【攻防世界】十九 --- fakebook

原创

通地塔 于 2020-12-29 18:28:47 发布 102 收藏

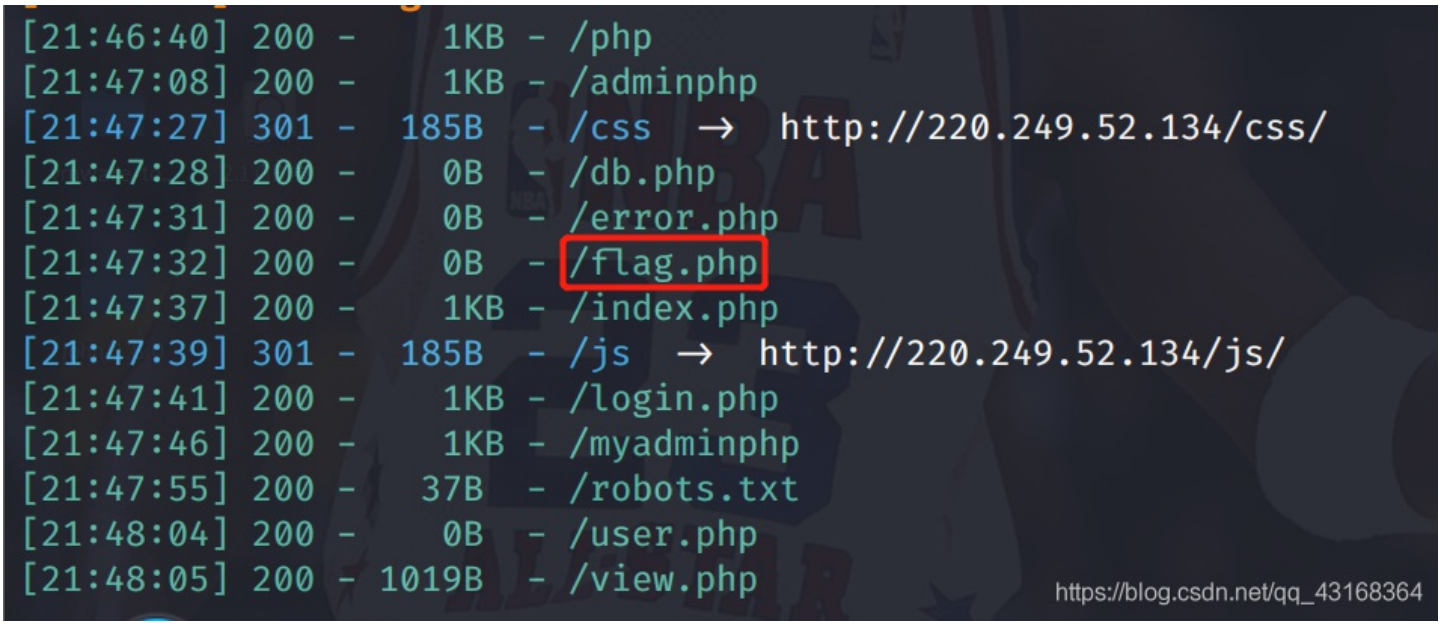
分类专栏： 攻防世界 文章标签： ctf 网络安全

本文链接：https://blog.csdn.net/qq_43168364/article/details/111240560

题目 — fakebook

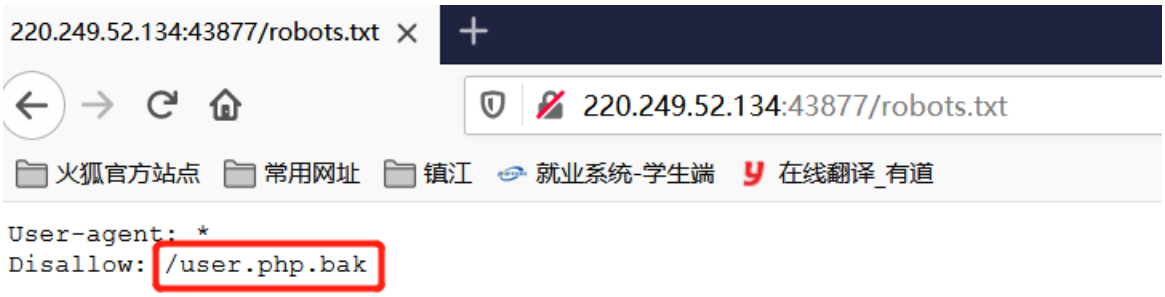
一、writeup

使用dirsearch扫描目录

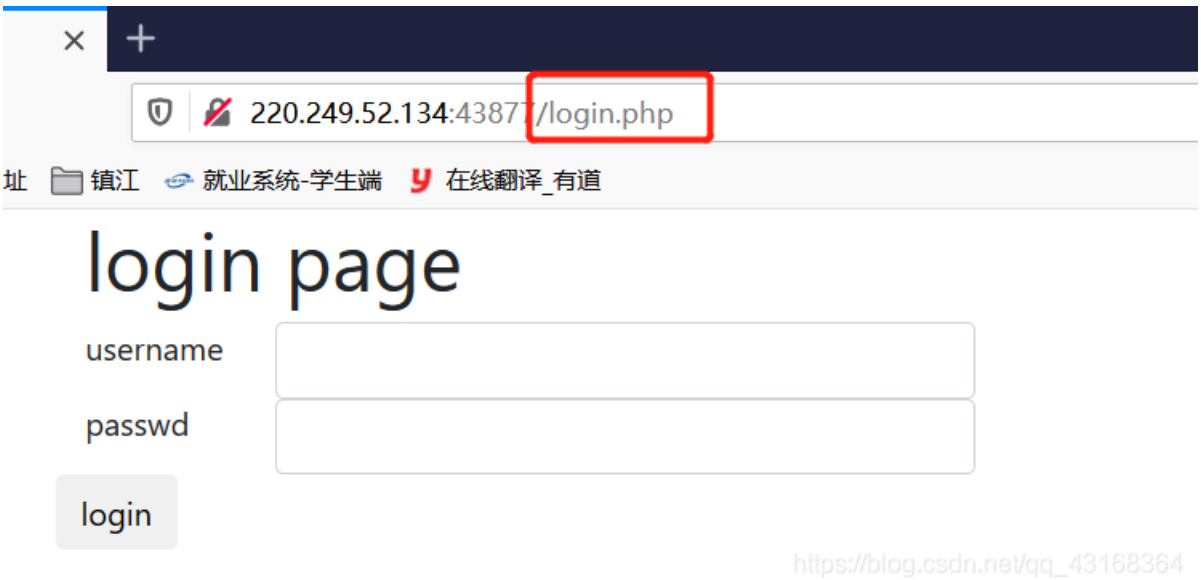


flag应该在`flag.php`文件中

访问`robots.txt`文件，得到了一个目录

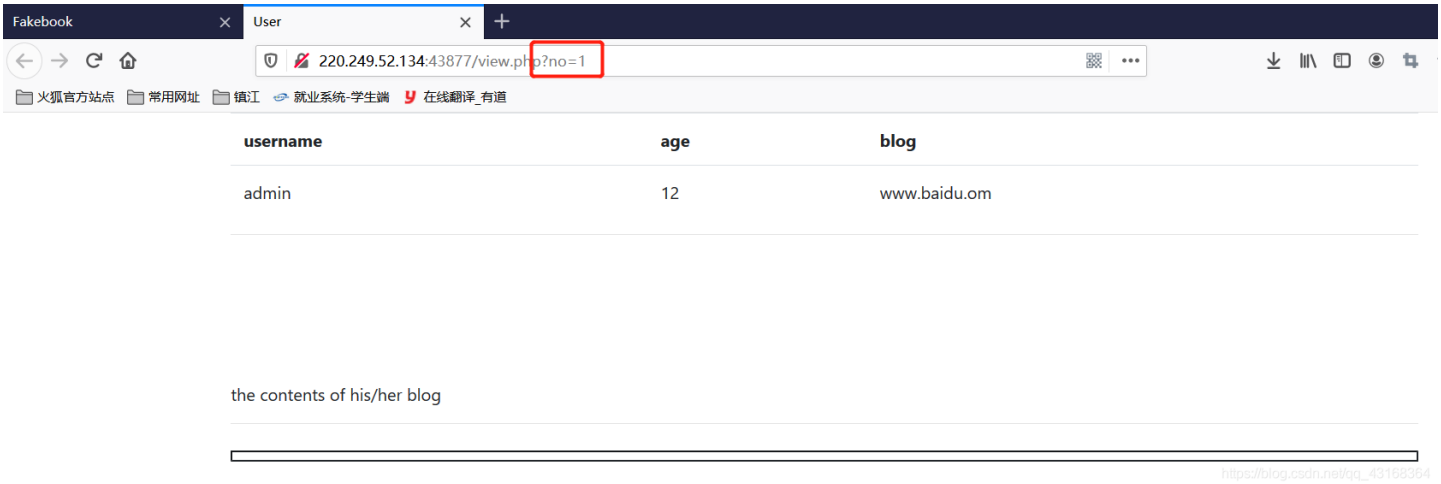


下来了一个文件，`user.php.bak`。该文件先丢在这里等下再审计，主页没有任何发现，前端代码中没有提示，login.php 也没有找到可以利用的点(可以尝试：sql，xss，逻辑漏洞，写入日志文件，爆破)

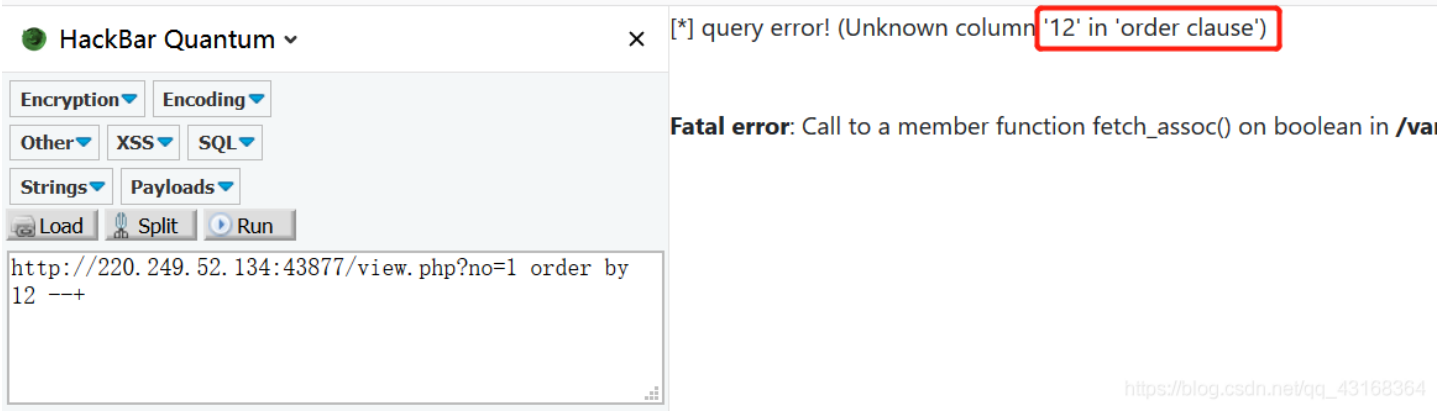


来到`join.php`页面注册了一个admin用户

从首页可以访问到admin用户的页面，发现存在一个查询字符串，其有sql注入漏洞，注入类型是数字型

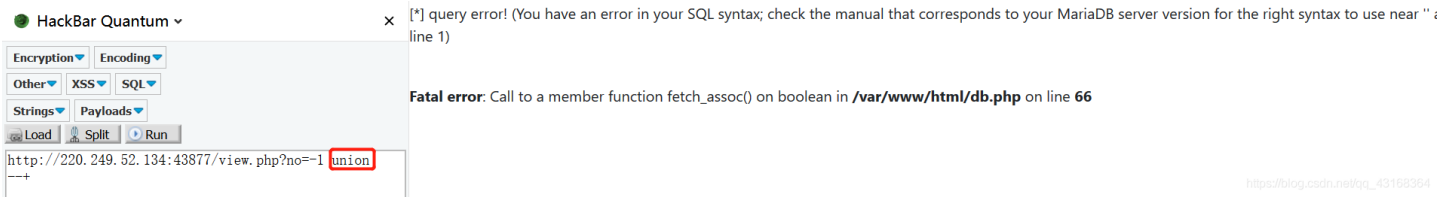


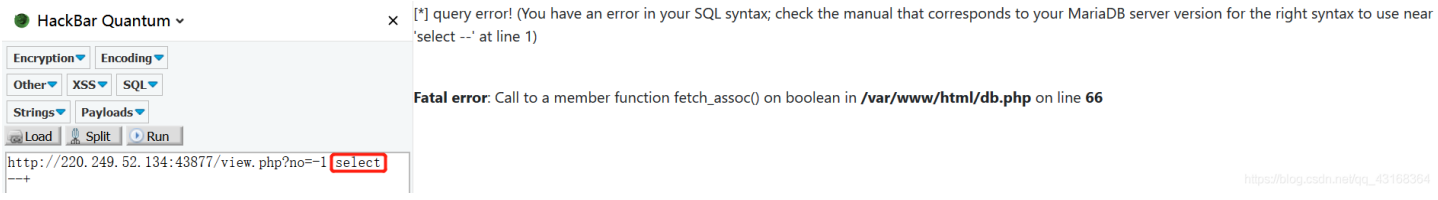
字段数：4，这里的报错给出了网站的根路径：**/var/www/html**



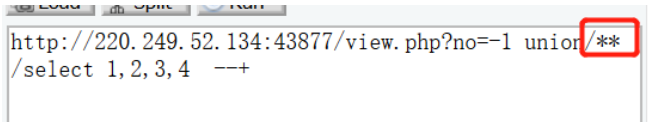
判断显示位时被过滤了

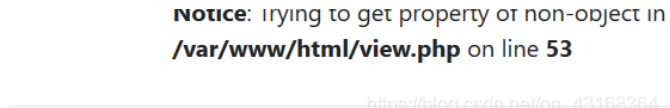
接下来判断其过滤方式，单独输入 `union` 和 `select` 都没有被过滤，只是提示语法错误





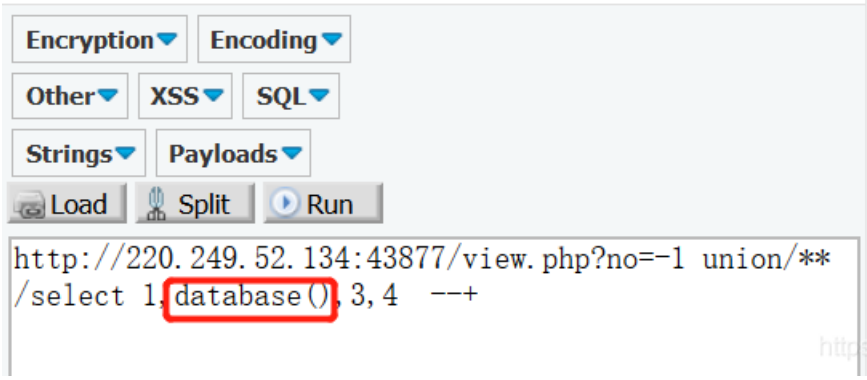
因此可以判断其过滤方式是：**基于组合的检测**，而非基于**关键字的检测**，这里使用 `/**/` 即可绕过检测

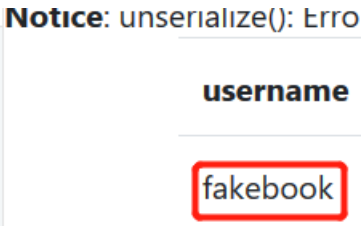




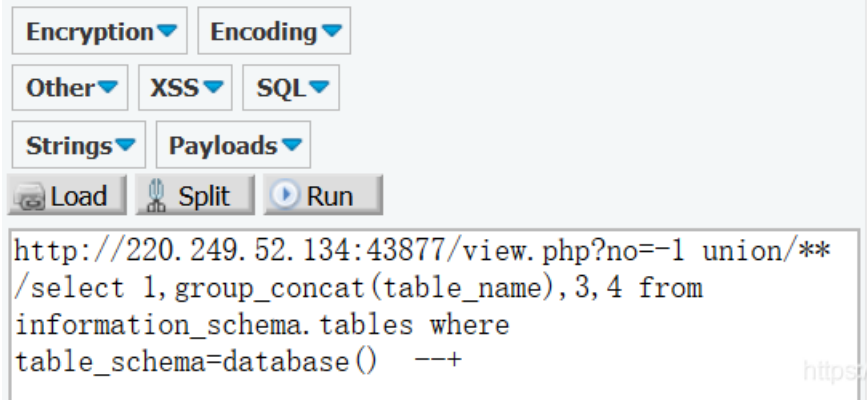
显示位：2

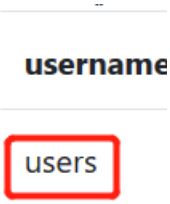
库名：fakebook



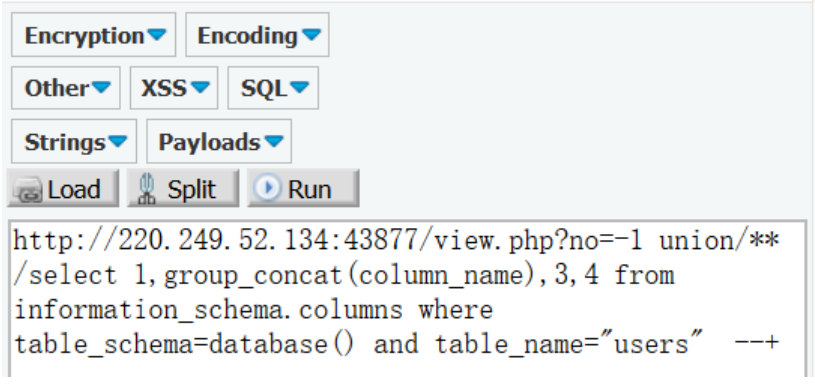


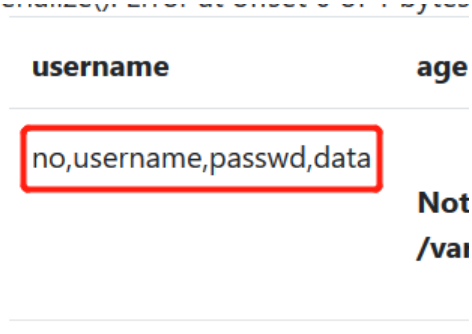
表名：users



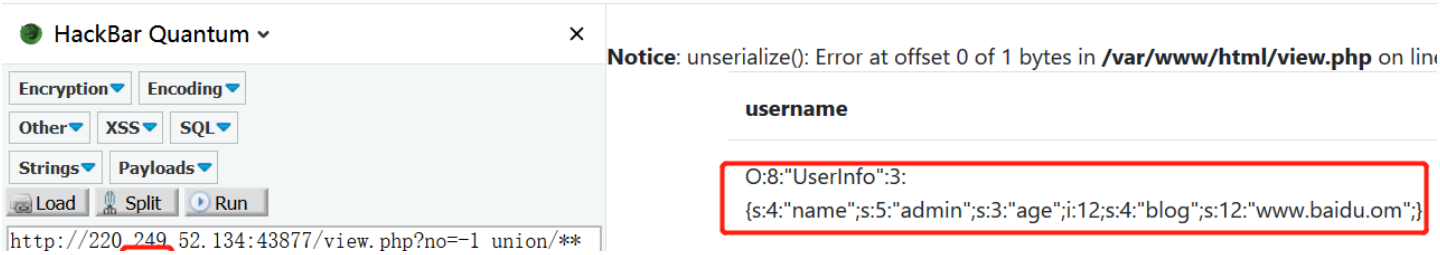


字段：no,username,passwd,**data**



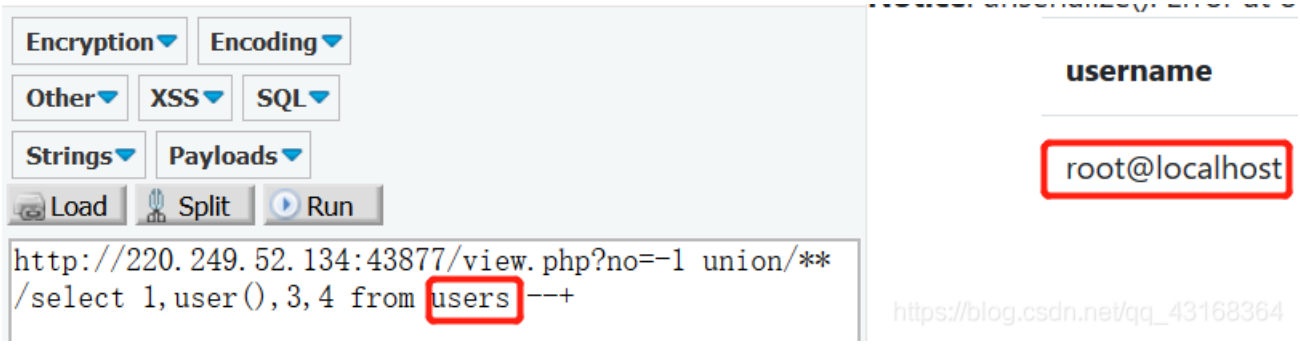


数据：no username passwd字段中的数据没有用，看看data字段中的数据

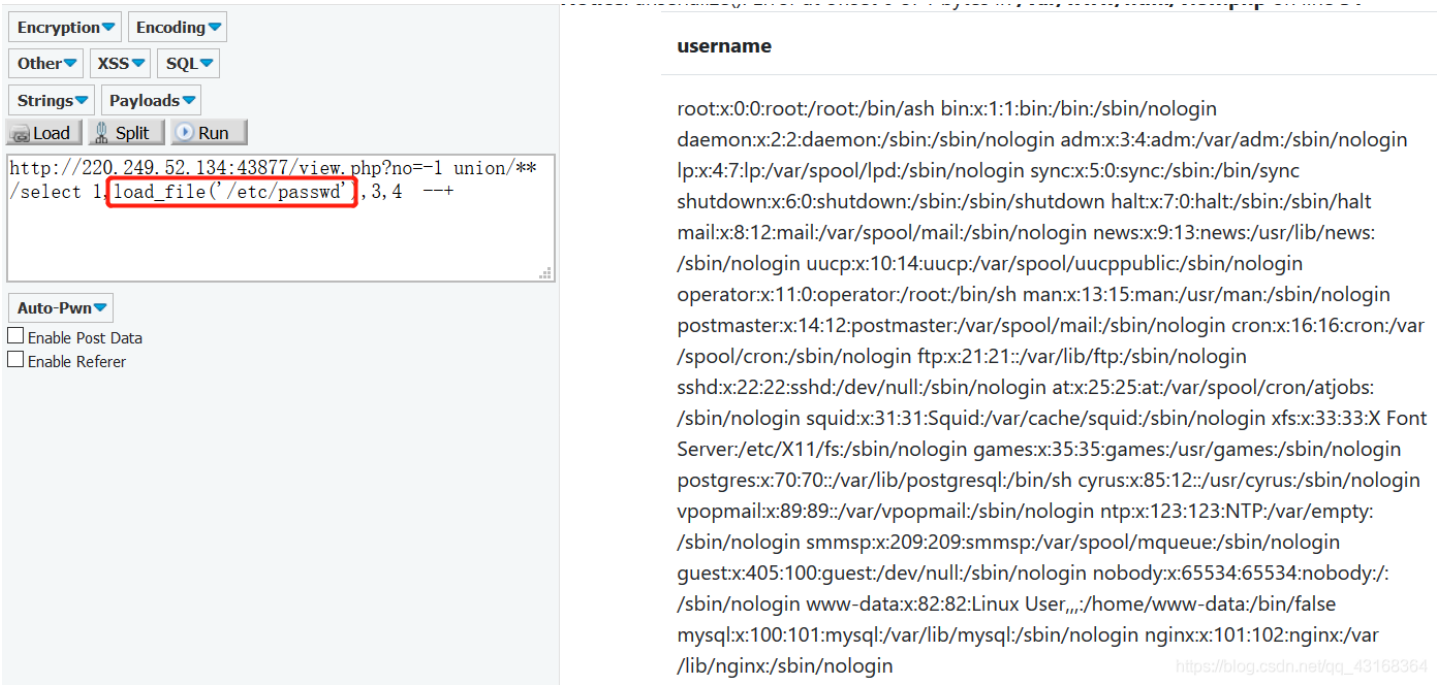


```
/select 1,data,3,4 from users --+
```

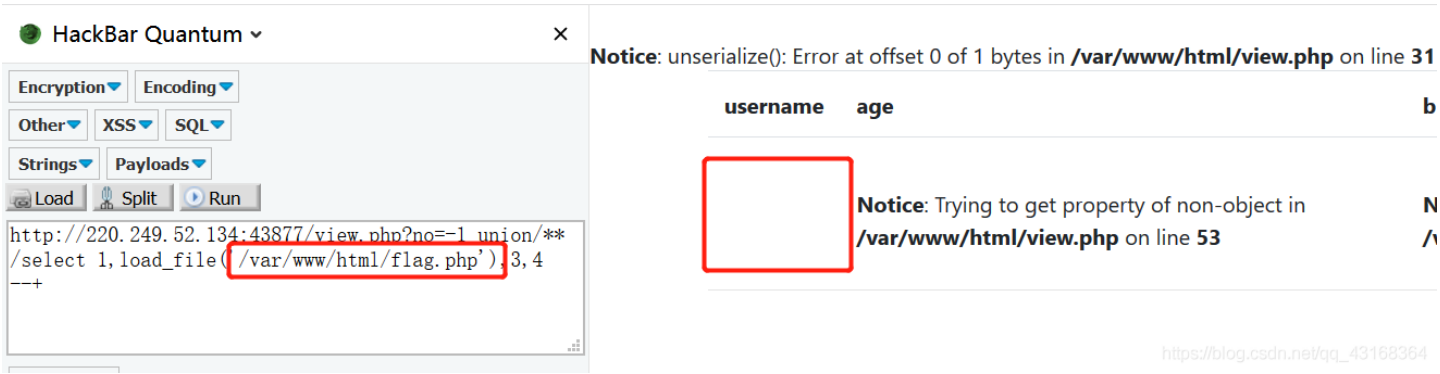
是一段序列化之后的内容，应该和`user.php.bak`中的代码有关，这里先放下。
来看看当前数据库的用户



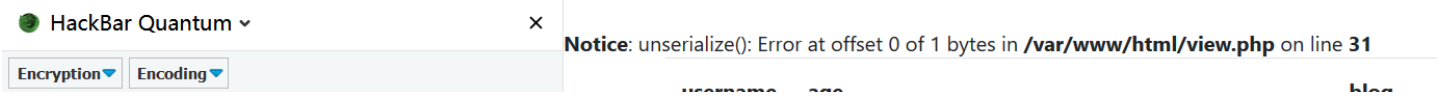
是root用户，使用 **load_file** 读文件成功，**into outfile** 写文件失败

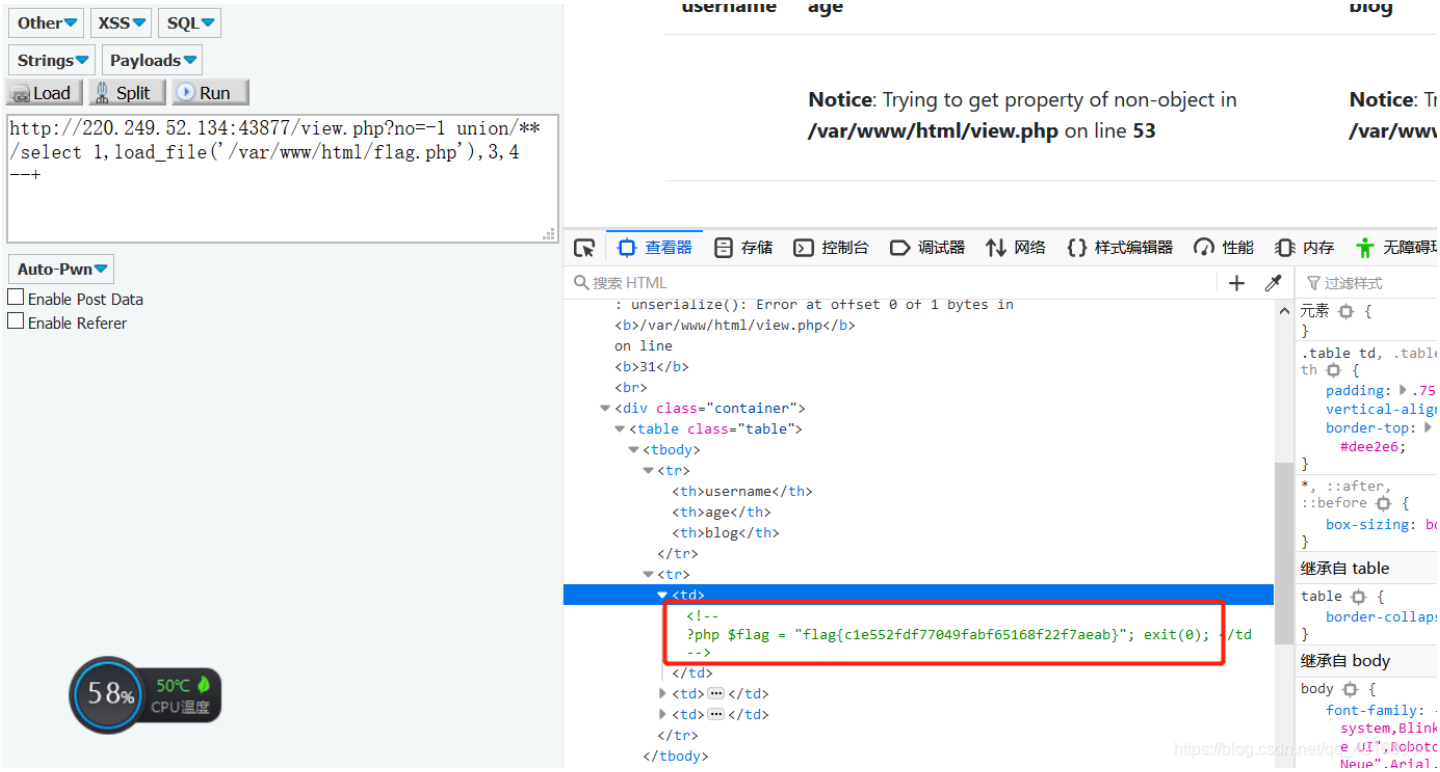


尝试读取：`/var/www/html/flag.php` 文件



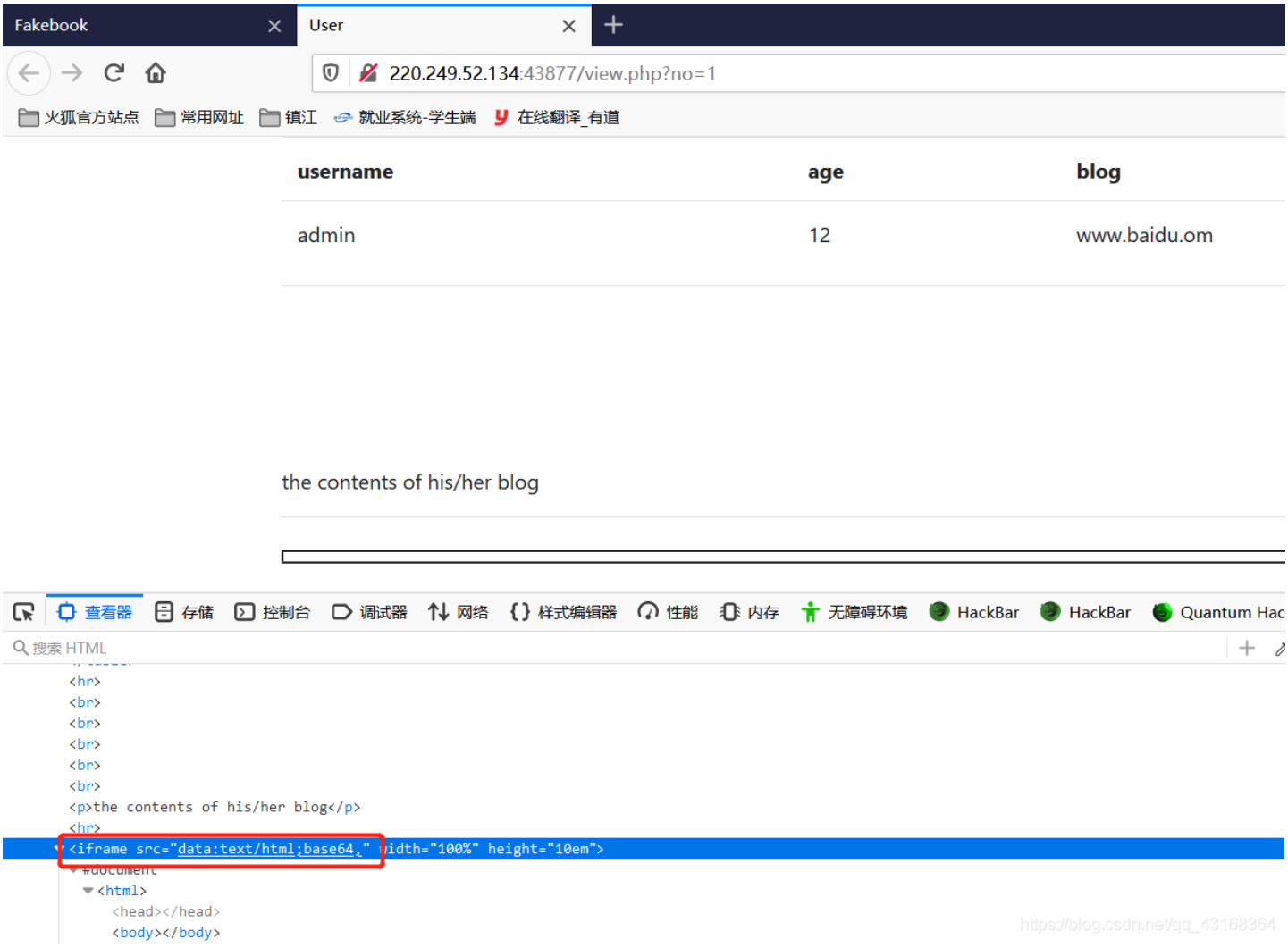
页面中没有显示，控制台查看前端代码





得到flag。这种方法应该是这道题的作弊解法吧。下来看看正确解法

admin用户的页面中的前端代码中有一个 iframe 标签很可疑



`src = data:text/html;base64,` 标签的内容通过 `data伪协议` 取得，注意一下这个标签。
然后需要关注的有两点

- 一、前面得到的user.php.bak文件
- 二、数据库的data字段的内容

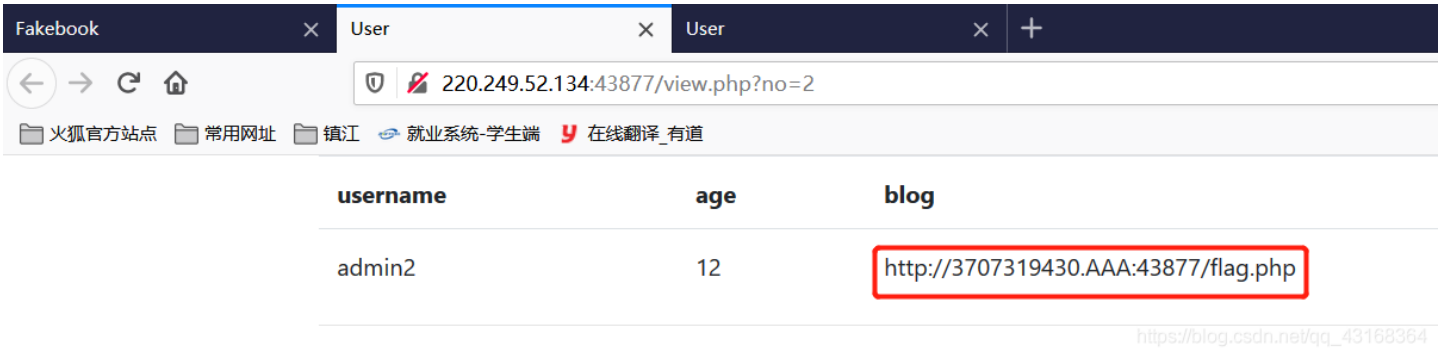
接下来对`user.php.bak`审计

```php
<?php


class UserInfo
{
 // 这三个变量的内容会显示到 view.php 页面中
    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)
    {
     // 初始化一个新的会话，返回一个cURL句柄，供curl_setopt(), curl_exec()和curl_close() 函数使用
        $ch = curl_init();

        // 设置curl传输选项
        curl_setopt($ch, CURLOPT_URL, $url); // 设置url
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1); // 获取到的信息以文件流形式返回，不直接返回

        // 执行一个句柄
        $output = curl_exec($ch);

        // 获取一个cURL连接资源句柄的信息
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE); // 得到响应状态码
        if($httpCode == 404) {
            return 404;
        }

        curl_close($ch);

        return $output;
    }

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

    public function isValidBlog ()
    {
        $blog = $this->blog;

        // 传入的blog网址，必须符合下面的正则表达式
        return preg_match("/^(((http(s?))\:\/\/)?)([0-9a-zA-Z\-]+\.)+[a-zA-Z]{2,6}(\:[0-9]+)?(\/\S*)?$/i", $blog
);
    }

}
```

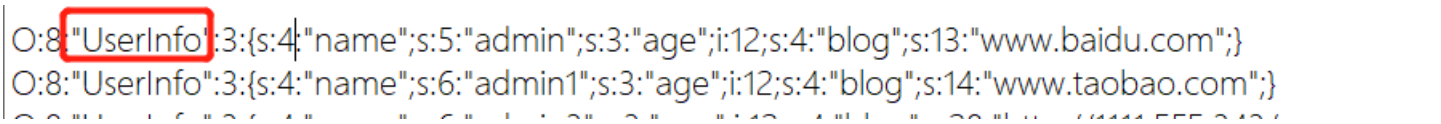
类名是：UserInfo。代码比较简单注意下面的函数就行：

- `curl_init()` — 初始化一个新的会话，返回一个cURL句柄，供curl_setopt(), curl_exec()和curl_close() 函数使用
- `curl_setopt($x, $y, $z)` — 设置一个cURL传输选项，为给定的cURL会话句柄设置一个选项
  - $x — 句柄 $y — 选项 $z — 选项的参数
  - CURLOPT_URL — 需要获取的URL地址，也可以在curl_init()函数中设置
  - CURLOPT_RETURNTRANSFER — 将curl_exec()获取的信息以文件流的形式返回，而不是直接输出
- `curl_exec()` — 执行一个句柄
- `curl_getinfo()` — 获取一个cURL连接资源句柄的信息
- `curl_close()` — 关闭连接

这里很明显存在 `ssrf漏洞` 只要我们提交的 blog 的内容符合 `isValidBlog函数` 中正则的匹配条件即可。这里本小白想了很久，但是就是无法满足正则匹配的中间的那一部分内容：`[a-zA-Z]{2,6}`，做到最好的程度就是：`http://3707319430.AAA:43877/flag.php` 3707319430是220.249.52.134转为int之后的结果

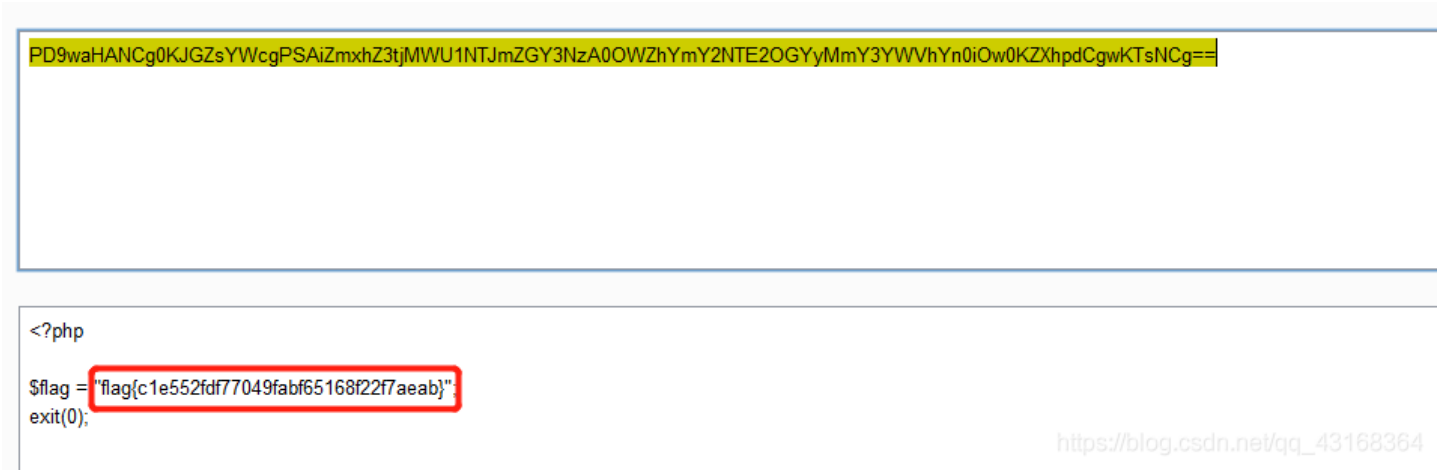


如果可以将AAA去除，就可以实现SSRF了，不知道有没有办法。。。这条路不通还有一条路，在data字段中。再来看看data字段存放的数据



是 UserInfo对象 序列化 之后的内容，显然数据库是通过它来渲染回显到页面中的内容的，我们这里构造一个序列化的字符串：`O:8:"UserInfo":3:{s:4:"name";s:6:"admin4";s:3:"age";i:12;s:4:"blog";s:29:"file:///var/www/html/flag.php";}` 去访问 `file:///var/www/html/flag.php` 文件，data的字段是4。

可以得到flag，iframe 标签中的后面的内容经过base64解码后也可以得到flag



二、知识点

php cURL函数

- PHP支持的libcurl库允许你与各种的服务器使用各种类型的协议进行连接和通讯
- PHP中使用cURL实现Get和Post请求的方法
- 为了使用PHP的cURL函数，需要安装 libcurl 包