

# Write-up: VBR (level1)

<https://dreamhack.io/wargame/challenges/1188>

[1. Challenge Info](#)

[2. Problem Description](#)

[3. Provided Files](#)

[4. Tools Used](#)

[5. Analysis Steps](#)

[5.1 VBR 분석](#)

[5.2 A: 파일 시스템 확인](#)

[5.3 B: 해당 볼륨의 크기 확인](#)

[5.4 C: 볼륨 시리얼 번호 확인](#)

[5.5 최종 정보 정리](#)

[6. Flag](#)

## 1. Challenge Info

- **Challenge Name:** VBR
- **Category:** Forensics
- **Difficulty Level:** Level 1
- **Platform:** DreamHack Wargame
- **Objective:**

주어진 VBR을 분석하고, 파일 시스템, 해당 볼륨의 크기, 볼륨 시리얼 번호를 분석하여 플래그를 완성하라.

플래그 형식: **DH{(A + B + C)}**

- (단, 더한 값을 십진수로 변환할 것)
- A: 파일 시스템이 FAT32면 **1**, NTFS면 **2**
- B: 해당 볼륨의 크기
- C: 볼륨 시리얼 번호

---

## 2. Problem Description

“주어진 VBR을 분석하고, 플래그를 계산하시오.”

- 제공된 파일: `vbr.bin`

---

## 3. Provided Files

- `vbr.bin` (Windows 파일 시스템의 VBR 덤프 파일, 바이너리 원시 이미지)

---

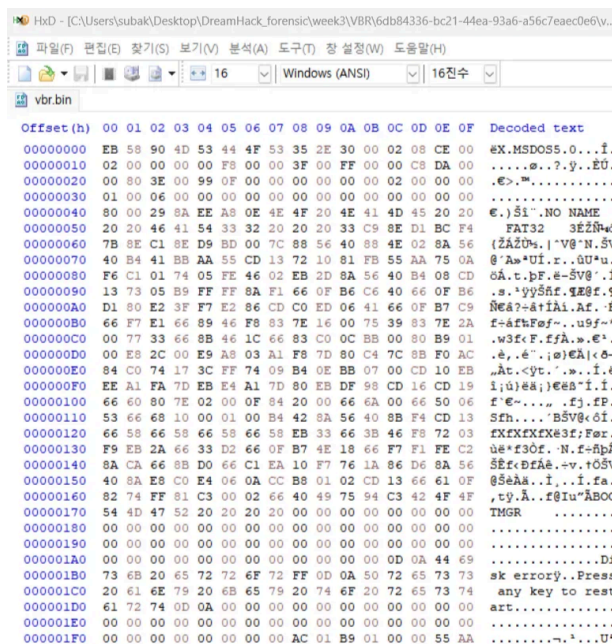
## 4. Tools Used

Tool	Version
FTK Imager	v4.7.8.31
HxD	v2.5.0.0

## 5. Analysis Steps

### 5.1 VBR 분석

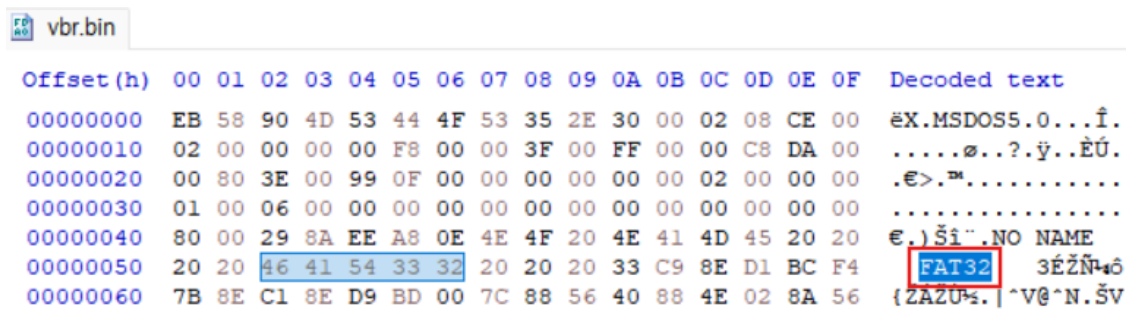
- 도구: HxD
- 제공된 **vbr.bin** 파일을 HxD로 열기



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 EB 58 90 4D 53 44 4F 53 35 2E 30 00 02 08 CE 00 ěX.MSDOS5.0...ĭ.
00000010 02 00 00 00 00 00 F8 00 00 3F 00 FF 00 00 C8 DA 00 .....ø...?.ÿ...ËÚ.
00000020 00 80 3E 00 99 0F 00 00 00 00 00 00 00 02 00 00 00 .E>..™.....
00000030 01 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040 80 00 29 8A EE A8 0E 4E 4F 20 4E 41 4D 45 20 20 20 €. )Ši~.NO NAME
00000050 20 20 46 41 54 33 32 20 20 20 33 C9 8E D1 BC F4 FAT32 3EŽN+o
00000060 7B 8E C1 8E D9 BD 00 7C 88 56 40 88 4E 02 8A 56 {ZAZU+. | ^V@^N.ŠV
```

### 5.2 A: 파일 시스템 확인

- Offset **0x52** 에서 **FAT32** 문자열 확인
- 파일 시스템: **FAT32**



```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 EB 58 90 4D 53 44 4F 53 35 2E 30 00 02 08 CE 00 ěX.MSDOS5.0...ĭ.
00000010 02 00 00 00 00 00 F8 00 00 3F 00 FF 00 00 C8 DA 00 .....ø...?.ÿ...ËÚ.
00000020 00 80 3E 00 99 0F 00 00 00 00 00 00 00 02 00 00 00 .E>..™.....
00000030 01 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00000040 80 00 29 8A EE A8 0E 4E 4F 20 4E 41 4D 45 20 20 20 €. )Ši~.NO NAME
00000050 20 20 46 41 54 33 32 20 20 20 33 C9 8E D1 BC F4 FAT32 3EŽN+o
00000060 7B 8E C1 8E D9 BD 00 7C 88 56 40 88 4E 02 8A 56 {ZAZU+. | ^V@^N.ŠV
```

## 5.3 B: 해당 볼륨의 크기 확인

- Total Sectors 추출 (FAT32 VBR 기준)
  - Offset `0x20 ~ 0x23` (4 bytes, Little Endian)
  - 값: `0x003E8000` = 4,096,000 (섹터 수)

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	CE	00	ëX.MSDOS5.0...î.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	C8	DA	00	.....ø...?.ÿ..ËÚ.
00000020	00	80	3E	00	99	0F	00	00	00	00	00	00	02	00	00	00	.€>™.....

- 섹터 크기 확인
  - Offset `0x0B ~ 0x0C` = `0x0200` = 512 bytes

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	CE	00	ëX.MSDOS5.0...î.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	C8	DA	00	.....ø...?.ÿ..ËÚ.

- 바이트 단위 용량으로 변환
  - 4,096,000 (섹터 수) × 512 (바이트) = `2,097,152,000 bytes`

## 5.4 C: 볼륨 시리얼 번호 확인

- FAT32의 시리얼 번호 확인
- Offset `0x43 ~ 0x46` (Little Endian, 4 bytes)
- 값: `0x0EA8EE8A` = `245,952,138`

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	CE	00	ëX.MSDOS5.0...î.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	C8	DA	00	.....ø...?.ÿ..ËÚ.
00000020	00	80	3E	00	99	0F	00	00	00	00	00	00	02	00	00	00	.€>™.....
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000040	80	00	29	8A	EE	A8	0E	4E	4F	20	4E	41	4D	45	20	20	€.)Si"NO NAME
00000050	20	20	46	41	54	33	32	20	20	20	33	C9	8E	D1	BC	F4	FAT32 3ÉŽŇ4ó

## 5.5 최종 정보 정리

- 파일시스템: FAT32 (`A = 1`)

- 해당 볼륨의 크기: 2,097,152,000 bytes ( B )
- C볼륨 시리얼 번호:0x0EA8EE8A = 245,952,138 ( C )

- 합산:

$$1 + 2,097,152,000 + 245,952,138 = 2,343,104,139$$

## 6. Flag

DH{2343104139}


축하합니다!

### 1 LEVEL 1 VBR

문제를 해결했습니다.

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

괜찮아요

 풀이 작성하기