DOI: 10.3745/KIPSTC.2008.15-C.3.141

포렌식 조사를 위한 윈도우 비스타 보안 체계 분석

황 성 호[†] · 남 현 우[††] · 박 능 수[†††] · 조 수 형[††††] · 홍 도 원[†††††]

요 약

2006년 겨울 Microsoft사에서 새롭게 출시한 Windows Vista는 기존의 Windows 운영체제와 비교해서 강력한 보안 메커니즘을 제공하고 있다. 하지만 컴퓨터가 범죄에 사용될 경우, 포렌식 관점에서는 새로운 보안 메커니즘으로 인하여 저장 장치에 저장되어 있는 범죄와 관련된 데이터를 획득하기가 더욱 힘들어진다. 본 논문에서는 Windows Vista에서 새롭게 사용하는 보안 메커니즘인 BitLocker에 대해 분석하고, 이전 Windows 버전에서 사용했던 UAC과 EFS에 대해서 변경된 점에 대해서 살펴보고 포렌식 관점에서 주요 보안 이슈를 살펴본다. 또한 포렌식 관점에서 활용할 수 있는 기타 Windows Vista 특징에 대해서 살펴본다.

키워드 : Windows Vista, BitLocker, TPM, Windows Forensics

Analysis of Windows Vista Security System for Forensic Examination

Seongho Hwang[†] · Hyunwoo Nam[††] · Neungsoo Park[†††] · Suhyung Jo[††††] · Dowon Hong[†††††]

ABSTRACT

Windows Vista published by Microsoft provides more powerful security mechanisms than previous Windows operating systems. In the forensics point of view, new security mechanisms make it more difficult to get data related to the criminals in a storage device. In this paper, we analyze BitLocker introduced as an new security mechanism in Windows Vista. Also, compared to the previous Windows operating systems, the changes and security issues of UAC and EFS in Windows Vista are discussed in the forensics point of view. Futhermore, we discuss other characteristics of Windows Vista useful for forensic examinations.

Key Words : Windows Vista, BitLocker, TPM, Windows Forensics

1. 서 론

Microsoft에서 2006년 겨울 새롭게 출시한 Windows Vista는 기존의 Windows 2000과 XP에 비해서 발전된 인터페이스와 사용자 편의 프로그램의 추가와 같이 많은 변화가 있었다. 하지만 무엇보다 가장 큰 변화는 강화된 보안 메커니즘이다. 기존에 Windows 운영체제에서 사용하는 보안 메커니즘은 사용자 계정 정보를 바탕으로 파일이나 폴더를 암호화 하는 EFS(Encrypting File System)과 로그인 하는 사용자가 소유한 권한에 따라서 프로그램 실행의 제한을 하는 UAC(User Account Control)가 있다. 이러한 보안 메커니즘은 기존에 많은 취약점을 가지고 있는 것으로 알려져 있다. Windows Vista에서는 기존의 보안 메커니즘의 보안상의 취약점을 보완해서 더욱 강한 보안 메커니즘을 만들었고, 새롭게 사용되는 BitLocker 보안 메커니즘[5,6,7,17,19]을 통해서 휴대용 컴퓨터와 개인용 컴퓨터에 저장되어 있는 개인이나 기업의 기밀 정보를 더욱 강력하게 보호할 수 있게 되었다.

하지만 더욱 강력해진 보안 메커니즘 때문에 반대로 컴퓨터가 범죄에 사용되거나 범죄와 관련된 중요한 정보가 컴퓨터에 저장되어 있는 경우, 포렌식 조사를 통해서 저장되어 있는 범죄 관련 데이터를 저장장치로부터 획득하는 것이 더욱 힘들어졌다. 이러한 Windows Vi.sta에서 포렌식 관점에서 발생할 수 있는 문제점을 대응하기 위해서 Windows Vista가 기존의 Windows 운영체제에서 변경된 점과 새로운 추가된 부분에 대한 분석이 필요하다.

본 논문에서는 기존의 Windows Vista에서 새롭게 사용되는 BitLocker의 데이터를 보호하는 암호화 방법, 동작 메커니즘, 보안을 위한 구성 요소들에 대해서 분석하고 포렌

※ 본 연구는 지식경제부 및 정보통신연구진흥원의 IT신성장동력핵심기술 개발사업의 일환으로 수행하였음. [2007-S019-02, 정보투명성 보장형 디지털 포렌식 시스템 개발]
† 준 회 원 : 건국대학교 컴퓨터공학부 석사
†† 준 회 원 : 건국대학교 컴퓨터공학부 석사과정
††† 종신회원 : 건국대학교 컴퓨터공학부 부교수 (교신저자)
†††† 정 회 원 : 한국전자통신연구원 암호기술연구팀 선임연구원
††††† 정 회 원 : 한국전자통신연구원 암호기술연구팀 팀장
논문접수 : 2007년 12월 27일
수 정 일 : 2008년 3월 10일
심사완료 : 2008년 3월 19일

식 관점에서 가장 중요하게 획득해야 하는 자료에 대해서 조사한다. 또한 이전 Windows에서부터 사용되어 왔지만 Windows Vista에서 보안적 취약점을 보완해서 더욱 강력해진 UAC[8,9,10,11]와 EFS[13,14,15]에 대한 분석하고 기존 Windows 운영체제와 비교하여 포렌식을 통해 데이터를 획득하기 위해서 확인해야 할 변경된 특징에 대해서 살펴본다. 본 논문의 구성은 2장에서는 새롭게 추가된 보안 메커니즘인 BitLocker에 대해서 분석하고 3장에서는 UAC를 4장에서는 EFS에 대해서 분석한다. 5장에서는 포렌식 관점에서 활용 가능한 Windows Vista의 자료를 살펴보고 6장에서 결론을 내린다.

2. Windows BitLocker

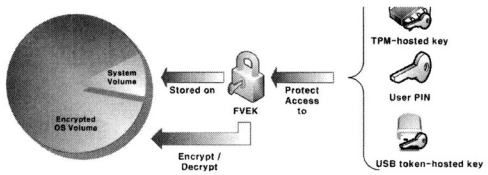
BitLocker는 기존에 Windows 운영체제에서 사용하던 보안 메커니즘과는 다른 새로운 방식의 보안 메커니즘이다[2]. Windows 운영체제에서 기존에 사용하던 EFS, RMS(Right Management System) 메커니즘은 사용자 로그인 정보를 바탕으로 작동하는 보안 메커니즘이다. 운영체제의 부팅이 완료된 후 사용자가 로그인한 정보를 바탕으로 적용되는 보안 메커니즘이기 때문에 악의적인 사용자에 의해서 로그인한 사용자 정보가 해킹당하거나 루트킷에 의해서 비정상적인 방법으로 로그인 할 수 있는 보안상의 취약점 있다. 이러한 보안상의 취약점을 해결하기 위해서 부팅 단계에서부터 부팅에서 필요한 컴포넌트들의 무결성을 검증하고 사용자로부터 입력 받는 정보를 바탕으로 운영체제를 부팅한다.

2.1 BitLocker 특징

2.1.1 디스크 볼륨 암호화

BitLocker가 기존의 Windows 운영체제에서 사용하는 보안 메커니즘과 가장 큰 차이점은 보안 메커니즘이 적용되는 대상이다. 기존에 사용하는 보안 메커니즘은 각 파일을 대상으로 사용자 로그인 정보를 바탕으로 보안 메커니즘을 적용하지만 BitLocker는 컴퓨터를 사용하는 사용자 정보와 상관없이 컴퓨터를 사용하기 전에 사용자가 가지고 있는 정보를 확인해서 디스크 볼륨 전체를 암호화하여 부팅과정에서 무결성이 보장된 후 디스크 복호화가 이루어지는 방식이다.

Windows Vista에서 디스크 볼륨은 운영체제가 설치되는 Windows OS 볼륨과 Windows OS 볼륨을 암호화하기 위하여 사용되는 FVEK(Full Volume Encryption Key)를 저장하는 시스템 볼륨으로 구성된다. 시스템 볼륨에 저장되는 FVEK는 외부로부터 입력받는 키에 의해서 보호된다. BitLocker 보안 메커니즘에서 부팅시 입력 가능한 키의 종류는 TPM 만을 사용하는 방법, PIN(Personal Identification Number)를 입력 받는 방법, USB 저장 장치에 저장되어 있는 startup-key를 사용하는 방법이 있다. (그림 1)은 BitLocker가 Windows 운영체제에 적용될 경우, 볼륨들의 구성과 이 볼륨들과 암호화 키 간의 관계를 보여주고 있다.

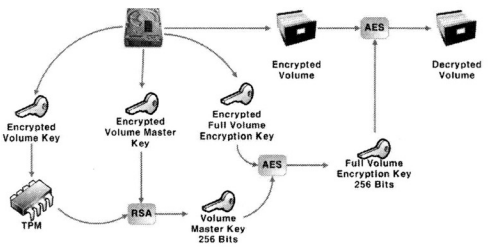


(그림 1) BitLocker 보안 메커니즘

2.1.2 TPM(Trusted Platform Module)

TPM은 기존의 사용하던 보안 메커니즘에서 가장 기본적인 문제인 보안 메커니즘이 동작하는 플랫폼의 신뢰성을 보장하기 위해서 사용한다. TPM은 Intel, Microsoft, Sun Microsystems와 같은 기업들이 모여 조직한 TCG(Trusted Computing Group)이 플랫폼에서 발생하는 신뢰성에 대한 문제를 해결하기 위해서 만든 하드웨어 칩이다. TPM은 서버, 모바일 디바이스, 개인용 컴퓨터 등의 다양한 플랫폼에 적용 가능하고 이들 플랫폼에서 설치해서 사용할 수 있는 Windows, Linux와 같은 운영체제와 같이 연동하여 사용할 수 있다.

Windows Vista에서 TPM은 BIOS, MBR 코드, Boot Sector 코드등과 같은 부팅과정에서 사용되는 구성 요소들에 대한 무결성을 검증과 BitLocker 보안 메커니즘에서 필요한 암호화 알고리즘을 TPM을 통해서 안전하게 사용할 수 있다. (그림 2)는 TPM이 있는 플랫폼에서 BitLocker를 적용할 경우, Windows Vista의 무결성을 보장하기 위한 TPM의 역할과 이를 통한 Windows 운영체제의 부팅과정을 설명한다.



(그림 2) TPM만을 사용하여 BitLocker를 적용한 경우

2.1.3 Diffuser+AES-CBC

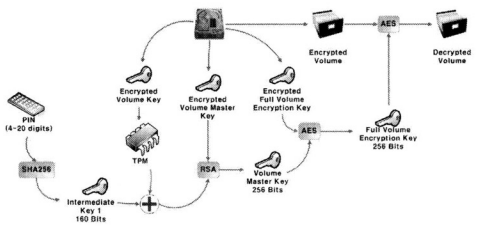
볼륨 전체를 암호화 하는 보안 메커니즘을 위해서는 아주 신뢰할 수 있는 보안 알고리즘을 필요로 하고 또한 많은 데이터를 암호화 하여야 하므로 보다 빠른 암호화 알고리즘을 적용해야 할 것이다. 따라서 기존에 암호화 메커니즘에서 사용하던 대칭 암호화 방법과 비대칭 암호화 방법들 중 한 가지 방법만으로는 이러한 암호화의 신뢰성과 신속성에 대한 문제를 모두 충족할 수 없기 때문에 볼륨 전체를 암호화하는 BitLocker에 적합하지 않다. Microsoft에서는 이러한 두 개의 문제를 해결하기 위해서 새롭게 diffuser+AES-CBC

암호화 알고리즘을 사용하였다[18]. diffuser+AES-CBC 알고리즘은 diffuser 알고리즘이 아직 보안상으로 확인이 안 된 단점을 가지고 있지만 이 diffuser 알고리즘이 깨지더라도 최소한 AES-CBC 알고리즘으로 보호할 수 있고 전체 볼륨을 보호하기에 다른 알고리즘에 비해서 빠른 속도를 가진 장점을 가지고 있다.

2.2 BitLocker를 적용한 시스템 볼륨 보안 체계

2.2.1 TPM만 적용된 BitLocker

BitLocker를 위해서 TPM 만을 적용할 경우 복호화 절차는 (그림 3)과 같다. 먼저 플랫폼에서는 부팅을 진행하고 있는 볼륨이 현재 사용하는 컴퓨터에서 암호화 된 볼륨인지 확인한다. 그리고 TPM을 통해서 시스템 볼륨에 암호화되어 있는 볼륨 키를 복호화 한다. 볼륨 키는 시스템 볼륨에 저장되어 있는 볼륨 마스터 키를 복호화 하고 복호화 된 볼륨 마스터 키는 FVEK를 복호화 하는데 사용된다. 이렇게 복호화 된 FVEK를 통해 BitLocker로 암호화 된 Windows OS 볼륨을 복호화 하기 때문에 BitLocker가 적용된 컴퓨터에서 암호화 된 볼륨을 다른 컴퓨터로 옮겨서 사용 하려고 할 경우 사용할 수가 없게 된다.



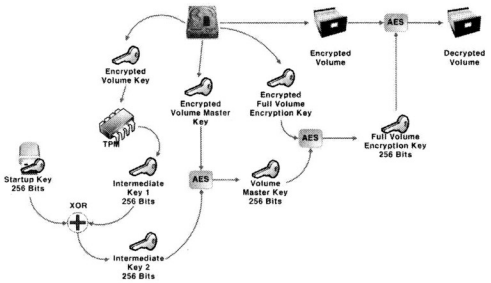
(그림 3) TPM과 PIN을 사용하는 BitLoker를 적용한 경우

2.2.2 TPM과 PIN을 사용하는 방법이 적용된 BitLocker

PIN과 TPM을 사용해 BitLocker를 적용한 방법은 (그림 3)과 같이 더 강력한 보안을 적용할 수 있다. 이 경우 BitLocker는 TPM뿐만 아니라 컴퓨터를 사용하려고 하는 사용자가 알고 있는 PIN 정보를 확인하게 된다. TPM만을 사용해 BitLocker를 적용한 방법과 차이점은 암호화 된 볼륨 마스터 키를 복호화 하기 위해서 TPM을 통해서 복호화 된 볼륨 키와 외부에서 입력 받은 PIN으로 만들어진 임시키를 함께 이용한다. 이때 PIN은 4-20 자리의 숫자로 구성되며, SHA256 해시 알고리즘을 통해서 임시키를 만들고 이중 160-bit와 볼륨 키로 키를 만들어서 볼륨 마스터 키를 복호화 하기 위한 키로 사용한다.

2.2.3 TPM과 startup-key를 사용한 방법이 적용된 BitLocker

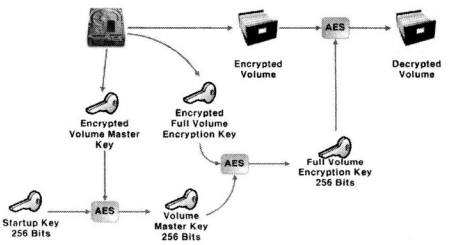
TPM과 startup-key를 사용해 BitLocker를 적용할 경우에는 (그림 4)와 같이 조금 더 복잡한 과정을 거쳐서 복호화 한다. TPM과 startup-key를 사용할 경우에는 컴퓨터를 사용하는 사용자가 컴퓨터를 사용하기 위해서 필요한 권한을 가지고 있는지 확인한다. 컴퓨터를 사용하기 위한 권한은 USB 저장 장치에 저장되어 있는 startup-key로 확인한다. 복호화 과정은 외부로부터 256-bit의 startup-key와 TPM을 통해서 복호화 된 볼륨 키와 XOR 연산을 해서 256-bit의 키로 볼륨 마스터 키를 복호화 한다.



(그림 4) TPM과 USB dongle(Startup-key)를 사용하여 BitLocker를 적용한 경우

2.2.4 TPM 없이 BitLocker 적용

TPM 없이 BitLocker를 Windows Vista에서 적용할 경우 TPM이 있는 경우보다 보안상에 취약점이 있다. 이 방법은 PIN을 사용할 수 없고 오직 USB dongle에 저장되어 있는 startup-key를 사용한 방법만으로 로그인이 가능하다. USB에 있는 startup-key만을 사용해서 볼륨 마스터 키를 복호화 하기 때문에 부팅 과정에서 사용하는 구성 요소들에 대한 무결성을 검증할 수가 없다.



(그림 5) TPM 없이 BitLocker를 적용한 경우

2.3 BitLocker를 적용한 데이터 볼륨 보안 체계

Windows Vista에서 다른 볼륨을 보호하기 위하여 EFS를 이용할 수 있다. 이러한 EFS 키는 기본적으로 OS 볼륨에 저장이 되므로, 만일 OS 볼륨에 BitLocker가 적용될 경우, EFS에 의해 보호되는 다른 데이터도 간접적으로 BitLocker에 보호를 받게 되는 것이다. 이러한 간접적인 방법 외에도 Windows OS 볼륨이 아닌 데이터 볼륨도 사용자가 직접 암호화 할 수 있다[4]. Windows OS 볼륨과 마찬가지로 NTFS 파일 시스템 형식으로 포맷되어 있는 볼륨이어야 하고,

manage-bde.wsf 명령어를 이용하여 명령줄(command line) 도구를 통해서 암호화 가능하다.이때 복호화 키는 Windows OS 볼륨 영역에 있는 레지스트리에 저장되어 있다. (HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FVEAutoUnlock\{VolumeGUID}). Windows OS 볼륨이 복호화 된 후에 데이터 볼륨은 자동으로 사용 가능해야 한다.

2.4 포렌식 관점에서 BitLocker

Windows Vista에서 처음 사용된 보안 메커니즘인 BitLocker는 기존의 포렌식 도구로는 해석이 불가능하다. 따라서 현재는 컴퓨터 사용자 도움 없이 적용 가능한 포렌식 방법이 없다. 기존에 사용하던 보안 메커니즘과는 적용 대상이나 적용 알고리즘이 다르기 때문에 저장장치에서 직접 데이터를 획득하기는 힘들다. 특히 TPM이 적용되어있는 시스템에 하드디스크의 경우 다른 컴퓨터로 옮겨 복호화 하는 것은 불가능하다. 따라서 자체 컴퓨터에서 BitLocker를 해제한 후에 디스크를 다른 컴퓨터의 logical drive로 마운트 하여야만 내용을 읽을 수가 있다. 또한 start-key나 PIN을 이용하는 경우 BitLocker를 해제하기 위하여서는 컴퓨터 사용자가 가지고 있는 이들 정보를 획득하는 것이 무엇보다 중요하다. 포렌식 관점에서 이러한 시스템에 대하여 다음의 데이터 획득을 목적으로 하여야 한다

- startup-key나 복구키(recovery key)가 저장되어 있는 USB
- PIN이 또는 복구 패스워드(recovery password)

특히 복구키나 복구 패스워드는 새로운 PC 업그레이드 또는 TPM 장애시 복구를 위하여 관리하는 정보이다. 이러한 정보를 획득할 경우 암호화되어 있는 디스크를 원활하게 복호화 할수 있을 것이다.

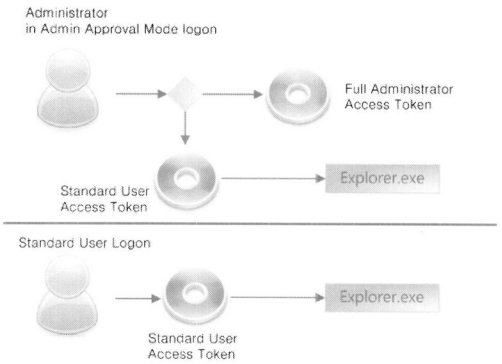
BitLocker가 적용되어 있는 데이터 볼륨의 경우, Windows OS 볼륨에 적용되어 있는 BitLocker를 해제한 후 Windows OS 볼륨에 있는 레지스트리를 통해서 데이터 볼륨의 복호화 키를 획득해야 한다. 이때 레지스트리는 EFS 암호화 메커니즘에 의해서 암호화 되어 있기 때문에 이를 우회하는 방법으로 키를 획득하는 것이 가능할 것이다.

3. UAC(User Account Control)

UAC는 컴퓨터에 로그온하는 사용자가 설치되어 있는 시스템 프로그램과 사용자 프로그램을 사용, 삭제, 프로그램 사용에 관한 권한을 다른 사용자에게 양도 등 로그인한 사용자 계정에 관한 다양한 설정을 통해서 Windows Vista 보안을 강화 한다.

3.1 UAC 특징

기존의 Windows 운영체제의 경우 사용자 계정은 '관리자 계정'과 '사용자 계정'으로 나뉜다. 관리자 계정으로 로그인



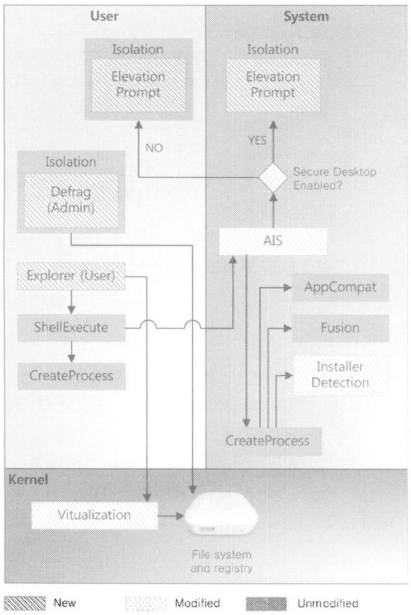
(그림 6) Windows Vista UAC 동작 메커니즘

할 경우 모든 리소스의 사용에 사용자 확인 없이 설치하거나 실행하는 것이 자유롭기 때문에 기존의 많은 Malware와 바이러스가 사용자의 의도와 상관없이 작동했다. 또한 사용자 계정이 관리자 권한을 가질 경우 관리자 계정과 비슷한 동작을 하는 것이 가능하기 때문에 사용자도 모르게 Malware와 바이러스 프로그램을 설치(silent 설치)하고 실행하는 것이 가능하다.

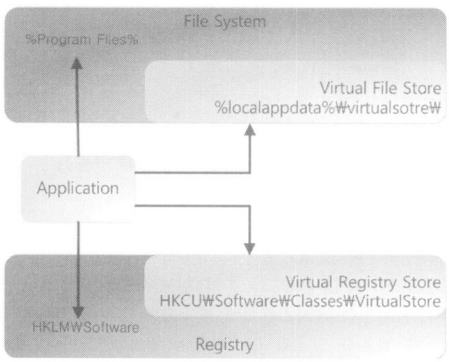
Windows Vista의 경우 모든 리소스들에 대한 권한을 가진 '관리자 계정'과 제한된 권한을 가진 '표준 사용자 계정'으로 나누어진다. 기존의 Windows 운영 체제는 관리자 계정의 경우 모든 리소스를 사용할 수 있는 액세스 토큰이 부여 되지만 Windows Vista의 경우 FAAT(Full Administrator Access Token)와 SUAT(Standard User Access Token) 두 가지 종류의 액세스 토큰이 주어진다. 표준 사용자 계정의 경우 SUAT 한 종류의 액세스 토큰만이 주어진다. FAAT와 SUAT 두 개의 토큰이 주어지는 관리자 계정의 경우 사용자 프로그램 실행에는 SUAT를 바탕으로 프로그램을 실행하고 시스템 프로그램의 경우 FAAT를 바탕으로 프로그램을 실행한다. (그림 6)은 Windows 운영체제에서 관리자 계정과 표준 사용자 계정이 프로그램을 실행할 때 사용되는 액세스 토큰의 구조를 나타낸 그림이다.

3.2 UAC 작동 메커니즘

UAC의 가장 큰 특징은 Windows Vista가 실행하는 모든 프로그램에 기반이 되는 보안 메커니즘이라는 것이다. 따라서 UAC는 사용자 프로그램과 시스템 프로그램 모두를 실행하기 위한 메커니즘과 프로그램을 실행하는 사용자의 액세스 토큰을 확인하는 메커니즘이 모두 필요하다. Windows Vista에서는 파일 시스템과 레지스트리에 접근하기 위해서 독립적인 실행 프로그램을 통해서 실행해야 한다. 또한 대부분의 Windows Vista에서는 explorer.exe가 SUAT를 기반으로 실행되기 때문에 이를 통해서 실행되는 사용자 프로그램 역시 SUAT 통해서 실행된다. 시스템 영역에서 실행하는 프로그램의 경우 기존의 Windows 운영체제의 메커니즘



(그림 7) Windows Vista UAC구조



(그림 8) UAC의 가상화

이 강화된 메커니즘을 사용한다. 만약 시스템 영역에서 실행해야할 프로그램은 관리자 계정의 경우 FAAT를 통해서 실행하고 표준 사용자 계정의 경우 FAAT를 얻기 위해서 explorer.exe와 독립된 입력 창을 통해서 관리자 권한을 획득해야 한다. (그림 7)은 Windows Vista에서 UAC의 구조와 UAC가 작동되는 메커니즘을 나타낸다.

가상화(Virtualization)는 표준 사용자 계정으로 로그인 했을 때 FAAT를 가지고 있지 않기 때문에 발생하는 호환성 문제를 해결한다. FAAT를 가져야 쓸 수 있는 레지스트리나 파일 시스템 공간에 데이터를 써야 할 경우 발생할 수 있는 호환성 문제를 표준 사용자가 접근할 수 있는 영역과

〈표 1〉 Windows Vista의 디렉터리 변경 사항

기존의 Windows 운영체제(2000, XP)	Windows Vista
\Documents and Settings*	\Users
\Documents and Settings\<user>	\Users\<user>
\Documents and Settings\<user>\Local Settings*	\Users\<user>\AppData\Local
\Documents and Settings\Recent*	\Users\<user>\AppData\Roaming\Microsoft\Windows\Recent
\Documents and Settings\Start Menu*	\Users\<user>\AppData\Roaming\Microsoft\Windows\Start
\Documents and Settings\Local Settings\History	\Users\<user>\AppData\Local\Microsoft\Windows\History
\Documents and Settings\Local Settings\Temporary Internet Files	\Users\<user>\AppData\Local\Microsoft\Windows\Temporary
\Documents and Settings\<user>\Application Data*	\Users\<user>\AppData
\Documents and Settings\<user>\Cookies*	\Users\<user>\AppData\Roaming\Microsoft\Windows\Cookies\Low
thumbs.db	\Users\<user>\AppData\Local\Microsoft\Windows\Explorer
\Recycled or \Recycler	\$Recycle.Bin
HKLM\Software	HKUS\<User SID>_Classes\VirtualStore\Machine\Software

그 데이터를 실제로 써야하는 영역을 매핑 시키는 방법으로 이를 해결한다. 예를 들어 표준 사용자가 C:\Program Files에 데이터를 써야할 경우 직접 저장하게 되면 모든 사용자에 영향을 받게 되므로 대신 %localappdata₩virtualstore에 데이터를 쓰게 되고 이를 C:\Program Files에 등록되지만 저장한 사용자만이 사용하게 된다. 또한 레지스트리 정보의 경우에도 마찬가지로 HKLM\SOFTWARE 영역에 데이터를 써야할 경우 대신에 HKCU\Software\Classes\VirtualStore에 데이터를 저장하고 이를 HKLM\SOFTWARE와 매핑하는 방식으로 FAAT 토큰 없이도 사용할 수 있게 한다.

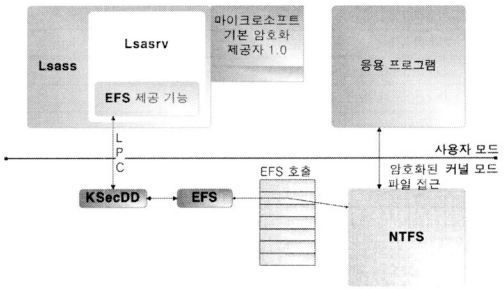
3.3 포렌식 관점에서 UAC

UAC는 운영체제의 부팅이 완료된 다음에 로그인 한 계정에 적용된다[12]. 따라서 BitLocker의 적용을 무효화 시킨 후 다른 컴퓨터로 저장 장치를 옮길 경우 기존의 포렌식 툴을 사용하여 데이터를 읽을 수 있다. 하지만 이전의 Windows 버전과 가상화 때문에 기존에 사용하던 디렉터리와 다른 변경된 디렉터리 경로명을 통해서 접근해야 한다. 데이터뿐만 아니라 레지스트리 정보 역시 기존과 다른 경로명을 통해서 접근해야 한다. 포렌식을 위해서 UAC를 보안 메커니즘은 경로의 변경을 확인하여야 한다. <표 1>은 Windows Vista의 디렉터리 변경 사항을 정리한 것이다.

4. EFS(Encrypting File System)

4.1 EFS 특징

Windows 운영체제가 다중 사용자를 위해서 계정을 만들 수 있고, 사용자 계정이 관리자 권한을 가진 경우 다른 사



(그림 9) EFS 구조

용자가 소유한 파일을 접근을 할 수 있기 때문에 이를 막기 위하여 각 사용자 별로 자신이 소유한 파일과 디렉토리를 다른 사용자가 볼 수 없도록 암호화를 할 수 있는 기능이 필요하다. EFS는 Windows 2000부터 적용된 보안 메커니즘이다. Windows 운영체제에서는 기본적으로 ACL(Access Control List)에 의해서 데이터 접근에 대한 검사를 하여 데이터를 보호한다. 하지만 운영체제를 거치지 않고 저장 장치에 물리적으로 접근하거나 다른 컴퓨터에 저장 장치를 옮겨서 설치할 경우, 저장 장치가 설치된 컴퓨터의 관리자 계정으로 로그인 할 때는 저장 장치에 있는 모든 데이터에 대하여 접근이 가능하기 때문에 이러한 경우에도 데이터를 보호할 수 있는 기능이 필요하다.

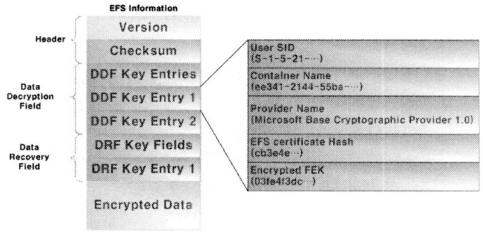
4.2 EFS 작동 메커니즘

4.2.1 EFS 구조

EFS는 Windows 2000 이후의 운영체제에서 사용하는 NTFS파일 시스템과 밀접한 관련이 있다[3,16]. 그 외에도 데이터를 암호화하기 위한 컴포넌트(Microsoft 기본 암호화 제공자 1.0), 사용자 세션 관리(LSASS : Local Security Authority Subsystem), I/O 관리자등과 같이 데이터를 암호화해서 저장하기 위한 다양한 컴포넌트들과의 연동이 필요하다.

4.2.2 EFS 파일 암호화 포맷

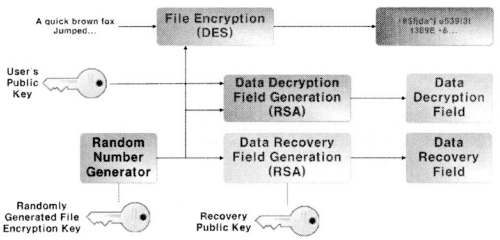
EFS는 NTFS파일 시스템을 기반으로 한다. NTFS파일 시스템은 파일을 저장할 때 데이터와 별도의 MFT(Master File Table)로 나누어 저장된다. MFT에는 파일의 다양한 속성 정보가 저장되고 EFS 파일 시스템을 위한 속성도 역시 MFT에 저장된다. 이 속성은 $EFS 속성이라 부른다. 이 속성 안에는 DDF(Data Description Filed)와 DRF(Data Recovery File) 항목을 가지고 있다. DDF는 파일에 대한 권한이 있는 사용자 마다 별도로 만들어지고, DRF는 파일을 복구할 권한이 있는 사용자마다 별도로 만들어진다.



(그림 10) EFS 정보의 구조와 키 엔트리

4.2.3 EFS 암호화 메커니즘

EFS는 사용자 정보를 바탕으로 암호화를 한다. 이때 파일에 대한 권한을 가진 사용자와 파일을 복구할 수 있는 권한을 가진 사용자 정보를 바탕으로 DDF와 DRF를 만든다. 이때 사용되는 알고리즘은 비대칭 암호화 알고리즘인 RSA 알고리즘으로 암호화 하고 실제 데이터는 DES 암호화 알고리즘을 바탕으로 암호화 한다.



(그림 11) EFS 암호화 메커니즘

4.3 포렌식 관점에서 EFS

EFS는 기존에 운영체제에서 사용되던 보안 메커니즘이고, Windows Vista에서도 기존의 윈도우즈와 변환점이 많지 않다. 차이점은 레지스트리에 저장되는 위치와 EFS를 통해서 적용 가능한 암호화 알고리즘의 종류 정도이다. 또한 파일을 복구할 수 있는 계정의 경우 사용자가 권한을 설정해 주지 않아도 복구하는 것이 가능하므로 관리자 계정을 알아내는 방법을 통해서 포렌식이 가능하다.

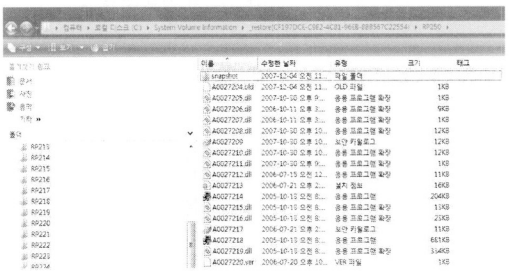
〈표 2〉 사용 가능한 암호화 알고리즘 종류

NTFS Version	Platform	Support	Value
5.0	Windows 2K	DESX	0x6604
5.1	Windows XP Pro	DESX, 3DES	0x6604
5.1	Windows XP Pro SP1	DESX, 3DES, AES	0x6603
5.2	Windows 2003	DESX, 3DES, AES	0x6610
6.0	Windows Vista	DESX, 3DES, AES, ...	0x6610

5. 포렌식 관점에서 추가적으로 봐야할 데이터

5.1 Shadow Copy와 Volume Snapshot 활용

기존의 Windows 운영체제에서 사용하던 "시스템 복원" 기능에 각 파일별 백업 기능이 추가되어서 Shadow Copy라는 기능과 Volume Snapshot 서비스라는 기능으로 사용된다. Shadow Copy는 볼륨 전체의 데이터를 백업해서 저장하거나, 변경된 점만 저장하는 방법을 선택적으로 사용하는 것이 가능하다. 이렇게 저장된 정보를 바탕으로 전체 볼륨을 복원하거나 특정 파일 별로 복원하는 것도 가능하다. Volume Snapshot 서비스는 모든 파일을 모니터링해서 파일들의 위치와 리스트를 업데이트 한다. Shadow Copy와 Volume Snapshot 기능의 정보는 해당 볼륨의 \System Volume Information 디렉터리에 저장된다.



(그림 12) Shadow copy & Volume Snapshot 저장

5.2 이벤트 로그의 활용

일반적으로 로그 파일들은 포렌식 조사에 있어 정보 획득을 위한 중요한 자료 중에 하나이다. 그러나 Windows NT와 같은 기존의 로그 서비스는 모든 로그 정보가 메모리에 올라와 있어 비효율적인 단점이 있었다. Windows Vista에서는 바이너리 XML 파일 포맷 형태로 이벤트 로그를 사용하고 이것은 새롭게 .evtx 확장되어 사용한다. 특히 Windows Vista에 새로운 로그 파일 포맷은 파일 header와 chunk 부분으로 구현이 되어 있고 기존의 방식과 비교하여 상대적으로 적은 시스템 자원을 사용할 수 있도록 운영 되고 있다[1]. 또한 이벤트 로그 데이터는 새로운 로그 포맷에 맞추어 \Windows\System32\winevt\Logs 디렉터리에 저장되어 있다. 이벤트 로그인 활용의 한 예로, 로그 파일에 고유한 매직 스티링(magic strings)과 블록 중심의 파일 정렬(block-oriented file layout)은 원시 파일 시스템 이미지에서 Vista 이벤트 로그 파일들의 파일 header와 chunk를 찾아내고자 하는 포렌식 조사에 도움이 될 수 있을 것 이다. 하지만 현재까지 기존의 포렌식 도구로는 새로운 이벤트 로그 파일을 복수 없고 Windows Vista Event Viewer(eventvwr.exe)로만 볼 수 있다.

이름	수정한 날짜	유형	크기
ACEEventLog	2007-12-06 오전 8:...	이벤트 로그	1,028KB
Application	2007-12-06 오전 8:...	이벤트 로그	1,092KB
DFS Replication	2007-12-05 오후 1:...	이벤트 로그	68KB
HardwareEvents	2007-12-05 오후 1:...	이벤트 로그	68KB
Internet Explorer	2007-12-05 오후 1:...	이벤트 로그	68KB
Key Management Service	2007-12-05 오후 1:...	이벤트 로그	68KB
Microsoft-Windows-Bits-Client%4Operational	2007-12-06 오전 8:...	이벤트 로그	1,028KB
Microsoft-Windows-CodeIntegrity%4Operational	2007-12-06 오후 3:...	이벤트 로그	1,028KB
Microsoft-Windows-Diagnosis-DPS%4Operational	2007-12-06 오전 8:...	이벤트 로그	1,028KB
Microsoft-Windows-Diagnosis-PLA%4Operational	2007-12-05 오후 1:...	이벤트 로그	68KB
Microsoft-Windows-Diagnostics-Networking%4Operational	2007-12-07 오전 11...	이벤트 로그	68KB
Microsoft-Windows-Diagnostics-Performance%4Operational	2007-12-06 오전 8:...	이벤트 로그	1,028KB
Microsoft-Windows-DriverFrameworks-UserMode%4Operational	2007-12-06 오전 8:...	이벤트 로그	1,028KB
Microsoft-Windows-GroupPolicy%4Operational	2007-12-06 오전 8:...	이벤트 로그	1,092KB
Microsoft-Windows-International%4Operational	2007-12-06 오전 8:...	이벤트 로그	68KB
Microsoft-Windows-Kernel-WHEA	2007-12-06 오전 8:...	이벤트 로그	68KB
Microsoft-Windows-LanguagePackSetup%4Operational	2007-12-06 오후 1:...	이벤트 로그	68KB

(그림 13) Windows Vista 로그 데이터

6. 결 론

Windows Vista 운영체제는 기존의 XP와 2000에 비해 강화된 보안 메커니즘을 사용한다. 사용자 입장에서는 강화된 보안 메커니즘의 사용으로 개인 정보 및 기업 정보를 안심하고 저장할 수 있지만, 반대로 포렌식적 관점에서 보면 강화된 보안 메커니즘에 의해서 저장 장치에 저장된 범죄와 관련된 데이터들을 확인하기가 어려워졌다. 또한 디렉터리의 변경이나 기존에 사용하던 보안 메커니즘의 구조도 변경되었기 때문에 기존에 사용하던 포렌식 도구를 이용하여 데이터를 획득하기가 힘들어졌다. 특히, BitLocker는 새롭게 나온 보안 메커니즘으로 기존에 사용하던 포렌식 도구로 데이터를 획득하는 것 자체가 불가능하다. 따라서 기존에 연구되었던 포렌식 방법에 Windows Vista에서 변경된 정보를 업데이트해서 포렌식을 위한 도구를 개발하여야 한다. 그 외 로그 데이터나 윈도우즈 자체 복원 프로그램을 통해서 데이터를 복원하기 위해서 분석하는 방법의 개발도 필요하다. 또한 BitLocker와 같은 경우 기존의 보안 메커니즘과 다른 방법을 사용하므로 BitLocker가 적용된 컴퓨터를 부팅하기 위해서 사용자가 소유한 PIN이나 시작 키, 복구 키 또는 복구 암호를 확보하는 것을 일차적 목표로 해야 한다. 궁극적으로 BitLocker 보안 메커니즘의 분석을 바탕으로 포렌식을 적용시킬 수 있는 방법을 연구해야 할 것이다.

참 고 문 헌

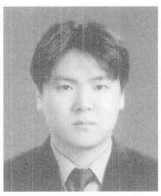
[1] Andreas Schuster, "Introducing the Microsoft Vista event log file format," Digital Investigation Vol.4, Supplement-1, Sep. 2007, pp.65-72.

[2] Douglas MacIver, "Penetration Testing windows Vistatm BitLockertm Drive Encryption," HITBSecConf2006, 2006

[3] Andrey Malyshev and Serg Vasilenkov, "Security Analysis of Microsoft Encrypting file System(EFS)," Black Hat Europe 2003 Conference, 2003.

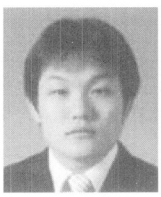
[4] Lance Mueller, "First Looks Basic Investigations of Windows Vista," Computer and Enterprise Investigations Conference 2007.

[5] Microsoft Corporation, "BitLocker Drive Encryption Technical Overview," http://technet.microsoft.com/en-us/windowsvista/aa906017.aspx, 2006.
[6] Microsoft Corporation, "Windows BitLocker Drive Encryption Frequently Asked Questions," http://technet2.microsoft.com/WindowsVista/en/library/58358421-a7f5-4c97-ab41-2bcc61a58a701033.mspx, 2006.
[7] Microsoft Corporation, "BitLocker Drive Encryption : Scenarios, User Experience, and Flow," http://www.microsoft.com/whdc/system/platform/hwsecurity/BitLockerFlow.mspx, 2006.
[8] Microsoft Corporation, "The Windows Vista and Windows Server 2008 Developer Story : Windows Vista Application Development Requirements for User Account Control(UAC)," http://msdn2.microsoft.com/en-us/library/aa905330.aspx, 2006
[9] Microsoft Corporation, "Understanding and Configuring User Account Control in Windows Vista," http://technet2.microsoft.com/WindowsVista/en/library/00d04415-2b2f-422c-b70e-b18ff918c2811033.mspx, 2006.
[10] Microsoft Corporation, "Windows User Account Control Step-by-Step Guide," http://technet2.microsoft.com/WindowsVista/en/library/0d75f774-8514-4c9e-ac08-4c21f5c6c2d91033.mspx, 2006.
[11] Microsoft Corporation, "Getting Started with User Account Control on Windows Vista," http://technet.microsoft.com/en-us/windowsvista/aa906022.aspx, 2006.
[12] Microsoft Corporation, "Windows Vista Developer Story(Help File)," http://msdn2.microsoft.com/en-us/windowsvista/aa904951.aspx, 2006.
[13] Microsoft Corporation, "How EFS Work," http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/distrib/dsck_efs_duwf.mspx, 2006.
[14] Microsoft Corporation, "Windows Data Protection," http://msdn2.microsoft.com/en-us/library/ms995355.aspx, 2006
[15] Jim Moeller, "Microsoft Windows Vista Forensic Jumpstart," Techno Security 2007.
[16] Mark E. Russinovich and David A. Solomon, 'Microsoft Windows Internals, 4th Edition,' Microsoft Press, 2006
[17] Microsoft TechNet, "BitLocker 드라이브 암호화를 사용하여 데이터를 보호하기 위한 핵심 사항," http://www.microsoft.com/technet/technetmag/issues/2007/06/BitLocker/default.aspx?loc=ko, 2007 June.
[18] Niels Ferguson, "AES-CBC+Elephant diffuser A Disk Encryption Algorithm for Windows Vista," http://download.microsoft.com/download/0/2/3/0238acaf-d3bf-4a6d-b3d6-0a0be4bbb36e/BitLockerCipher200608.pdf, 2006.
[19] Shon Eizenhoefer, "BitLocker Drive Encryption Hardware Enhanced Data Protection," Microsoft WinHEC 2006.
[20] 정준석, 정원용, '임베디드 개발자를 위한 파일시스템의 원리와 실습,' 한빛미디어, 2006.



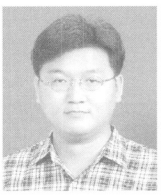
황 성 호

e-mail : seongho.hwang@nhncorp.com
2006년 건국대학교 컴퓨터공학과(학사)
2008년 건국대학교 컴퓨터공학부(공학석사)
2008년~현 재 NHN 어플리케이션 보안팀
관심분야 : 임베디드 시스템, 컴퓨터 보안 시스템 등



남 현 우

e-mail : namhw@konkuk.ac.kr
2007년 건국대학교 컴퓨터공학과(학사)
2008년~현 재 건국대학교 컴퓨터공학부 (석사과정)
관심분야 : 임베디드 시스템, 운영체제, 멀티미디어 정보 검색 등

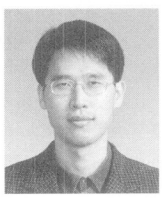


박 능 수

e-mail : neungsoo@konkuk.ac.kr
1991년 연세대학교 전기공학과(학사)
1993년 연세대학교 대학원 전기공학과(석사)
2002년 미국 Univ. of Southern California, 전기공학과 컴퓨터공학(공학박사)
2002년~2003년 삼성전자 책임연구원
2003년~현 재 건국대학교 컴퓨터공학부 부교수
관심분야 : 컴퓨터구조, 임베디드 시스템, 병렬시스템, 멀티미디어 컴퓨팅, 컴퓨터보안 등

조 수 형

e-mail : shjo@etri.re.kr
1997년 경북대학교 전자공학과(학사)
1999년 경북대학교 대학원 전자공학과(석사)
2007년 충북대학교 대학원 전자계산학과 (이학박사)
2000년~현 재 한국전자통신연구원 암호기술연구팀 선임연구원
관심분야 : 정보보호, 디지털 포렌식, 네트워크 보안 등



홍 도 원

e-mail : dwhong@etri.re.kr
1994년 고려대학교 수학과(학사)
1996년 고려대학교 대학원 수학과(석사)
2000년 고려대학교 대학원 수학과(이학박사)
2000년~현 재 한국전자통신연구원 암호기술연구팀 팀장
관심분야 : 암호학, 정보보호, 디지털 포렌식 등