KIPS Tr. Comp. and Comm. Sys.
Vol.5, No.12 pp.491~498 pISSN: 2287-5891

https://doi.org/10.3745/KTCCS.2016.5.12.491

A Study of Acquisition and Analysis on the Bios Firmware Image File in the Digital Forensics

Seung Hoon Jeong† · Yun Ho Lee†† · Sang Jin Lee†††

ABSTRACT

Recently leakages of confidential information and internal date have been steadily increasing by using booting technique on portable OS such as Windows PE stored in portable storage devices (USB or CD/DVD etc). This method allows to bypass security software such as USB security or media control solution installed in the target PC, to extract data or insert malicious code by mounting the PC's storage devices after booting up the portable OS. Also this booting method doesn't record a log file such as traces of removable storage devices. Thus it is difficult to identify whether the data are leaked and use trace-back technique. In this paper is to propose method to help facilitate the process of digital forensic investigation or audit of a company by collecting and analyzing BIOS firmware images that record data relating to BIOS settings in flash memory and finding traces of portable storage devices that can be regarded as abnormal events.

Keywords : Digital Forensic, BIOS Firmware Image, NVRAM Variable Area, BIOS Boot Sequence

디지털 포렌식 관점에서 BIOS 펌웨어 이미지 파일 수집 및 분석에 관한 연구

정 승 훈† · 이 윤 호†† · 이 상 진†††

요 약

최근 Windows PE와 같은 포터블 OS를 USB, CD/DVD 등의 이동식 저장매체에 저장하여 부팅하는 기법으로 기밀자료 및 내부정보가 유출되는 사례가 증가하고 있다. 이동식 저장매체를 이용한 이 부팅 기법은 타깃 PC에 설치된 USB 보안, 매체제어솔루션 등의 보안 소프트웨어의 우회가 가능하고, 부팅 후 PC의 저장매체를 마운트하여 정보 추출 및 악성코드 삽입 등의 행위가 가능하며, 이동식 저장매체의 사용흔적과 같은 로그기록이 남지 않는 특징이 있어 자료유출여부 확인과 역추적이 어렵다. 이에 본 논문에서는 플래시 메모리에서 BIOS 설정과 관련된 데이터가 기록되는 BIOS 펌웨어 이미지를 수집 및 분석하여 이상행위로 추정할 수 있는 이동식 저장매체를 이용한 부팅 흔적을 찾아 기업의 감사 또는 디지털 포렌식 수사를 수행하는데 도움이 될 수 있는 방안을 제시한다.

키워드 : 디지털포렌식, 바이오스 펌웨어 이미지, NVRAM Variable Area, BIOS 부팅 순서

1. 서 론

기밀자료 및 기술유출사고는 주로 전·현직 직원, 협력업체 등에 의해 발생하며, 이로 인한 국가 및 기업의 금전적인 피해가 꾸준히 증가하고 있다[1]. 이에 조직은 중요한 자료를 보호하기 위해 네트워크 기반의 정보보호 시스템을 구축하고, 엔드포인트 보안을 위해 백신, 매체제어 솔루션 등 다양한 보안 솔루션을 도입하여 운영하고 있다. 하지만 이러한 보안 솔루션을 운영하더라도 우회할 수 있는 다양한 기법과 취약점 등이 존재한다[2].

다양한 기법 중 hiren's bootcd와 같이 포터블 OS인 Windows PE를 USB, CD/DVD 등의 이동식 저장매체에 저장하여 부팅을 할 수 있는 기법이 있다[3]. 이는 이동식 저장매체를 이용하여 포터블 OS를 부팅하면 부팅한 타깃 PC의 저장매체를 마운트 할 수 있어 정보 추출 및 악성코드 삽입 등의 악성행위가 가능해진다. 이 기법의 특징은 기존 OS에 설치된 보안 솔루션을 우회할 수 있으며, 기본적으로 OS에서 이동식 저장매체를 인식하였을 때 생성되는 로그

※ 본 연구는 미래창조과학부 및 한국인터넷진흥원의 "2016년 고용계약형 정보보호 석사과정 지원사업"의 연구결과로 수행되었음.
† 준 회 원 : 고려대학교 정보보호대학원 금융보안학과 석사과정
†† 비 회 원 : 고려대학교 정보보호대학원 정보보호학과 박사과정
††† 종신회원 : 고려대학교 정보보호대학원 교수
Manuscript Received : September 28, 2016
Accepted : October 25, 2016
* Corresponding Author : Sang Jin Lee(sangjin@korea.ac.kr)

또는 아티팩트가 생성되지 않아 이상행위 여부확인 및 역추적에 어려움이 있다.

이동식 저장매체를 이용하여 PC를 부팅하려면 BIOS 설정(Configuration)에 진입하여 부팅순서(Boot Sequence)를 이동식 저장매체로 설정해야 한다. 과거에는 BIOS에 대한 데이터가 ROM(Read Only Memory)에 저장되어 있어 BIOS 부팅 설정에 대한 변경 이력을 확인할 수 없었지만, 최근에는 이러한 변경이력을 플래시 메모리 내에 기록하고 있으며, BIOS 펌웨어 볼륨내의 NVRAM(Non-Volatile RAM) Variables Area를 통해 확인할 수 있다.

본 논문에서는 포터블 OS를 USB와 같은 이동식 저장매체를 통하여 부팅한 흔적을 찾기 위해, 이전에는 확인하지 않았던 BIOS 설정과 관련된 데이터가 저장된 BIOS 펌웨어 이미지를 수집 및 분석하여, 이상행위 여부를 판단할 수 있는 방법을 제안하고, 기업 또는 정부기관 등에서 이상행위 탐지 및 디지털 포렌식 수사를 진행하는데 도움이 될 수 있는 방안을 제안한다.

2. 관련 연구

BIOS(Basic Input/Output System)는 OS(Operation System)와 하드웨어사이에서 입출력을 담당하기 위한 저수준의 소프트웨어와 드라이버로 이루어진 펌웨어이다. BIOS는 디스크, 비디오, USB 포트, 네트워크 장치 등과 같은 주변장치들의 설정을 읽어오거나 변경할 수 있는 프로그램이며, 읽기 및 쓰기가 가능한 플래시메모리에서 BIOS의 설정 값을 읽어 부팅을 진행한다.

미국국립표준기술연구소(NIST, National Institute of Standards and Technology)에서는 BIOS 펌웨어의 무결성 유지를 위한 지침을 발표했다. BIOS 펌웨어에 악성코드가 삽입되거나 사용자의 설정 변경으로 인해 시스템이 비정상적으로 동작하여 서비스 거부(DoS, Denial of Service) 공격이 발생할 수 있으므로 BIOS 펌웨어 보안의 필요성을 제시하였고, 관리자가 클라이언트 PC의 BIOS 설정 값이 무엇인지 판단하는 것이 중요하다고 언급하였다. 또한, BIOS 부팅 설정과 관련된 항목에서 부팅순서가 변경되었는지, 컴퓨터가 하드디스크 이외의 장치에서 부팅 되었는지에 대한 주기적인 점검이 필요하다고 권고하였다[4].

BIOS 펌웨어 보안에 대한 연구는 BIOS 펌웨어를 조작하는 공격 방법과 이를 탐지 및 방어하는 연구가 주를 이루었으며, Johannes Stüttgen[5]는 펌웨어 조작 기술의 다양한 예시와 메모리를 조사하는 과정에서 펌웨어 레벨의 위협을 식별하기 위한 방법을 제안하였다.

또한 BIOS설정 값의 변경을 막기 위해 BIOS 진입 시 비밀번호를 설정할 수 있다. 하지만 CMOS 배터리의 제거, 마더보드 점퍼 조작, 소프트웨어 등을 통해 비밀번호를 초기화 할 수 있으며, 일부 제조사에서는 별도의 방법을 제공하고 있어 BIOS의 비밀번호를 쉽게 무력화 시킬 수 있다[6].

이와 같이 기존에는 BIOS 펌웨어 이미지의 변조를 식별하고 탐지하는 방법이 주로 연구되었지만, 디지털 포렌식 관점에서 BIOS 펌웨어 이미지에 기록되는 데이터의 의미를 분석하고 활용하는 내용은 전무하다.

3. BIOS 펌웨어 볼륨 이미지

BIOS 펌웨어 볼륨 이미지가 저장된 플래시 메모리는 가장 일반적인 펌웨어 볼륨을 위한 비휘발성 장치이다. UEFI (Unified Extensible Firmware Interface) 스펙에서는 데이터 또는 코드를 저장하기 위한 기본 스토리지를 펌웨어 볼륨이라고 정의하고 있다[7]. 펌웨어 볼륨 이미지의 구조는 제조사마다 조금씩 상이할 수 있으며, 본 논문에서는 전 세계적으로 65% 이상 보급되어 있는 AMI(American Megatrends, Inc.)의 BIOS를 대상으로 구조를 기술하였다[8].

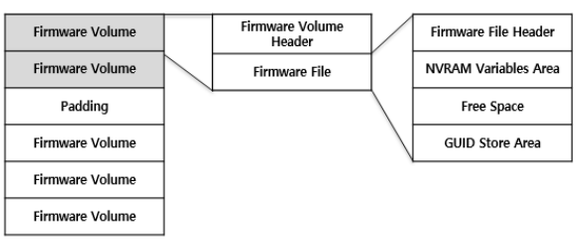


Fig. 1. Firmware Volume Image

펌웨어 볼륨 이미지를 추출하면 Fig 1.과 같이 여러 개의 볼륨으로 구성되어 있으며, 첫 번째, 두 번째 펌웨어 볼륨에는 펌웨어 볼륨헤더와 펌웨어 파일이 존재하고, 펌웨어 파일에는 펌웨어 파일 헤더와 NVRAM Variables Area, Free Space, GUID Store Area로 구성되어 있다. 나머지 펌웨어 볼륨에는 펌웨어를 실행시키기 위한 코드와 데이터로 구성되어 있다.

3.1 Firmware Volume

각 펌웨어 볼륨은 펌웨어 파일 시스템으로 구성되어 있으며, 16바이트 크기인 고유의 GUID(Globally Unique Identifier)로 펌웨어 볼륨 이미지에서 각 볼륨을 구분한다. GUID에서 상위 8바이트는 리틀엔디안(Little Endian)방식, 하위 8바이트는 빅엔디안(Big Endian)방식으로 표기된다. 펌웨어 볼륨을 식별하기 위한 GUID는 “8C8CE578-8A3D-4F1C-9935-896185C32DD3”이다[7].

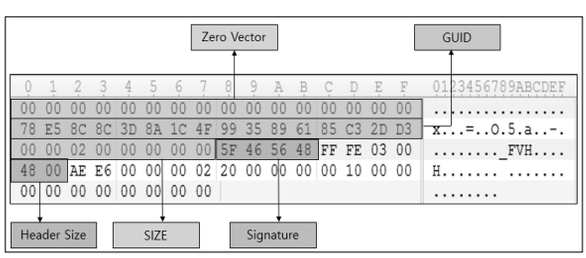


Fig. 2. Firmware Volume Header

Fig. 2는 펌웨어 볼륨 헤더의 구조를 나타내며, 첫 16바이트는 제로벡터로 되어 있고, 펌웨어 볼륨을 식별하는 GUID, 볼륨의 크기, 펌웨어 볼륨의 시그니처인 "_FVH" 문자열, 펌웨어 볼륨의 헤더 사이즈 등을 확인할 수 있다.

3.2 Firmware File

펌웨어 파일은 펌웨어 볼륨에 저장된 데이터와 코드를 말한다. 펌웨어 파일은 헤더와 데이터를 포함하며, GUID "CEF5B9A3-476D-497F-9FDC-E98143E0422C"로 식별한다.

Fig. 3은 펌웨어 파일의 헤더영역이며, GUID, 체크섬(Checksum), 타입(Type), 속성(Attributes), 크기(Size), 상태(State) 속성을 가지며, 헤더 뒤에 데이터가 기록된다.

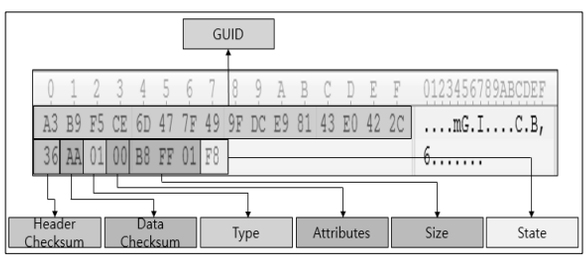


Fig. 3. Firmware File Header

3.3 NVRAM Variables Area

Fig. 4는 NVRAM Variables Area이며 변경되는 BIOS 설정 정보를 기록하는 영역이다. NVRAM Variables Area는 펌웨어 파일 뒤에 데이터가 순차적으로 기록되며, 헥사값 "0x5241564e"인 문자열 "NVAR"이라는 시그니처로 시작되는 특징이 있다. 데이터 영역에는 부팅과 관련된 설정정보가 유니코드 문자열로 기록된다.

NVRAM Variables Area에 데이터를 기록하는 방식은 첫 번째 펌웨어 볼륨의 NVRAM Variables Area에 BIOS 설정에 대한 히스토리데이터가 모두 기록되어 공간이 다 차면, 두 번째 펌웨어의 NVRAM Variables Area에 기록된다. 만약 두 개의 펌웨어 볼륨의 NVRAM Variables Area에 데이터가 모두 기록되면, 기존에 기록되어 있는 데이터에 덮어씌워지면서 기록을 하게 된다.

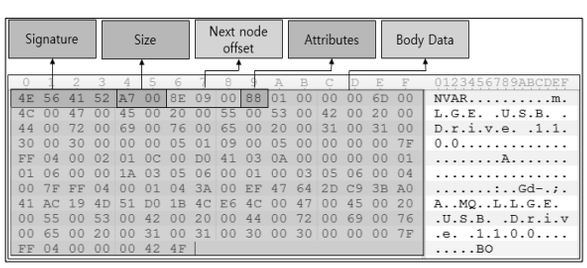


Fig. 4. NVRAM Variable Area

3.4 Free Space

Fig. 5와 같이 펌웨어 볼륨 이미지에는 0xFF로 패딩된 Free Space영역이 존재한다. Free Space영역은 펌웨어 볼륨 이미지 내에서 각 펌웨어 볼륨 사이에 존재하며, 각 펌웨어 볼륨 내에도 존재한다. BIOS 설정과 관련된 데이터는 펌웨어 볼륨 내의 미할당 영역에 순차적으로 기록된다. 즉, 펌웨어 볼륨 내에 있는 미할당 영역은 BIOS 부팅 설정의 데이터가 기록될수록 공간이 줄어들게 된다.

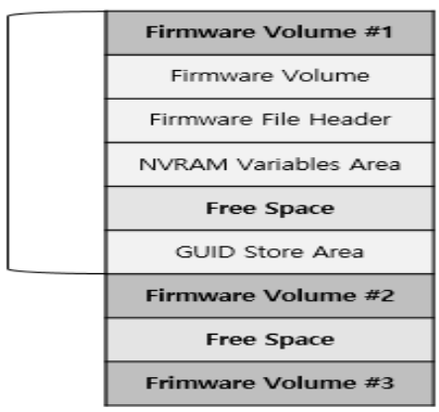


Fig. 5. Free Space in Firmware Volume Image

3.5 GUID Store Area

GUID는 128비트의 16진수로, 각 펌웨어 볼륨과 펌웨어 파일을 식별하고, NVRAM Variable Area에서 초기에 설정된 부팅과 관련된 각 변수를 식별하기 위해 사용하며, GUID Store Area는 각 변수를 식별하기 위해 사용된 GUID를 저장한 영역으로, NVRAM Variable Area가 포함된 펌웨어 볼륨의 제일 마지막 부분에 존재한다.

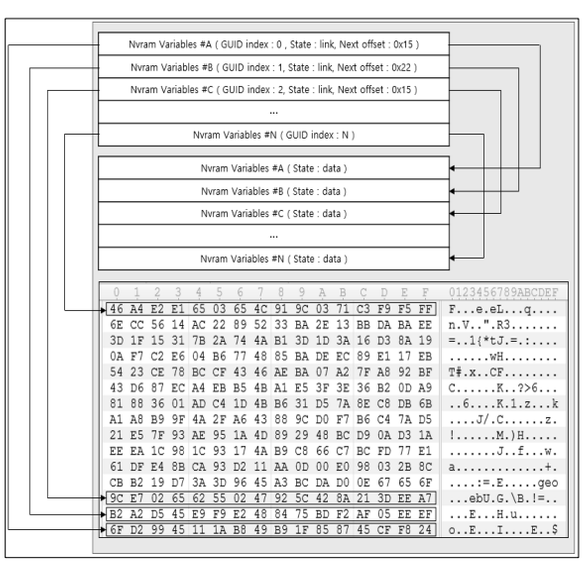


Fig. 6. GUID Store Area

Fig. 6은 NVRAM 파일에 존재하는 GUID Store Area를 나타낸다. 초기에 NVRAM 변수에 해당하는 GUID를 펌웨어 볼륨 아래에 기록하고, 아래서부터 순차적으로 기록된다. 예를 들면 NVRAM Variable #B의 GUID index는 1이고, 이

에 해당하는 GUID는 GUID Store Area에서 맨 아래에서 2번째 값이다. 또한 상태(State)값이 링크(link)인데 이는 #B에 해당하는 변수에 새로운 값이 기록되었을 경우 상태 값이 데이터에서 링크로 변경되면서 다음 오프셋이 기록된다.

다음 오프셋의 의미는 현재 NVRAM Variables #B의 오프셋에서 다음 오프셋만큼 더한 위치에 새로운 값이 기록되어 있다는 것을 의미한다. 그래서 NVRAM Variables #B의 현재 오프셋과 다음 오프셋을 더하면, NVRAM Variables #B의 새로운 값이 갱신된 곳의 위치를 계산할 수 있다. GUID index 값은 최초 기록될 때만 속성 값을 확인할 수 있으며, 새로운 데이터는 다음 오프셋을 더하면 그곳의 위치 값을 확인할 수 있다.

4. BIOS 설정 값 변경 실험

본 논문에서는 H/W 제조사인 SAMSUNG, MSI, DELL, HP를 대상으로 실험을 진행하였고, BIOS 제조사인 AMI (American Megatrends Inc.), Dell, HP의 BIOS 펌웨어 이미지를 추출하여 실험하였으며 자세한 실험대상의 상세정보는 Table 1과 같다.

AMI의 BIOS 펌웨어 이미지는 AMI에서 제공하는 'Afuwin' 도구를 이용하여 추출하였고[9], DELL, HP의 BIOS 펌웨어 이미지는 'SLIC_ToolKit' 도구로 펌웨어 영역의 오프셋을 확인한 후, 해당 영역의 오프셋을 Read하여 펌웨어를 추출하였다[10].

데이터를 분석하기 위해 실험방법은 BIOS 설정에서 부팅 순서를 변경해 가면서 NVRAM Variable값이 어떻게 기록되고, 이동식 저장매체를 이용하여 OS를 부팅하였을 경우 어떤 기록이 남는지 실험을 통해 확인하였다.

Fig. 7은 정상적인 부팅과 이상행위로 간주할 수 있는 비정상적인 부팅과정을 보여준다. 기술정보, 업무기밀 또는 국가기밀을 취급하는 기업, 국가기관 등에서는 업무용 PC에서 BIOS 설정에 접근하는 것만으로도 이상행위로 간주할 수 있으며, 부팅이 가능한 이동식 저장매체를 이용하여 OS를 부팅하게 된다면 자료유출, 데이터 변조 및 악성코드삽입 등의 영향을 받을 수 있다.

본 실험에서는 비교 데이터 셋을 만들기 위해 정상적인 부팅을 한 후 BIOS 펌웨어 이미지를 추출하였다. 이후 BIOS 설정에서 부팅 순서를 변경하여 USB로 부팅을 한 후, 다시 BIOS 펌웨어 이미지를 추출하는 과정을 반복적으로 진행하였고, 추출한 BIOS 펌웨어 이미지는 바이트 단위로 비교분석하였다.

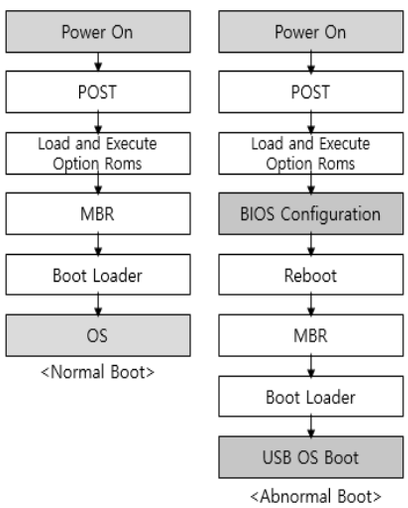


Fig. 7. Normal vs Abnormal Boot Process

5. 실험 결과

5.1 USB 저장매체를 통한 부팅 흔적

실험에 사용한 USB는 SANDSIK와 LG제조사의 USB를 사용하였고, 레지스트리는 "HKLM\SYSTEM\ControlSet00#\Enum\USBSTOR"에서 확인한 정보이며, 상제정보는 다음 Table 2와 같다.

AMI BIOS는 NVRAM Variable Area에서 "0x5241564e" 시그니처를 시작으로 데이터가 기록되며 Fig 8, 9는 SAMSUNG 데스크톱에서 USB를 이용하여 OS를 부팅하였을 때, Fig. 10, 11은 MSI 노트북에서 USB를 이용하여 OS를 부팅하였을 때, NVRAM Variable Area에 해당 USB의 벤더명(Vendor Name), 제품명(Product Name), 버전(Version)이 기록되는 것을 확인하였다. DELL BIOS는 "0xAA55" 시그니처를 시작으로 데이터가 기록되며, DELL 서버에서는 Fig. 12와 같이 USB를 이용하였을 때 제품명(Product Name)만 기록되는 것을 확인하였다.

Table 1. Test Environment Information

BIOS Vendor	H/W	H/W Type	OS Name	BIOS Version	UEFI Firmware	Firmware Address
AMI	SAMSUNG	Desktop	Windows 8.1	POOLES.075.140905.XJ	O	0xFFD00000 to 0xFFFFFFFF
AMI	MSI	Notebook	Windows 7	E16GAIMK.10A	O	0xFFD00000 to 0xFFFFFFFF
DELL	Dell Inc.	Server	Windows Server 2012 R2	Dell Inc. 2.5.4	X	0xFFA60000 to 0xFFFFFFFF
HP	HP	Workstation	Windows 7	M60 v01.62	O	0xFF600000 to 0xFFFFFFFF

Table 2. USB Information

Vendor Name	SANDISK	LGE
Product Name	CRUZER_BLADE	USB_DRIVE
Version	1.26	1100
Serial Number	4C530310030211109391	2012011800000055

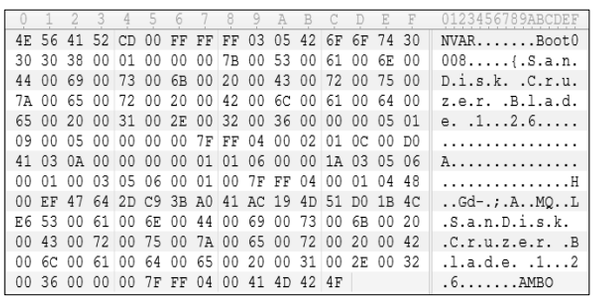


Fig. 8. Samsung Desktop SANDISK USB Log

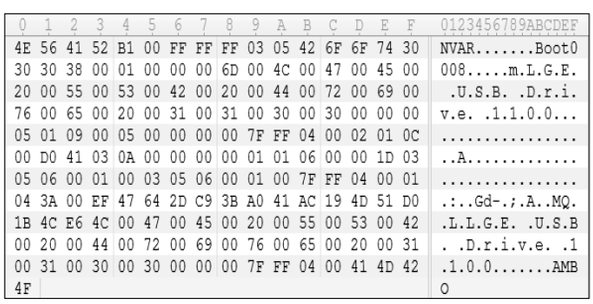


Fig. 9. Samsung Desktop LGE USB Log

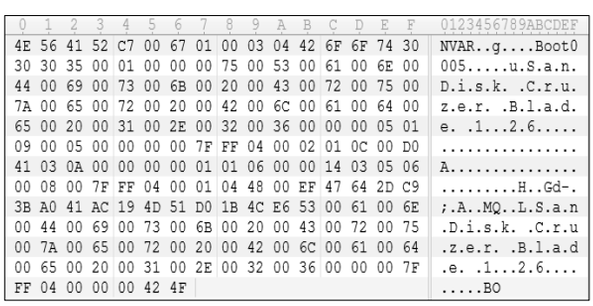


Fig. 10. MSI Laptop SANDISK USB Log

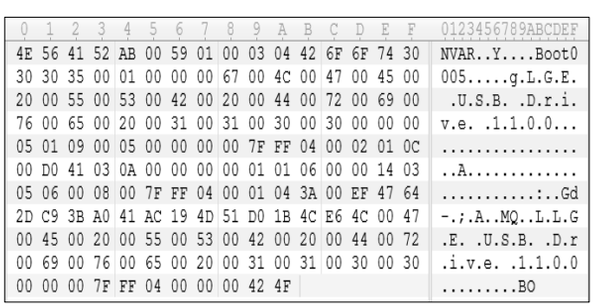


Fig. 11. MSI Laptop LGE USB Log

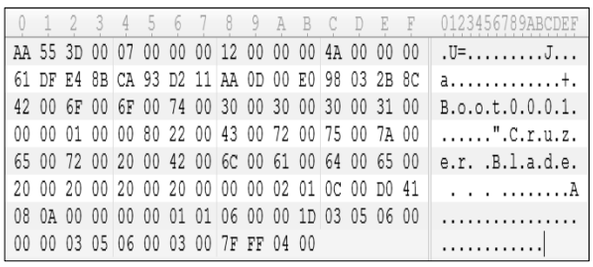


Fig. 12. DELL Server SANDISK USB Log

HP BIOS는 “0xAA55” 시그니처를 시작으로 데이터가 기록되며, AMI와 DELL BIOS와는 다르게 HP BIOS는 Fig. 13, 14와 같이 USB의 시리얼 넘버(Serial Number)가 기록되는 것을 확인하였다. 시리얼 넘버는 같은 벤더사의 제품이라도 USB 마다 갖는 고유한 식별 값이며 이를 통해 사용된 USB를 보다 정확하게 식별할 수 있다.

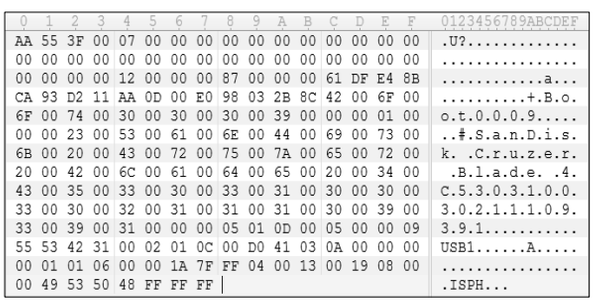


Fig. 13. HP Workstation SANDISK USB Log

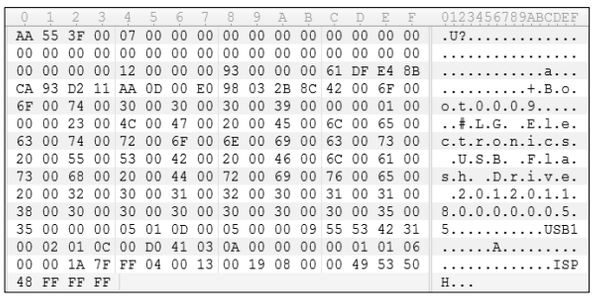


Fig. 14. HP Workstation LGE USB Log

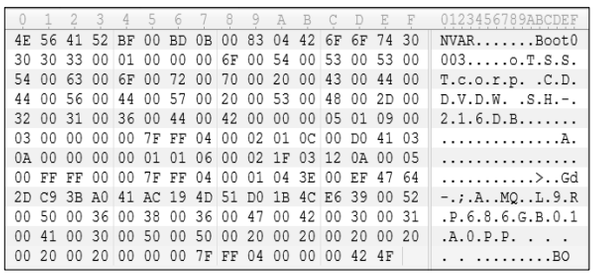
5.2 CD/DVD 저장매체를 통한 부팅 흔적

이번 실험은 CD 저장매체에 Windows PE를 저장하여 부팅 가능한 CD를 생성한 후, 외장형 CD/DVD ROM과 내장형 CD/DVD ROM으로 부팅 순서를 설정하여 테스트를 진행하였다. 실험에 사용한 CD/DVD ROM의 정보는 Table 3과 같다.

Table 3. CD/DVD Device Information

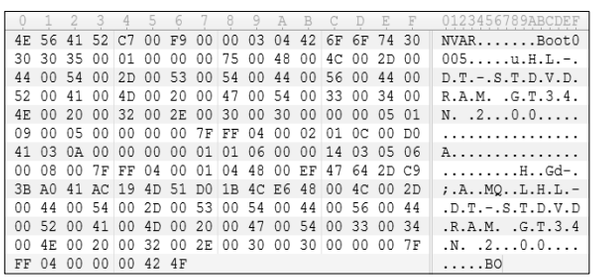
Type	Product Name
Internal	TSSTcorp CDDVDW SH-216DB
External	HL-DT-ST DVDRAM GT34N USB Device

Fig. 15, 16은 SAMSUNG 데스크톱의 내·외장형 CD/DVD ROM을 이용하여 부팅하였을 경우에 기록되는 데이터 이며, CD/DVD를 이용하여 부팅한 기록도 확인하였다.



0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
4E	56	41	52	BF	00	BD	0B	00	83	04	42	6F	6F	74	30	NVAR.....Boot0
30	30	33	00	01	00	00	00	6F	00	54	00	53	00	53	00	003.....o.T.S.S.
54	00	63	00	6F	00	72	00	70	00	20	00	43	00	44	00	T.c.o.r.p. .C.D.
44	00	56	00	44	00	57	00	20	00	53	00	48	00	2D	00	D.V.D.W. .S.H.-.
32	00	31	00	36	00	44	00	42	00	00	00	05	01	09	00	2.1.6.D.B.....
03	00	00	00	00	7F	FF	04	00	02	01	0C	00	D0	41	03	.............A.
0A	00	00	00	00	01	01	06	00	02	1F	03	12	0A	00	05	....
00	FF	FF	00	00	7F	FF	04	00	01	04	3E	00	EF	47	64	..........>..Gd
2D	C9	3B	A0	41	AC	19	4D	51	D0	1B	4C	E6	39	00	52	-.;.A..MQ..L.9.R
00	50	00	36	00	38	00	36	00	47	00	42	00	30	00	31	.P.6.8.6.G.B.0.1
00	41	00	30	00	50	00	50	00	20	00	20	00	20	00	20	.A.0.P.P. . . .
00	20	00	20	00	00	00	7F	FF	04	00	00	00	42	4F		. . ..BO

Fig. 15. Internal CD/DVD ROM



0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
4E	56	41	52	C7	00	F9	00	00	03	04	42	6F	6F	74	30	NVAR.....Boot0
30	30	35	00	01	00	00	00	75	00	48	00	4C	00	2D	00	005.....u.H.L.-.
44	00	54	00	2D	00	53	00	54	00	44	00	56	00	44	00	D.T.-.S.T.D.V.D.
52	00	41	00	4D	00	20	00	47	00	54	00	33	00	34	00	R.A.M. .G.T.3.4.
4E	00	20	00	32	00	2E	00	30	00	30	00	00	00	05	01	N. .2...0.0.....
09	00	05	00	00	00	00	7F	FF	04	00	02	01	0C	00	D0	....
41	03	0A	00	00	00	00	01	01	06	00	00	14	03	05	06	A...............
00	08	00	7F	FF	04	00	01	04	48	00	EF	47	64	2D	C9	.........H..Gd-.
3B	A0	41	AC	19	4D	51	D0	1B	4C	E6	48	00	4C	00	2D	;.A..MQ..L.H.L.-
00	44	00	54	00	2D	00	53	00	54	00	44	00	56	00	44	.D.T.-.S.T.D.V.D
00	52	00	41	00	4D	00	20	00	47	00	54	00	33	00	34	.R.A.M. .G.T.3.4
00	4E	00	20	00	32	00	2E	00	30	00	30	00	00	00	7F	.N. .2...0.0....
FF	04	00	00	00	42	4F										.....BO

Fig. 16. External CD/DVD ROM

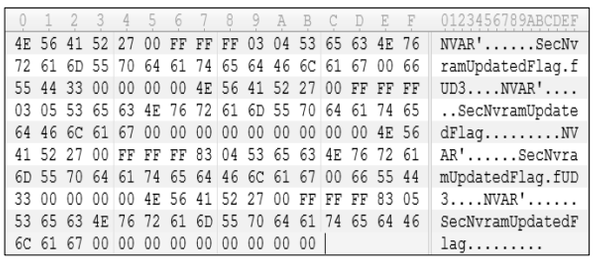
5.3 BIOS 펌웨어 이미지 추출도구의 무결성 여부

AMI에서 제공하는 'afuwin' 도구와 'SLIC_ToolKit' 도구를 이용하여 BIOS 펌웨어 이미지를 추출하였을 때 무결성이 유지되는지를 확인하였다. 실험은 도구를 이용하여 각 제조사 및 BIOS 별로 연속적으로 BIOS 펌웨어 이미지를 추출하여 Hash값을 비교하였다.

Table 4. Results of Integrity Test of Tools

Type	afuwin	SLIC_ToolKit
SAMSUNG (AMI)	X	O
MSI (AMI)	O	O
HP (HP)	Unavailable Extraction	O
DELL (DELL)	Unavailable Extraction	O

실험결과 Table 4와 같이 'SLIC_ToolKit' 도구는 4개 제조사의 BIOS 펌웨어를 추출할 수 있으며, 무결성이 유지되는 것을 확인하였고 'afuwin' 도구는 AMI BIOS만 추출할 수 있으며, SAMSUNG 데스크톱의 BIOS는 무결성이 유지되지 않는 것을 확인하였다. 그 이유는 SAMSUNG 데스크톱의 BIOS 펌웨어 이미지를 추출할 때마다 Fig. 17과 같이 156바이트의 일정한 데이터가 추가적으로 기록되는 것을 확인하였다.



0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
4E	56	41	52	27	00	FF	FF	FF	03	04	53	65	63	4E	76	NVAR'......SecNv
72	61	6D	55	70	64	61	74	65	64	46	6C	61	67	00	66	ramUpdatedFlag.f
55	44	33	00	00	00	00	4E	56	41	52	27	00	FF	FF	FF	UD3....NVAR'....
03	05	53	65	63	4E	76	72	61	6D	55	70	64	61	74	65	..SecNvramUpdate
64	46	6C	61	67	00	00	00	00	00	00	00	00	00	4E	56	dFlag.........NV
41	52	27	00	FF	FF	FF	83	04	53	65	63	4E	76	72	61	AR'......SecNvra
6D	55	70	64	61	74	65	64	46	6C	61	67	00	66	55	44	mUpdatedFlag.fUD
33	00	00	00	00	4E	56	41	52	27	00	FF	FF	FF	83	05	3....NVAR'......
53	65	63	4E	76	72	61	6D	55	70	64	61	74	65	64	46	SecNvramUpdatedF
6C	61	67	00	00	00	00	00	00	00	00	00					lag.........

Fig. 17. BIOS Firmware Image Data Generated During The Extraction

SAMSUNG데스크톱의 BIOS 펌웨어 파일은 추출 시 마다 파일의 무결성이 유지되지는 않지만, 고정적으로 추가되는 문자열 이므로 법적 증거로 제출되는 것이 아닌 기업의 감사 및 책임추적의 목적이라면 충분이 활용이 가능하다.

MSI 노트북에서 연속적으로 5회 추출하였을 때 Table 5와 같이 MD5 해시값이 모두 동일하며 무결성이 유지되는 것을 확인하였다.

Table 5. MD5 Hash Value

Count	MD5
1	73073FF432405A5D7F1DA3907BD1A7EC
2	73073FF432405A5D7F1DA3907BD1A7EC
3	73073FF432405A5D7F1DA3907BD1A7EC
4	73073FF432405A5D7F1DA3907BD1A7EC
5	73073FF432405A5D7F1DA3907BD1A7EC

실험결과를 정리하면 Table 6과 같이 AMI, DELL, HP BIOS 제조사 별로 제품명이 모두 기록되는 것을 확인하였고, 시리얼 넘버는 HP의 경우에만 기록되는 것을 확인하였다. 그리고 NVRAM Area에서 데이터가 기록될 때 사용되는 시그니처는 DELL과 HP가 동일한 것을 확인하였다.

Table 6. Result of Test

BIOS	AMI	DELL	HP
Data	O	O	O
Signature	0x5241564e	0xAA55	0xAA55
Vendor Name	O	X	O
Product Name	O	O	O
Version	O	X	X
Serial Number	X	X	O

6. NVRAM 데이터 활용방안

NIST 800-155 표준을 보면 BIOS 설정에 진입 또는 변경하는 것만으로도 이상행위로 판단하고 있다. 국가기관 또는 기업 등에서 각 사용자 PC의 BIOS 설정 데이터를 주기적으로 추출하고 분석하여 해당 사용자가 이상행위를 하였는지 확인할 수 있는 방법을 제안한다.

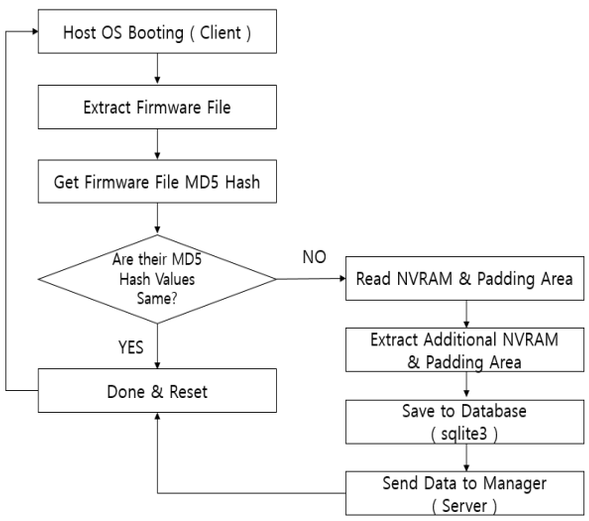


Fig. 18. Detection Method of Abnormal Activity

Fig. 18은 BIOS 펌웨어 이미지를 이용해서 이상행위를 판단할 수 있는 프로세스를 표현한 것이다. 최초 호스트 PC가 부팅되면 BIOS 펌웨어 이미지를 추출하고, 이 파일에 대한 MD5 해시값을 계산한다. 이 후에 BIOS 펌웨어 이미지에서 NVRAM Variables Area 및 패딩 영역를 추출하여 저장한다. 이후 다음 부팅 시에 BIOS펌웨어 이미지 추출 및 MD5 해시값을 계산한 후, 이전에 저장한 값과 비교하여 같으면 프로그램을 종료하고, 다르면 추가된 NVRAM Variables Area과 패딩 영역이 있는지 조사하여 추가된 데이터를 데이터베이스에 저장하고 이 데이터를 관리자에게 바로 전송하여 경고(Alert)를 보여준다. 이러한 방법을 통하여 거의 실시간에 가깝게 이상행위 사용자를 식별하는데 있어 정탐률을 올릴 수 있으며, 현직 직원 및 퇴사자, 외주직원 등의 호스트 PC를 조사할 때 이상행위에 대한 책임을 추궁할 수 있는 근거를 마련할 수 있다.

7. 도구 개발

본 논문에서는 포터블 OS를 이용하여 자료유출 및 변조하기 위해 선행되어야 되는 BIOS 설정변경의 흔적을 모니터링하고 BIOS 펌웨어 이미지에 악성코드를 삽입하는 공격 등의 이상행위를 탐지하기 위해 도구를 개발하였다.

Table 7. Execution Result Table of Tool

Name	Value
Host Name	jason-PC
HOST_IP	192.168.3.138
HOST_MAC	00:0c:29:1a:46:87
OS_Version	Windows-7-6.1.7600-SP0
File_Name	bios_firmware.rom
Firmware_ID	1APTJ015
BIOS_Version	E16GAIMK.10A
Creation_Time	2016-07-23 15:34:54
MD5_Hash	e59bbdc1be7233bda8f7c12e3d258ed4
NVAR_START_INDEX	30536
NVAR_SIZE	30598
NVAR_DATA	LGE USB Drive 1100A
Padding_Data	None

주요 기능으로는 호스트 OS가 부팅을 할 때마다 추출된 BIOS 펌웨어에서 변경된 데이터가 있다면, 이를 탐지하여 추가된 데이터를 데이터베이스에 저장 및 관리자에게 전달해주는 기능이다. 비교가 되는 데이터 영역은 NVRAM Variable Area와 Padding Area이며, NVRAM Variable Area의 데이터를 통해 이상행위를 하였는지에 대한 여부를 탐지할 수 있고, Padding Area의 데이터를 통해 악성코드 삽입 여부를 Table 7과 같이 확인할 수 있다.

도구의 결과로 나온 Fig. 19와 같이 Host Name, Host_IP, Host_MAC, OS_Version 정보를 통해 해당 사용자의 호스트임을 식별하고, Firmware_ID, BIOS_Version 정보를 통해 펌웨어 정보를 확인할 수 있다. 또한 BIOS 펌웨어 이미지에서 기록되지 않는 시간 값을 보완하기 위해, BIOS 펌웨어 이미지 파일의 생성시간을 기록하여 이상행위를 한 시점을 판별할 수 있도록 보완하였고, NVRAM Variables Area 및 Padding Area를 분석하여 이상행위에 대한 흔적을 확인할 수 있다.

HOST_NAME	HOST_IP	HOST_MAC	OS_VERSION	FILE_NAME	FIRMWARE_ID	BIOS_VERSION	CREATION_TIME	MD5_HASH	NVAR_START_INDEX	NVAR_SIZE	NVAR_DATA
Click here to define a filter											
jason-PC	192.168.3.142	00:0c:29:1a:46:87	Windows-7-6.1.7600-SP0	reboot_2.rom	1APTJ015	E16GAIMK.10A	2016-07-23 15:34:54	8b2bfb4ffa35afe7be7499e909206138	75300	75362	NVAR HobRomImage
jason-PC	192.168.3.142	00:0c:29:1a:46:87	Windows-7-6.1.7600-SP0	reboot_2.rom	1APTJ015	E16GAIMK.10A	2016-07-23 15:34:54	8b2bfb4ffa35afe7be7499e909206138	75362	75398	NVAR 9 J
jason-PC	192.168.3.142	00:0c:29:1a:46:87	Windows-7-6.1.7600-SP0	reboot_2.rom	1APTJ015	E16GAIMK.10A	2016-07-23 15:34:54	8b2bfb4ffa35afe7be7499e909206138	75398	75498	NVAR2
jason-PC	192.168.3.142	00:0c:29:1a:46:87	Windows-7-6.1.7600-SP0	reboot_2.rom	1APTJ015	E16GAIMK.10A	2016-07-23 15:34:54	8b2bfb4ffa35afe7be7499e909206138	75498	75598	NVAR2
jason-PC	192.168.3.142	00:0c:29:1a:46:87	Windows-7-6.1.7600-SP0	reboot_2.rom	1APTJ015	E16GAIMK.10A	2016-07-23 15:34:54	8b2bfb4ffa35afe7be7499e909206138	75598	75698	NVAR2
jason-PC	192.168.3.142	00:0c:29:1a:46:87	Windows-7-6.1.7600-SP0	reboot_2.rom	1APTJ015	E16GAIMK.10A	2016-07-23 15:34:54	8b2bfb4ffa35afe7be7499e909206138	75698	75738	NVAR
jason-PC	192.168.3.142	00:0c:29:1a:46:87	Windows-7-6.1.7600-SP0	reboot_2.rom	1APTJ015	E16GAIMK.10A	2016-07-23 15:34:54	8b2bfb4ffa35afe7be7499e909206138	76092	76446	NVAR Boot0008 mLGE USB Drive 1100 A Gd A MQ LLGE USB Drive 11 00 AMBO

Fig. 19. Tool for Detection of Abnormal Activity

8. 결론 및 향후 연구

부팅이 가능한 이동식 저장매체를 이용하여 포터블 OS를 부팅하면, 해당 PC에 적용된 매체제어 솔루션과 같은 보안소프트웨어를 우회할 수 있고, 로그 또는 아티팩트가 기록되지 않아 내부 저장매체의 자료유출 및 악성코드 삽입 등의 이상행위 여부 확인 및 흔적을 찾는데 어려움이 존재했다. 하지만 BIOS 펌웨어 이미지를 수집하고 NVRAM Variables Area의 분석을 통해 이상행위로 간주될 수 있는 BIOS 부팅 순서 설정 변경 흔적을 확인할 수 있다. 또한 BIOS 진입 패스워드를 우회할 수 있는 방법이 많기 때문에, 이러한 데이터를 추출 및 분석하고 주기적인 모니터링을 통해 이상행위 여부 판별이 필요하다.

NVRAM Variable Area의 데이터를 이용하여 디지털포렌식 수사에서의 선별 수집 시 추가적인 데이터로 활용될 수 있으며, 기업에서의 이상행위를 판별하는데 도움이 될 수 있다.

향후 본 논문에서 실험한 제조사 이외에 다양한 제조사의 BIOS 펌웨어 이미지 파일 수집 및 분석과 펌웨어 이미지 파일 업데이트와 같은 안티포렌식 행위분석에 대한 추가적인 연구가 필요하다.

References

[1] NIS [Internet], http://www.nis.go.kr/AF/1_5_1_1.do.

[2] Sun-Mi Kim, SoonOh Hong, and Kang Seok Lee, “The secure operation stratedgy of Data Leakage Prevention System,” *Proceedings of Symposium of the Korean Institute of Communications and Information Sciences*, Korea Institute of Communication Sciences, pp.1639-1640, 2009.

[3] Hirensbootcd [Internet], http://www.hirensbootcd.org/.

[4] Andrew Regenscheid, “BIOS Integrity Measurement Guidelines,” National Institute of Standards and Technology, sp800-155, 2011.

[5] J. Stüttgen, “Acquisition and analysis of compromised firmware using memory forensics,” *Digital Investigation*, Vol.12, Sup.1, pp.S50-S60, 2015.

[6] Dell [Internet], http://www.dell.com/support/article/us/en/19/SLN284985.

[7] UEFI, “VOLUME 3: Platform Initialization Shared Architectureal Elements,” Version 1.4, 2015.

[8] Stortrends [Internet], http://stortrends.com/st_downloads/StorTrends_Customer_Value_Proposition.pdf.

[9] AMI, Afuwin [Internet], https://ami.com/ami_downloads/Apti o_4_AMI_Firmware_Update_Utility.zip.

[10] BIOS, SLIC_ToolKit [Internet], http://www.bios.net.cn/e/enews?enews=DownSoft&classid=60&id=448&pathid=0&pass=44d02f9d28e3295b1eed8911519a23d9&p=:::.

정 승 훈

e-mail : dabster@korea.ac.kr

2013년 한남대학교 컴퓨터공학과(학사)

2015년~현 재 고려대학교 정보보호대학원 금융보안학과 석사과정

관심분야 : Digital Forensic, Information Security

이 윤 호

e-mail : yuno21@korea.ac.kr

2001년 부경대학교 전자계산학과(학사)

2011년~현 재 고려대학교 정보보호대학원 정보보호학과 박사과정

관심분야 : Digital Forensic, Mobile Forensic, Information Security

이 상 진

e-mail : sangjin@korea.ac.kr

1987년 고려대학교 수학과(학사)

1989년 고려대학교 수학과(석사)

1994년 고려대학교 수학과(박사)

1989년~1999년 ETRI 선임연구원

1999년~현 재 고려대학교 정보보호대학원 교수

2008년~현 재 고려대학교 디지털포렌식연구센터 센터장

관심분야 : Digital Forensic, Steganography, Hash Function