

# CUTE\_TIGER

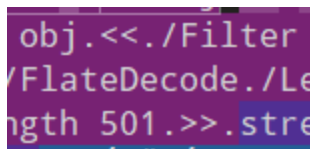
☰ 태그	포렌식
⚙ 상태	진행 중

## 문제풀이

### ▼ CUTE\_TIGER

<https://blog.forensicresearch.kr/4>

[참고자료] pdf 자료구조 (stream 뒤의 암호문 복호화)



```
obj.<<./Filter  
/FlateDecode./Le  
ngth 501.>>.stre
```

1. stream - end stream 사이의 암호문 복호화 시도.

```
import zlib
import sys
import binascii

def FlateDecode(data):
    return zlib.decompress(UP2UP3(data))

def UP2UP3(string):
    if sys.version_info[0] > 2:
        return bytes([ord(x) for x in string])

if __name__ == "__main__":
    data = ""
    result = FlateDecode(data)
    print(result)
```

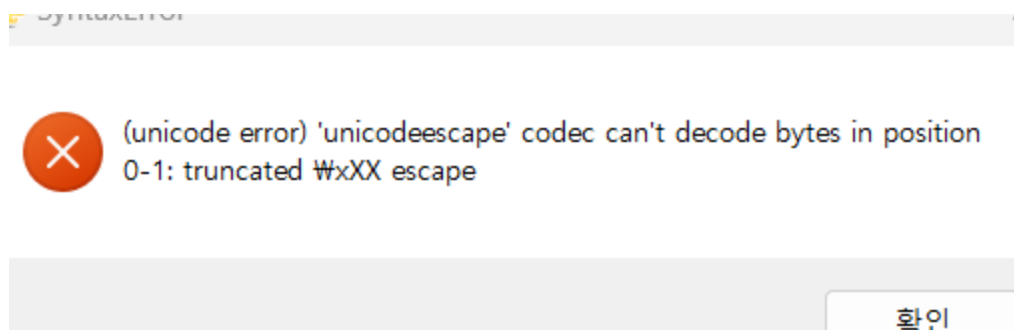
어떤 이유에서인지 hexa 코드가 붙여 넣기가 되지 않아 일일이 옮겨적지 않으면 안 되는 상황이 되었다. hxd를 경유해 복붙하려니 '올바르지 않은 문자열' 탓에 입력이 불가능하다는 에러 메시지가 출력되었다.

0A	78	9C	65	52	CB	6E	83	30	10	BC	FB	2B	7C	4C	0F	.xœRĒnf0.¼û+ L.
11	2F	03	8D	84	90	A2	44	95	38	F4	A1	D2	7E	00	D8	./...„.çD•8ô;ô~.Ø
0B	B5	54	8C	65	CC	81	BF	AF	F1	D2	B4	49	2C	81	E5	.µTŒeİ.¿¬ñÔ'I,.â
D9	99	9D	91	D7	C1	A9	3A	57	4A	5A	1A	BC	99	91	D7	Û™.‘×Á@:WJZ.¼™‘×
60	69	27	95	30	30	8D	B3	E1	40	5B	E8	A5	22	11	A3	`i'•00.³á@[è¥".£
42	72	BB	9D	FC	9F	0F	8D	26	81	13	D7	CB	64	61	A8	Br»..üŸ..&..×Ēda"
54	37	92	A2	A0	C1	BB	2B	4E	D6	2C	74	77	14	63	0B	T7'ç Á»+NÖ,tw.c.
0F	24	78	35	02	8C	54	3D	DD	7D	9E	6A	77	AE	67	AD	.\$x5.ŒT=Ÿ}žjw@g-
BF	61	00	65	69	48	CA	92	0A	E8	5C	A3	E7	46	BF	34	¿a.eiHĒ'.è\fcF¿4
03	D0	C0	CB	F6	95	70	75	69	97	BD	D3	FC	31	3E	16	.ĐÀĒö•pui-½Ōü1>.
0D	34	F6	E7	08	C3	F0	51	C0	A4	1B	0E	A6	51	3D	90	.4öç.ĀđQĀ▪...!Q=.
22	74	AB	A4	C5	93	5B	25	01	25	6E	EA	9B	AA	ED	F8	"t«▪Ā"[%.%nê,ªíø
57	63	3C	FB	D1	B1	C3	30	4E	3D	7B	C3	D9	2F	EB	D2	Wc<ûN±ĀŌN={ĀŬ/ěŌ
34	0A	3D	2D	4A	90	7D	2E	3D	98	21	98	FB	2D	49	10	4.-=J.}.=~!~û-I.
3C	22	78	46	30	F7	60	CC	50	87	66	2C	DA	CC	B0	7D	<"xF0÷`İP±f,Ŭİ°}
7C	9B	89	61	DF	0C	1B	32	14	65	57	09	93	BB	84	0C	>%aß..2.eW."»„.
7D	19	FA	66	E8	9B	62	EC	14	63	67	18	3B	4D	11	3C	}..úfè>bì.cg.;M.<
F8	2D	8F	AF	C2	DC	5D	50	7A	42	DA	E1	BF	FD	7A	B1	ø-.¬ĀŬ]PzBŬá¿ýz±
EB	FC	2F	53	E3	B3	31	6E	60	FE	91	F8	49	AD	33	92	ëü/Să³1n`p'øI-3'
0A	2F	FF	48	8F	7A	55	AD	DF	0F	AA	56	R7	87	0A		ïH>II-RªV.±

'stream' 내 hexa 코드 + 문자열.

```
File C:\Users\wromi\OneDrive\바탕 화면\내역\SWING 2023\1학기 여름
.py", line 6, in FlateDecode
    return zlib.decompress(UP2UP3(data))
zlib.error: Error -3 while decompressing data: unknown compression method
```

hexa값 입력 뒤 오류.



```

if __name__ == "__main__":
    data = "78 9C CD 55 CF 6B D0
    d = data.split(" ")
    s = ''.join(d)
    result = FlateDecode(s)
    print(result)

```

```

File "C:\Users\wonniwon\OneDrive\work\python\line 6, in FlateDecode
    return zlib.decompress(UP2UP3(data))
zlib.error: Error -3 while decompressing data: u

```

flatedecode(복호화 코드)에서 지속적 오류 발생.

pdf 파일 탐지 툴 peepdf를 다운로드 받아보자.

<https://eternal-todo.com/tools/peepdf-pdf-analysis-tool>

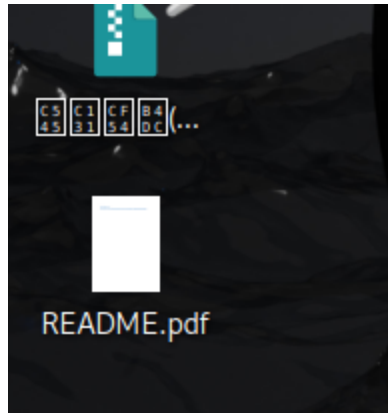
peepdf 다운로드 과정에서 문제 발생(usr/bin에 python파일 부재 문제 같은데 정확히는 모르겠다.)

pypdf 를 다운로드 받았으나 생각했던 툴이 아니라 uninstall했다.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF
FF	D8	FF	E0	00	10	4A	46	49	46	00	01	01	00	00	01	ÿÿà...JFIF.....
00	01	00	00	FF	DB	00	43	00	04	04	04	04	04	04	04	...ÿÛ.C.....
04	04	04	06	06	05	06	06	08	07	07	07	07	08	0C	09	.....
09	09	09	09	0C	13	0C	0E	0C	0C	0E	0C	13	11	14	10	.....
0F	10	14	11	1E	17	15	15	17	1E	22	1D	1B	1D	22	2A	....."..."*
25	25	2A	34	32	34	44	44	5C	FF	DB	00	43	01	04	04	%%*424DD\ÿÛ.C...
04	04	04	04	04	04	04	04	06	06	05	06	06	08	07	07	.....

pdf 속 링크에 있는 이미지를 010 editor로 분석해 봤으나 특이점은 없다.

+) 8.11 칼리 리눅스 다운로드. 리눅스 안에 pdfid라는 툴이 내장되어 있었음.



칼리 리눅스 시스템 안에서 포렌식 해보자.

pdfid -e(extra data) README.pdf

```
/JS 0
/JavaScript 0
/AA 0
/OpenAction 0
/AcroForm 0
/JS 0
```

우선 악성코드는 보이지 않음.

```
/Type0 1
/URI system 4
/w 3
```

그 외: URI 4개.

jpg 이미지 링크일 것 같긴 한데....일단 확인은 해 보자.

<https://mdin1.tistory.com/155>

URI 내용물 확인: pdf-parser 사용.

```

(kali@kali)-[~/3sctf]
$ pdf-parser --stats README.pdf
This program has not been tested with this version of Python (3.11.9)
Should you encounter problems, please use Python version 3.11.1
Comment: 4
XREF: 1
Trailer: 1
StartXref: 1
Indirect object: 31
Indirect objects with a stream: 11, 13, 27
  12: 9, 11, 12, 7, 13, 16, 19, 25, 27, 24, 4, 31
/Catalog 1: 1
/ExtGState 2: 17, 18
/Font 3: 8, 6, 14
/FontDescriptor 2: 10, 15
/OBJR 2: 23, 26
/Page 1: 5
/Pages 1: 2
/StructElem 6: 28, 20, 29, 30, 21, 22
/StructTreeRoot 1: 3
Search keywords:
/URI 2: 19, 25

```

파이썬 버전 3.11.1 설치해야 할 듯. 아니면 툴을 업데이트하거나.

어쨌든... URI가 있는 오브젝트: 19, 25.

두 개나 있었나?

```

/Subtype /Link
/Rect [69.75 738.21 525.55 756.87]
/BS
  <<
    /W 0
  >>
/F 4
/A
  <<
    /Type /Action
    /S /URI
    /URI (https://give-me-in-june.tistory.com/entry/%EA%B7%80%EC%97%AC%EC%9A%B4-%EB%B3%B4%ED%98%B8)
  >>
/StructParent 0
>>

```

먼저 19번 오브젝트의 URI.

```
Referencing:
<<
  /Subtype /Link
  /Rect [69.75 711.54 479.1 738.21]
  /BS
    <<
      /W 0
    >>
  /F 4
  /A
    <<
      /Type /Action
      /S /URI
      /URI (https://give-me-in-june.tistory.com/entry/%EA%B7%80%EC%97%AC%EC%9A%B4-%EB%B3%B4%ED%98%B8)
    >>
  /StructParent 1
>>
```

25번 오브젝트의 uri....똑같잖아ㅋㅋ;  
 왜 두 개로 카운트한 건지 이해가 안 간다.

## 권한이 없거나 존재하지 않는 페이지입니다.

궁금하신 사항은 [고객센터](#)로 문의해 주시기 바랍니다.

이전화면

? CTF가 끝나서 잠궜나?

<https://give-me-in-june.tistory.com/entry/귀여운-보호>

pdf에 달린 링크는 멀쩡히 작동한다.

음....

<https://hackingstudypad.tistory.com/132>

그럼 스테가노그래피로 접근해보자. jpg 파일을 위의 방법으로 카빙해보겠다.

```
valid_lft forever preferred_lft forever
(kali㉿kali)-[~/3sctf]
$ steghide extract -sf image.jpg
Enter passphrase: 
```

리눅스 툴 steghide를 적용하자 다음과 같이 passphrase(비번)을 입력하라는 문구가 나왔다.

이럴 때 비번을 알아내는 도구가 stegcrack이라길래 추가설치.

```
Copyright (C) 2024 - Luke Paris (Paradox15)
valid_lft forever preferred_lft forever
StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.
StegSeek can be found at: https://github.com/RickdeJager/stegseek
Error: Wordlist '/usr/share/wordlists/rockyou.txt' does not exist!
(kali㉿kali)-[~/3sctf]
$ 
```

땀....stegseek이라는 새 툴로 업데이트해서 stegcracker를 더 이상 지원하지 않는다고 한다.

```
valid_lft forever preferred_lft forever
(kali㉿kali)-[~/3sctf]
$ stegseek --seed image.jpg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek
[i] Found (possible) seed: "e9dd5dc7"
Plain size: 63.0 Byte(s) (compressed)
Encryption Algorithm: rijndael-128
Encryption Mode: cbc
(kali㉿kali)-[~/3sctf]
$ 
```

steghide로 숨긴 무언가로 의심되는 것 발견. cbc로 암호화되어있다는 모양이다.

```
(kali@kali)-[~/3sctf]
$ stegseek image.jpg /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 99.54% (132.8 MB)
[!] error: Could not find a valid passphrase.
```

stegseek에서 기본 제공해주는 암호문 파일 'rockyou.txt'를 이용해 암호 추출 시작. 도중에 오류가 발생했다.

```
(kali@kali)-[~/3sctf]
$ stegseek image.jpg /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 99.74% (133.1 MB)
[!] error: Could not find a valid passphrase.
```

```
(kali@kali)-[~/CTF/3Sctf]
$ stegseek image.jpg /usr/share/wordlists/rockyou.txt
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek

[i] Progress: 99.54% (132.8 MB)
[!] error: Could not find a valid passphrase.
```

재시도 했으나 여전히 100%를 채우지 못 하고 실패. passphrase를 찾을 수 없다는 것 같은데... 보통은 100% 전수 조사한 뒤 passphrase 존재여부를 알려 주지 않나?

```
(kali@kali)-[~/3sctf]
$ stegcracker image.jpg /usr/share/wordlists/rockyou.txt
StegCracker 2.1.0 - (https://github.com/Paradoxix/StegCracker)
Copyright (c) 2024 - Luke Paris (Paradoxix)

StegCracker has been retired following the release of StegSeek, which
will blast through the rockyou.txt wordlist within 1.9 second as opposed
to StegCracker which takes ~5 hours.

StegSeek can be found at: https://github.com/RickdeJager/stegseek

Counting lines in wordlist..
Attacking file 'image.jpg' with wordlist '/usr/share/wordlists/rockyou.txt'..
2368/14344392 (0.02%) Attempted: boobiesha
```



구버전인 'stagcracker'로도 시도는 해보자.

작동방식을 보니 사람들이 흔히 사용하는 비밀번호를 주먹구구식으로 일일이 전부 대입해 보나봄. 14344392개를 입력하면 웬만하면 하나쯤은 얻어걸리겠거니 싶기도 하다. 다만 난수에 가까운 비밀번호를 사용하는 경우에는 무의미하지 않을까? (내가 이 경우다.)

```
(kali㉿kali)-[~/3sctf]
$ stegseek --seed image.jpg
StegSeek 0.6 - https://github.com/RickdeJager/StegSeek
[i] Found (possible) seed: "e9dd5dc7"
Plain size: 63.0 Byte(s) (compressed)
Encryption Algorithm: rijndael-128
Encryption Mode: cbc
(kali㉿kali)-[~/3sctf]
$
```

정 안 되면 아까 찾은 이걸 cbc 모드로 복호화해보자. 플래그가 되기에는 너무 짧은 것 같긴 한데.... 아무튼 밀쳐야 본전이니까.

구버전이 열심히 돌리는 동안 새로운 툴 설치.

stegoveritas.

```
(kali㉿kali)-[~/bin]
$ /home/kali/.local/bin/stegoveritas image.jpg
ERROR:StegoVeritas:Cannot find file "/home/kali/.local/bin/image.jpg"
(kali㉿kali)-[~/bin]
$ /home/kali/.local/bin/stegoveritas image.jpg
zsh: exec format error: /home/kali/.local/bin/stegoveritas
(kali㉿kali)-[~/bin]
$
```

?

```

(kali㉿kali)-[~/bin]
$ ./stegoveritas
zsh: exec format error: ./stegoveritas

(kali㉿kali)-[~/bin]
$

```

아니 그냥 stegoveritas [파일명]하면 실행이 안 됨.

뭘 어쩌자는 건지....

```

(kali㉿kali)-[~/bin]
$ sudo stegoveritas image.jpg
sudo: stegoveritas: command not found

(kali㉿kali)-[~/bin]
$ ./stegoveritas image.jpg
zsh: exec format error: ./stegoveritas

(kali㉿kali)-[~/bin]
$

```

일단 stegoveritas....는 관두기로 했다.

그냥 아까 추출해낸 e9dd5dc7 이걸 한번 복호화해보자. 플래그 치고는 많이 짧은 한데...  
혹시 모르니까.

```

Output
éŸ]ç

```

복호화 결과.

## 참고자료

## 6. StegSeek:

StegSeek is a lightning fast steghide cracker that can be used to extract hidden data from files.

StegSeek can also be used to extract steghide metadata without a password, which can be used to test whether a file contains steghide data.

StegSeek uses a wordlist that you provide to crack hidden data in the file:

```
stegseek <filename> <wordlist>
```

Check whether a file contains steghide data without a password:

```
stegseek -- seed <filename>
```

Source: <https://github.com/RickdeJager/stegseek>

Stegseek 사용법

/usr/share/wordlists/rockyou.txt