

BrokenHearted

32기 안현진

포렌식 문제 BrokenHearted를 풀어보기로 했다.

챌린지 0명 해결함 X


BrokenHearted

500

2024 3S ctf 문제제작에 참여하게 되어 너무나 긴장한 제작자. 결국 플래그를 운반중 어딘가에 떨어뜨리는 중대한 실수를 하고마는데.. 없어진 플래그를 찾아주세요!

BrokenHea...

플래그 제출



CTF



Hint

파일을 다운로드 받으면 jpg파일과 텍스트 파일이 보인다.

Why don't you search for file carving?

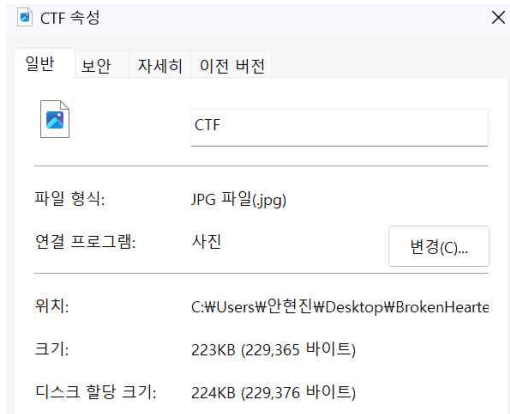
file carving에 대해 검색해보라는 힌트를 얻었다. file carving이 뭔지 몰라 검색해보니 “비할당영역을 대상으로 알려진 파일 헤더(File header, Signature)와 푸터(File footer, EOF(End Of File) Marker)를 이용하여 복원을 하는 기술”이라고 한다.

jpeg 파일은 파일의 일부분만 복구해도 해당 부분만큼은 복원이 된다고 하니, jpeg 파일 특성을 알아보고, 파일에 있는 CTF 이미지 파일을 파일 카빙 해보자.

1. jpeg 파일 살펴보기

먼저 파일 정보를 확인해보았다. jpg와 jpeg는 같은 것을 가리키는 용어이며, 디지털 이미지를 손실 압축하여 파일 크기를 줄이는 데 사용되는 파일들이다. .jpg/.jpeg/.jpe/.jif/.jfif/.jfi 파일 확장자는 모두 JPEG 이미지에 해당한다.

jpeg 파일은 비교적 작은 크기를 유지하면서, 1680만 개의 색상을 표현할 수 있다고 한다. jpeg 이미지의 최대 크기는 6만 5535 픽셀인데, CTF jpg 파일의 크기는 223KB이다.



2. foremost tool 사용

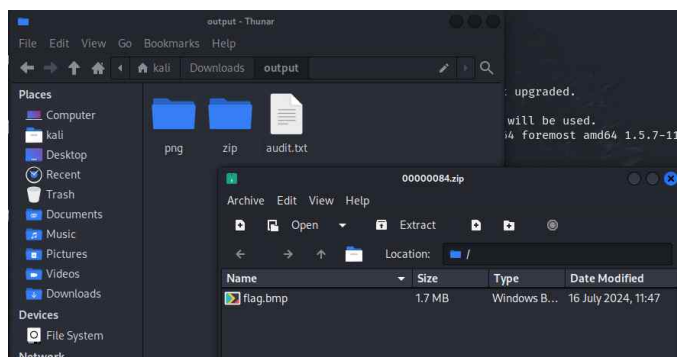
kali 리눅스에서 사용했다. **#apt-get install foremost** 명령어를 사용하여 툴을 다운받았다.

```
(kali@kali) - [~/Downloads]
$ sudo apt-get install foremost
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  foremost
0 upgraded, 1 newly installed, 0 to remove and 682 not upgraded.
Need to get 42.5 kB of archives.
After this operation, 104 kB of additional disk space will be used.
Get:1 http://http.kali.org/kali kali-rolling/main amd64 foremost amd64 1.5.7-11+b2 [42.5 kB]
Fetched 42.5 kB in 1s (28.5 kB/s)
Selecting previously unselected package foremost.
(Reading database ... 398443 files and directories currently installed.)
Preparing to unpack .../foremost_1.5.7-11+b2_amd64.deb ...
Unpacking foremost (1.5.7-11+b2) ...
Setting up foremost (1.5.7-11+b2) ...
Processing triggers for man-db (2.12.0-3) ...
Processing triggers for kali-menu (2023.4.7) ...
```

이제 foremost를 사용해보자. **#foremost -t all -i CTF.jpg** 명령어를 사용하여 파일을 추출했다.

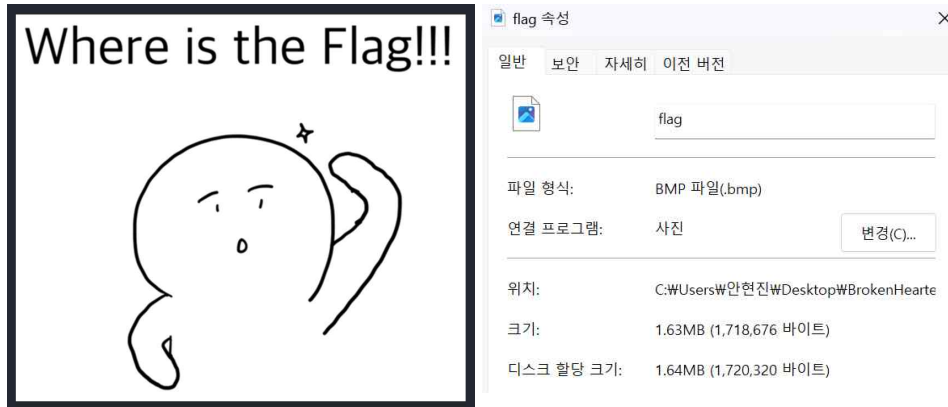
-> -t all: 추출될 파일의 타입을 지정하는 부분이다. 어떤 타입의 파일이 추출될지 모르니 타입을 all로 지정했다.

-> -i CTF.jpg: 분석할 파일명을 적는 부분이다(기본값=stdin)



파일 추출이 끝나면, 'output' 폴더가 생긴다. 폴더를 열어보면 위의 사진과 같은 파일과 텍스트 문서가 있다(다른 사진들도 있다)

zip폴더안에 flag.bmp라는 이미지 파일이 보인다. 열어보자. flag라고 적혀있어서 flag값이 있을거라 기대했는데, 딱히 flag로 보이는 문자열을 보이지 않는다.



이 이미지의 확장자가 bmp라고 되어있다. bmp는 “마이크로소프트의 윈도우와 IBM의 OS/2 운영체제를 위해 개발된 비트맵 포맷”이라고 한다. 근데 이 사진의 크기가 꽤 크다. 근데 bmp 파일은 대부분 파일 크기가 크다고 한다.

2. WinHex tool 사용

CTF.jpg를 파일카빙해서 bmp파일을 포함해 총 3장의 사진을 얻었다, 그래서 이번에는 Window에서 WInHex 툴을 사용해서 파일 카빙을 해보려고 한다.

flag.bmp, hint1.jpg, hint2.png을 카빙해서 헤더를 얻은 파일은 hint.jpg 파일뿐이었다. 그런데 뭔가 결과가 이상하다. 그래서 foremost에서 카빙 성공한 CTF.jpg파일을 다시 카빙해보았는데, foremost의 결과와는 조금 달랐다. 그래서 다시 foremost를 사용해 hint1 파일을 카빙해보려고 한다.

하지만 foremost를 사용해도 결과는 크게 다르지 않았다. 계속 hint1과 flag의 그림만 나올 뿐이다.

3.HxD 사용

flag.bmp파일을 HxD를 사용해서 분석해보았다. 먼저 파일 시그니처가 올바른지 확인했는데, 모두 맞는 번호들이다.

그리고 텍스트 문자열에 flag관련 정보가 있을까해서 텍스트 검색을 해보았다. 그랬더니 진짜

```
001A3940 20 30 30 30 30 30 31 30 30 20 74 6F 20 30 30 30 00000100 to 000
001A3950 30 35 30 30 30 2E 20 54 68 65 20 66 6C 61 67 20 05000. The flag
001A3960 62 65 67 69 6E 73 20 77 69 74 68 20 46 45 20 61 begins with FE a
001A3970 6E 64 20 63 6F 6E 73 69 73 74 73 20 6F 66 20 61 nd consists of a
001A3980 20 74 6F 74 61 6C 20 6F 66 20 32 31 36 20 62 79 total of 216 by
001A3990 74 65 73 2E tes.
```

```
FF 60 E0 CF 1C 01 00 94 39 1A 00 08 00 00 00 66 y`aĩ..."9.....f
6C 61 67 2E 62 6D 70 EC DC 3B 48 1C 41 1C 07 E0 lag.bmpiÜ;H.A..à
```

4. Hint.bmp파일 분석

flag 힌트대로 파일의 중간정도에 FE로 시작하는 16진수들이 있다. 한 줄에 16개의 16진수가 있으니 216 나누기 16을 하면 13.5라는 값이 나온다. 그래서 13줄 + 14번째 줄의 16진수 8개를 잘라내서 FE는 0으로, FF는 1로 이진수 변환을 했다.

간단한 파이썬 코드를 사용하여 변환했다. 변환한 이진수 값을 아스키로 변환해주는 사이트를 통해 해석하면 3S{}형식으로 된 플래그가 출력된다.

☐ 0x/0b prefix

ASCII text

```
3S{f06ens1cs_1s_Re61ly_FUn}
```

Hex (bytes)

```
33 53 7B 66 30 36 65 6E 73 31 63 73 5F 31 73 5F 52 65 36 6C 6C 79 5F
46 55 6E 7D
```

Binary (bytes)

```
001100110001011000110111001101011110011000101110011010111101010010
0110010100110110011011000110110001111001010111101000110010101010110
111001111101
```

<참고자료>

[File Carving이란? :: hacking_security \(tistory.com\)](#)

[파일카빙\(winhex활용\) \(tistory.com\)](#)

[Chapter 3-3. Digital Forensics Tool에 대한 간단한 소개 \(tistory.com\)-도구](#)

[\[디지털 포렌식\] N0Named wargame Left Side B | HxD 도구 사용 \(tistory.com\)](#)

[\[포렌식\] N0Named Wargame \[C\] Left Side B 문제 풀이 | HxD 사용, LSB 변조 :: 보안하는 백선비 \(tistory.com\)](#)