

[304 – Find a confidential file]

Write-Up

작성자	박혜미
분석 일자	2024.05.22
작성 일자	2024.05.23
분석 대상	Atom_disk_0.dd Atom_USB.dd
문서 버전	1.0
작성자 E-mail	parkm0708@naver.com

0. 목차

1. 문제3

2. 분석 도구3

3. 환경3


4. Write-Up.....4

5. Flag.....9

6. 별도 첨부 10

7. Reference 11

1. 문제

URL	
문제 내용	<p>The person, who is suspected of leaking a company's core technology, has a high level of computer knowledge, so he has safely stored the company's secret technical information in his PC. An investigator secured 1 hard disk and 1 USB storage of the suspect, but the USB was already formatted. There are some traces of browsing the company's confidential file, 'Small Modular Reactor.png', on the disk, but the original file does not exist. Find the confidential file hided by the suspect.</p> <p>Questions Note, the suspect used Windows 10. You should analyze the following questions based on UTC+9 time zone.</p> <ol style="list-style-type: none"> What is the exact capacity of the encrypted volume? (40 points) When was the volume (#1's encrypted volume) encrypted? (40 points) Calculate SHA1 hash value of the confidential file (Small Modular Reactor.png). (220 points) <p>(한글 번역은 [6. 별도 첨부]에 기재)</p>
문제 파일	 Atom_disk_0.dd-A ND-Atom_USB.dd.
문제 유형	other forensics
난이도	3 / 3

2. 분석 도구

도구명	다운로드 링크	Version
FTK Imager	https://www.exterro.com/digital-forensics-software/ftk-imager	4.7.1.2
HxD	https://mh-nexus.de/en/hxd/	2.5
Autopsy	https://www.autopsy.com/download/	4.21.0
DCode	https://www.digital-detective.net/dcode/	5.5
PowerShell	-	1.0

3. 환경

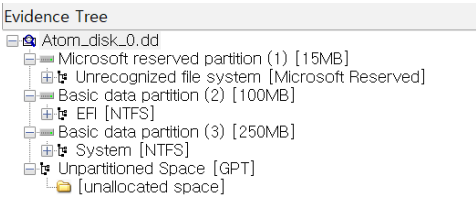
OS
Windows 11 Home

4. Write-Up

파일명	Atom_disk_0.dd
용량	2GB
SHA256	8d2e1eced08706c986adf1b26a77143ab0dfdfcdfab0e3f40a15a702292f08a6
Timestamp	2021-06-30 17:12:58

파일명	Atom_USB.dd
용량	131MB
SHA256	fdce51feeeef070dfc67e606634c63080e3acd1089518d68e057f0995fcc8602
Timestamp	2021-06-30 17:11:00

1. 암호화된 볼륨의 크기



[그림 1] FTK Imager로 열린 Atom_disk_0.dd 파일

Atom_disk_0.dd 파일을 FTK Imager 를 사용해 열어보았더니 파티션이 3 개가 존재하는 것을 알 수 있다. 해당 파티션을 다 합치면 375MB 인데 해당 파일의 용량은 2GB 이다. 따라서 문제 파일에 숨겨진 것이 존재한다고 볼 수 있다.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	33	C0	8E	D0	BC	00	7C	8E	C0	8E	D8	BE	00	7C	BF	00	3A2D4..jZAZ0%.jZ.
00000010	06	B9	00	02	FC	F3	A4	50	68	1C	06	CB	FB	B9	04	00	..G0Mh...E0+..
00000020	BD	BE	07	80	7E	00	00	7C	0B	0F	85	0E	01	83	C5	10	4h...fVv.h.h.
00000030	E2	F1	CD	18	88	56	00	55	C6	46	11	05	C6	46	10	00	4h...fVv.h.h.
00000040	B4	41	BB	AA	55	CD	13	5D	72	0F	81	FB	55	AA	75	09	4h...fVv.h.h.
00000050	F7	C1	01	00	74	03	FE	46	10	66	60	80	7E	10	00	74	4h...fVv.h.h.
00000060	26	66	68	00	00	00	00	66	FF	76	08	68	00	00	68	00	4h...fVv.h.h.
00000070	7C	68	01	00	68	10	00	B4	42	8A	56	00	8B	F4	CD	13	4h...fVv.h.h.
00000080	9F	83	C4	10	9E	EB	14	B9	01	02	BB	00	7C	8A	56	00	4h...fVv.h.h.
00000090	8A	76	01	8A	4E	02	8A	6E	03	CD	13	66	61	73	1C	FE	4h...fVv.h.h.
000000A0	4E	11	75	0C	80	7E	00	80	0F	84	8A	00	B2	80	EB	84	4h...fVv.h.h.
000000B0	55	32	E4	8A	56	00	CD	13	5D	EB	9E	81	3E	FE	7D	55	4h...fVv.h.h.
000000C0	AA	75	6E	FF	76	00	E8	8D	00	75	17	FA	B0	D1	E6	64	4h...fVv.h.h.
000000D0	E8	83	00	B0	DF	E6	60	E8	7C	00	B0	FF	E6	64	E8	75	4h...fVv.h.h.
000000E0	00	FB	B9	00	BB	CD	1A	66	23	00	75	3B	66	81	FB	54	4h...fVv.h.h.
000000F0	43	50	41	75	32	81	F9	02	01	72	2C	66	68	07	BB	00	4h...fVv.h.h.

[그림 2] HxD로 열린 Atom_disk_0.dd

해당 파일의 숨겨진 것을 찾기 위해 HxD 를 사용하여 Atom_disk_0.dd 파일을 연다. 해당 파일이 4GB 미만이기 때문에 사용할 수 있다.

[WHS-2] .iso

000001A0	07 20 0E 70 05 72 01 73 09 0E 07 20 73 73 73 73	g operating sys
000001B0	65 6D 00 00 00 63 7B 9A 00 00 00 00 00 00 00 00	em...c{š.....
000001C0	02 00 EE FE 7F 04 01 00 00 00 FF FF FF FF 00 00	..ip.....ÿÿÿÿ..
000001D0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
000001F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AAU*

[그림 3] NTFS 파일 시스템

HxD 로 살펴보면 MBR 의 시그니처 55 AA 와 4 개의 파티션이 존재하는 것을 확인할 수 있다. 해당 내용을 보고 NTFS 파일 시스템을 사용하고 있다는 것을 알 수 있다. 또한 파일 시스템의 타입 값이 EE 이므로 GPT 를 사용하고 있다.

00000400	16 E3 C9 E3 5C 0B B8 4D 81 7D F9 2D F0 02 15 AE	.ãÄÄ\..M.)ü-8..@	섹터 2
00000410	B1 85 6B 83 B2 07 08 4E A0 15 CA 03 8D E8 4F EF	±_kf²..N .Ê...èOi	
00000420	22 00 00 00 00 00 00 00 FF 7F 00 00 00 00 00 00	".....ÿ.....	
00000430	00 00 00 00 00 00 00 00 4D 00 69 00 63 00 72 00M.i.c.r.	
00000440	6F 00 73 00 6F 00 66 00 74 00 20 00 72 00 65 00	o.s.o.f.t. .r.e.	
00000450	73 00 65 00 72 00 76 00 65 00 64 00 20 00 70 00	s.e.r.v.e.d. .p.	
00000460	61 00 72 00 74 00 69 00 74 00 69 00 6F 00 6E 00	a.r.t.i.t.i.o.n.	
00000470	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00000480	A2 A0 D0 EB E5 B9 33 44 87 C0 68 B6 B7 26 99 C7	c ðeÄ³3D+Ähŧ·š™Ç	
00000490	EE 6B F2 4B 92 22 D9 45 84 05 1D DB D5 C9 6A AA	ikòK' "ÜE...ÜÖËj*	
000004A0	00 80 00 00 00 00 00 00 FF 9F 03 00 00 00 00 00	.€.....ÿÿ.....	
000004B0	00 00 00 00 00 00 00 00 80 42 00 61 00 73 00 69 00€B.a.s.i.	
000004C0	63 00 20 00 64 00 61 00 74 00 61 00 20 00 70 00	c. .d.a.t.a. .p.	
000004D0	61 00 72 00 74 00 69 00 74 00 69 00 6F 00 6E 00	a.r.t.i.t.i.o.n.	
000004E0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
000004F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	
00000500	A2 A0 D0 EB E5 B9 33 44 87 C0 68 B6 B7 26 99 C7	c ðeÄ³3D+Ähŧ·š™Ç	
00000510	87 49 9E 58 B8 FC 85 48 A5 4C F6 E7 1F DC 02 55	+IëX,ü_HŧLôç.Ü.U	
00000520	00 A0 03 00 00 00 00 00 FF 6F 0B 00 00 00 00 00ÿo.....	
00000530	00 00 00 00 00 00 00 00 42 00 61 00 73 00 69 00B.a.s.i.	
00000540	63 00 20 00 64 00 61 00 74 00 61 00 20 00 70 00	c. .d.a.t.a. .p.	
00000550	61 00 72 00 74 00 69 00 74 00 69 00 6F 00 6E 00	a.r.t.i.t.i.o.n.	
00000560	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	

[그림 4] 섹터 2

NTFS 는 1 섹터 아래에 백업 내용을 담고 있기에 섹터 2 로 이동한다. 해당 내용을 보니 [그림 1]에서 보았던 것처럼 파티션이 3 개 존재하는 것을 확인할 수 있다.

173FFFF0	00 00 00 00 00 00 8A 01 A7 01 BF 01 00 00 55 AAŠ.Š.¿...U*	
17400000	EB 58 90 2D 46 56 45 2D 46 53 2D 00 02 08 00 00	ëX. -FVE-FS-.....	섹터 761,856
17400010	00 00 00 00 00 00 F8 00 00 3F 00 FF 00 00 A0 0B 00ø...?.ÿ... ..	
17400020	00 00 00 00 00 E0 1F 00 00 00 00 00 00 00 00 00ä.....	
17400030	01 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00	
17400040	80 00 29 00 00 00 00 4E 4F 20 4E 41 4D 45 20 20	€.)....NO NAME	
17400050	20 20 46 41 54 33 32 20 20 20 33 C9 8E D1 BC F4	FAT32 3ÉŽŦ46	

[그림 5] 비트락커

지금까지의 정보를 이용하여 NTFS 의 시작 위치들을 탐색해본 결과 섹터 761,856 에서 비트락커를 발견했다. 해당 섹터 시작 위치부터 파일의 끝까지 선택하여 새 파일에 복사, 붙여넣기를 한다.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	2D	46	56	45	2D	46	53	2D	00	02	08	00	00	ëX.-FVE-FS-.....
00000010	00	00	00	00	00	F8	00	00	3F	00	FF	00	00	A0	0B	00ø...?.ÿ... ..
00000020	00	00	00	00	E0	1F	00	00	00	00	00	00	00	00	00	00à.....
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	80	00	29	00	00	00	00	4E	4F	20	4E	41	4D	45	20	20	€.)....NO NAME
00000050	20	20	46	41	54	33	32	20	20	20	33	C9	8E	D1	BC	F4	FAT32 3ÉŽÑ*4ø
00000060	7B	8E	C1	8E	D9	BD	00	7C	A0	FB	7D	B4	7D	8B	F0	AC	{ŽĂŽŮ* . ů}´<ø~
00000070	98	40	74	0C	48	74	0E	B4	0E	BB	07	00	CD	10	EB	EF	"@t.Ht.´.»...í.ëi
00000080	A0	FD	7D	EB	E6	CD	16	CD	19	00	00	00	00	00	00	00	ý)ëæí.í.....
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	3B	D6	67	49	29	2E	D8	4A	83	99	F6	A3	39	E3	D0	01	;ÖgI).0Jf™6ē9ãD.
000000B0	00	00	20	02	00	00	00	00	00	00	A0	05	00	00	00	00
000000C0	00	00	20	09	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

[그림 6] Atom_disk_0.dd의 비트락커 부분

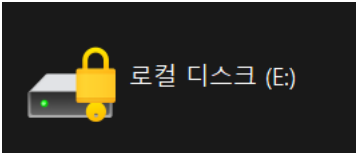
해당 파일 시스템은 FAT32 를 따른다는 것을 확인할 수 있다. 또한 선택된 부분에서 끝 지점(x9200000)을 계산하여 이동한다.

09200000	2D	46	56	45	2D	46	53	2D	39	00	02	00	04	00	04	00	-FVE-FS-9.....
09200010	00	00	40	1F	00	00	00	00	00	00	00	00	10	00	00	00	...ë.....
09200020	00	00	20	02	00	00	00	00	00	00	A0	05	00	00	00	00
09200030	00	00	20	09	00	00	00	00	00	00	21	02	00	00	00	00
09200040	46	03	00	00	01	00	00	00	30	00	00	00	46	03	00	00	F.....0...F...
09200050	6E	CE	6D	EF	D7	37	3B	4A	98	19	15	F3	97	07	25	44	nîmî×7;J~..ó-.%D
09200060	0A	00	00	00	04	80	04	80	F5	3B	80	04	F0	64	D7	01€..ëð;€..ðd×.
09200070	46	00	07	00	02	00	01	00	44	00	45	00	53	00	4B	00	F.....D.E.S.K.

[그림 7] 볼륨 크기

오프셋을 이동하여 해당 위치에 도달했고, 이곳에서 볼륨의 크기를 구할 수 있다. **볼륨의 크기는 x1F400000 이므로 500MB 이다.** 따라서 x9200000 부터 500MB 만큼 선택하여 .vhd 확장명으로 저장한다.

2. 볼륨(#1 의 암호화된 볼륨)은 언제 암호화되었습니까?



[그림 8] BDE 이미지의 E드라이브

E 드라이브로 비트락커 파티션으로 인식되는 것을 확인했다.

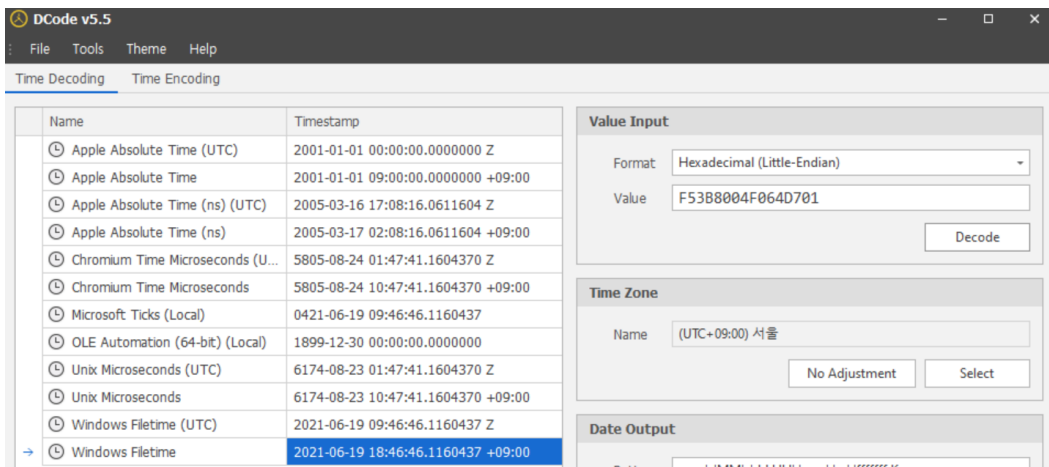
[WHS-2] .iso

```

02200000 2D 46 56 45 2D 46 53 2D 39 00 02 00 04 00 04 00 -FVE-FS-9.....
02200010 00 00 40 1F 00 00 00 00 00 00 00 10 00 00 00 ..@.....
02200020 00 00 20 02 00 00 00 00 00 00 A0 05 00 00 00 .. .....
02200030 00 00 20 09 00 00 00 00 00 00 21 02 00 00 00 .. .....!...
02200040 46 03 00 00 01 00 00 00 30 00 00 00 46 03 00 00 F.....0...F...
02200050 6E CE 6D EF D7 37 3B 4A 98 19 15 F3 97 07 25 44 nĩmĩ*7;J~..ó-.%D
02200060 0A 00 00 00 04 80 04 80 F5 3B 80 04 F0 64 D7 01 .....€.€5;€.8d×.
02200070 46 00 07 00 02 00 01 00 44 00 45 00 53 00 4B 00 F.....D.E.S.K.
02200080 54 00 4F 00 50 00 2D 00 52 00 43 00 55 00 52 00 T.O.P.-.R.C.U.R.
02200090 56 00 39 00 33 00 20 00 41 00 74 00 6F 00 6D 00 V.9.3. .A.t.o.m.
022000A0 20 00 36 00 2F 00 31 00 39 00 2F 00 32 00 30 00 .6./..9./..2..
  
```

[그림 9] FVE metadata header

볼륨이 암호화된 시각은 FVE 메타데이터 헤더에서 알 수 있다.



[그림 10] 암호화된 시각

DCode 를 사용하여 볼륨이 암호화된 시각을 구할 수 있다. 볼륨이 암호화된 시각은 2021-06-19 18:46:46.1160437 (+09:00)이다.

3. 기밀 파일의 SHA1 해시값 계산(Small Modular Reactor.png)

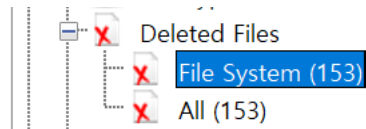
```

00000190 71 91 4B 7E 65 65 3C 01 02 00 08 00 01 00 E2 BD q'K~ee<.....
000001A0 BD 17 9C 91 BD 41 A1 06 CE 69 F3 31 AF 72 C0 D0 4.α~A;..ĩó1~AD
000001B0 2F 16 F0 64 D7 01 00 00 08 AC 00 00 00 03 00 /.8d×.....
  
```

[그림 11] 키 값

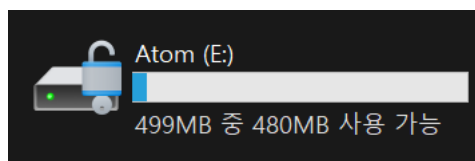
해당 부분이 키 값이다. 키는 17BDBDE2-919C-41BD-A106-CE69F331AF72 라고 읽는다. 하지만 해당 키 값이 옳지 않다고 뜬으로 키는 복호화 해야 할 것 같다.

[WHS-2] .iso



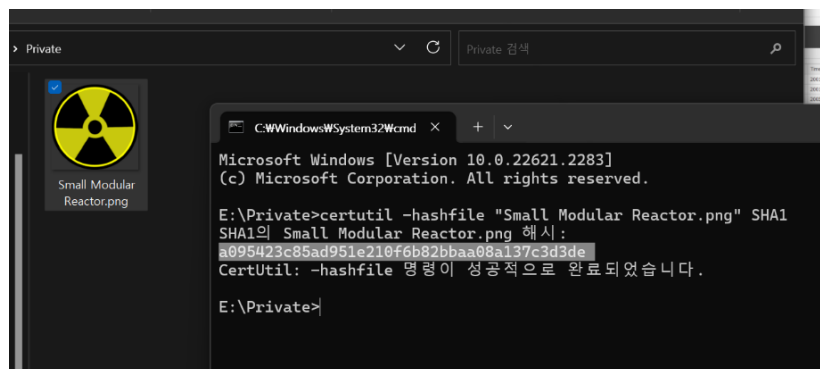
[그림 12] Autopsy로 연 Atom_USB.dd

Autopsy 를 사용하여 Atom_USB.dd 파일을 열어봤다. 보니 많은 파일이 삭제된 것을 볼 수 있다. 이곳에서 파일을 탐색하며 복호화 전 키가 17BDBDE2-919C-41BD-A106-CE69F331AF72 인 것을 찾고, 복호화된 키가 574783-056749-579095-431145-710270-240900-174637-560197 인 것을 알아낼 수 있다.



[그림 13] 열린 Atom 디스크

하지만 복구키를 입력해 보아도 7 번째 자리가 틀리다고 나온다. 해당 7 번째 자리를 복호화하기 위해 코드를 입력한다. 코드는 [recovery_code] 파일로 첨부해 두었고, [6. 별도 첨부]에도 첨부해 두었다. 해당 파일을 PowerShell ISE 를 열어 실행해야 한다. 그러면 7 번째 자리의 숫자가 174636 이라고 나오고, 최종적인 키 값은 574783-056749-579095-431145-710270-240900-174636-560197 이다.



[그림 14] Small Modular Reactor.png

Atom 디스크 안을 보면 Private 폴더가 존재하고 그 안에 Small Modular Reactor.png 사진이 있다. 해당 파일을 SHA1 값을 구하면 **a095423c85ad951e210f6b82bbaa08a137c3d3de** 이다.

5. Flag

- 1. 500MB
- 2. 18:46:46.1160437 (+09:00)
- 3. a095423c85ad951e210f6b82bbaa08a137c3d3de

6. 별도 첨부

- 문제

기업의 핵심 기술 유출 혐의를 받고 있는 이 씨는 컴퓨터 지식 수준이 높아 기업의 비밀 기술 정보를 PC에 안전하게 저장해 왔습니다. 수사관이 용의자의 하드디스크 1 개와 USB 저장고 1 개를 확보했지만 USB는 이미 포맷된 상태였습니다. 디스크에 회사의 비밀 파일인 'Small Modular Reactor.png'을 열람한 흔적이 일부 남아 있지만 원본 파일은 존재하지 않습니다. 용의자가 숨겨둔 비밀 파일을 찾아보세요.

Questions 참고로 용의자는 윈도우 10을 사용했습니다. UTC+9 시간대를 기준으로 다음 질문을 분석해야 합니다.

1. 암호화된 볼륨의 정확한 용량은 얼마입니까? (40 포인트)
2. 볼륨(#1의 암호화된 볼륨)은 언제 암호화되었습니까? (40 포인트)
3. 기밀 파일의 SHA1 해시값 계산(Small Modular Reactor.png) (220 점)

- recovery_code

```
1 $count_drive_letter = "E:"
2
3 $recovery_key1 = "574783-056749-579095-431145-710270-240900-"
4 $recovery_key7 = ""
5 $recovery_key8 = "-560197"
6
7 for ($count = 0; $count -le 1000000; $count++){
8     $recovery_key7 = $count.ToString()
9
10     if ($recovery_key7.Length -lt 6){
11         $padd = "0" * (6 - $recovery_key7.Length)
12         $recovery_key7 = $padd + $recovery_key7
13     }
14
15     $beauty_recovery_key = $recovery_key1 + $recovery_key7 + $recovery_key8
16     $recovery_key = $beauty_recovery_key.replace("-", "")
17
18     try{
19         Unlock-BitLocker -MountPoint $count_drive_letter -RecoveryPassword $recovery_key -ErrorAction Stop
20         Write-Output "[${count}] Matched!! : $beauty_recovery_key"
21         break
22     }
23     catch [System.Runtime.InteropServices.COMException] {
24         Write-Output "[${count}] Failed Recovery Key: $beauty_recovery_key"
25     }
26 }
```

7. Reference

- [URL]