

What File Type 2

2nd Hack The Pods Write-up

問題概要

問題文

ファイルが開けないし、なんかファイルサイズが大きいよ～！うえ～ん！

file: flag.mp3

概要

200点 - Forensics

What File Typeの派生問題

音声ファイルであるはずのflag.mp3が開けない

ファイルサイズが大きいらしい

まずは愚直に `$file`

問題名が What File Type 2 なので, ファイルの種類に関する可能性がある

`file` コマンドはそのファイルの種類を教えてくれる

まずは愚直に `$ file`


問題名が What File Type 2 なので、ファイルの種類に関する可能性がある

`file` コマンドはそのファイルの種類を教えてくれる

```
vagrant@ubuntu-focal:~/what-file-type-2$ file flag.mp3
flag.mp3: PNG image data, 300 x 300, 8-bit/color RGBA, non-interlaced
```

PNG image らしい

拡張子を `.png` に変更してみる



http-ctf{Do_not_trust_
file_extensions}

flag.png

これはWhat File Typeと同じ

これを提出しても当然 Incorrect

ファイルの種類は画像だけどこれ以上進めない...

> なんかファイルサイズが大きいよ～！

これに着目する

画像はバイナリファイルなので基本的に
画像の大きさとファイルサイズは比例する

ファイルサイズを比べてみる

```
$ du -bh flag.mp3
```

```
vagrant@ubuntu-focal:~/what-file-type$ du -bh flag.mp3  
8.1K    flag.mp3
```

What File Type : flag.mp3 -> 8.1KB

```
vagrant@ubuntu-focal:~/what-file-type-2$ du -bh flag.mp3  
231K    flag.mp3
```

What File Type 2 : flag.mp3 -> 231KB

ls -l とかでもファイルサイズは見える

ファイルサイズは全然違うのに画像サイズは同じ...？

What File Type : flag.mp3 -> 8.1K

What File Type 2 : flag.mp3 -> 231KB

```
vagrant@ubuntu-focal:~/what-file-type$ file flag.mp3
flag.mp3: PNG image data, 300 x 300, 8-bit/color RGBA, non-interlaced
```

```
vagrant@ubuntu-focal:~/what-file-type-2$ file flag.mp3
flag.mp3: PNG image data, 300 x 300, 8-bit/color RGBA, non-interlaced
```

どちらも 300 x 300 のPNG imageとされている

2つのファイルのバイナリ差分を調べる

```
$ cmp -l flag1.mp3 flag2.mp3
```

```
vagrant@ubuntu-focal:~$ cmp -l flag1.mp3 flag2.mp3  
cmp: EOF on flag1.mp3 after byte 8235
```

flag1.mp3が8235バイトでEOF (End Of File) らしい

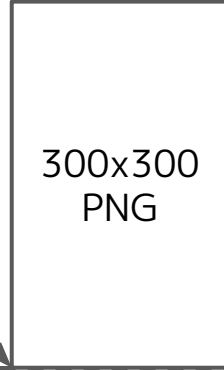
つまり300x300のPNG画像部分は差分がなく

flag2.mp3はまだ続いているということ

What File Type: flag.mp3 → flag1.mp3

What File Type 2: flag.mp3 → flag2.mp3

What File Type
flag.mp3



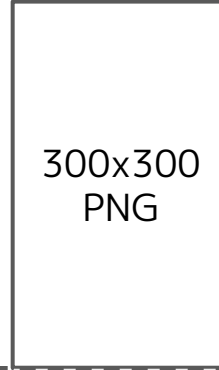
EOF



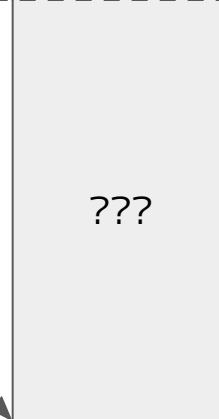
8235



What File Type 2
flag.mp3



EOF



何かしらのファイルが結合されている

Linux バイナリ 分割 取り出す Forensics [検索]

binwalkコマンド

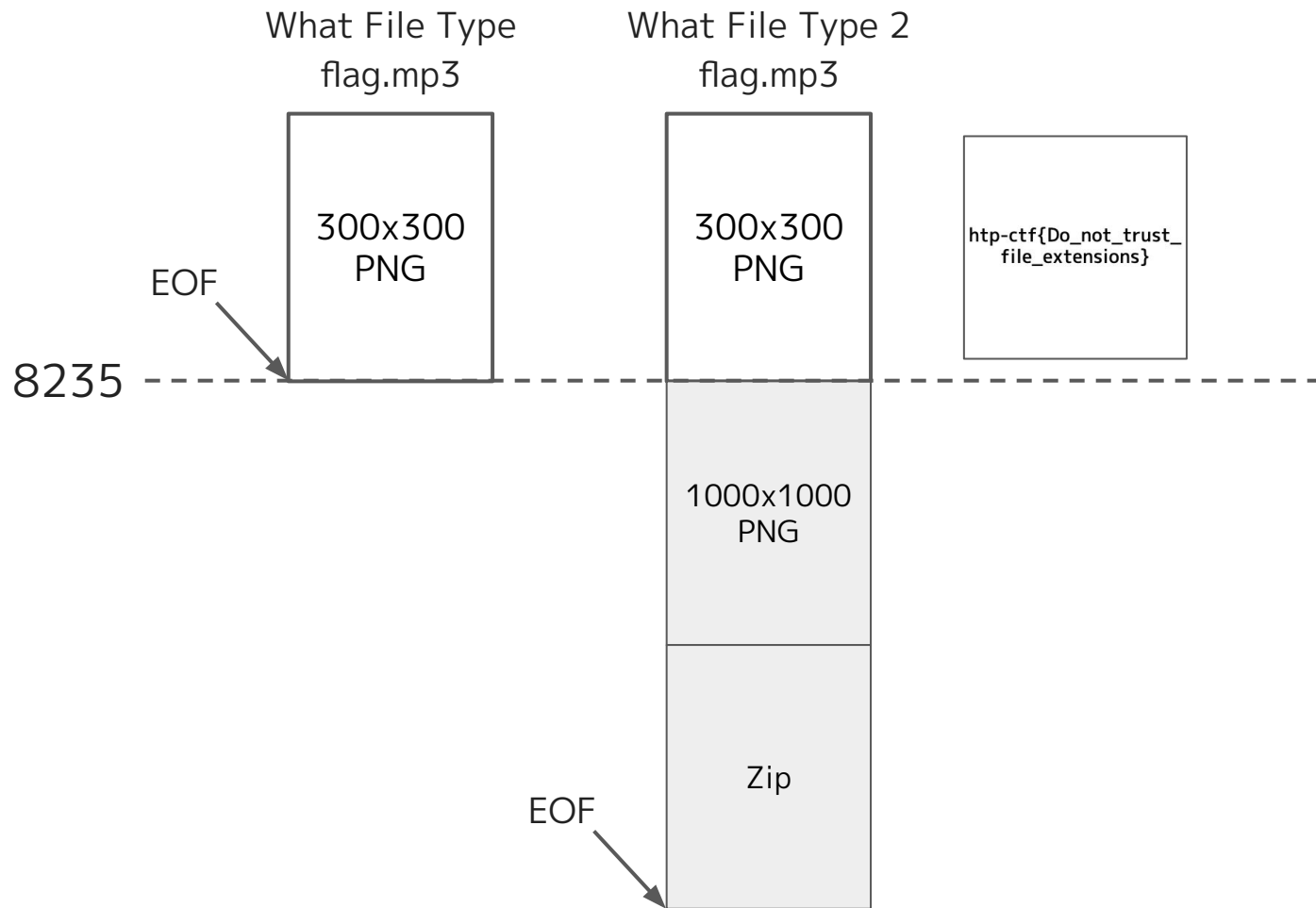
連結されているバイナリなどを確認，抽出できる

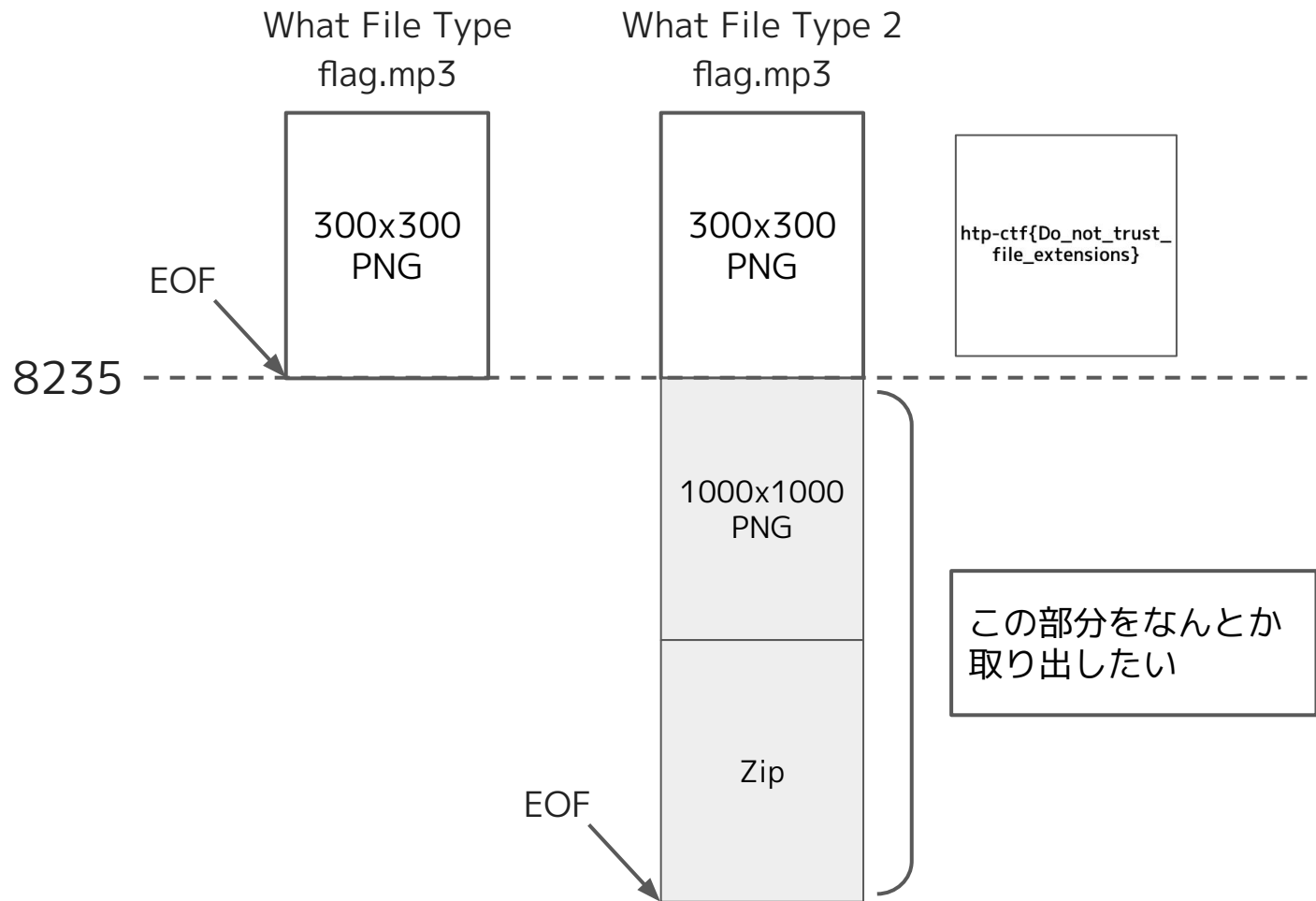
```
vagrant@ubuntu-focal:~/what-file-type-2$ binwalk flag.mp3
```

DECIMAL	HEXADECIMAL	DESCRIPTION

0	0x0	PNG image, 300 x 300, 8-bit/color RGBA, non-interlaced
41	0x29	Zlib compressed data, default compression
8235	0x202B	PNG image, 1000 x 1000, 8-bit/color RGBA, non-interlaced
8276	0x2054	Zlib compressed data, default compression
124475	0x1E63B	Zip archive data, at least v2.0 to extract, compressed size: 111667, uncompressed size: 131948, name: share2.pcapng
236296	0x39B08	End of Zip archive, footer length: 22

どうやら 300x300PNG, 1000x1000PNG, Zip という風に結合されている





結合ファイルから抽出

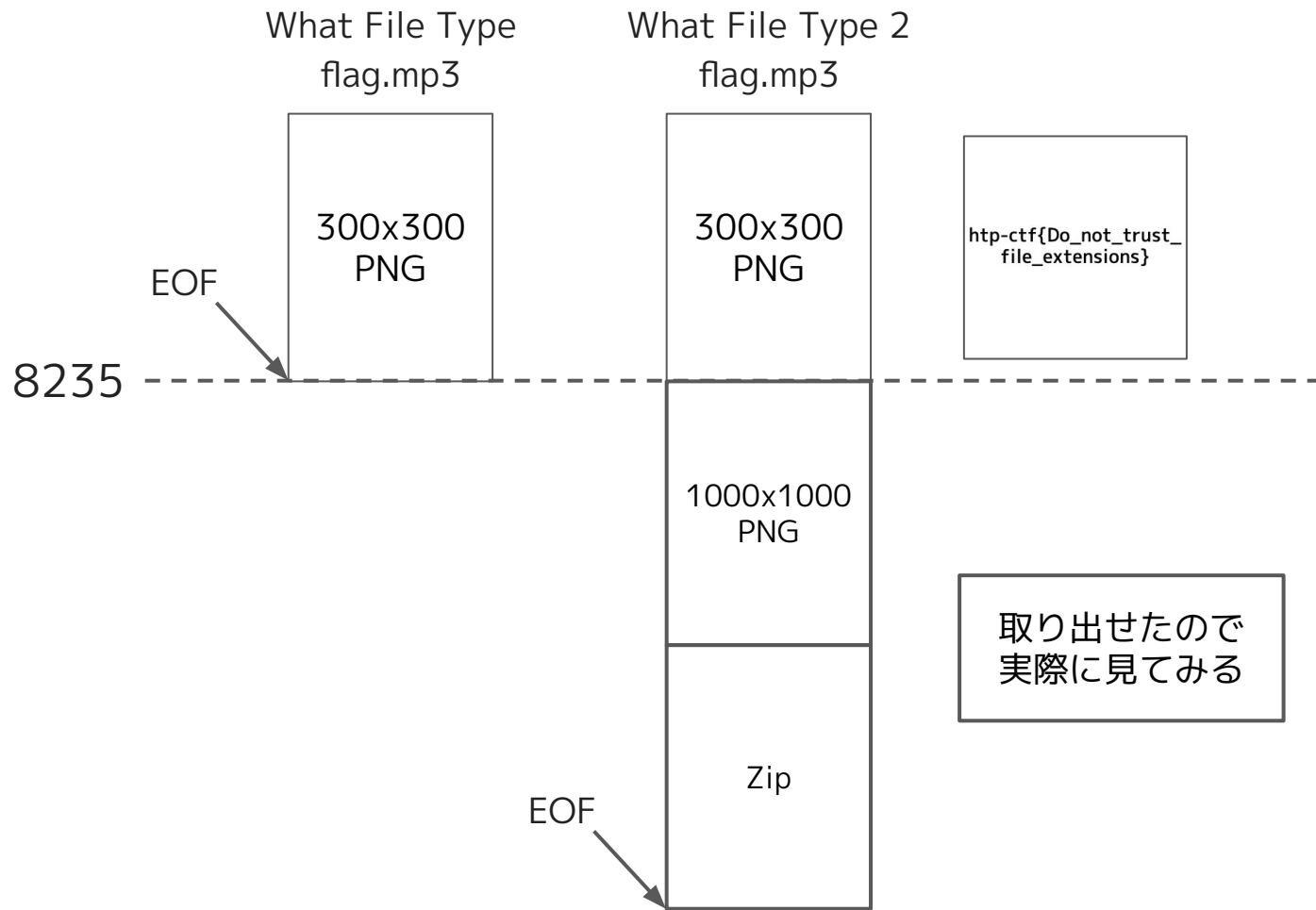
binwalk

```
$ binwalk -D=".*" flag.mp3
```

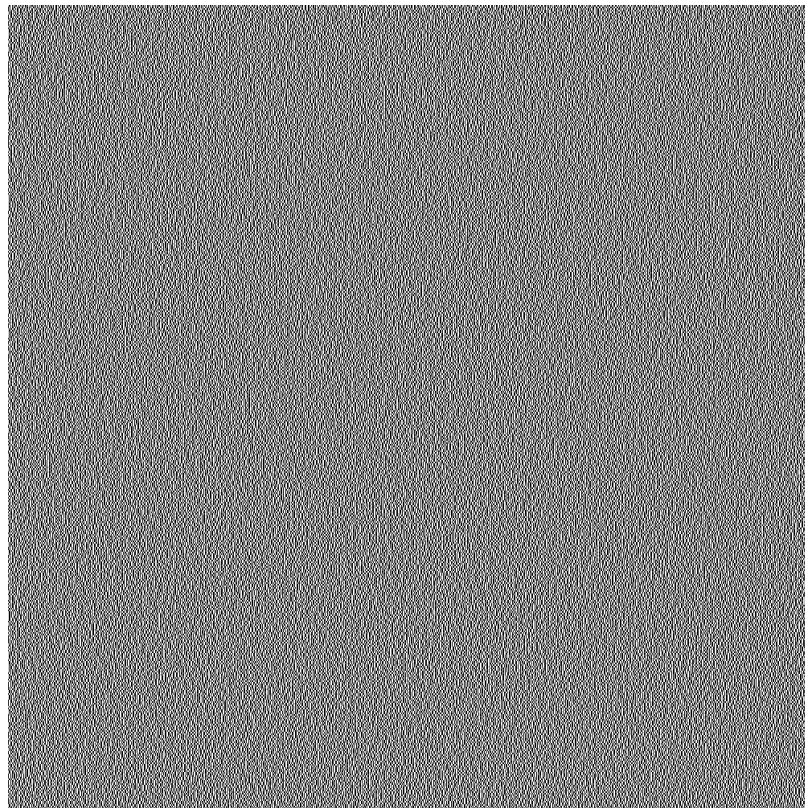
foremost

```
$ foremost flag.mp3
```

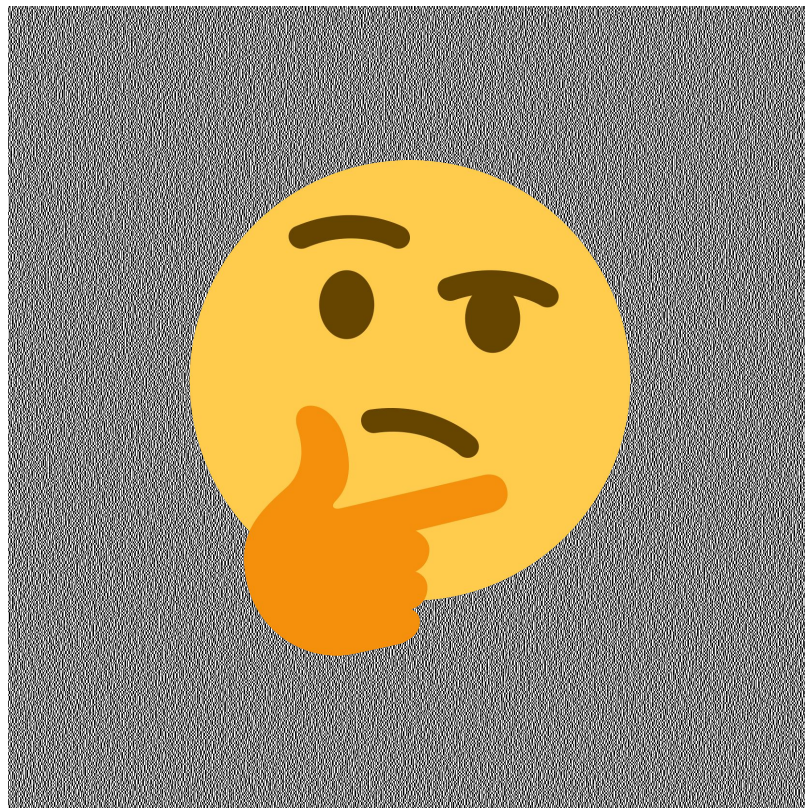
どちらも標準では入っていないので入れる必要がある



1000x1000 PNGの方



1000x1000 PNGの方

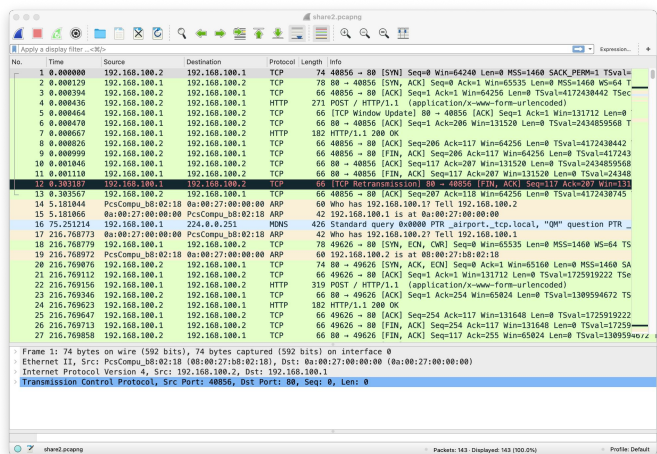


Zipファイルの方

とりあえず展開する（ダブルクリックでもunzipコマンドでも）

と、share2.pcapが出てくる

.pcapはWiresharkで開くことができる



やりとりを読む

中身はHTTPで数回のやりとりをしているのみ

その中でPNGをGETしている部分がある

122	237.123633	192.168.100.1	192.168.100.2	TCP	1517	80 → 40000 [ACK] Seq=117
123	237.123633	192.168.100.1	192.168.100.2	HTTP	93	HTTP/1.1 200 OK (PNG)
124	237.123740	192.168.100.2	192.168.100.1	TCP	66	40000 → 80 [ACK] Seq=117

Wiresharkの左上の File > Export Object > HTTP > share2.png > save

からやり取りしてる画像を保存できる

やりとりを読む

他の部分はこの問題のヒントとなる会話をしている

> hey, I can not read image file. What should I do ?

> 画像読めないんだけど、どうしたらいい？

< Oh sorry, Please access '/share2' endpoint.

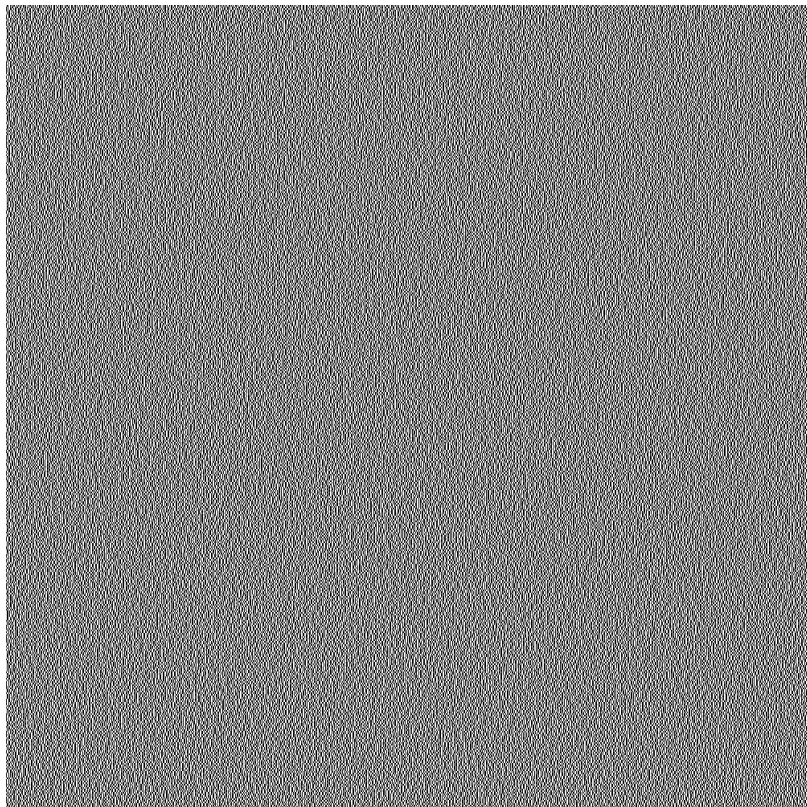
< You can get share2.png and you can view this picture.

< おっとごめん、/share2にアクセスしてshare2.pngをDLすれば見えるようになるよ

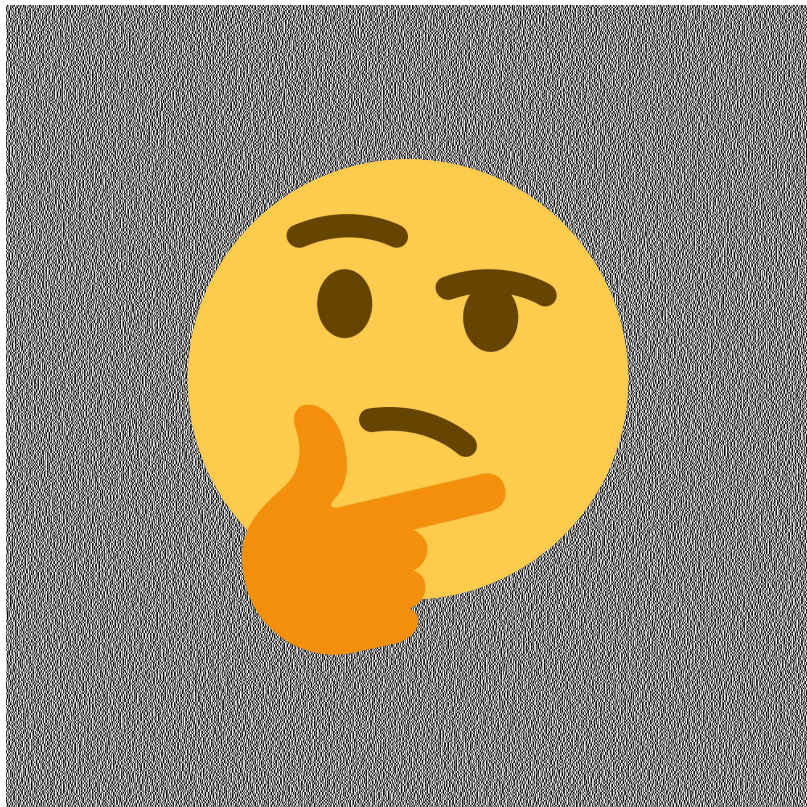
> THX, I viewed it.

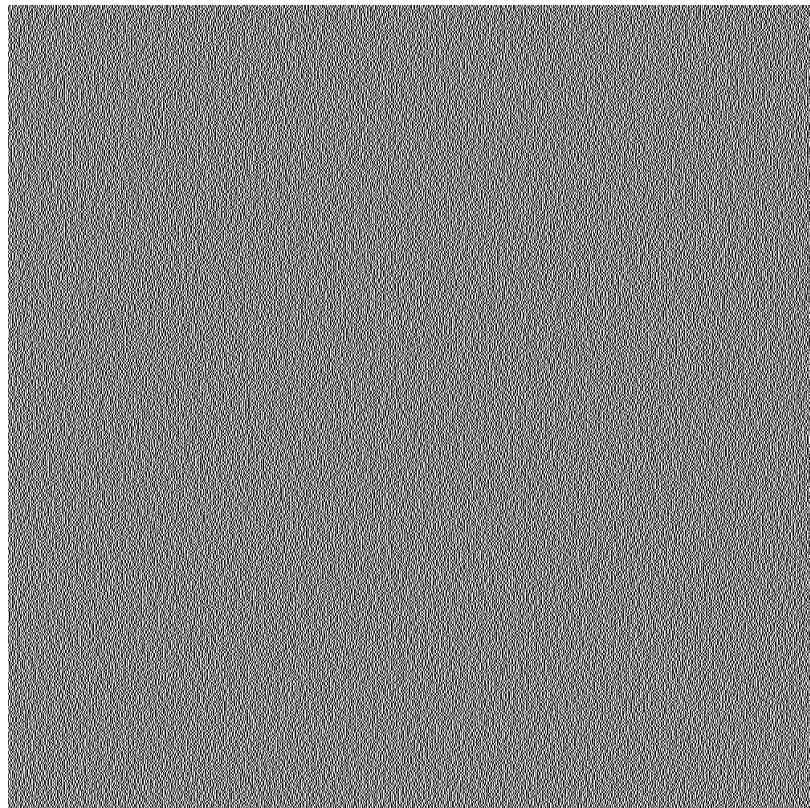
> ありがとう、見えたよ

share2.png

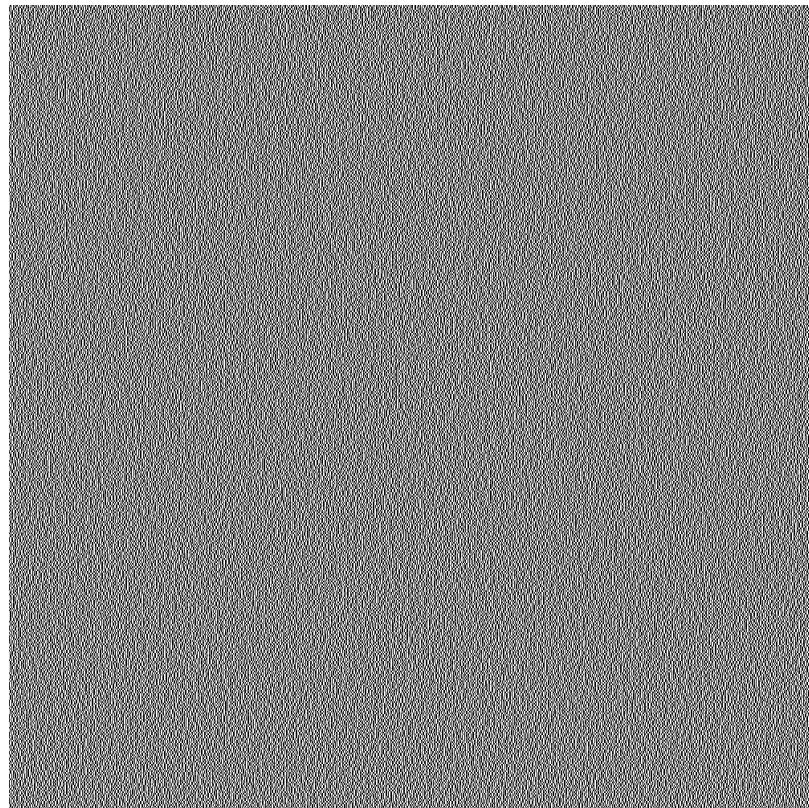


share2.png

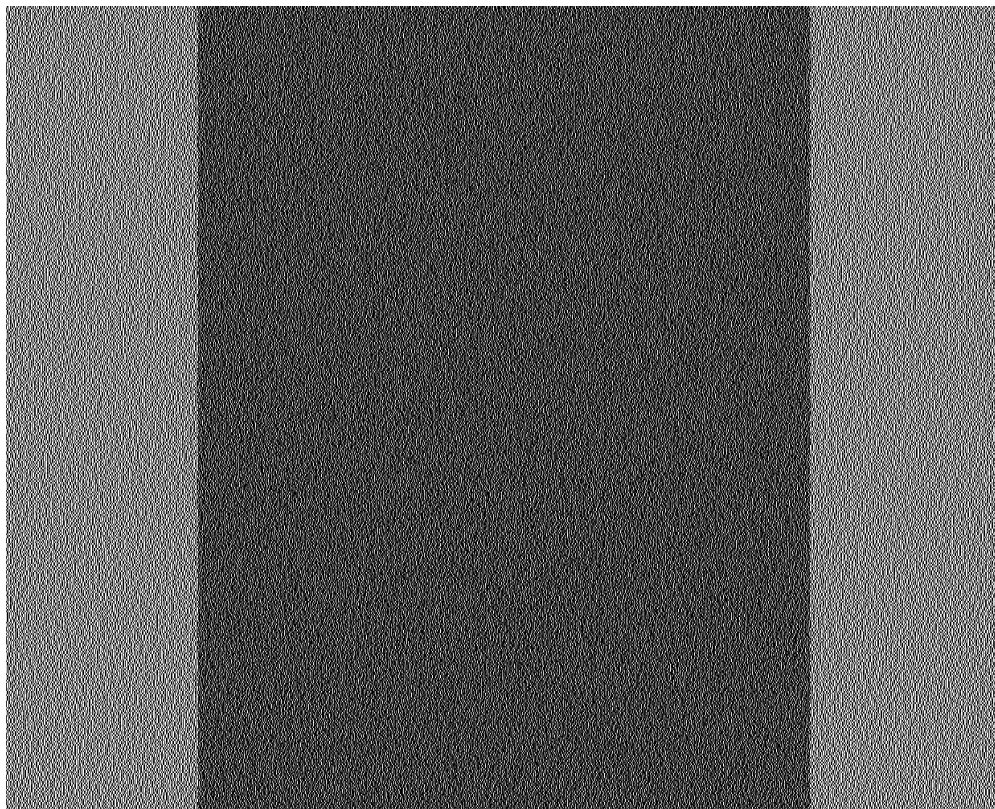




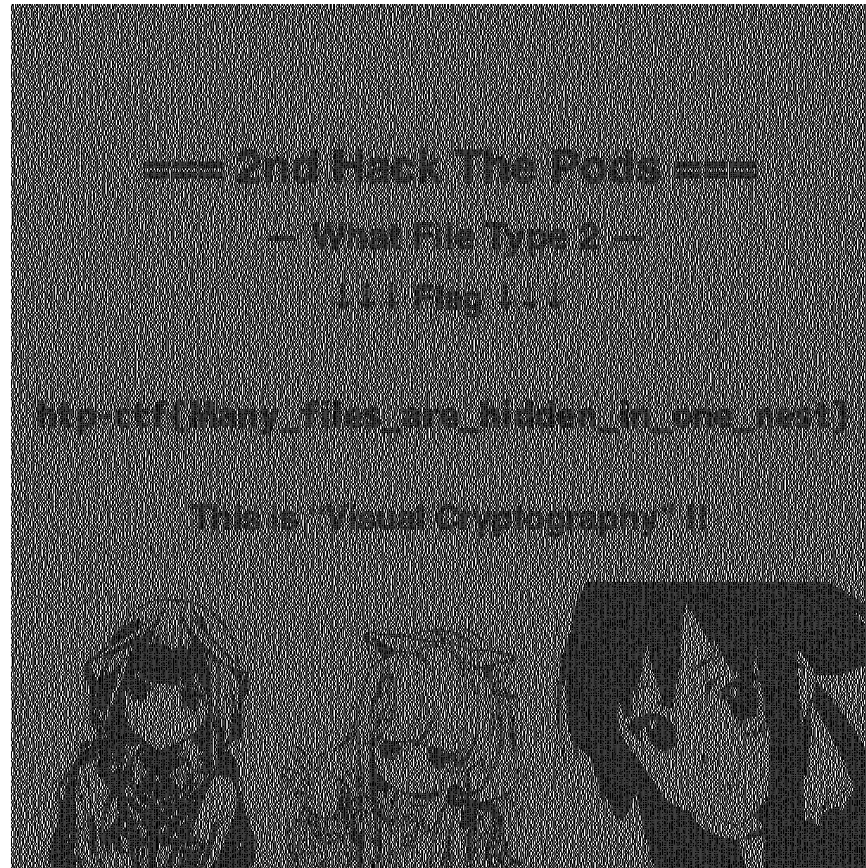
share1.png



share2.png



どちらも透過画像なので、重ねてみる



http-ctf{Many_files_are_hidden_in_one_nest}

=== 2nd Hack The Pods ===

— What File Type 2 —

↓↓↓ Flag ↓↓↓

http-ctf{Many_files_are_hidden_in_one_nest}

This is “Visual Cryptography” !!



http-ctf{Many_files_are_hidden_in_one_nest}

おきもち

決め打ちしようとする初見殺しされる問題

でも地道に調査すればまあわかる問題だと思う

最後の画像を重ねるやつは知らないといけないかも

正直楽しくなっちゃっただけなので、このCTFで出すべきじゃなかった感

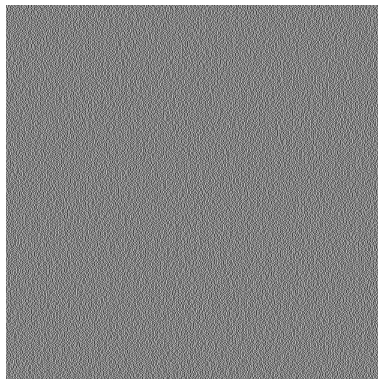
Visual Cryptography

視覚的な情報を暗号化する方法

今回は単純に 1 枚を 2 枚のshareに分割している

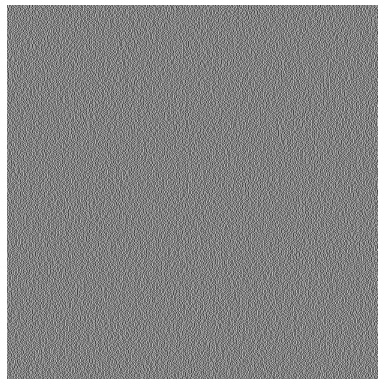


=



share1

+



share2



しっかり http-ctf と読むことができる

original
2 x 2 parts

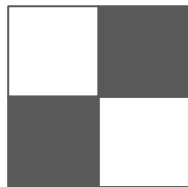
share1

share2

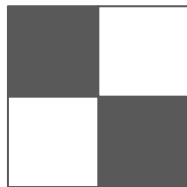
share1+2



乱数で分岐



+



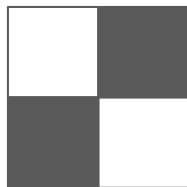
=



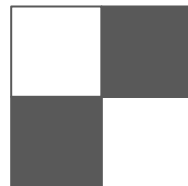
完全に黒



+



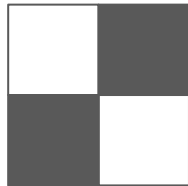
=



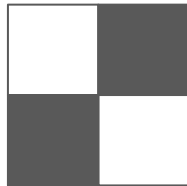
白に見える



乱数で分岐



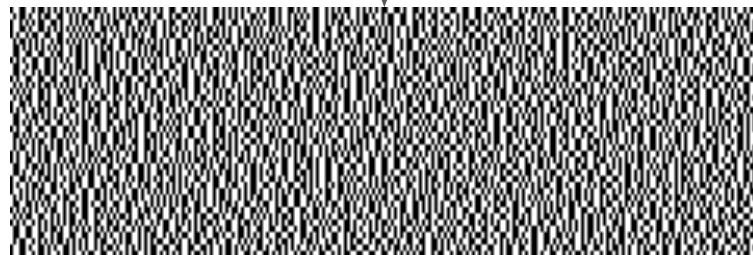
+



=



けいおん！



=== 2nd Hack The Pods ===

— What File Type 2 —

↓↓↓ Flag ↓↓↓

`http-ctf{Many_files_are_hidden_in_one_nest}`

This is “Visual Cryptography” !!



=== 2nd Hack The Pods ===

— What File Type 2 —

↓↓↓ Flag ↓↓↓

`http-ctf{Many_files_are_hidden_in_one_nest}`

This is “Visual Cryptography” !!

