

Write-up: Corrupted Disk Image (level1)

<https://dreamhack.io/wargame/challenges/1189>

[1. Challenge Info](#)

[2. Problem Description](#)

[3. Provided Files](#)

[4. Tools Used](#)

[5. Analysis Steps](#)

[5.1 디스크 이미지 마운트](#)

[5.2 VBR 시그니처 확인 및 복사](#)

[5.3 FTK Imager로 복구 확인](#)

[5.4 플래그 확인 및 해시 계산](#)

[5.5 최종 정보 정리](#)

[6. Flag](#)

1. Challenge Info

- **Challenge Name:** Corrupted Disk Image
- **Category:** Forensics
- **Difficulty Level:** Level 1
- **Platform:** DreamHack Wargame
- **Objective:**

주어진 디스크의 이미지를 복원하여 플래그를 완성하라.

플래그 형식: **DH{something}**

- something의 길이: 32자

2. Problem Description

“디스크 이미지가 열리지 않습니다...!

주어진 디스크 이미지를 복원하여 플래그를 구해주세요.”

- 제공된 디스크 이미지: `CorruptedDiskImage.E01`

3. Provided Files

- `CorruptedDiskImage.E01` (Windows 시스템의 디스크 이미지 파일, E01 포맷)

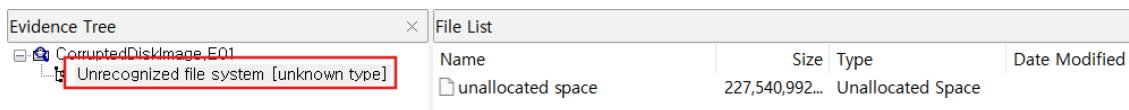
4. Tools Used

Tool	Version
FTK Imager	v4.7.8.31
HxD	v2.5.0.0
Windows PowerShell	v5.1.26100.4202

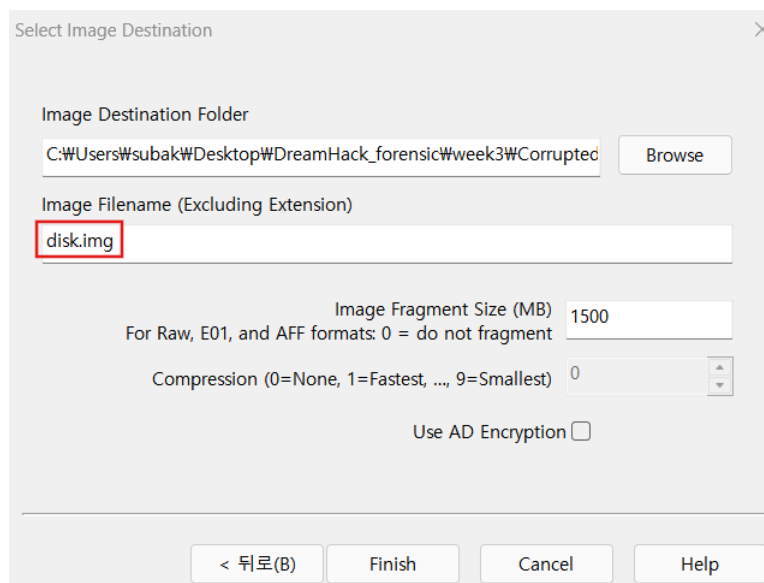
5. Analysis Steps

5.1 디스크 이미지 마운트

- 도구: FTK Imager
- 제공된 **CorruptedDiskImage.E01** 파일을 FTK Imager로 열기
 - Unrecognized file system 표시



- Export Disk Image → Raw(dd) 선택 → **disk.img** 로 저장



5.2 VBR 시그니처 확인 및 복사

- 도구: HxD
- 저장한 **disk.img** 파일을 HxD로 열어 NTFS VBR 시그니처를 찾아 덮어써 손상된 VBR을 복구

- VBR 시그니처 확인

- 문자열 검색에서 NTFS 입력하여 VBR 영역 찾기
- EB 52 90 4E 54 46 53 값이 VBR 시그니처임을 확인

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
0D8FFDE0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	yyyyyyyyyyyyyyyyyy
0D8FFDF0	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	oooooooooooooooooooo
0D8FFE00	EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00	ëR.NTFS
0D8FFE10	00	00	00	00	00	F8	00	00	3F	00	FF	00	00	48	19	01ø...?.ÿ..H..

- VBR 시그니처 복사 및 덮어쓰기

- 시그니처가 있는 영역(EB 52 90 4E 54 46 53 이후의 VBR 전체)을 복사하여 손상된 앞 부분에 덮어씀

disk.img.001

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00	ëR.NTFS
00000010	00	00	00	00	00	F8	00	00	3F	00	FF	00	00	48	19	01ø...?.ÿ..H..
00000020	00	00	00	00	80	00	00	00	FF	C7	06	00	00	00	00	00€...ÿÇ.....
00000030	AA	A6	00	00	00	00	00	00	02	00	00	00	00	00	00	00	*
00000040	F6	00	00	00	01	00	00	00	A5	48	A0	F0	7C	A0	F0	F2	ö.....¥H ð ðò
00000050	00	00	00	00	FA	33	C0	8E	D0	BC	00	7C	FB	68	C0	07ú3ÀŽĐ*. úhÀ.
00000060	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA
00000070	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA

Offset (h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA
00000010	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA
00000020	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA
00000030	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA
00000040	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA
00000050	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA
00000060	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA
00000070	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AA	AAAAAAAAAAAAAAAA

Context menu options:

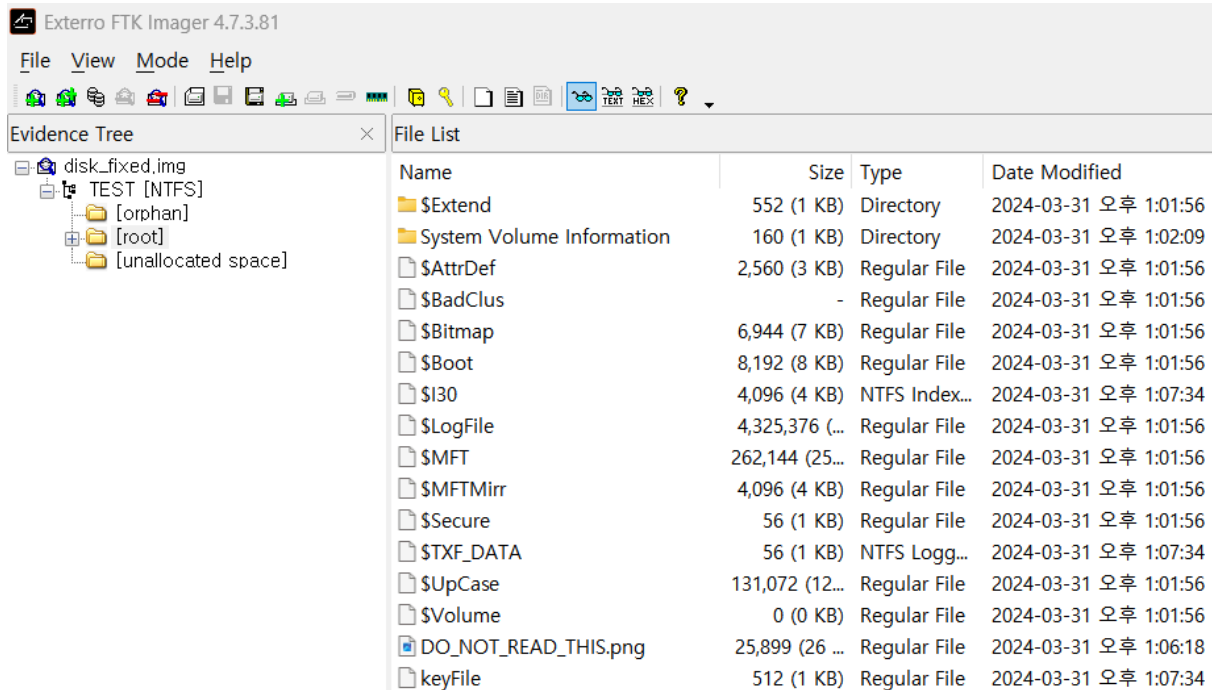
- 되돌리기(U) Ctrl+Z
- 잘라내기(T) Ctrl+X
- 복사(C) Ctrl+C
- 붙여넣기 삽입(I) Ctrl+V
- 붙여넣기 쓰기(W) Ctrl+B**

- 덮어쓴 이미지 다른 이름으로 저장

- VBR 덮어쓰기가 완료되면, 다른 이름으로 저장을 선택하여 disk_fixed.img 로 저장

5.3 FTK Imager로 복구 확인

- 도구: FTK Imager
- 저장한 `disk_fixed.img` 를 FTK Imager로 열어 VBR 복구 결과 확인
 - 정상적으로 NTFS 파티션 및 파일 구조가 인식되는지 확인하여 복구 여부 검증



5.4 플래그 확인 및 해시 계산

- 도구: FTK Imager
- 경로: `C:\DO_NOT_READ_THIS.png`
 - `DO_NOT_READ_THIS.png` 파일 확인
 - 이미지를 열어보면 플래그가 다음과 같이 작성되어 있음

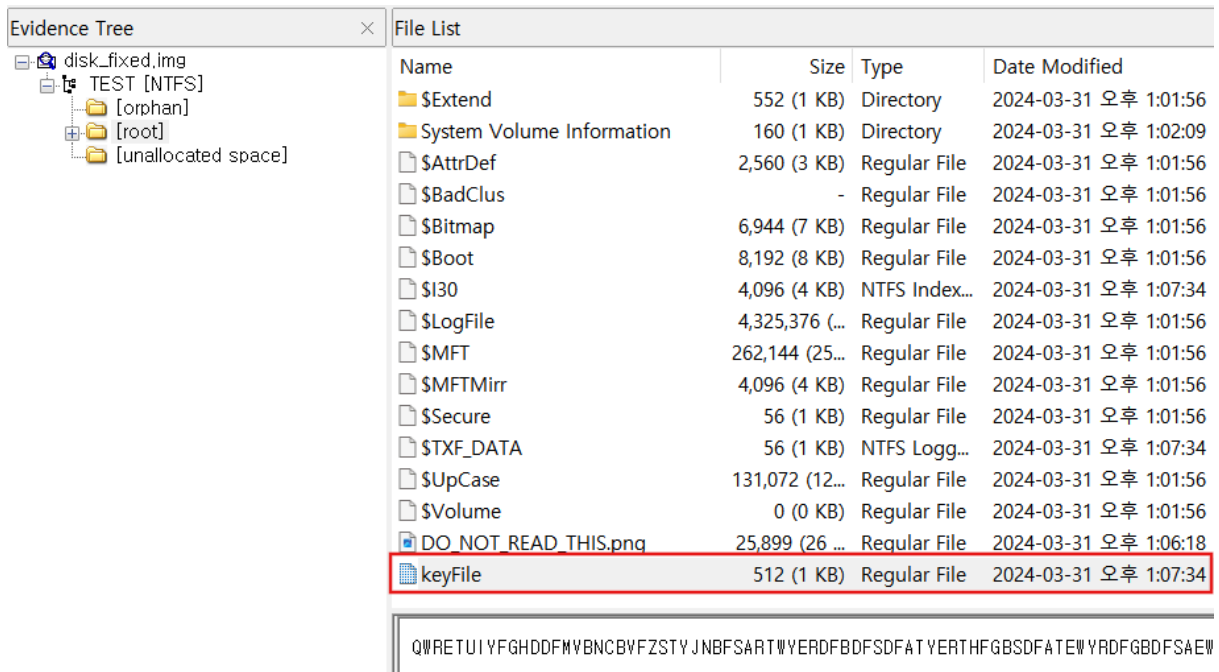
The screenshot shows the FTK Imager interface. On the left, the 'Evidence Tree' pane displays the disk structure: 'disk_fixed.img' containing a 'TEST [NTFS]' volume with subfolders '[orphan]', '[root]', and '[unallocated space]'. The main 'File List' pane shows a table of files and directories. The file 'DO_NOT_READ_THIS.png' is highlighted with a red box. Below the file list, a large red-bordered box contains handwritten text.

Name	Size	Type	Date Modified
\$Extend	552 (1 KB)	Directory	2024-03-31 오후 1:01:56
System Volume Information	160 (1 KB)	Directory	2024-03-31 오후 1:02:09
\$AttrDef	2,560 (3 KB)	Regular File	2024-03-31 오후 1:01:56
\$BadClus	-	Regular File	2024-03-31 오후 1:01:56
\$Bitmap	6,944 (7 KB)	Regular File	2024-03-31 오후 1:01:56
\$Boot	8,192 (8 KB)	Regular File	2024-03-31 오후 1:01:56
\$I30	4,096 (4 KB)	NTFS Index...	2024-03-31 오후 1:07:34
\$LogFile	4,325,376 (...)	Regular File	2024-03-31 오후 1:01:56
\$MFT	262,144 (25...)	Regular File	2024-03-31 오후 1:01:56
\$MFTMirr	4,096 (4 KB)	Regular File	2024-03-31 오후 1:01:56
\$Secure	56 (1 KB)	Regular File	2024-03-31 오후 1:01:56
\$TXF_DATA	56 (1 KB)	NTFS Logg...	2024-03-31 오후 1:07:34
\$UpCase	131,072 (12...)	Regular File	2024-03-31 오후 1:01:56
\$Volume	0 (0 KB)	Regular File	2024-03-31 오후 1:01:56
DO_NOT_READ_THIS.png	25,899 (26 ...)	Regular File	2024-03-31 오후 1:06:18
keyFile	512 (1 KB)	Regular File	2024-03-31 오후 1:07:34

Handwritten text in the red box:

Flag is
DH{ Sha-256 (keyFile) }
HA HA HA HA

- 도구: Windows PowerShell
- `C:\keyFile` 파일을 FTK Imager를 이용해 추출



- 추출한 `keyFile` 에 대해 PowerShell에서 SHA-256 해시 확인
- 명령어:

```
Get-FileHash keyFile -Algorithm SHA256
```

- KeyFile의 SHA256 해시:

```
E71E2B1230FD090AEBD3A347310ACAC611E0161684FB4B7703135B6CC91BB7AC
```

```
PS C:\Users\subak\Desktop\DreamHack_forensic\week3\CorruptedDiskImage> Get-FileHash keyFile -Algorithm SHA256
```

Algorithm	Hash	Path
SHA256	E71E2B1230FD090AEBD3A347310ACAC611E0161684FB4B7703135B6CC91BB7AC	C:\Users\subak\Desktop\DreamH...

5.5 최종 정보 정리

something: `E71E2B1230FD090AEBD3A347310ACAC611E0161684FB4B7703135B6CC91BB7AC`

- 플래그 제출 시 대문자로 제출하면 실패
 - 소문자로 변경하여 제출

6. Flag

DH{e71e2b1230fd090aebd3a347310acac611e0161684fb4b7703135b6cc91bb7ac}

축하합니다!

LEVEL 1 Corrupted Disk Image 문제를 해결했습니다.

대단해요. 문제를 어떻게 해결하셨나요?
풀이를 작성하면 포인트까지 받을 수 있어요.

괜찮아요

 풀이 작성하기