



# [Job Interview] Write-Up

작성자	윤지원
분석 일자	2024.05.24
작성 일자	2024.05.24
분석 대상	Image_forensic.e01
문서 버전	1.0
작성자 E-mail	<a href="mailto:yoonsjw0827@gmail.com">yoonsjw0827@gmail.com</a>

0. 목차

1. 문제 .....3

2. 분석 도구 .....3

3. 환경 .....3

4. Write-Up.....4


5. Flag.....7

6. 별도 첨부 .....8

7. Reference .....9



1. 문제

URL	<a href="https://www.root-me.org/en/Challenges/Forensic/Job-interview">https://www.root-me.org/en/Challenges/Forensic/Job-interview</a>
문제 내용	You are invited to an interview for a forensics investigator position at the NSA. For your first technical evaluation they ask you to analyze this file. Prove to them that you're a fitting candidate for this job.
문제 파일	 image_forensic.e01
문제 유형	Disk forensics
난이도	2 / 3

2. 분석 도구

도구명	다운로드 링크	Version
Autopsy	<a href="https://www.autopsy.com/download/">https://www.autopsy.com/download/</a>	4.21.0
HxD	<a href="https://mh-nexus.de/en/downloads.php?product=HxD20">https://mh-nexus.de/en/downloads.php?product=HxD20</a>	2.5

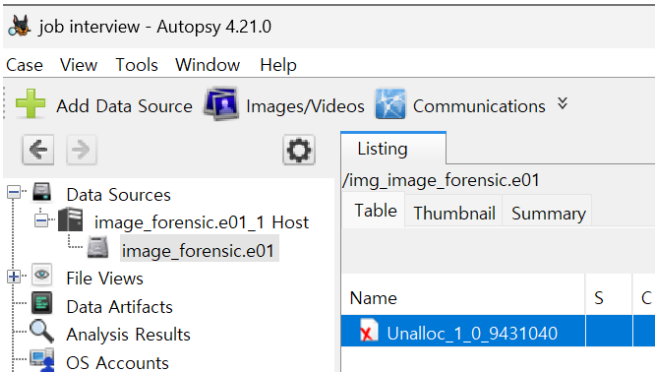
3. 환경

OS
Windows 11 64-bit

## 4. Write-Up

파일명	Image_forensic.e01
용량	237KB
SHA256	b35f4cd4bad19301e6970b30c1c713883b657858ef86d2b7247272c9d0f23591
Timestamp	2024-05-24 13:20:24

파일을 분석하라는 간단한 문제이다. 기존에는 이미지 파일 분석에 FTK Imager 를 이용했는데, 이번에는 다른 툴을 이용해보고 싶어서 Autopsy 를 사용해서 image\_forensic.e01 을 열어보았다.



[사진 1] Auptosy로 image\_forensic.e01 파일을 연 모습

증거 파일 안에서 정체를 모르겠는 파일 하나만이 존재하였다. 따라서 extract file을 통해 추출해 주었다. 이것이 어떤 파일인지 알기 위해 HxD를 이용해서 살펴보았다.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000060	00	00	00	00	30	30	30	30	37	37	30	00	30	30	30	30	...0000770.0000
00000070	30	30	30	00	30	30	30	30	31	37	34	37	00	30	30	30	000.0001747.0004
00000080	33	37	35	37	37	35	34	00	31	32	37	33	34	35	36	35	3757754.12734565
00000090	33	31	33	00	30	31	32	32	33	32	00	20	30	00	00	00	313.012232. 0...
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000C0	00	00	00	00	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	.....
00000100	00	75	73	74	61	72	20	20	00	72	6F	6F	74	00	00	00	..ustar..root...
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000120	00	00	00	00	00	00	00	00	00	00	76	62	6F	78	73	66	.....vboxsf..

[사진 2] HxD로 Unalloc 파일을 연 모습

파일을 열자마자 특별히 눈에 띄는 것은 ustar와 root, vboxsf밖에 없었다. 따라서 이들에 대해 검색해보니 ustar이 tar 유형 파일의 확장자가 될 수 있다는 것을 알아냈다. 따라서 tar 파일에 대해 알아보니, 압축 해제를 한 다음에 분석해야 한다는 사실을 알 수 있었다. 따라서 cmd에서 다음과 같은 명령어를 입력하여 압축 해제를 해주었다.

**tar -xvf Unalloc\_1\_0\_9431040**

## [WHS-2] .iso

```
C:\Users\윤지원\OneDrive\바탕 화면\화이트햇프로젝트\문제>tar -xvf Unalloc_1_0_9431040  
x bcache24.bmc
```

### [사진 3] tar 압축 해제를 위해 cmd에 입력

압축 해제를 해주니 bcache24.bmc 라는 파일이 생성되었다. 검색을 통해 bmc 파일이 이미지 파일의 유형들 중 하나라는 것을 알게 되었다. 그러나 일반적인 이미지 파일과 달리 그냥 열리지는 않아서 더 검색해보니, bitmapcacheviewer 라는 것으로 열어야 하는데 현재 이것을 구할 수 있는 방법이 없었다. 그래서 더 알아본 결과 <https://github.com/ANSSI-FR/bmc-tools> 에서 bmc-tools 라는 도구를 발견할 수 있었다. 그래서 이를 이용하여 bcache24.bmc 를 열기 위해 cmd 에 git clone 을 이용하여 해당 레포지토리와 연결시켜 도구를 다운받았다. 명령어는 다음과 같다.

**git clone https://github.com/ANSSI-FR/bmc-tools**

```
C:\Users\윤지원>git clone https://github.com/ANSSI-FR/bmc-tools  
Cloning into 'bmc-tools'...  
remote: Enumerating objects: 82, done.  
remote: Counting objects: 100% (43/43), done.  
remote: Compressing objects: 100% (35/35), done.  
remote: Total 82 (delta 18), reused 27 (delta 8), pack-reused 39  
Receiving objects: 100% (82/82), 37.95 KiB | 4.74 MiB/s, done.  
Resolving deltas: 100% (35/35), done.
```

### [사진 4] git clone 연동

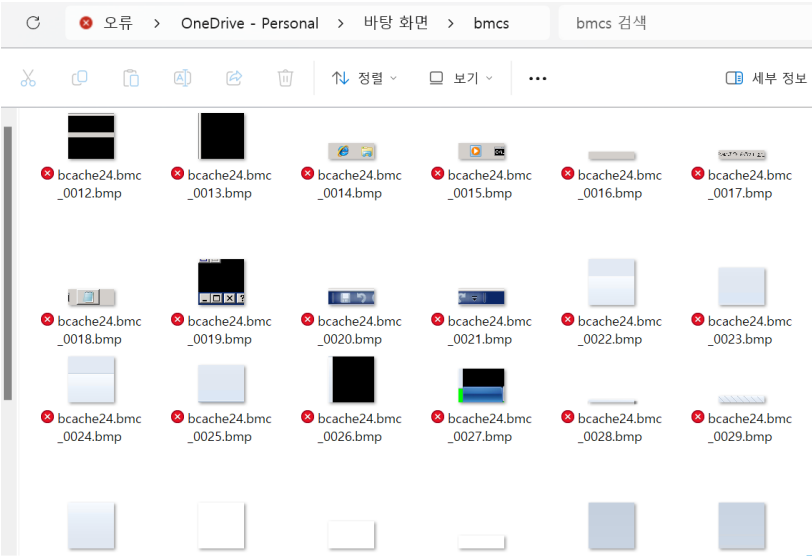
그 다음 python을 이용하여 해당 도구를 실행시켜주고, -s에는 분석할 파일을, -d에는 분석한 결과를 저장할 폴더를 지정해주면 된다. 그러나 처음에는 앞의 주소까지 넣어야 하는지 모르고 단순히 파일명들만 넣었다가 안돼서 헤매기도 했었다. 파일과 폴더의 주소들을 "로 묶어서 입력해주면 다음과 같이 정상적으로 파일들이 export되는 것을 확인할 수 있다. 명령어는 다음과 같다.

**python bmc-tools/bmc-tools.py -s "분석할 파일 위치" -d "저장할 폴더 위치"**

```
C:\Users\윤지원>python bmc-tools/bmc-tools.py -s "C:\Users\윤지원\OneDrive\바탕 화면\화이트햇프로젝트\문제\bcache24.bmc"  
-d "C:\Users\윤지원\OneDrive\바탕 화면\bmc"  
[+++] Processing a single file: 'C:\Users\윤지원\OneDrive\바탕 화면\화이트햇프로젝트\문제\bcache24.bmc'.  
[===] 575 tiles successfully extracted in the end.  
[===] Successfully exported 575 files.
```

### [사진 5] 도구를 통해 bcache24.bmc 압축 해제

[사진 6]과 같이 내가 지정한 bmcs 폴더에 압축 해제된 bmp 파일들이 들어간 모습을 볼 수 있다. 이 파일들을 쭉 살펴보다가 **bcache24.bmc\_0182.bmp, bcache24.bmc\_0183.bmp, bcache24.bmc\_0184.bmp** 파일들에 flag 값이 나뉘어 적혀 있는 것을 발견할 수 있었다.



[사진 6] bmcs 폴더에 압축 해제된 bmp 파일들이 저장된 모습

Yeah  
RdP\_

[사진 7] bcache24.bmc\_0182.bmp

this is the  
l3av3s\_Tra

[사진 8] bcache24.bmc\_0183.bmp

flag:  
c3S

[사진 9] bcache24.bmc\_0184.bmp

따라서 이 세 개의 사진들을 합쳐보면 다음과 같은 문구가 출력된다.

**Yeah this is the flag:**

**RdP\_l3av3s\_Trac3S**

# 5. Flag

RdP\_I3av3s\_Trac3S

## 6. 별도 첨부



## 7. Reference

- <https://brownbears.tistory.com/161>