라우드가 각광받음에 따라 클라우드 하이퍼바이저 구조에서 발전되어 새로운 위협으로 다가왔다. 하이퍼 바이저가 관리하는 VM을 이용, DKOM 기반 루트킷을 생성함으로써 VM 커널 기반의 루트킷 공격이다. 이 공격은 권한 수준을 관리자 권한으로 격상할 수 있기에 클라우드 환경에서 큰 위협이다[20].



(그림 4) DKOM 기반 이중 연결 리스트 조작을 통한 은닉 [19]

3. 안티 포렌식 대응 방안

3.1 데이터 삭제 파일 복구 기법

3.1.1 데이터 와이핑에 의해 삭제된 파일 복구 기법

최근까지 연구된 데이터 와이핑에 의해 삭제된 파일을 복구하는 방법 중 플래시 메모리에 적용하는 기법은 FDS(Flash Memory Directory Scanning) 기법[21], FSS(Flash Memory Sector Scanning) 기법, 웨어 레벨링 기법을 이용한 복구 기법[22], 무효 데이터 영역이 가지는 특성을 이용한 복구기법[23] 등이 있다.

<표 2> 와이핑에 의해 삭제된 파일 복구방안

구 분	복구 방안
FDS	플래시 메모리에 잔존되는 이전 메타 데이터를 이용한 복구
FSS	원본 파일이 잔존하는 현상과 파일의 내부적 특성을 이용한 복구
웨어레벨링	플래시 메모리의 특성인 웨어레벨링 기법을 이용한 복구
무효영역	플래시 메모리의 무효 데이터 영역이 가지는 특성과 파일의 내부적 특성을 이용한 복구

위 연구에서 웨어레벨링에 의해 블록 지우기 횟수(ECN)가 낮은 번호가 먼저 쓰이고, 파일을 기록할 때 논리 주소가 순차적으로 쓰여, 정렬된 블록들에서 논리블

록번호(LBN)이 순차적이지 않는 블록은 다른 파일의 블록이 될 수 있다는 특징[22], 원본파일의 LBN과 와이핑 파일의 LBN의 논리 주소를 비교하여 와이핑 파일을 기준으로 원본파일의 논리주소가 서로 다른 블록은 와이핑과 관련된 영역일 수 있다는 특징[22] 무효 데이터 영역에서 와이핑 데이터와 원본 파일은 동일한 논리주소를 가지기 때문에, 가비지 컬렉션 테이블의 사상정보에서 논리주소가 중복되는 영역은 와이핑과 관련된 영역일 수 있다는 특징[23] 등을 찾을 수 있었다.

하지만 위 기법들은 메타데이터의 의존도가 높고, 이전 메타데이터에서 와이핑과 관련된 메타데이터를 구분하기 어려워 와이핑과 관련된 메타데이터를 찾는데 시간적 소비가 증대하는 등의 제약이 따른다.[23] 그러므로 최근 연구된 기법의 단점을 보완하기 위해 머신러닝 기반의 이상탐지기법의 적용 방안을 다음과 같이 제안하고자 한다.

이상탐지(Anomaly detection)란 어떤 특정한 도메인에서 일반적으로 예상되는 특성을 따르지 않는 데이터나, 정상으로 규정된 데이터와 다른 특징을 가지는 데이터를 찾아내는 것으로 개념을 정의하며, 세부적으로 지도학습 기반 이상탐지와 비지도학습 기반 이상탐지로 분류되며, 지도학습 기반의 이상탐지 알고리즘은 SVM(Support vector machine), Bayesian network, K-nearest neighbors 등이 있고, 비지도학습 기반 이상탐지 알고리즘은 Clustering, Principal Component Analysis, Autoencoder 등의 종류가 있다 [24].

이전 연구에서 나타난 특징들을 시그니쳐화한 후 정상 데이터와 비정상 데이터로 구분하고 라벨링하여 지도학습 기반 이상탐지 알고리즘 중 학습한 영역을 기반으로 새로운 데이터가 어느 카테고리에 속할지 판단하고 이진 선형 분류 모델로 보여주는 SVM(Support vector machine) 알고리즘을 기반으로 이상탐지 기법을 적용한다면 플래시 메모리에서 와이핑 의심 파일을 찾는데 필요한 노력과 시간을 현저히 줄여 보다 빠르게 와이핑 파일을 찾아 복원하는데 도움이 될 것으로 여겨진다.

3.1.2. Pseudo-labeling 기반 준지도 학습을 이용한 로그삭제 진단 모델

로그 삭제 공격을 보완하기 위해 로그 저장 장치에 백업 서버를 추가하는 구조가 제안되었지만 실시간 온라인 상태의 백업 서버를 추가 운영해야 되며 물리적, 논리적으로 안전을 보장하지못하며 비효율적이다 [2]. 그러므로 로그 저장 장치에 Pseudo-labeling 기반 준지도 학습 을 이용한 로그삭제 진단 모델 적용 방안을 다음과 같이 제안한다.

준지도 학습은 일부분의 정답라벨데이터를 이용하여 정답이 없는 라벨데이터를 증강한 후 예측된 라벨에 대해 UDA등과 같은 방식을 통해 정답이 없는 라벨데이터의 손실을 최소화 한다. 로그삭제 진단 모델의 예로 윈도우의 이벤트 로그를 데이터 값으로 설정하고자 한다[25] .

Pseudo-labeling은 가장 확률 높은 라벨을 가상 라벨의 형태로 부여하는 기법으로서, 라벨값이 부족한 데이터의 한계를 극복하기 위한 방법이다[26].윈도우의 이벤트 로그는 이 시스템의 전반적인 모든 활동을 기록하며 EVTX파일 형식으로 저장하며 이벤트 뷰어를 통해 볼 수 있다. EVTX 파일은 다수의 청크(Chunk)가 이어져 있는 형태로, 이 청크 내부에 이벤트 레코드(Record) 기록이 존재한다.

삭제 로그가 남는 이벤트 로그는 시스템(System.evtx)로그와 보안(Security.evtx) 로그로 이 로그들을 지울 경우 이벤트 로그 삭제 여부와 감사 로그 삭제 여부가 남는다.

감사 설정을 설정한 이벤트 로그를지우면 보안로그에 기록이 남게 되고, 나머지 로그들을 삭제할 경우 시스템 로그에 기록이 남기 때문에, 만약 다른 이벤트 로그를 다 지우고 시스템 로그를 마지막에 지운다면 시스템 로그에 대한 기록만 남아 이벤트 로그를 은닉이 가능하다.삭제 관련 로그는 <표 2>과 같다[27].

<표 2> 삭제 관련 로그[27]

이벤트 로그	이벤트 ID	설명
보안	1102	감사 로그가 지워졌습니다.
시스템	104	System 로그 파일이 삭제되었습니다.
		Application 로그 파일이 삭제되었습니다.
		Setup 로그 파일이 삭제되었습니다.
		그 외 로그들의 삭제 기록

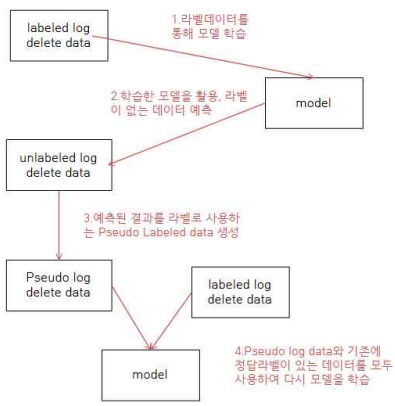
이러한 EVTX는 로그를 삭제해도 남는 기록이므로

Record 자체를 지워도 한 개의 삭제한 이벤트라는 기록이 남는다. 즉 System 로그를 아무리 지워도 System 로그가 삭제되었다는 이벤트(event ID : 104)(그림 5)와 같이 남는다. 즉 이러한 로그삭제 기록은 공격자의 안티포렌식 행위로 취급할 수 있다.

(그림 5) 로그 삭제 이벤트

(그림 6)과 같이 로그 삭제 기록을 정답데이터로 라벨링하여 정답라벨 데이터(labeled log delete data)를 만든 후 데이터를 통해 모델을 학습하고, 학습한 모델을 통해 라벨이 없는 데이터를 예측하여, 예측한 데이터(unlabeled log delete data)를 라벨로 사용하는 Pseudo Labeled log data 생성한다.

생성한 Pseudo Labeled log data와 기존의 정답라벨 데이터를 모두 사용하여 다시 모델을 학습한다. 이러한 Pseudo-labeling 기반 준지도 학습 기법 모델을 적용한다면 로그 삭제 의심 여부를 쉽게 파악 할 수 있을 것으로 예상된다.



(그림 6) 로그삭제 진단 모델

3.1.3 선별적 trim 명령 기법을 이용한 ssd 데이터 복구 가능성 증진

trim 기능으로 삭제된 데이터는 복구 가능성이 낮고 복구율이 제한적이다. trim 기능으로부터 데이터를 선별적으로 관리하여 데이터 완전 삭제를 방지할 수 있는 데이터 관리 기법을 제안한다. ssd trim 과정 중 선별된 데이터들은 trim list에 보관되어 trim이 적용되지 않은 데이터와 같이 garbageCollection & Wear-Leveling 이하 영역에 가지 않고 데이터가 보관된다. 사용자에게는 trimlist의 데이터들이 보이지 않지만 데이터 완전 삭제를 방지하고 데이터를 보관할 수 있다. 성능 저하 없이 기존 trim 기법 그대로 사용할 수 있다.[28]

이렇게 선별적으로 데이터를 관리하면 trim 기능을 사용했더라도 trimlist에 남은 범죄 의심 데이터들을 확인하여 포렌식 성공률을 높일 수 있다.

다만 적확하게 데이터를 선별할 수 있도록 공공기반 데이터를 기반으로 데이터 마이닝 기법을 이용해 효율적이고 정확한 알고리즘을 연구하고 개발 할 필요가 있다.[29]

3.2.1 암호화 관련 기법

윈도우11 버전으로 업그레이드 시 TPM 2.0이 필수로 장착되어야 하는데, 레지스트리 편집기를 통해 이를 우회하는 방법을 통해 TPM 2.0 없이 업그레이드 하는 방법이 있다. 이를 반대로 이용하여 TPM 2.0을 인식하지 못하게 하여 암호 알고리즘상의 오류를 유도하여 암호화를 해제하는 방법 및 물리적으로 TPM 2.0의 메모리부분을 읽어 저장된 키를 찾는 방법을 고려해볼 수 있으나 유효성 검증이 이뤄지지 않았기 때문에 적용에 있어 부적합하다.

또한 키로거를 이용하여 공격자가 복호화 시 복호화 키를 입력할 때의 키보드 활동기록을 저장하는 방법을 통해 공격자가 설정해 놓은 복호화 키를 가로채는 방법이 있지만 키로거가 준비되어 있는 환경 및 한 번 이상의 복호화가 강제되는 한계로 인해 대응법으로서의 범용성이 부족하다. 따라서 암호화를 활용한 안티포렌식 대응보다는 예방에 중점을 둬야 한다.

3.2.2 난독화 관련 기법

Taint Analysis는 프로그램 내의 정보의 흐름을 분석하기 위한 기술로, Taint 된 데이터와 연관된 모듈, 데이터 흐름을 지속적으로 추적하는 분석기법이다. Taint Analysis를 사용하여 분석하고자 하는 데이터가 코드 실행에 따라 호출되는 파일, 모듈 및 메모리 위치 등을 모두 파악할 수 있다[30]. Themida의 API 난독화를 해제하려면 API를 분석해 API 주소를 난독화 한 값을 저장하는 과정과 IAT에 등록된 함수를 분석하여 난독화 된 주소에 접근하는 순간을 추적해 나간다면 IAT에 저장된 주소를 원본 API 주소로 복구할 수 있다[31]. 이 후 메모리를 덤프해 재조립하면 API 난독화를 해제한 실행 파일을 얻을 수 있는데 이러한 추적과정을 Taint Analysis 기술을 적용하여 추적한다면 Themida의 API 난독화를 보다 쉽게 해제할 수 있을 것으로 예상된다.

3.3 각종 은닉 데이터 탐지 기법

3.3.1 PE파일 헤더 제거를 통한 은닉의 대응

PE파일의 헤더 제거를 통한 데이터의 은닉에 대응하기 위한 방법으로 가상 주소의 연속된 영역에 대한 정보를 담는 구조체인 VAD를 선별하고, 선별한 가상 메모리 영역에서 Section Header 시그니처를 모두 스캐닝하고 탐지된 시그니처 간의 Offset 간격이 Section Header 크기인 40 bytes의 배수인지 확인하여 Section Table을 탐색하는 방법으로 탐지 가능하다.

하지만 속도와 정확도 측면에서 편차가 크며, 지정한 시그니처 조합에 따라서 탐지율 또한 달라질 수 있는 한계가 존재한다. [15] 이 방법을 통해 비록 안정성 측면에서의 부족함이 발생할 수는 있으나, 확인되는 시그니처 조합을 지속적으로 보충함으로써 안정성을 높여 PE파일 헤더 제거를 통한 은닉을 무효화하여 정상적인 구조체처럼 분석을 재게할 수 있다.

3.3.2 Bootice 활용 은닉에 대응

MBR의 파티션 엔트리의 제거를 통해 은닉한 데이터를 탐지하기 위해 MBR영역의 반복적인 존재를 확인하고, Bootice의 시그니처의 존재 여부, 정상 DATA 파티션의 위치, 파티션의 위치값의 순차적 변화 여부를 점검함으로써 탐지할 수 있다. FAT32 파일 시스템으로 구성된 은닉 파티션 또한 동일하게 확인이 가능하며, 은닉파티션 특정 위치에서만 생성되기에 해당 위치에 VBR이 발견되는 것으로 탐지할 수 있다. [16] 이 대응법을 통해 Bootice를 활용해 은닉된 데이터를 복구 가능하며, 이는 FAT32 파일 시스템으로 은닉된 데이터 또한 복구 가능하기에 대응법의 범용성을 기대할 수 있다.

3.3.3 플래시 메모리 기반 은닉의 대응

플래시 메모리를 컨트롤러 조작으로 은닉한 데이터를 탐지하기 위해 강구한 대응책은 PID와 VID값을 확인, PID와 일치하는 USB의 존재 여부를 점검하고 수신 패킷과 같은 패킷을 재전송하는 재사용 공격을 이행함으로써 은닉영역을 탐지 및 영역의 데이터 확인까지 가능하다. 즉, PID 확인을 통한 은닉영역 존재 점검과 은닉 영역 USB 이미징 분석 절차를 추가함으로써 보완할 수 있다. 하지만 해당 보완책은 전패킷의 개별 확인 가능 여부에 따른 제약과 자동화 미흡의 번거로움이 있기에 자동화 기법 개발에 초점을 맞추어 추가 연구를 요한다. [17]

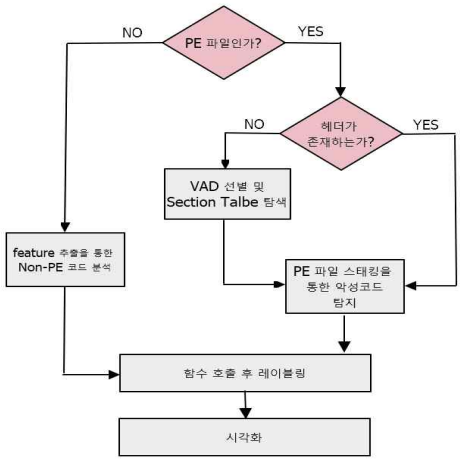
현재로써도 시간과 인력의 소모가 크다는 제약을 감수한다면 충분히 플래시 메모리의 은닉 데이터의 탐지가 가능하며, 자동화 연구가 성공적으로 이루어 질 시 개별 플래시 메모리를 수동 점검하는 번거로움이 해결되어 효율적으로 더 많은 증거 수집이 진행됨을 기대할 수 있다.

3.3.4 PE와 Non-PE 탐지 과정

시스템에 학습 데이터가 입력되면 각 파일의 바이너리를 읽어서 PE 파일, Non-PE 파일 여부를 구별한다. PE 파일에 경우, 바이너리 feature와 함께 LIEF 패키지를 통해 추출한 PE 관련 feature도 추출한다. Non-PE 파일에 경우, 바이너리 feature만 추출한다.[32] 이렇게 PE 파일인지 여부에 따라 먼저 분류하고, PE 파일 일 시 헤더의 제거 여부를 기준으로 한번 더 분류한다.[32], [33] 이렇게 분류한 파일들을 기준에 맞게 분석을 진행하고, 분석으로 얻은 정보를 레이블링, 레이블링한 데이터를 학습시키고, 이 함수와 정보들의 관계도를[34] 시각화 [35] 함으로써 PE 파일과 NON-PE 파일에 대해 쉽게 분석할 수 있는 하나의 절차가 완성

될 것이다.

이 모든 과정을 하나의 절차화시켜 포렌식에서 활용하면, PE파일과 헤더가 제거된 PE파일, Non-PE 파일에 대해 모두 악성코드를 탐지하고, 코드 내 함수들의 관계도를 생성하여 악성코드를 탐지하고, 근원과 문제를 분석하는데 유용할 것이다.



(그림 6) 은닉 악성 코드 탐지를 위한 절차

4. 결 론

4차 산업혁명 시대에 디지털기기 보급의 대중화와 함께 이를 대상으로 한 사이버 범죄가 증가함에 따라 디지털 데이터 증거 확보의 중요성이 대두되고 있다. 하지만 디지털 데이터 증거 확보의 어려움은 디지털 데이터를 조작, 삭제, 난독화 등 분석 시간을 증가시키거나 불가능하게 만드는 안티 포렌식 기법을 활용하기 때문이다.

이러한 안티 포렌식은 디지털 포렌식 측면에서 증거물의 회손 및 차단하는 일련의 행위로 정의하며, 안티 포렌식 기법으로 데이터 파괴, 데이터 암호화, 데이터 은닉, 데이터 변조 등으로 분류된다.

따라서 본 연구에서는 안티포렌식 기법을 데이터 은닉 및 삭제 (난독화 및 암호화)로 분류하여 최근 연구 동향을 조사 및 분석하고 향후 안티포렌식의 연구 방향을 제안하였다.

참고문헌

[1] Ryan Harris, “Arriving at an anti-forensics consensus Examining how to define and control the anti-forensic problem”, Digital Forensic Research Workshop, Digital Investigation Elsevier, 2006.

[2] 박주호, "지연 검증 기반의 로그 시스템 보안을 위한 로그 저장 기법에 관한 연구." 국내석사학위논문 단국대학교 대학원, pp.1-28, 2015.

[3] 윤지수, 이경렬, “안티 포렌식 신종기법에 대한 형사법적 대응방안”, 刑事政策 제32권, 제4호, 2021.

[4] B. Dipert, M. Levy, “Designing with Flash Memory”, Annabooks Publisher Poway, CA, USA, 1993.

[5] J. Kim, J. M. Kim, S. H. Noh, S. L. Min, Y. Cho, “A Space-Efficient Flash Translation Layer for CompactFlash Systems”, IEEE Transactions on Consumer Electronics, vol.48, no2, pp.366-375, 2002.

[6] L. P, Chang, T. W. Kuo, S. W. Lo, “Real-Time Garbage Collection for Flash-Memory Storage Systems of Real-Time Embedded Systems”, ACM Transactions on Embedded Computing Systems, vol.3, no.4, pp.837-863, Nov. 2004.

[7] G. G. Richard III, V Roussev, “Scalpel: A frugal, High Performance File Carver”, 2005 DFRWS Published by Citeseer, 2005.

[8] Joel Reardon, Claudio Marforio, Srdjan Capkun, David Basin, “Secure Deletion on Log-structured File Systems”, Cornell University Library, Jun. 2011.

[9] Lee Jaehong, Oh Jinha, Kim Seokhyeon, Lee Sangho, Huh Junyeong, Cho Yugeun, Hong Jiman, "A Secure Deletion Method for NAND Flash File System," "Korean Institute of Information Scientists and Engineers," vol.14, no.3, May 2008.

[10] 이상호, 신명섭, 박동주, “낸드 플래시 메모리에서 블록치환을 이용한 삭제파일 복구 방지”, 정보과학회논문지, 제39권, 제6호, pp.372 - 379, 2012.

[11] 황성호, 남현우, 박능수, 조수형, 홍도원, "포렌식 조사를 위한 윈도우 비스타 보안 체계 분석", 정보처리학회논문지C, 15(3), pp.141-148, 2008.

[12] 박진성, 서승희, 김역, 이창훈, "포렌식 분석을 위한 LockMyPix의 미디어 파일 복호화 방안 연구", 디지털포렌식연구, 14(3), pp.269-278, 2020.

[13] Collberg, Christian, Clark Thomborson, and Douglas Low. A taxonomy of obfuscating transformations. Department of Computer Science, The University

of Auckland, New Zealand, 1997.

[14] Jae-hwi Lee, Jaehyeok Han, Min-wook Lee, Jae-mun Choi, Hyunwoo Baek, Sang-jin Lee.A Study on API Wrapping in Themida and Unpacking Technique.Journal of the Korea Institute of Information Security & Cryptology, 27(1), pp.67-77. (2017).

[15] Seunghee Seo, Yeog Kim, Changhoon Lee..Countering Portable Executable File Header Removing based Anti-Memory Forensic Techniques.Journal of Digital Forensics ,15(2),50-59.(2021).

[16] Pyo-Gil Hong, Dohyun Kim, "A Study of an anti-forensics and counter anti-forensics of Bootice's hidden area", Journal of Digital Forensics , 16(1), pp.24-36. 2022.

[17] So-Hee Kim, Jaehyeok Han, Sangjin Lee.A, "Study on Detecting Data Hiding Area of Removable Storage Device Based on Flash Memory", Journal of Digital Forensics, 12(2), pp.21-29, 2018.

[18] Jae-bum Sim, Jung-chan Na, "Detecting a Driver-hidden Rootkit using DKOM in Windows OS", 한국정보처리학회 학술대회논문집, 21(1), pp.391-394. 2014.

[19] Jiwon Choi, Bongkyo Moon, "Study on Detection Method and Development of the Kernel Mode Rootkit", 한국정보처리학회 학술대회논문집, 23(1), pp.233-236, 2012.

[20] Tae Woo Kim, Sang Kee Suk, Jong Hyuk Park. (2020). Consideration for Improving the Vulnerability of the Cloud Hypervisor Architecture. 한국정보처리학회 학술대회논문집, 27(1), 238-241.

[21] M. S. Shin, D. J. Park, "A File Recovery Technique for Digital Forensics on NAND Flash Memory," Journal of KIISE : Databases, vol.37, no.6, pp.292-299, Dec. 2010.

[22] Y. B. Lim, M. S. Shin, D. J. Park, "A Recovery Technique against File Wiping for Digital Forensic on NAND Flash Memory," Journal of KIISE : Databases, vol.39, no.2, pp.100-108, Dec. 2012.

[23] 신명섭, 오지현, 박동주, "플래시 메모리에서 디지털 포렌식을 위한 파일 와이핑 증거 확보기법", 정보과학회논문지, 제39권, 제5호, pp.271 - 278, 2012.

[24] 오미애, 박아연, 김용대, 진재현, "기계학습(Machine Learning) 기반 이상탐지(Anomaly Detection) 기법 연구" 한국보건사회연구원, 연구보고서 2018-12, pp.1 - 193, 2018.

[25] Jong-Sik Kim, Dae-Seong Kang, "A Study on Fire Data Generation and Recognition Rate Improvement using F-guessed and Semi-supervised Learning", The Journal of Korean Institute of Information Technology, 20(12), pp.123-134, 2022.

[26] 박나무. "Pseudo-label 기반 준 지도 학습을 사용한 비정형 의료 텍스트 정보 추출", 석사학위논문, 연세대학교 대학원, pp.1-150, 2020.

[27] 강세림. "윈도우 이벤트 로그(EVTX) 분석 및 포렌식 활용방안", 석사학위논문 국민대학교 일반대학원, pp.1-150, 2019.

[28] 황현호, 박동주, "디지털 포렌식 관점에서 SSD TRIM 명령의 선별적 복구", 정보처리학회논문지, 제4권 제9호, 2015.

[29] 김준우, 손중권, 이상한, "데이터 마이닝의 범죄수사 적용 가능성", 대한수사과학지, 제1권 제2호, 2006.

[30] 이태진, 오주형, 정현철, "Taint Analysis 기반 악성코드 탐지 방안", 한국정보과학회 학술발표논문집, 38(1D), 109-110, 2011.

[31] 이재휘, 이병희, 조상현, "최신 버전의 Themida가 보이는 정규화가 어려운 API 난독화 분석방안 연구", 정보보호학회논문지, 29(6), pp.1375-1382, 2019.

[32] KyungHyun Han, Seong Oun Hwang, "Implementation of Improved Malware Detection System for Analysis of Non-PE Files", 대한전자공학회 학술대회, pp.738-740, 2020.

[33]]Jin-Young Cho, Eun-Gi Ko, Hye-Bin Yoo, Mi-Ri Cho, Chang-Jin Seo, "The Development of Malware Detection System Based on PE file and Stacking", 대한전기학회워크샵, pp.,101-102, 2020.

[34] DaeYoub Kim, "Generating Call Graph for PE file", Journal of IKEEE, 25(3), pp.451-461, 2021.

[35] Youngjoo Lee, Jeongjin Lee.A, "Similarity Measurement and Visualization Method for the Analysis of Program Code", Journal of Korea Multimedia Society, 16(7), pp.802-809, 2013.

[36] 서동수, "Visualization Techniques for Massive Source Code", The Journal of Korean Association of Computer Education, 18(4), pp.63-70, 2015.

〔저자소개〕

한 현 동 (Han Hyun Dong)

현 재 중부대학교 정보보호학과 학부생

email : ghkdzm0869@gmail

조 영 준 (Cho Young Jun)

현 재 중부대학교 정보보호학과 학부생

email : as5961996@naver.com

조 재 연 (Cho Jae Yeon)

현 재 중부대학교 정보보호학과 학부생

email : sdrjseka246@naver.com

김 세 온 (Kim Se On)

현 재 중부대학교 정보보호학과 학부생

email : ksomaine@gmail.com

한 완 섭 (Han Wan Seop)

현 재 중부대학교 정보보호학과 학부생

email : wanseop1507@naver.com

최 용 준 (Choi Yong Jun)

현 재 중부대학교 정보보호학과 학부생

email : yjrain0806@naver.com

이 정 훈 (Lee Jeong Hun)

현 재 중부대학교 정보보호학과 학부생

email : jinwhnael1@naver.com

김 민 수 (Minsu Kim)

2004년 컴퓨터공학사
2012년 보안관리학석사
2015년 산업보안학박사
현 재 중부대학교 정보보호학과 조교수

email : fortcom@hanmail.net