情報保護學會論文誌
第 22 卷 第 6 號, 2012. 12

소프트웨어 기반 보안 USB에 대한 취약성 분석 방법론

김 민 호,[1†] 황 현 욱,[2‡] 김 기 범,[2] 장 태 주,[2] 김 민 수,[3] 노 봉 남[1]
[1]전남대학교, [2]ETRI 부설연구소, [3]목포대학교

Vulnerability Analysis Method of Software-based Secure USB

Minho Kim,[1†] Hyunuk Hwang,[2‡] Kibom Kim,[2] Taejoo Chang,[2] Minsu Kim,[3] Bongnam Noh[1]
[1]Chonnam National University, [2]The Attached Institute of ETRI, [3]Mokpo National University

요 약

USB 메모리가 보편화됨에 따라 보안 USB 제품들이 일반화 되고 있다. 보안 USB는 장치 기반의 접근제어, 저장된 파일의 암호화 등 다양한 방식으로 데이터를 보호하고 있다. 따라서 포렌식 관점에서 분석자가 데이터에 접근하기 위해서는 많은 어려움이 존재하여 데이터 복호화가 필요하다.

본 논문에서는 보안이 적용된 이동식 저장 매체에 대한 취약성 검증을 위해 소프트웨어 방식의 데이터 암·복호화 기술을 연구하고 이에 대한 분석 메커니즘을 제안한다. 보안 메커니즘이 적용된 USB 저장장치를 대상으로 데이터 복호화를 위한 취약점 분석을 수행하였으며, 그 결과 암호화가 적용된 보안 USB 제품에 대해서 패스워드 없이 원본 파일을 추출할 수 있는 취약점이 존재함을 확인할 수 있었다.

ABSTRACT

The modern society with the wide spread USB memory, witnesses the acceleration in the development of USB products that applied secure technology. Secure USB is protecting the data using the method as device-based access control, encryption of stored files, and etc. In terms of forensic analyst, to access the data is a lot of troubles. In this paper, we studied software-based data en/decryption technology and proposed for analysis mechanism to validation vulnerability that secured on removable storage media. We performed a vulnerability analysis for USB storage device that applied security mechanism. As a result, we found vulnerabilities that extracts a source file without a password.

Keywords: Digital Forensic, Secure USB, Data encryption, Vulnerability

I. 서 론

오늘날 USB 저장장치는 대부분의 직장인이 소지하고 있는 열쇠고리의 필수 아이템으로 자리 잡을 정도로 보편화되어 있다. 또한 다양한 디바이스, 네트워크, 애플리케이션 및 정보 자산들은 방대하고 복잡해졌으며, 이와 동시에 저장장치에 대한 취약점이 지속적으로 노출되어 데이터 보안 중요성이 커지게 되었다[1]. 이러한 문제점을 해결하기 위해 이동식 저장매체를 보호하는 기술이 연구되고 있다. 하지만 이런 보호 기술이 사이버 범죄에서 발생할 수 있는 증거 은닉에 사용되고 있어 수사 기관에서 증거 수집에 큰 어려움이 되고 있다[2]. 따라서 사이버 범죄에 사용된 보안 저장매체에서 원본 파일을 추출할 수 있거나, 적용된 기술 및 제품을 알아낼 수 있을 경우 증거 수집에 큰 도움이 된다.

또한 전 세계는 단일 시장으로 통합되고 있고 기업 간 경쟁이 심해져 조직이 가진 기밀 정보의 유출은 해당 조직의 경쟁력을 좌우하는 매우 중요한 문제로 인식되고 있다. 하지만 현재까지도 내부직원에 의한 기

접수일(2012년 6월 21일), 수정일(2012년 9월 27일), 게재확정일(2012년 11월 6일)
† 주저자, linz@lsrc.jnu.ac.kr
‡ 교신저자, hhu@ensec.re.kr

밀 유출 가능성이 78.9%가 '가능'하다고 응답하며, 기업들의 산업기밀 유출에 대한 부재도 심각한 상황이다[3]. 이에 정보유출 방지에 도움이 되는 저장장치 보안 기술이 지속적으로 연구되고 있는 단계이며, 이미 많은 시제품이 나오고 점차 시장이 확대되고 있는 과정이다. 하지만 이러한 정보 유출 방지 기술은 용의자들이 범죄 사실을 은폐하기 위한 도구로 이용할 수 있어 포렌식 수사 시, 증거를 획득하는데 많은 어려움이 있다[4]. 이러한 문제점을 해결하기 위해 은폐된 데이터를 읽고 복구할 수 있는 연구가 반드시 필요하다.

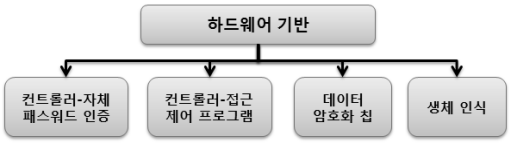
본 논문에서는 데이터 보호 기술이 적용된 저장매체에 대한 사용자 접근제어 기술 및 데이터 암·복호화 기술을 연구하고 이에 대한 분석 방법을 제안한다. 제안한 분석 방법을 보안 USB를 대상으로 적용하여 사용하고 있는 암호화 메커니즘을 확인할 수 있었으며, 이 과정에서 패스워드 없이 원본 파일을 추출할 수 있는 취약점이 존재함을 확인할 수 있었다.

논문의 구성은 2장에서 현재 사용되고 있는 이동식 저장매체에 대한 보안 기술을 소개하고 3장에서 이러한 제품들의 암·복호화 과정에 대한 분석 메커니즘을 제안한다. 4장에서는 분석 메커니즘을 활용하여 상용 제품들에 대한 취약성 검증을 실험한 결과를 기술하고, 5장에서 결론을 맺는다.

II. 관련 연구

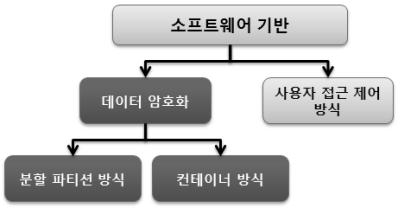
2.1 보안 USB 동향

보안 USB의 작동 방식은 하드웨어 방식과 소트프웨어 방식으로 나누어진다. 하드웨어 방식은 [그림 1]과 같이 플래시 드라이브 컨트롤을 이용하여 보안이 요구되는 데이터에 접근하는 Flash Drive Controller 방식을 취하는 방법과 암호화 칩을 기반으로 데이터를 암호화 시키는 Encryption Chip 방식, 지문이나 생체 인식 등을 사용하는 방식으로 나눌 수 있다. 컨트롤러-자체 패스워드 인증 방식의 경우 이동식 저장 장치에 패스워드 입력 및 인증 기능이 포함된 제품을 의미하며, 컨트롤러-접근 제어 프로그램 방식의 경우 PC에 설치된 접근 제어 프로그램 상에서 인증하는 방식을 의미한다. 하드웨어 기반 보안 USB의 경우 컨트롤러-접근 제어 프로그램 방식에 한하여 인증을 우회하는 기법에 대한 연구가 이미 진행되어있으며 컨트롤 명령을 분석함으로서 원본 파일에 접근이 가능하다[5].



[그림 1] 하드웨어 방식의 보안 USB

접근 제어 프로그램의 동작 과정은 보안 프로그램이 실행되었을 때 USB 플래시 드라이브를 확인하고 명령어 전송을 통해 USB로부터 접근 제어 패스워드를 전송받아 사용자가 입력한 패스워드와 비교하는 과정을 통해 인증을 수행한다. 이 과정에서 접근 제어용 패스워드가 노출되는 취약성이 존재할 수 있다[6].



[그림 2] 소프트웨어 방식의 보안 USB

소프트웨어 방식의 경우 [그림 2]와 같이 데이터 보안 기술에 따라 차이는 있지만 인증 소프트웨어 상에서 데이터 자체를 암호화하는 방식과 사용자 접근 제어 방식으로 이루어진다. 소프트웨어 방식의 보안 취약점 연구는 암호화된 데이터를 복구하는 방법이 아닌, 대부분 사용자 접근 제어 방식을 사용하는 소프트웨어에서 인증을 우회하는 방식으로 진행되어왔다. 본 논문에서는 소프트웨어 방식의 데이터 암호화 메커니즘 자체에 대한 분석을 통해 취약점을 분석하는 방법을 제안한다.

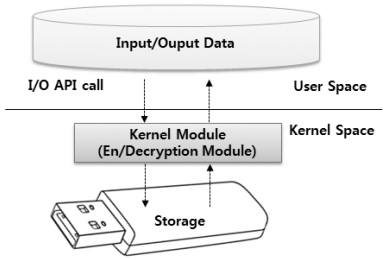
2.2 소프트웨어 방식 USB 저장장치 암호화 및 복호화 기법

2.2.1 실시간 암호화 방식(OTFE: On-The-Fly Encryption)

대부분의 소프트웨어 방식의 전체적인 저장장치 암호화 기술(FDE: Full Disk Encryption)은 On-The-Fly Encryption(이하 OTFE)를 기반으로 한다[7]. OTFE는 실시간 암호화(Real-Time Encryption)라고도 하는데, 실시간으로 저장장치의 데이터를 복호화/암호화하기 때문이다. 이것은 저장장치에서 데이터

를 가져올 때 복호화를 하고, 저장 장치에 저장하기 직전에 암호화를 수행한다[8]. 하지만 이것이 사용자에게는 보이지 않기 때문에 사용자는 일반적인 저장장치와 똑같은 방법으로 사용할 수 있다.

[그림 3]과 같이 OTFE는 일반적으로 운영체제의 I/O계층의 API를 후킹하여 사용한다. 즉 저장장치의 일반적인 행위에 드라이버를 사용하여 암호화/복호화(이하 암·복호화) 모듈을 끼워 넣는 것이다. 그래서 사용자는 일반적인 저장장치와 별 다른 차이점을 느끼지 못하고 사용할 수 있다.



[그림 3] OTFE 구조

운영체제에서 저장장치에 접근하기 전에 커널 모듈을 지나가게 되고 모듈에서 암호화 알고리즘을 사용하여 실제적인 암·복호화가 이루어지게 된다. 설치된 드라이버는 항상 운영체제와 함께 로드되어야 정상적인 암·복호화가 가능하다. 만약 드라이버가 지워지거나 이름이 변경되면 윈도우가 부팅되는 중간에 에러가 발생하여 부팅은 실패한다.

2.2.2 선택적 파일 암호화 방식

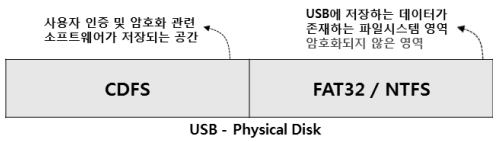
선택적 파일 암호화 방식은 사용자의 선택에 따라 특정 파일이나 다수의 파일에 대한 암호화를 수행하는 기법이다. 보안 USB에 설치된 프로그램을 이용하여 특정 파일에 대한 암호화를 수행하면 USB 내에 암호화된 파일이 생성되는 형태이다. 다수의 파일에 대한 암호화를 지원할 경우 하나의 이미지 파일에 다수의 파일을 암호화하는 경우와 각각의 파일마다 암호화된 파일이 생성되는 두 가지 경우로 나누어진다. 암호화는 프로그램 상에서 암호화를 수행하는 시점에서 이루어지며, 암·복호화 메커니즘은 드라이버가 아닌 프로그램 자체에 위치하기 때문에 분석이 용이하다.

III. USB 보안 메커니즘 분석

보안 USB는 저장된 데이터를 보호하기 위해 간단한 패스워드 인증방식에서 파티션 암호화, 데이터 암호화 방식을 사용해 USB 보안을 강화하고 있다. 본 논문에서는 이러한 보안 메커니즘을 분석하여 다음과 같이 크게 3가지 방법으로 분류한다.

3.1 사용자 접근 제어 방식 저장 장치

사용자 접근 제어 방식은 [그림 4]와 같이 사용자 데이터에 대한 암호화를 수행하지 않고 사용자 인증만을 수행하여 인증이 성공하였을 경우에만 데이터가 저장된 파일시스템 영역을 마운트하는 방식이다. 저장장치 내부 공간을 분할하여 사용자 인증 및 암호화 관련 소프트웨어가 저장되는 공간인 CDFS 영역과 데이터가 저장되는 파일시스템 영역으로 사용하는 것은 분할 파티션 방식과 유사하지만, 파일시스템 영역에 저장되는 데이터에 대한 암호화가 수행되지 않아 사용자 접근 제어만 우회하면 원본 파일에 접근할 수 있는 취약성이 존재한다.



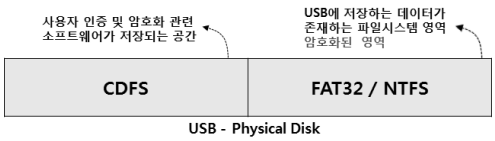
[그림 4] 사용자 접근 제어 방식의 저장 장치

사용자 접근 제어 방식을 사용하는 이동식 저장 매체의 보안 수준이 매우 낮기 때문에 전송하는 패스워드에 대한 암호화를 수행하지 않는 프로그램도 존재하며 이 경우 프로세스의 메모리상에 패스워드의 원문이 노출되기 때문에 매우 취약하다[6].

3.2 분할 파티션 방식 저장 장치

분할 파티션 방식은 [그림 5]와 같이 이동식 저장 매체의 공간을 분할하여 일부는 사용자 인증 및 암호화 관련 소프트웨어가 저장되는 공간인 CDFS영역으로, 나머지 부분은 암호화된 데이터가 저장될 파일시스템 영역(FAT32/NTFS)으로 사용한다. 매체가 PC와 연결되면 사용자 인증 과정을 거친 후 실제 데이터가 저장된 파티션을 마운트 하여 암·복호화 과정을 수행하게 된다. 파일시스템 영역은 암·복호화 기법

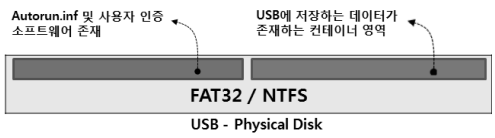
에 따라 파일시스템 전체를 암호화 시킨 OTFE 방식과 선택적으로 파일을 암호화하여 개별 파일을 저장하는 두 가지 방식이 사용되고 있다[9,10]. CDFS 영역에 암호화를 수행하는 프로그램이 존재하기 때문에 이를 분석하여 암·복호화 메커니즘 및 취약점 검증 등을 수행할 수 있다. 또한 암·복호화는 주로 드라이버 파일에서 이루어지기 때문에 이를 분석하는 작업이 추가로 필요하다.



[그림 5] 분할 파티션 방식의 저장 장치

3.3 컨테이너 방식 저장 장치

컨테이너 방식은 [그림 6]과 같이 파티션을 따로 분할하지 않고 데이터 저장 영역을 컨테이너 파일로 지정하는 방식이다. 저장 매체 내부에 사용자 인증 및 암·복호화를 수행하는 소프트웨어가 포함되어 있으며, 오토런 방식으로 소프트웨어가 자동 실행되는 방식을 취한다. 저장 매체 자체는 FAT32나 NTFS 등의 파일시스템을 사용하며, 사용자의 선택에 따라 특정 파일이나 다수의 파일에 대한 암호화를 수행하면 컨테이너 파일이 생성되고 이를 저장 매체의 파일시스템 공간에 저장한다. 대부분의 보안 장치에서는 다수의 파일을 암호화하고자 할 때 하나의 이미지 파일로 모아서 암호화를 수행하는 방식을 취한다. 분할 파티션 방식의 저장 장치와 마찬가지로 암호화를 수행하는 프로그램이 파일시스템 내에 존재하기 때문에 이를 분석하여 암·복호화 메커니즘 및 취약점 검증을 수행할 수 있다. 또한 드라이버 파일이 존재하지 않고 프로그램 자체에서 암·복호화가 수행되기 때문에 분석이 용이하나 별도의 암호화 모듈을 사용하는 경우가 많기 때문에 암호화 모듈에 대한 분석이 추가적으로 필요하다.



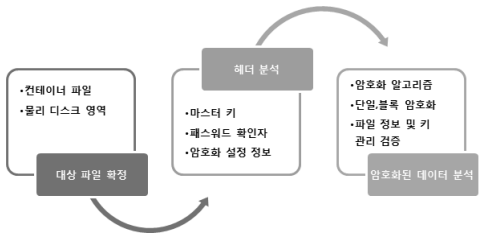
[그림 6] 컨테이너 방식의 저장 장치

IV. 취약성 분석 메커니즘

본 논문에서 제안하는 보안 이동식 저장 매체에 대한 취약성 분석은 크게 이미지 파일 분석, 암·복호화 메커니즘 분석, 원본 파일 추출 가능성 분석으로 나누어진다. 이미지 파일 분석은 [그림 7]과 같이 보안 프로그램을 통해 생성된 암호화된 데이터에 대한 분석을 수행하는 과정을 뜻한다. 이때 암호화된 데이터는 데이터 보안 기술에 따라 그 대상이 다를 수 있다. 이미지 파일 분석을 통해 암호화에 관련된 정보를 수집할 수 있는데, 이미지 파일 전체가 암호화되어있는 경우 암·복호화 메커니즘에 대한 분석이 선행되어야한다. 메커니즘에 대한 분석은 보안 프로그램에 대한 리버스 엔지니어링을 통해 암·복호화 메커니즘을 분석하는 과정을 뜻하며, 이 과정을 통해 암호화 알고리즘과 취약성의 존재 가능성 등을 알아낼 수 있다.

4.1 이미지 파일 분석

이미지 파일은 프로그램을 통해 암호화 과정을 거쳐 생성되는 데이터를 의미한다. 데이터 보안 기술에 따라 이미지 파일은 저장 매체의 파일시스템이나 별도의 파티션 영역에 존재한다. 따라서 우선 보안 저장 매체 내부에 존재하는 대상 파일을 이미지 파일로 가져오는 과정을 거친 다음, 이미지 파일 자체에 대한 분석을 수행해야한다.



[그림 7] 이미지 파일 분석 과정

4.1.1 대상 파일 확정

저장 장치 암·복호화 기법이나 데이터 보안 기술에 따라 대상 파일의 위치에 차이가 생긴다. 분할 파티션이나 사용자 접근 제어 방식의 저장 장치의 경우 암호화된 이미지 파일이 저장되는 영역이 논리 드라이브로 생성되지 않는다. 따라서 WinHex와 같은 도구를 이용하여 물리 디스크에 접근하여 데이터를 가져올 수 있다. 컨테이너 방식의 경우 논리 드라이브로 연결된 파일시스템 내부에 파일 형태로 존재하기 때문에 이를

분석할 수 있다.

4.1.2 이미지 헤더 분석

이미지 파일이 암호화 정보가 저장된 헤더 영역과 데이터 영역으로 분리되어 있을 경우 헤더 분석 과정이 필요하다. 헤더에는 암호화를 수행하기 위한 옵션이나 사용자 인증을 위한 패스워드 확인자(password verifier) 등이 존재한다. 대부분의 보안 프로그램은 암호화된 이미지를 생성할 때 데이터만을 암호화하여 이미지 파일을 만들기보다 정보가 포함된 헤더 부분을 포함하여 이미지 파일을 생성하는 방식을 취하고 있기 때문에 암·복호화 메커니즘을 확인하거나 취약성을 확인하는데 헤더를 분석하는 과정은 큰 도움이 될 수 있다.

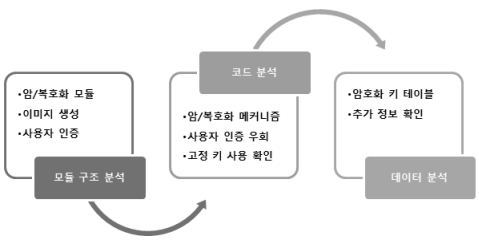
이미지 헤더의 경우 보안 프로그램에서 고정 키로 암호화를 수행하는 경우가 존재하는데, 리버스 엔지니어링을 통해 고정 키와 암호화 알고리즘을 알아낼 경우 이를 복호화 하는 것이 가능하여 데이터를 복호화 하기 위한 마스터 키나 패스워드 확인자, 파일 정보 등이 노출되는 취약성이 존재한다.

4.1.3 암호화된 데이터 분석

이미지 파일에 존재하는 실제 데이터 영역인 암호화된 데이터를 분석함으로서 암호화 알고리즘이나 블록 암호화를 사용하는 등의 정보를 얻을 수 있다. 단순히 하나의 암호화된 데이터만을 가지고 분석을 수행하는 것보다는 다수의 암호화된 데이터를 이용하여 비교 분석을 수행하면 고정 키 사용 확인 등의 추가적인 정보 획득이 가능하다.

4.2 암 · 복호화 메커니즘 분석

암·복호화 메커니즘에 대한 분석은 주로 프로그램에 대한 리버스 엔지니어링을 통해 이루어진다. 본 논문에서 제안하는 메커니즘 분석 과정은 〔그림 8〕과 같이 우선 프로그램에 대한 모듈 구조를 확인하고 각각의 모듈에 대한 코드 분석과 사용하는 데이터 분석을 통해 암호화 알고리즘을 알아내어 이미지 파일 분석 결과와 종합하여 취약성 여부를 판단하는 것이다.

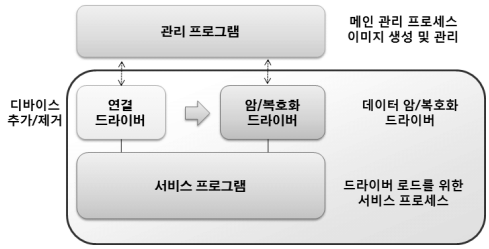


〔그림 8〕 암·복호화 메커니즘 분석 과정

4.2.1 보안 USB 관련 프로그램 모듈 구조 분석

암·복호화 메커니즘 분석에 앞서 우선 각각의 프로그램과 모듈의 구조에 대한 분석을 수행해야한다. 프로그램의 규모가 클수록 리버스 엔지니어링을 하는데 많은 자원이 소모되므로 모듈 구조 분석을 통해 분석하고자 하는 대상을 정한 다음 코드 분석과 데이터 분석을 수행하는 것이 효율적이다.

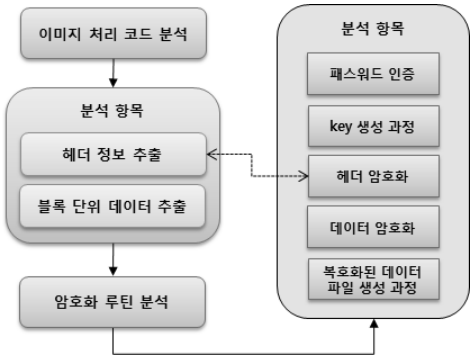
선택적 파일 암호화 방식을 사용하는 보안 프로그램의 경우 관리 프로그램에 암·복호화 코드가 존재하기 때문에 분석에 어려움이 없다. OTFE 방식을 사용하는 보안 프로그램의 모듈 구조는 〔그림 9〕와 같이 사용자가 이미지에 대한 암·복호화를 수행하기 위한 관리 프로그램과 드라이버에 대한 로드를 위한 서비스 프로그램, 데이터에 대한 암·복호화를 수행하기 위한 드라이버 모듈, 연결 드라이버 모듈 등으로 구성된다. 암·복호화 코드는 암·복호화 드라이버 모듈에 존재하며, 관리 프로그램에서 이미지 파일을 생성하는 부분에도 존재할 수 있다. 따라서 관리 프로그램과 암·복호화 드라이버 모듈에 대해 분석을 수행해야한다.



〔그림 9〕 모듈 구조 분석

4.2.2 코드 분석

코드 분석은 모듈 구조 분석을 통해 암·복호화 메커니즘이 존재할 것으로 생각되는 모듈 및 프로그램의 내부에 대한 리버스 엔지니어링 과정을 뜻한다. 코드 분석은 암·복호화 메커니즘을 비롯하여 사용자 인증 과정이나 헤더 제작 과정을 포함한다.



〔그림 10〕 코드 분석 방법

코드 분석은 〔그림 10〕과 같이 이미지 파일에서 헤더 및 데이터 정보를 추출하는 부분과 암호화된 데이터를 복호화하는 부분으로 나누어진다. 이미지 처리 부분에서는 헤더 정보 추출 코드를 분석하여 키를 관리하는 방식에 취약점 존재 여부를 확인할 수 있으며, 블록 단위 데이터를 추출하는 부분에서는 CBC, ECB와 같은 암호화 모드에 대해 일부분 파악하는 것이 가능하다. 헤더 정보가 암호화되어 있는 경우 데이터 암호화와의 비교 분석을 통해 사용 키나 암호화 함수와 같은 추가적인 정보를 얻을 수 있다. 암호화 루틴 분석 부분은 메커니즘에 대한 분석을 수행하는 것으로서 패스워드 인증 과정, 키 생성과정, 헤더 암호화 과정, 데이터 암호화 과정, 복호화된 데이터 파일 생성과정을 포함한다. 패스워드 인증 과정이나 키 생성 과정에서 노출되지 않아야할 데이터 관리의 문제점이나, 헤더 및 데이터 암호화 과정에서 고정 키 사용 여부 등을 분석한다. 대부분의 암·복호화는 서로 유사한 방식으로 이루어지기 때문에 복호화 과정이나 암호화 과정 중 한 부분만 분석한다면 나머지 부분에 대한 분석은 쉽게 수행할 수 있다. 또한 원본 데이터의 복구가 불가능하더라도 헤더 및 데이터 내에 존재하는 파일 명, 크기, 생성 시간 등과 같은 추가적인 정보를 추출할 수 있는 가능성을 판단할 수 있다.

4.2.3 데이터 분석

데이터 분석은 코드 분석 과정에서 암호화 알고리즘을 사용하는 함수로 추측되는 부분이 발견되었을 때 입력되는 인자들을 분석하는 과정을 뜻한다. 암호화 알고리즘의 특성 상 키 테이블이 존재하며, 이를 암호화를 수행하기 위한 함수에 인자로서 전달해야한다. 따라서 이러한 테이블을 분석 및 비교하는 과정을 통해 암호화 알고리즘을 알아낼 수 있다. 예를 들어 AES 암호화 알고리즘을 사용하는 경우 〔그림 11〕과 같은 테이블 값을 사용한다.

```
#define RT \
\
    V(51,F4,A7,50), V(7E,41,65,53), V(1A,17,A4,C3), V(3A,27,5E,96), \
    V(3B,AB,6B,CB), V(1F,9D,45,F1), V(AC,FA,58,AB), V(4B,E3,03,93), \
    V(20,30,FA,55), V(AD,76,6D,F6), V(88,CC,76,91), V(F5,02,4C,25), \
    V(4F,E5,D7,FC), V(C5,2A,CB,D7), V(26,35,44,80), V(B5,62,A3,8F), \
    V(DE,B1,5A,49), V(25,BA,1B,67), V(45,EA,0E,98), V(5D,FE,C0,E1), \
```

〔그림 11〕 FIPS-197 AES Table

```
; int dword_42D578[]
dword_42D578    dd 51F4A750h, 7E416553h, 1A17A4C3h, 3A275E96h
                                        ; DATA XREF: sub_422F50+E3↑r
                                        ; sub_422F50+156↑r ...
                dd 3BAB6BCBh, 1F9D45F1h, 0ACFA58ABh, 4BE30393h
                dd 2030FA55h, 0AD766DF6h, 88CC7691h, 0F5024C25h
                dd 4FE5D7FCh, 0C52ACBD7h, 26354480h, 0B562A38Fh
                dd 0DEB15A49h, 25BA1B67h, 45EA0E98h, 5DFEC0E1h
```

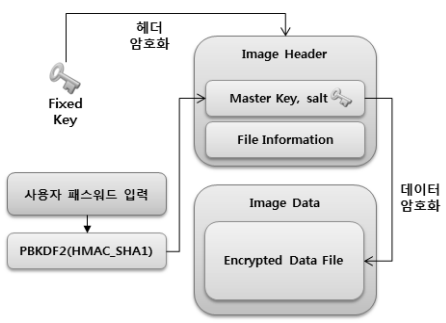
〔그림 12〕 프로그램 내부 데이터의 AES Table

이를 이용하여 프로그램 내부를 리버스 엔지니어링하는 과정에서 〔그림 12〕와 같이 나타나는 데이터를 통해 해당 암호화 알고리즘의 사용 여부를 확인할 수 있다.

4.3 취약성 분석

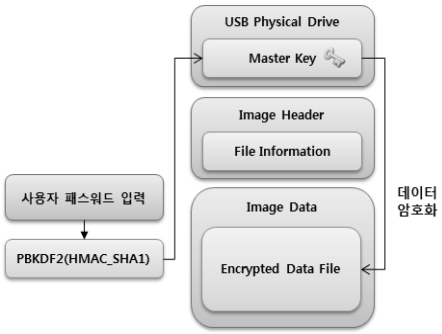
취약성 분석은 이미지 파일 분석과 암·복호화 메커니즘 분석 결과를 이용하여 패스워드 없이 원본 파일을 추출할 수 있는 가능성의 존재 여부를 확인하는 과정이다. 또한 원본 파일 추출이 불가능하더라도 이용될 수 있는 다양한 정보들을 수집할 수 있는 가능성 또한 포함된다.

소프트웨어 방식의 보안 이동식 저장 매체에 대한 취약성은 다음과 같이 헤더 암호화 메커니즘에 취약성이 존재하는 경우, 접근 가능한 영역에 마스터키가 존재하는 경우, 고정된 키로 데이터 영역을 암호화한 경우로 나눌 수 있다.



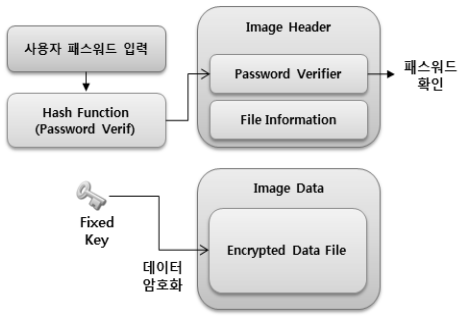
〔그림 13〕 헤더 암호화 메커니즘에 취약성 존재

헤더 암호화 메커니즘 상 취약성은 사용자 패스워드를 이용하여 마스터키를 생성한 이후 이를 헤더 영역에 저장하고 이를 고정 키를 사용하여 암호화한 경우이다. 〔그림 13〕과 같이 고정키를 사용하여 헤더를 암호화하였기 때문에 이를 복호화하여 마스터키를 추출할 수 있다. 따라서 추출된 마스터키를 이용하여 원본 데이터를 복호화 할 수 있다.



〔그림 14〕 접근 가능한 영역에 마스터 키 존재

데이터를 복호화 할 수 있는 마스터키가 접근 가능한 USB의 물리 드라이브에 존재하는 경우로서 주로 OTFE 방식에서 주로 나타나는 취약점이다. 〔그림 14〕와 같이 USB에 논리 드라이브로 할당되지 않은 섹터에 사용자 패스워드를 이용하여 생성된 마스터키가 저장되어 있기 때문에 직접 해당 섹터에 접근하여 마스터키를 추출하면 원본 데이터를 복호화 할 수 있다.



〔그림 15〕 고정된 키로 데이터 영역 암호화

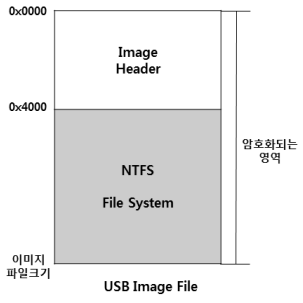
사용자 입력 패스워드를 이용하여 패스워드 확인자를 생성하고, 데이터는 고정된 키를 이용하여 암호화하는 경우이다. 〔그림 15〕와 같이 사용자가 데이터를 복호화하는 경우 입력된 패스워드와 기존 저장된 패스워드 확인자와 비교를 통해 인증을 수행하며, 인증이 성공한 경우 고정된 키를 이용하여 데이터를 복호화한다. 보안 수준이 매우 낮기 때문에 소프트웨어 방식의 보안 이동식 저장 매체에서 이와 같은 취약점이 발견될 확률은 낮다. 일부 보안 수준이 낮은 소프트웨어나 하드웨어 방식 보안 장치의 경우 사용자 입력 패스워드를 코드 상에 저장하거나, 특정 명령을 전송 시 스니핑을 통해 패스워드가 노출되거나, 메모리상에서 평문 패스워드를 획득할 수 있는 경우도 존재한다.

V. 성능 평가

논문에서 제안하는 분석 메커니즘의 효용성을 증명하기 위해 실제 5개의 상용 제품들에 대한 취약성 분석 실험을 수행한 결과, 2개의 제품에서 원본 파일 추출이 가능한 취약성이 발견되었다. 본 논문에서는 발견한 취약점 중 A사의 소프트웨어 방식 보안 이동식 저장 매체에 대한 취약성 분석 및 검증 실험 과정을 기술한다.

5.1 취약성 분석 실험

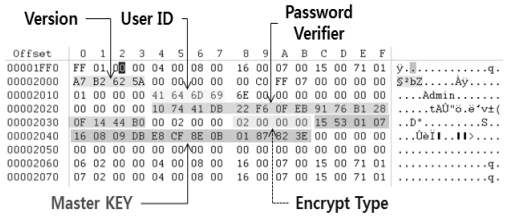
취약성 분석 실험을 수행하기 위해 보안 USB 제품에 대한 이미지 파일 분석과 암·복호화 메커니즘 분석을 수행하였다. 이미지 파일은 이미지 헤더와 암호화된 파일시스템 영역으로 나누어지며, 헤더는 AES 128bit 알고리즘을 사용하여 암호화된 상태이다.



〔그림 16〕 이미지 파일 구조

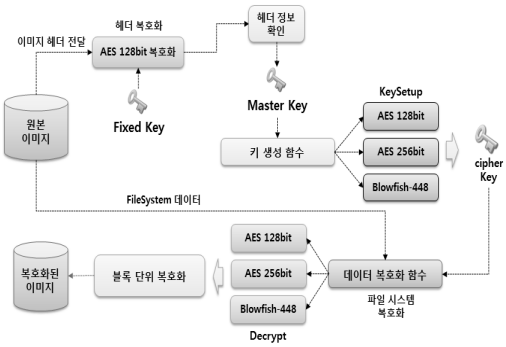
〔그림 16〕은 암호화된 이미지 파일의 구조를 나타낸다. 이미지 파일은 암호화 정보를 포함하고 있는 헤더 부분과 실제 데이터 부분으로 나누어지며, 실제 데이터 부분은 패스워드에 의해 생성된 마스터키에 의해 암호화된다. 이미지 헤더는 패스워드와 관계없이

고정 키를 이용하여 암호화되기 때문에 이를 복호화하는 것이 가능하다. 〔그림 17〕은 복호화된 이미지 헤더를 나타낸다.



〔그림 17〕 이미지 헤더 구조

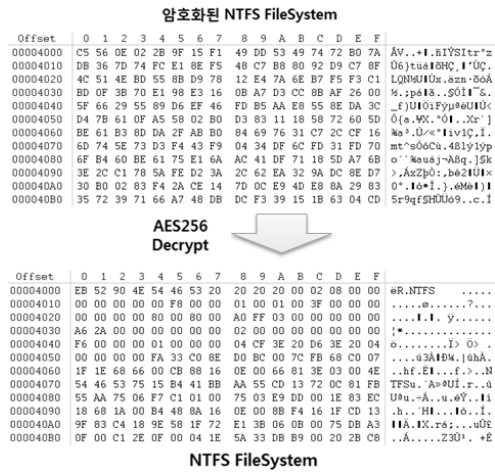
이미지 헤더는 사용자 인증을 위한 패스워드 확인자, 이미지 버전, 사용자 ID, 마스터 키, 암호화 타입 등의 정보를 저장하고 있다. 마스터 키는 암호화된 파일시스템 영역을 복호화 하기 위한 데이터로서 사용자 인증 과정에서 패스워드 확인자가 일치할 경우 마스터 키를 이용하여 암호화된 데이터 영역에 대한 복호화를 수행한다. 따라서 〔그림 18〕과 같이 인위적으로 헤더를 복호화하여 마스터키를 추출하고 이를 이용하여 암호화된 데이터 영역을 복호화 하면 원본 파일을 추출할 수 있는 취약점이 존재하게 된다. 따라서 암·복호화 메커니즘 분석을 통해 복호화 메커니즘을 그대로 구현하면 마스터키를 이용하여 원본 파일을 추출할 수 있다.



〔그림 18〕 데이터 복호화 메커니즘

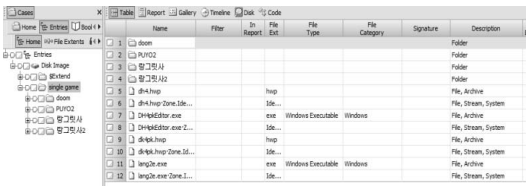
5.2 실험 결과

실험 결과 AES 256bit로 암호화된 이미지에 대해 원본 추출 실험을 수행한 결과 〔그림 19〕와 같이 복호화가 이루어진 것을 확인할 수 있었다.



〔그림 19〕 복호화된 파일시스템

NTFS 파일시스템으로 생성된 암호화된 이미지를 복호화한 결과 NTFS 파일시스템의 시그니처를 확인할 수 있었으며, Encase를 이용하여 〔그림 20〕과 같이 실제 저장된 파일들을 얻을 수 있었다.



〔그림 20〕 Encase를 이용한 파일 확인

VI. 결 론

본 논문에서는 보안 USB 저장 매체의 취약성을 분석하기 위한 방법을 제안하였고, 이를 이용한 분석을 통해 원본 파일을 추출할 수 있는 취약점을 소개하였다. 보안 USB 취약성 분석 방법은 이미지 파일 분석, 암·복호화 메커니즘 분석으로 나누어진다. 이미지 파일 분석은 보안 프로그램을 통해 생성된 이미지 파일에 대한 분석을 통하여 암·복호화 메커니즘을 위한 패스워드 확인자나 마스터키를 획득할 수 있는 가능성을 확인하는 과정이며, 암·복호화 메커니즘 분석 과정을 통해 사용하는 암호화 알고리즘을 확인하고 이 과정에서 취약성이 발견될 수 있는 가능성을 확인한다. 이러한 방법론을 통해 실제 검증을 수행 하였다. 그 결과 패스워드 여부와 관계없이 직접 복호화를 수행하여 암호화된 데이터에서 원본 파일을 추출할 수 있음

을 확인하였다. 향후 이러한 보안 취약점을 제거할 수 있는 연구를 수행할 계획이다.

참 고 문 헌

〔1〕 USB 보안 기술 및 제품 동향, 주간기술동향 통권 1380호, 정보통신연구진흥원, 2009년 1월

〔2〕 Simson Garfinkel, "Anti-Forensics: Techniques, Detection and Countermeasures," 2nd International Conference on i-Warfare and Security, pp. 77-84, Nov. 2007.

〔3〕 정선훈, 한기인, 신현우, "중소기업 산업기밀관리 실태조사 보고서," 한국산업기술진흥협회, 2010년 12월

〔4〕 C. Hargreaves and H. Chivers, "Recovery of Encryption Keys from Memory Using a Linear Scan," The Third International Conference on Availability Reliability and Security, pp. 1369-1376, Mar. 2008.

〔5〕 Jewan Bang, Byeongyeong Yoo, and Sanjin Lee, "Secure USB Bypassing Tool," DFRWS 2010, pp. 114-120, Aug. 2010.

〔6〕 정한재, 최윤성, 전응렬, 양비, 김승주, 원동호, "보안 USB 플래시 드라이브의 취약점 분석과 CC v3.1기반의 보호프로파일 개발," 한국정보보호학회 논문지, 17(6), pp. 99-119, 2007년 12월

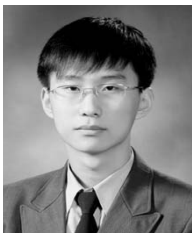
〔7〕 Robin Snyder, "Some Security Alternatives for Encrypting Information on Storage Devices," InfoSecCD '06 Proceedings of the 3rd annual conference on Information security curriculum development, pp. 79-84, 2006.

〔8〕 J.Alex Halderman and Seth D.Schoen, "Cold-boot attacks on encryption keys," Communications of the ACM vol.52, no. 5, pp. 91-98, May. 2009.

〔9〕 Stefan Balogh and Matej Pondelik, "Capturing Encryption Keys for Digital Analysis", IEEE:International Conference on Intelligent Data Acquisition and Advanced Computing Systems, vol 2, pp. 759-763. Sep. 2011.

〔10〕 이혜원, 박창욱, 이근기, 김권엽, 이상진, "포렌식 관점에서의 보안 USB 현황분석," 한국방송공학회 동계학술대회, pp.63-65, 2008년 2월

〈著者紹介〉



김 민 호 (Min-ho Kim) 학생회원
2011년 2월: 전남대학교 전자컴퓨터공학부(공학사)
2011년 3월 ~ 현재: 전남대학교 정보보안협동과정 석사과정
〈관심분야〉 디지털 포렌식, 시스템 보안, 악성코드 탐지, 취약점 분석

사 진

황 현 욱 (Hyunuk Hwang) 정회원
2000년 2월: 조선대학교 정보통신공학과 졸업(공학사)
2002년 2월: 조선대학교 전자공학과 졸업(공학석사)
2004년 8월: 전남대학교 정보보호협동과정 졸업(이학박사)
2004년 9월 ~ 현재: ETRI 부설연구소 선임연구원
〈관심분야〉 디지털 포렌식, 사이버보안, 정보보호

사 진

김 기 범 (Kibom Kim) 정회원
1994년 2월: 제주대학교 정보공학과 졸업(공학사)
1996년 8월: 고려대학교 전산과학과 졸업(이학석사)
2001년 2월: 고려대학교 전산과학과 졸업(이학박사)
2001년 1월 ~ 2004년 7월: (주)이씨오 개발부장
2004년 8월 ~ 현재: ETRI 부설연구소 선임연구원
〈관심분야〉 디지털 포렌식, 사이버보안, 정보보호

사 진

장 태 주 (Taejoo Chang) 정회원
1982년 2월: 울산대학교 전기공학과 졸업(공학사)
1990년 8월: 한국과학기술원 전기및전자공학과 졸업(공학석사)
1998년 2월: 한국과학기술원 전기및전자공학과 졸업(공학박사)
1982년 1월 ~ 2000년 1월: 국방과학연구소 선임연구원
2000년 2월 ~ 현재: ETRI 부설연구소 책임연구원
〈관심분야〉 디지털 포렌식, 암호프로세서설계, 정보보호, 통계학적 신호처리

김 민 수 (Minsoo Kim) 종신회원
1993년: 전남대학교 전산통계학과 (이학사)
1995년: 전남대학교 전산통계학과 (이학석사)
2000년: 전남대학교 전산통계학과 (이학박사)
2000년 ~ 2001년: 한국인터넷진흥원 선임연구원
2001년 ~ 2004년: 전남대학교 연구교수
2005년 ~ 현재: 목포대학교 정보보호학과 부교수
〈관심분야〉 침입탐지, 디지털 포렌식스, 데이터마이닝, 악성코드 분석

노 봉 남 (Bong-Nam Noh) 종신회원
1987년: 전남대학교 수학교육과 (이학사)
1982년: KAIST 전산학과 (이학석사)
1994년: 전북대학교 전산과 (이학박사)
1983년 ~ 현재: 전남대학교 전자컴퓨터공학부 교수
2000년 ~ 현재: 시스템보안연구센터 소장
〈관심분야〉 디지털 포렌식, 시스템 및 네트워크 보안, 정보사회와 사이버 윤리