

key=KoJexaU5P

- 然后发现还是这样的，然后继续go了好几把，仍然无止境，这个时候就想到了the fastest speed。。。估计要写脚本跑了，不然再怎么写也没脚本快啊，应该是设计的提交时间的限制。。。接下来去完成py脚本。

py脚本

```
import requests
import base64
'''-----'''
url = "http://ctf5.shiyanbar.com/web/10/10.php"
key = requests.get(url)
a = key.headers['FLAG']
print(a)
b = base64.b64decode(a).decode().split(':')[1]
print(b)
'''-----'''
value={'key':b}
post = requests.post(url,data=value)
#value = "key="+b
#headers = {'Content-Type': 'application/x-www-form-urlencoded'}
#post = requests.post(url,data=value,headers=headers)
print(post.text)
print(post.headers)
```

- 没想到还是把自己给坑了，原来都喜欢把post数据以字符串进行提交，然后会加上contet-Type的头部的，这次看抓包过程中没有，就没加了，一直出不了效果，然后换成了字典的形式马上就好了，也不需要头部，服了。。。又踩坑了。
- 开wireshark抓包，分别做了三次测试，用字典提交数据为第一次，第二次就是字符串，第三次是字符串加表单方式，结果为第一次和第三次相同，如果字符串没有头部去定义提交表单方式则结果不同

No.	Time	Source	Destination	Protocol	Length	Info
68	7.084460	192.168.2.141	106.2.25.10	HTTP	216	GET /web/10/10.php HTTP/1.1
70	7.109921	106.2.25.10	192.168.2.141	HTTP	518	HTTP/1.1 200 OK (text/html)
78	7.147974	192.168.2.141	106.2.25.10	HTTP	67	POST /web/10/10.php HTTP/1.1 (application/x-www-
81	7.174151	106.2.25.10	192.168.2.141	HTTP	275	HTTP/1.1 200 OK (text/html)
30...	23.873338	192.168.2.141	106.2.25.10	HTTP	216	GET /web/10/10.php HTTP/1.1
30	23.898451	106.2.25.10	192.168.2.141	HTTP	518	HTTP/1.1 200 OK (text/html)
30...	23.923590	192.168.2.141	106.2.25.10	HTTP	67	POST /web/10/10.php HTTP/1.1
30...	23.948225	106.2.25.10	192.168.2.141	HTTP	518	HTTP/1.1 200 OK (text/html)
37...	158.928497	192.168.2.141	106.2.25.10	HTTP	216	GET /web/10/10.php HTTP/1.1
37...	158.954772	106.2.25.10	192.168.2.141	HTTP	518	HTTP/1.1 200 OK (text/html)
37...	158.982668	192.168.2.141	106.2.25.10	HTTP	67	POST /web/10/10.php HTTP/1.1 (application/x-www-
37...	159.008248	106.2.25.10	192.168.2.141	HTTP	275	HTTP/1.1 200 OK (text/html)

- 对比报文的详细内容。

File Data: 13 bytes

HTML Form URL Encoded: application/x-www-form-urlencoded

Form item: "key" = "qec4uobVg"

Key: key

Value: qec4uobVg

0000 fc 7c 02 16 05 06 a0 c5 89 a4

0010 00 35 68 b2 40 00 80 06 4b cf

0020 19 0a ca 13 00 50 79 50 54 e4

0030 02 00 87 51 00 00 6b 65 79 3d

0040 62 56 67

Wireshark · Packet 3068 · Wi-Fi

Connection: keep-alive\r\n

> Content-Length: 13\r\n

\r\n

[Full request URI: http://ctf5.shiyanbar.com/web/10/10.php

[HTTP request 1/1]

[Response in frame: 3071]

File Data: 13 bytes

> Data (13 bytes)

Data: 6b65793d61714e547252644a33

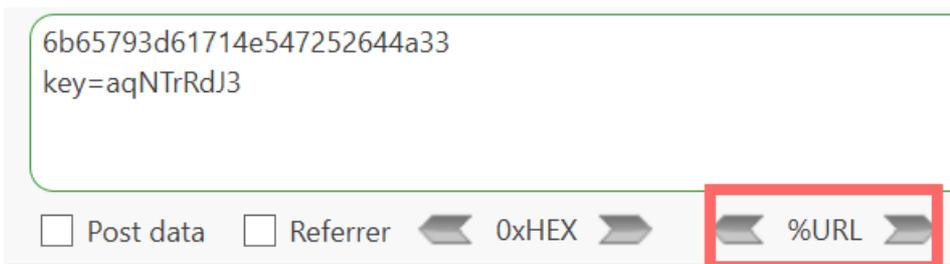
Data: 6b65793d61714e547252644a33
[Length: 13]

- 将报文进行url解密后发现，格式是正确的。。。

▼ Data (13 bytes)

Data: 6b65793d61714e547252644a33
[Length: 13]

0010	70 68 70 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f	php HTTP /1.1..Ho
0020	73 74 3a 20 63 74 66 35 2e 73 68 69 79 61 6e 62	st: ctf5 .shiyanb
0030	61 72 2e 63 6f 6d 0d 0a 55 73 65 72 2d 41 67 65	ar.com.. User-Age
0040	6e 74 3a 20 70 79 74 68 6f 6e 2d 72 65 71 75 65	nt: pyth on-reque
0050	73 74 73 2f 32 2e 32 32 2e 30 0d 0a 41 63 63 65	sts/2.22 .0..Acce
0060	70 74 2d 45 6e 63 6f 64 69 6e 67 3a 20 67 7a 69	pt-Encod ing: gzi
0070	70 2c 20 64 65 66 6c 61 74 65 0d 0a 41 63 63 65	p, defla te..Acce
0080	70 74 3a 20 2a 2f 2a 0d 0a 43 6f 6e 6e 65 63 74	pt: */*. .Connect
0090	69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d	ion: kee p-alive.
00a0	0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a	.Content -Length:
00b0	20 31 33 0d 0a 0d 0a 6b 65 79 3d 61 71 4e 54 72	13....k ey=aqNTr
00c0	52 64 4a 33	RdJ3



- 经过测试后，发现如果不定义头部中的提交表单方式默认会以类似“multipart/form-data”这种方式进行提交表单，这种一般是在上传文件时才会用到，我感觉问题在于获取表单数据的函数上，可能获取不到form-data这种方式提交的数据表，有机会再去证实了。

总结

- 这问题碰到过，但是解决了就解决了，这次又踩了同一个坑，不过这次理解得更透彻了一些，还没完全理解，差一次证实，需要去写一个php尝试一下。
- 好好的web题变成了python题，又从python题到了web题。。。无语~~