**1. Executive Summary**

This report details the outcome of a red-team-aligned, controlled malware simulation executed in a Windows 10 Pro virtual environment. The purpose was to mimic post-exploitation behaviors—specifically credential theft, stealthy process injection, and persistence—to validate incident response (IR) playbooks, test memory forensics tooling (Volatility 3), and demonstrate detection gaps in live system monitoring.

The attacker scenario focused on the use of mimikatz to dump credentials from LSASS memory and the use of schtasks to achieve persistence. Payloads were injected into legitimate processes, such as splunkd.exe, to illustrate common defense evasion techniques.

Volatile memory was captured immediately post-execution and examined offline to simulate real-world DFIR (Digital Forensics and Incident Response) processes. Key artifacts were recovered and correlated with MITRE ATT&CK techniques. Host-based images (Process Explorer, Autoruns, Task Scheduler, TCPView, and Registry snapshots) further validated memory artifacts.

From a broader business risk standpoint, this simulation represents threat actor behaviors aligned with credential harvesting and lateral movement stages of the attack lifecycle. Failure to detect such activity may result in privilege escalation, unauthorized access to critical assets, ransomware deployment, or exfiltration of sensitive data.

This document provides:

* A complete technical breakdown of the infection chain
* Memory forensics findings with supporting screenshots and hash evidence
* IOC correlation and MITRE mapping
* Stakeholder-specific recommendations
* Lessons learned for improving detection and incident response maturity

**2. Objective**

* Emulate adversarial post-exploitation behaviors including credential dumping and persistence.
* Capture RAM post-infection to preserve volatile evidence.
* Analyze memory using Volatility 3 and validate against endpoint-level artifacts.
* Provide an industry-grade IR report suitable for security audits, red/blue team exercises, and compliance documentation.

**3. Environment Overview**

**Host System:**

* Windows 10 Home
* Isolated, with no internet access

**Virtual Lab:**

* **Victim VM**: Windows 10 Pro (Target for credential theft)
* **Attacker VM**: Kali Linux (Volatility 3 + tools)

**Security Controls:**

* VM Snapshots pre-/post-execution
* No external network communication (Air-gapped simulation)
* Memory capture: 20250602.mem using Belkasoft RAM Capturer

**4. Attack Timeline**

|  |  |
| --- | --- |
| **Timestamp (UTC)** | **Description** |
| 05:18:55 | Victim system powered on |
| 05:20:03 | User login and explorer.exe process starts |
| 05:27:46 | Attacker initiates shell via cmd.exe |
| 05:30:00 | mimikatz.exe executed for credential dumping |
| 05:32:00 | schtasks creates persistence via SystemUpdater task |
| 05:48:13 | powershell.exe launched (possible secondary payload) |
| 05:53:38 | Memory captured for forensic review |

**5. Forensic Findings (Volatility 3)**

**5.1 Process List (windows.pslist)**

* 170 processes found
* **Suspicious Observations:**
  + No direct instance of mimikatz.exe in memory — consistent with short-lived or renamed execution
  + splunkd.exe contained anomalous RWX segments (potential shellcode injection)

**5.2 Command Line Activity (windows.cmdline)**

schtasks /create /tn SystemUpdater /tr C:\InfectionSimulation\mimikatz.exe /sc ONLOGON /ru SYSTEM

* Confirms use of LOLBin (schtasks) to execute Mimikatz under SYSTEM privileges
* Visible in screenshot: cmd.exe shell with execution trace

**5.3 Scheduled Task Analysis (windows.schedtasks)**

* **Task Name**: SystemUpdater
* **User**: SYSTEM (as per Task Scheduler snapshot)
* **Trigger**: At logon of any user
* **Action**: Execute C:\InfectionSimulation\mimikatz.exe
* **Privilege**: Highest (elevated)
* **Evidence**: Screenshot confirms presence in Task Scheduler GUI

**5.4 Memory Injection (windows.malfind)**

* RWX memory segment located:
  + **Process**: splunkd.exe (PID 3116)
  + **Memory Range**: 0x22ae5b60000 – 0x22ae5b6ffff
  + **Indicators**: Suspicious instructions, prologues typical of shellcode staging
* Consistent with shellcode injection to execute payloads under trusted process context

**5.5 Credential Dumping – mimikatz sekurlsa::logonpasswords**

* Extracted Credentials:

|  |  |  |  |
| --- | --- | --- | --- |
| User | NTLM Hash | SHA1 Hash | DPAPI Key |
| Sumit | 9218809b443d9dd59bcc9efd18572af9 | 32e5028f8b147f4d8c374bcd02d82bbe5b9e13c3 | 32e5028f8b147f4d8c374bcd02d82bbe |
| testuser | 58a478135a93ac3bf058a5ea0e8fdb71 | 0d7d930ac3b1322c8a1142f9b22169d4ee9e855 | 0d7d930ac3b1322c8a1142f9b22169d4 |

* Screenshots corroborate live credential dump in CLI and text log

**5.6 Registry & Autoruns**

* Registry Key Found:
  + HKCU\Software\Microsoft\Windows\CurrentVersion\Run
  + Value: MicrosoftEdgeAutoLaunch\_... (Benign entry — no malicious autorun found)
* Verified via Autoruns screenshot

**5.7 Network Behavior – TCPView Analysis**

* **No active C2 channels**
* All ports either localhost or 192.168.x.x
* No beaconing or outbound anomalies detected during live session
* TCPView screenshots confirm lack of active or remote sessions

**6. Indicators of Compromise (IOCs)**

|  |  |  |
| --- | --- | --- |
| Type | Value | Source |
| File Path | C:\InfectionSimulation\mimikatz.exe | cmdline.txt |
| Scheduled Task | SystemUpdater | GUI + Timeline |
| Injected Proc | splunkd.exe (PID 3116) | malfind |
| Memory Segment | 0x22ae5b60000 – 0x22ae5b6ffff | malfind |
| NTLM Hash | 9218809b443d9dd59bcc9efd18572af9 | mimikatz |
| NTLM Hash | 58a478135a93ac3bf058a5ea0e8fdb71 | mimikatz |
| SHA256 (exe) | 8d91fa5f2f7e404a48bc9e07be331364264a9b843ea9c1f3a31fa1b4abebd9bc | certutil hash |
| SHA256 (memory) | 1256e83e726d2c647b8bfe66d3c7332a92f8a4b7f88c47e2b90d9a7bccefef2d | RamCapture |

**7. MITRE ATT&CK Mapping**

|  |  |  |  |
| --- | --- | --- | --- |
| Tactic | Technique | ID | Evidence |
| Credential Access | OS Credential Dumping (mimikatz) | T1003.001 | Mimikatz, extracted hashes |
| Persistence | Scheduled Task/Job | T1053.005 | SystemUpdater via schtasks |
| Defense Evasion | Portable Executable Injection | T1055.002 | splunkd.exe RWX memory segment |

**8. Recommendations by Stakeholder**

**Security Operations Center (SOC):**

* Deploy IOC hashes into SIEM for real-time alerting
* Enable Volatility/YARA integration for suspicious memory regions
* Detect usage of schtasks with SYSTEM privilege via Sysmon Event ID 1 + 4698

**IT Operations:**

* Restrict scheduled task creation to privileged users only
* Harden PowerShell with Constrained Language Mode
* Enable LSASS protection via RunAsPPL (registry + GPO enforcement)

**Endpoint Security Team:**

* Use AppLocker to prevent execution from non-standard paths (e.g., C:\InfectionSimulation)
* Monitor for PE injections in non-browser, non-AV processes
* Conduct memory-based scanning using EDR tools with RWX flagging

**Governance & Incident Response:**

* Conduct quarterly memory analysis drills
* Ensure IR documentation includes memory acquisition SOPs
* Introduce tabletop exercises based on scenarios like this one

**9. Conclusion**

This simulation replicated a realistic APT-inspired attack involving post-exploitation credential harvesting and stealth persistence using built-in Windows tools. Volatility 3 confirmed its capability to extract and correlate forensic evidence even in the absence of disk-based artifacts.

Key detection gaps were identified in ephemeral binaries, injected shellcode, and abuse of scheduled tasks. Integration of memory forensics into routine IR workflow is essential to bridge detection gaps missed by traditional AV/EDR alone.

**Lessons Learned:**

* Memory-only attacks are viable and stealthy
* Scheduled task creation under SYSTEM should trigger alerts
* LSASS memory remains a critical asset for attackers

Organizations should enforce a layered approach including hardening, behavioral monitoring, memory forensics, and regular red-teaming to stay resilient against credential access and persistence threats.