Memory Forensics Project

Student Names: Devarshi Mahadevwala

Mentor: Ahmad Ghobeiti (ahmad.ghobeiti@senecapolytechnic.ca)

Introduction

This report documents a complete end-to-end memory forensics investigation conducted within a controlled virtualized environment. The objective of this project was to replicate a real-world credential-theft scenario, acquire multiple forensic memory images during different stages of the attack, and perform in-depth analysis using Volatility Frameworks 2 and 3. The investigation followed a structured incident-response methodology beginning with controlled infrastructure setup, followed by the deployment of an attack vector (Mimikatz), multi-stage memory acquisition using DumpIt, and comprehensive forensic examination of three discrete memory captures representing the *pre-attack*, *active attack*, and *post-attack* system states.

The primary goal of this exercise was to simulate how attackers leverage credential-dumping tools to escalate privileges and how defenders and forensic analysts reconstruct the attack timeline by examining volatile memory. By analyzing memory artifacts—such as active processes, process trees, handle objects, loaded DLLs, registry hives, network connections, and LSASS interactions—the investigation demonstrates how credential theft leaves forensic traces even when overt indicators (like running malicious processes) have disappeared.

The project provides a realistic representation of how memory forensics is used during incident response to (1) detect malicious activity, (2) confirm credential compromise, (3) recover persistence mechanisms, and (4) correlate activity across multiple system states. The resulting forensic narrative closely resembles what digital forensics teams produce in professional settings such as SOC environments, consulting engagements, and breach investigations.

Table of Content

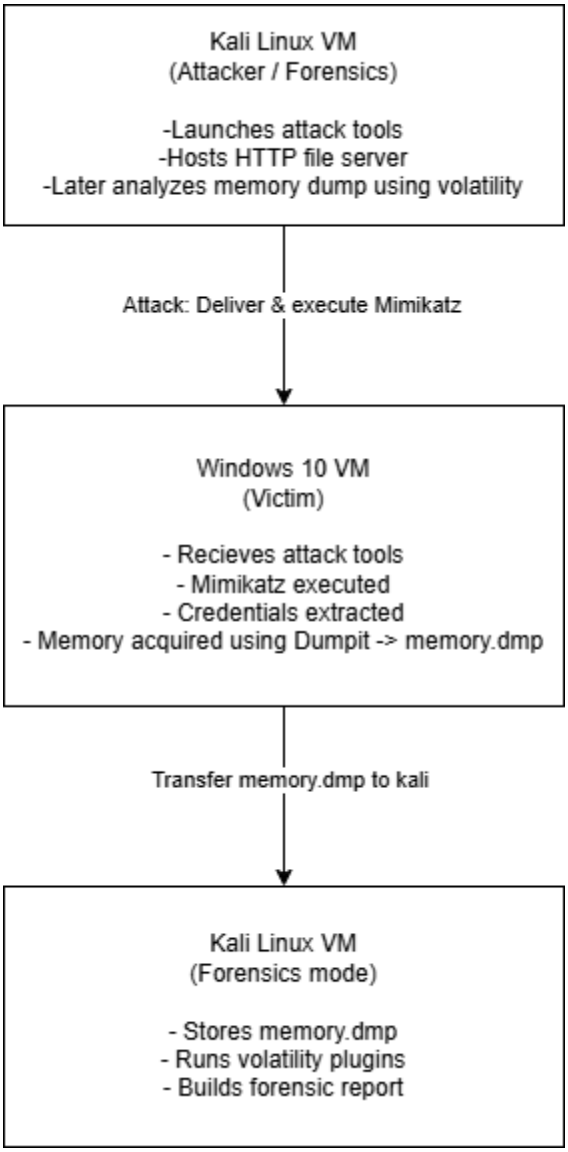
Phase 1 – Infrastructure Setup (Final, Professional & Technical Version)

This phase documents the complete preparation of the attacker (Kali Linux) and victim (Windows 10) environments prior to deployment of the attack. The objective of Phase 1 is to create a controlled, isolated, and reproducible environment for offensive testing and post-incident forensics.

1. Virtual Lab Overview & Architecture

Purpose

To establish an isolated attacker–victim environment suitable for controlled credential extraction, memory acquisition, and forensic investigation.



Explanation

This diagram summarizes the lab structure:

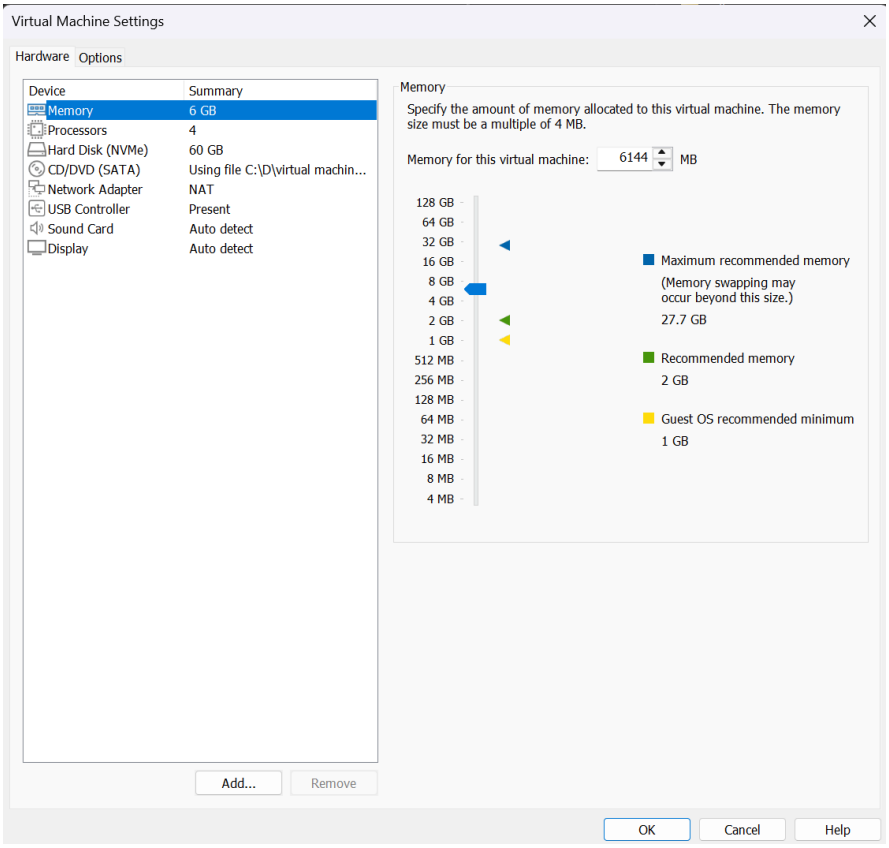
Kali Linux (attacker) ↔ Windows 10 (victim)

- Connected only through **Host-Only Adapter**
- No external internet connectivity
- Both VMs maintain **snapshot checkpoints** for repeatability

The architecture ensures the attack is contained without risk to external networks.

2. Virtual Machine Configuration

2.1. Kali Linux Configuration



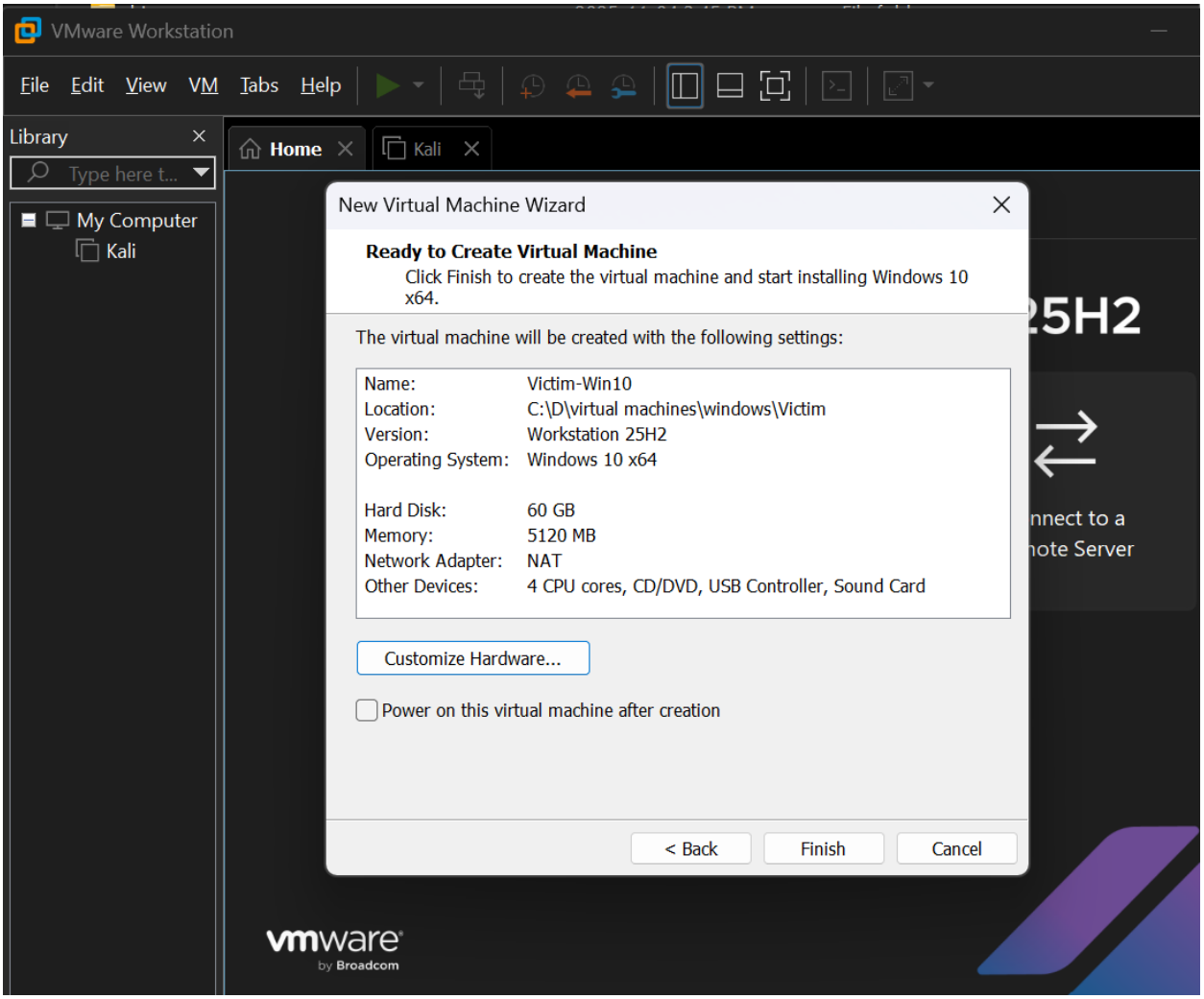
Explanation

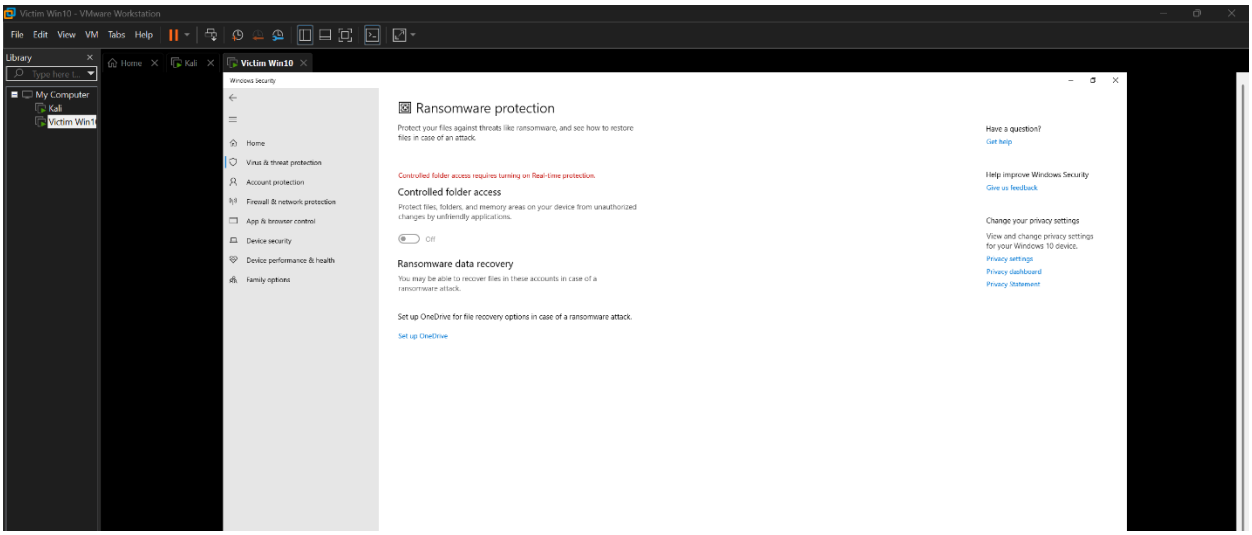
This screenshot documents the Kali Linux VM configuration:

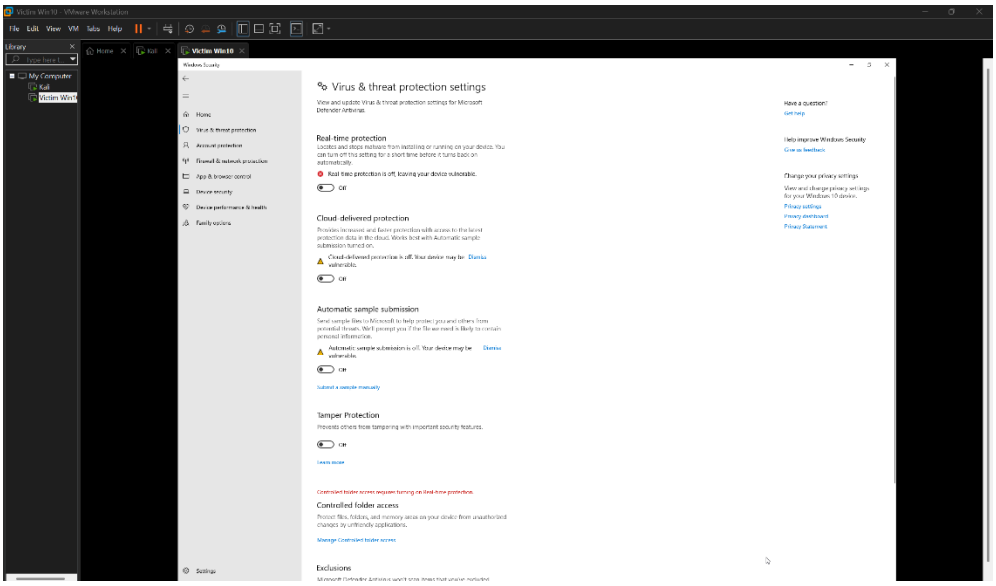
- Network mode set to **Host-Only Adapter**
- Adequate CPU & RAM allocation
- Virtual hardware configured for tool compilation, HTTP service hosting, and running forensic toolchains.

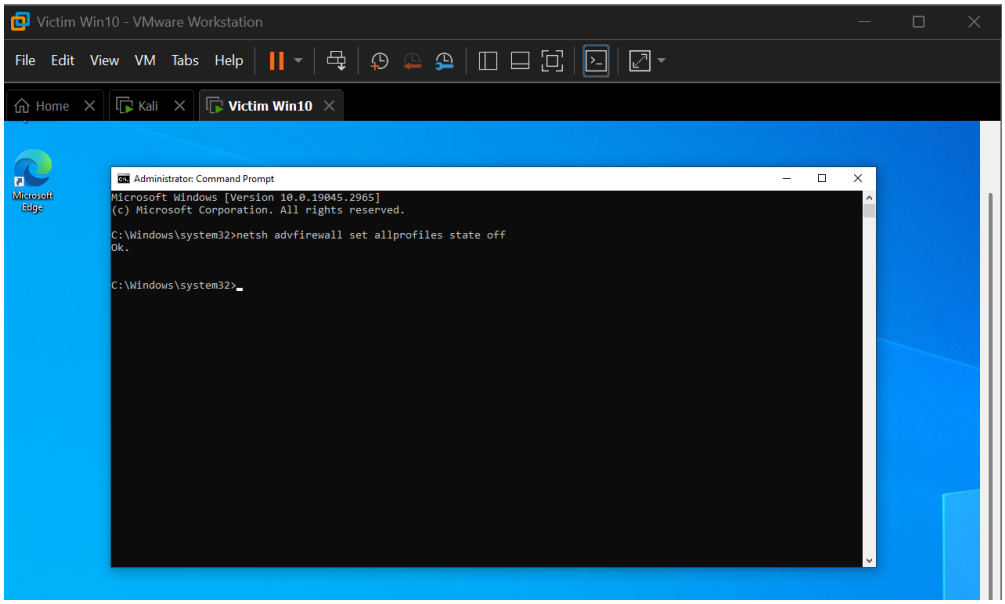
This acts as the attacker platform for **Mimikatz deployment**, **DumpIt execution**, and **Volatility analysis**.

2.2. Windows 10 Configuration (Victim Machine)









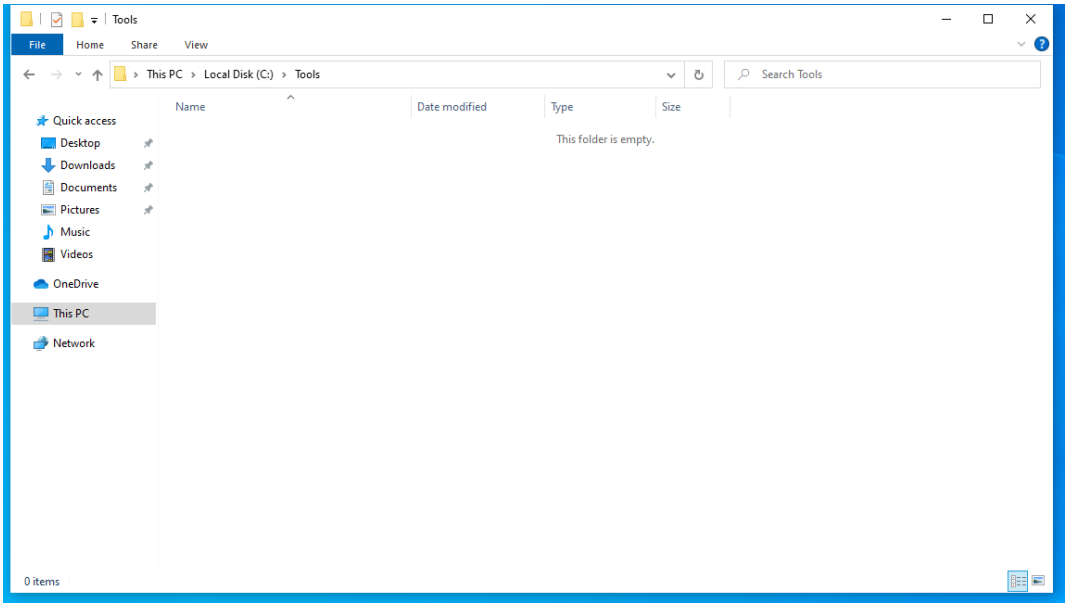
Explanation

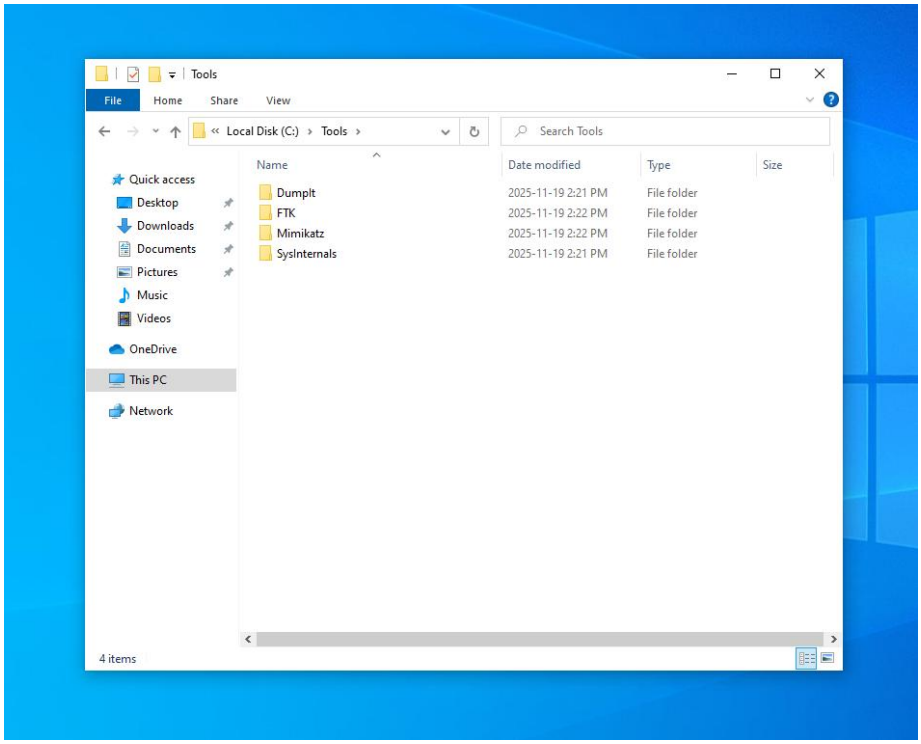
The Windows 10 victim machine is configured to simulate a realistic vulnerable endpoint:

- **Controlled Folder Access disabled**
  → Allows the attacker to place tools (DumpIt, Mimikatz) without Microsoft Defender interference.
- **Security protections and real-time scanning disabled**
  → Ensures the attack tools execute without obstruction.
- **Windows Firewall turned off**
  → Required for enabling inbound/outbound communication during attack staging.

These configurations are necessary to enable **credential dumping**, memory acquisition, and remote interaction.

3. Creation of Required Directories & Tool Placement





```
Upgrading: 0, Installing: 0, Removing: 0, Not Upgradi

┌──(vol3)─(dbmahadevwala㉿kali)-[~]
└─$ tree ¬/Forensics -L 2
¬/Forensics
├── MemoryDumps
├── Tools
└── Volatility

4 directories, 0 files

┌──(vol3)─(dbmahadevwala㉿kali)-[~]
└─$
```

Explanation

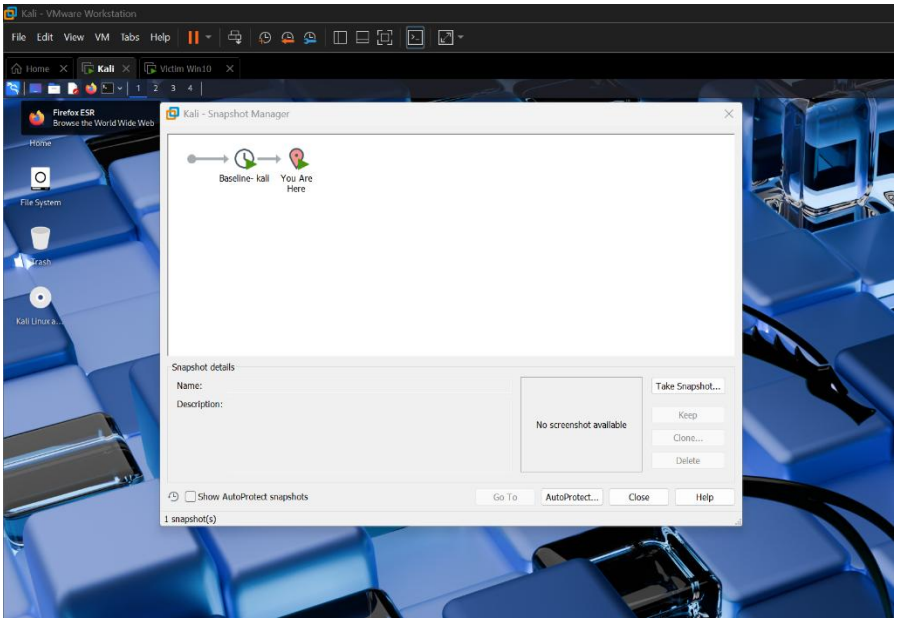
Dedicated folder structures were created to maintain clean segregation between:

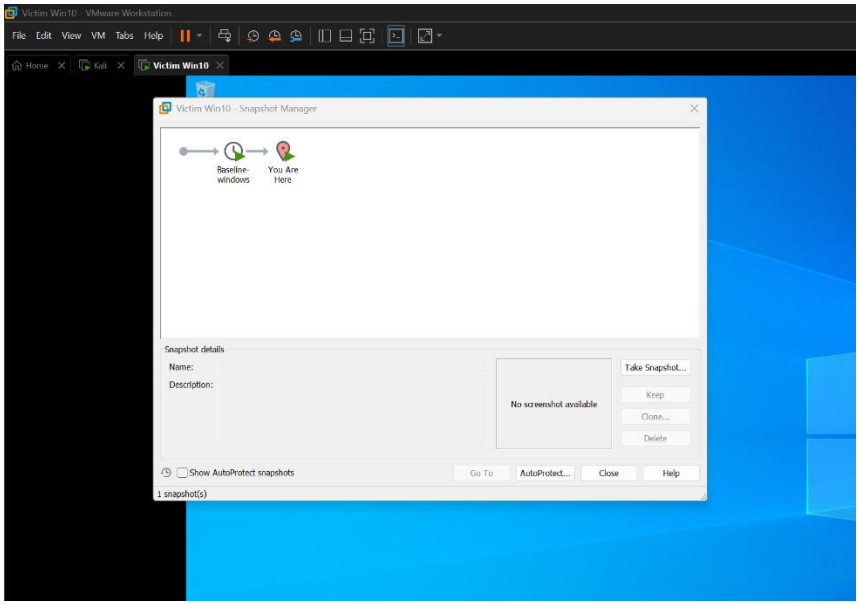
- **Attack Tools** (DumpIt.exe, mimikatz.exe)
- **Forensics Tools**
- **Generated Memory Dumps**

This organization is crucial for:

- Repeatability of experiments
- Clean pre-attack baseline captures
- Minimizing contamination of analysis artifacts

4. Snapshot Creation for Reproducibility





Explanation

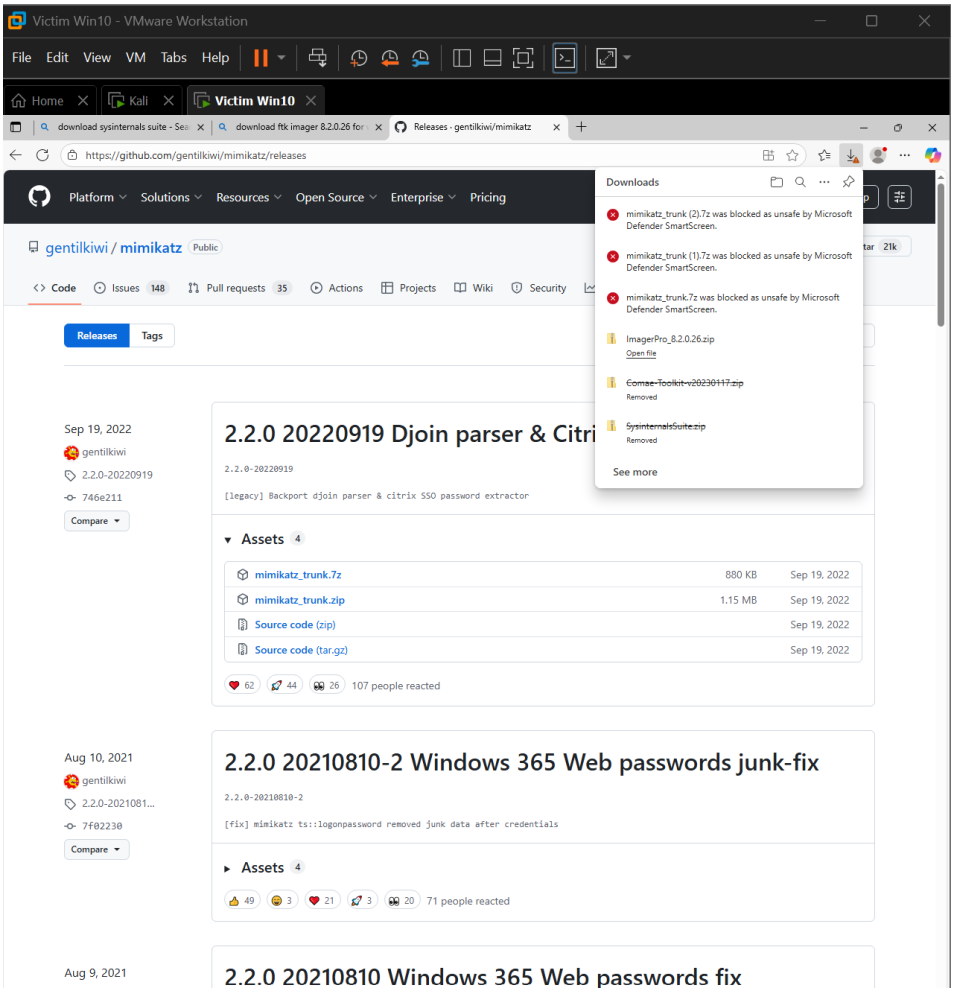
Snapshots were captured after configuring both VMs but before deploying any attack tools. This ensures:

- The **baseline state is preserved**
- Memory dumps represent clean and attack states accurately
- Any configuration mistakes can be reverted instantly

Snapshots form the foundation of **Phase 3 comparative analysis**.

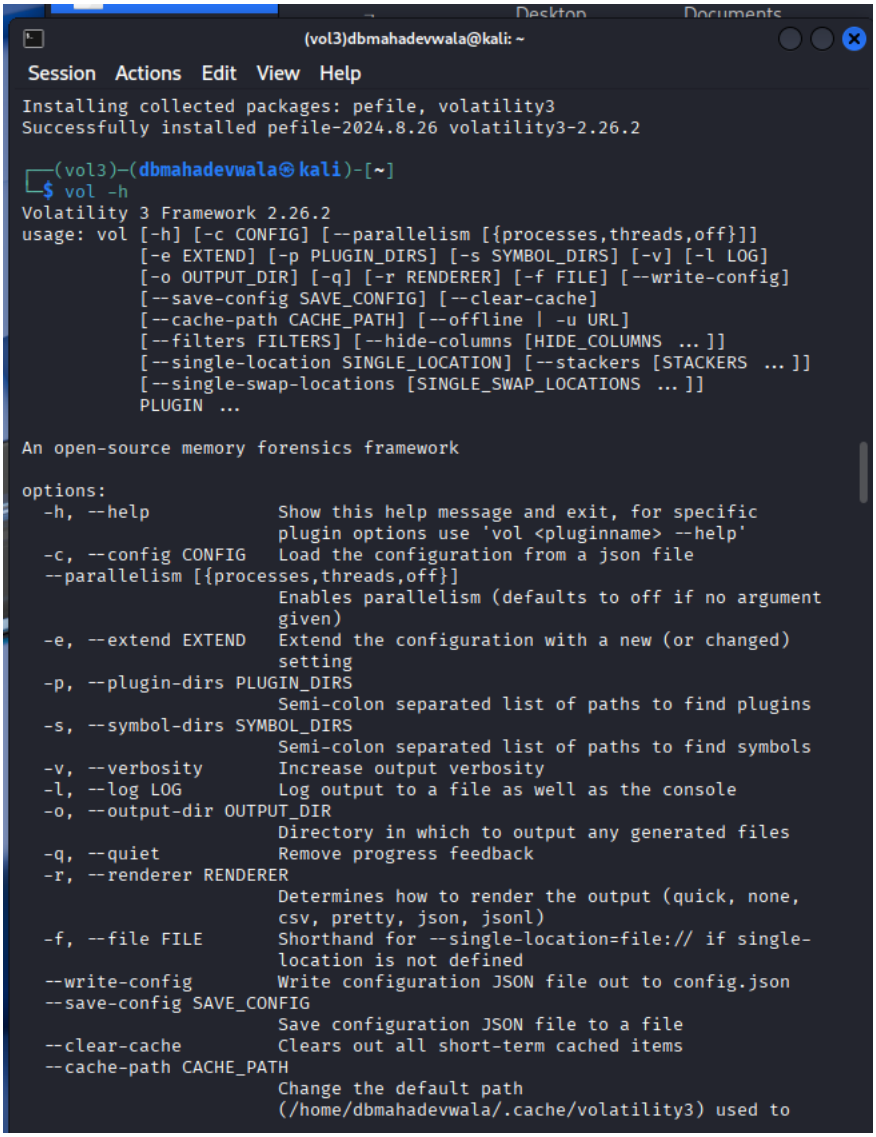
5. Installation & Validation of Required Tools

5.1. Mimikatz Restrictions (Verification)



Windows Defender blocks Mimikatz downloads under normal conditions.
This screenshot demonstrates why **security controls had to be temporarily disabled**.
This is not an attack action—only documentation of baseline Defender behavior.

5.2. Installing Volatility 3 (Kali)

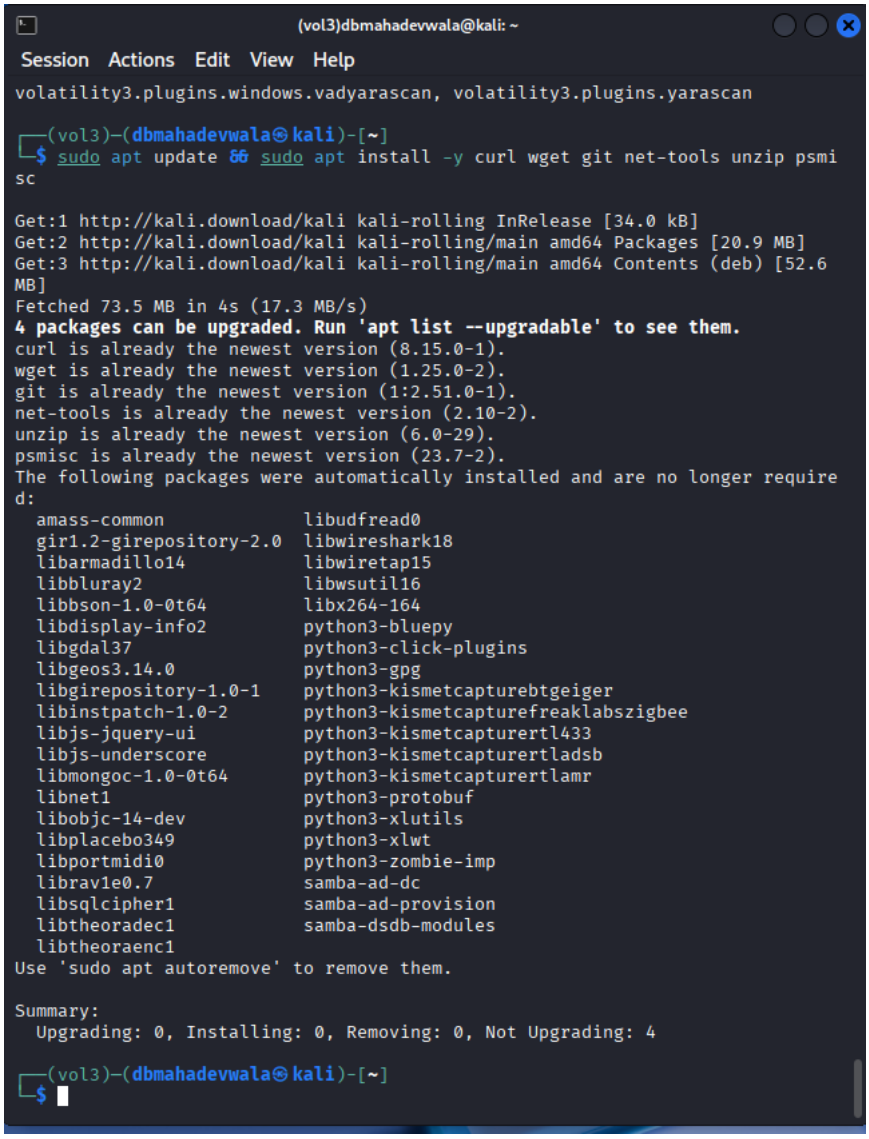


Volatility 3 is installed in Kali as the primary tool for:

- Memory parsing
- Process reconstruction
- Handle and module enumeration
- Credential extraction analysis (PSList, DLLList, Handles, Netstat)

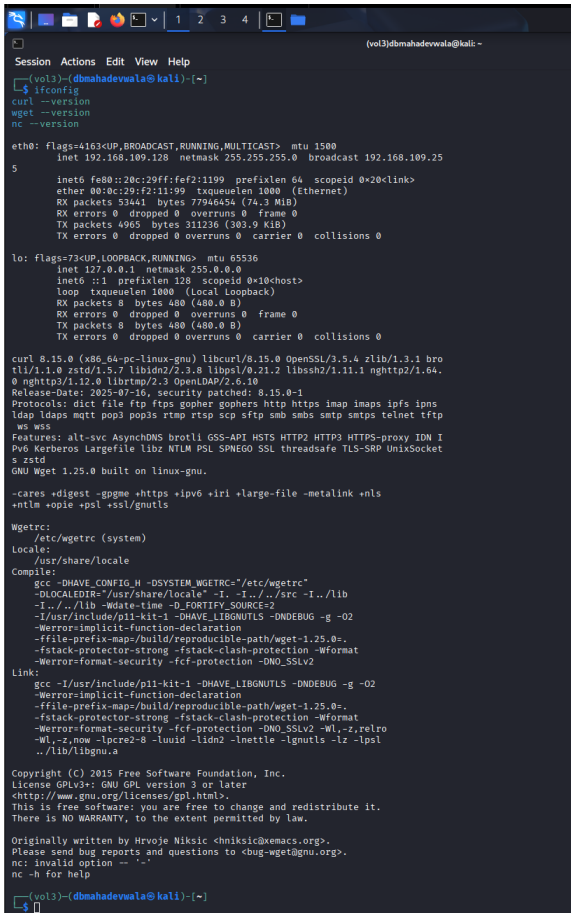
This prepares the forensic side of the lab.

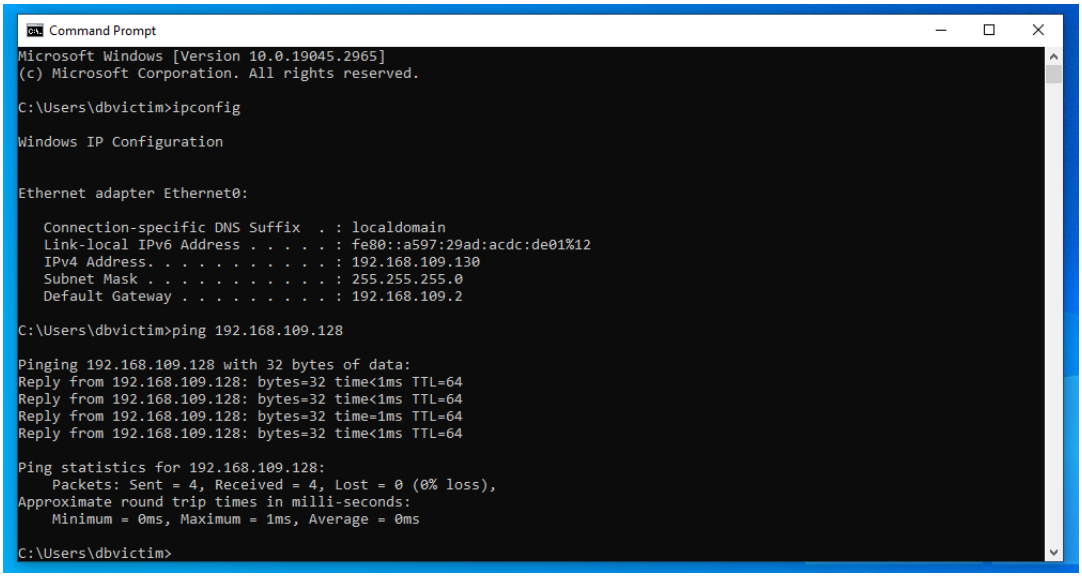
5.3. Updating Tooling & Dependencies in Kali

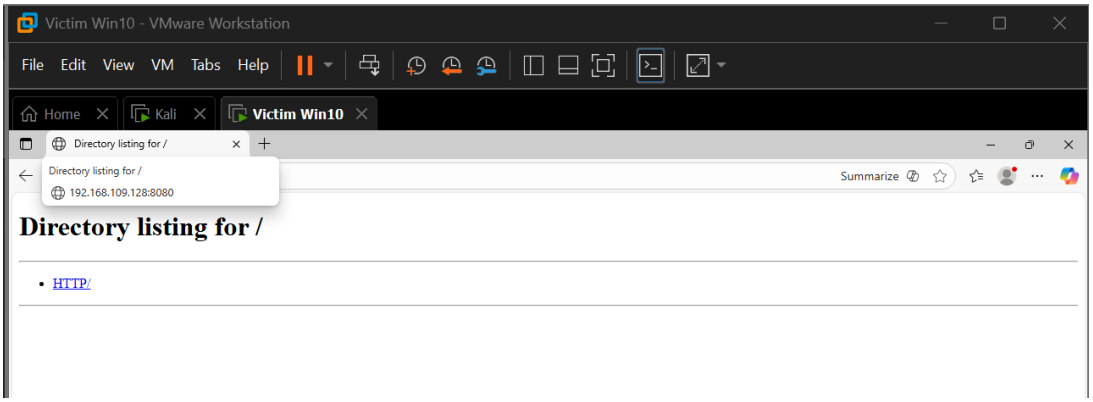


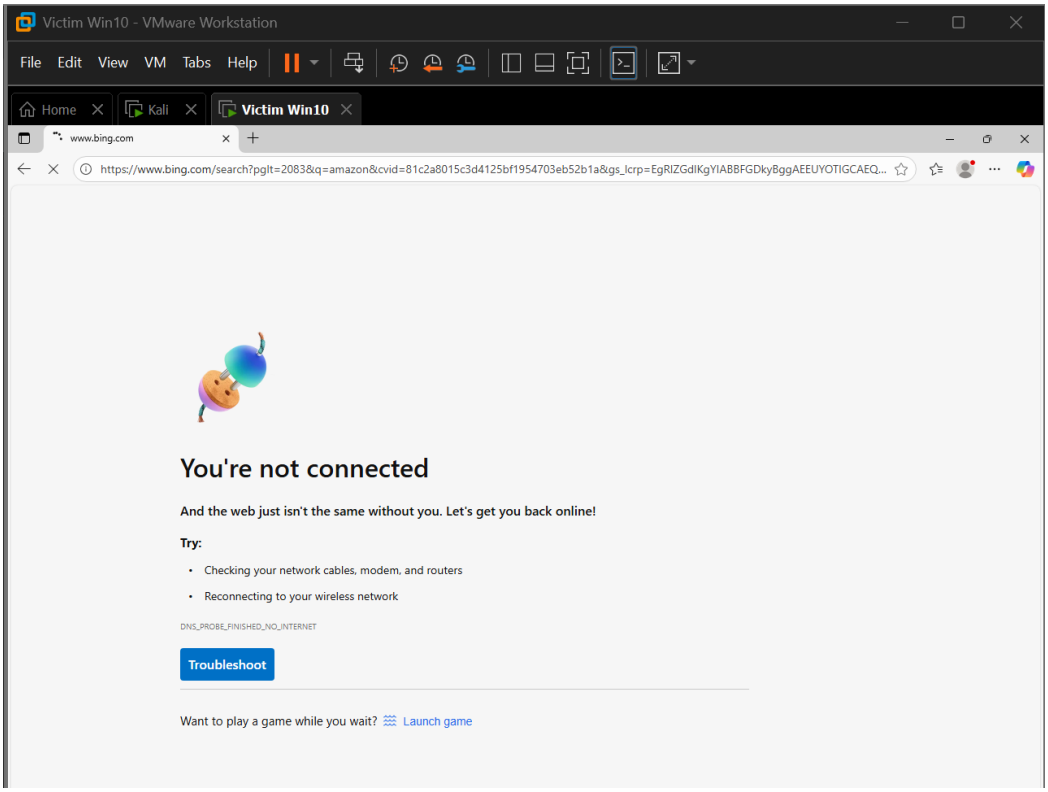
This ensures Python, networking libraries, Volatility dependencies, and HTTP server modules are up-to-date.

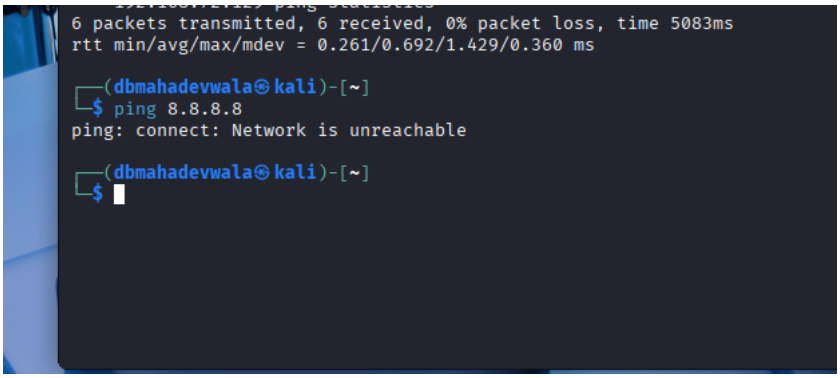
6. Network Configuration & Connectivity Tests




Command Prompt
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.
C:\Users\dbvictim>ipconfig
Windows IP Configuration
Ethernet adapter Ethernet0:
Connection-specific DNS Suffix . : localdomain
Link-local IPv6 Address . . . . . : fe80::a597:29ad:acdc:de01%12
IPv4 Address. . . . . : 192.168.109.130
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.109.2
C:\Users\dbvictim>ping 192.168.109.128
Pinging 192.168.109.128 with 32 bytes of data:
Reply from 192.168.109.128: bytes=32 time<1ms TTL=64
Reply from 192.168.109.128: bytes=32 time<1ms TTL=64
Reply from 192.168.109.128: bytes=32 time=1ms TTL=64
Reply from 192.168.109.128: bytes=32 time<1ms TTL=64
Ping statistics for 192.168.109.128:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms
C:\Users\dbvictim>


Victim Win10 - VMware Workstation
File Edit View VM Tabs Help
Home
Kali
Victim Win10
Directory listing for /
192.168.109.128:8080
Summarize
Directory listing for /
HTTP/





Explanation

This section validates controlled connectivity between attacker and victim:

What we verified:

✓ Both machines communicate bidirectionally via **Host-Only network**
✓ No external internet access (confirmed via browser tests and failed DNS pings)
✓ HTTP server on Kali serves content accessible from Windows
✓ ICMP reachability confirmed both ways

Why this matters:

- Ensures **attack traffic stays inside the lab**
- Allows file transfer from attacker to victim
- Guarantees realistic but safe offensive testing

7. Final Phase 1 Validation Summary

At the end of Phase 1, we confirmed that:

- Both VMs are properly configured
- Tools are installed and structured
- Security controls are disabled intentionally
- Network environment is isolated
- Snapshots are in place
- Connectivity is established
- Internet is isolated from both machines
- Directory structure is prepared for upcoming phases

This completes Phase 1 and provides a stable foundation for Phase 2 (attack execution and memory dumps).

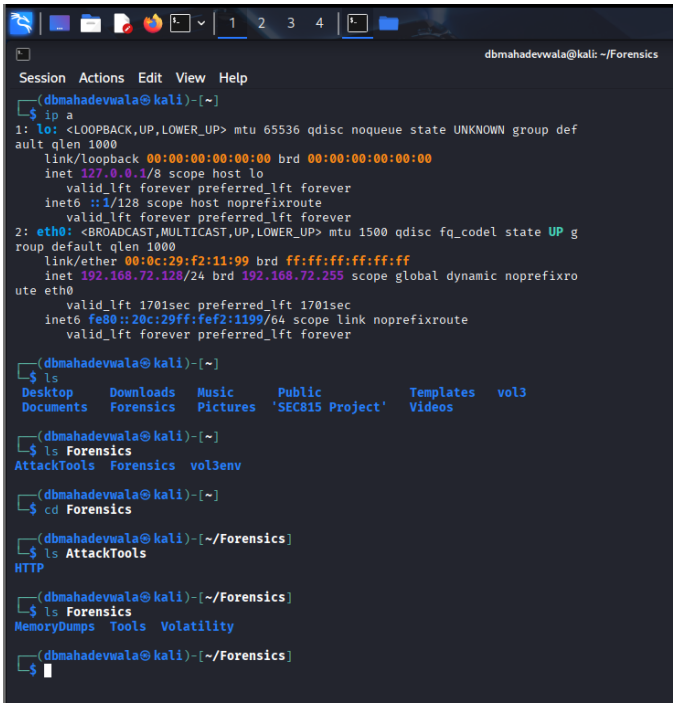
Phase 2 – Attack Delivery, Execution, and Memory Acquisition

This phase documents the preparation, execution, and capture of malicious activity within the controlled laboratory environment. The attacker (Kali Linux) delivered a sequence of payloads to the victim (Windows 10), executed credential-theft tools, and captured memory artifacts before and after the attack. This section outlines the methodology, tools used, and supporting screenshots that demonstrate each procedural step.

2.1 Preparation of the Attacker Environment

2.1.1 Verifying Attacker Network Connectivity

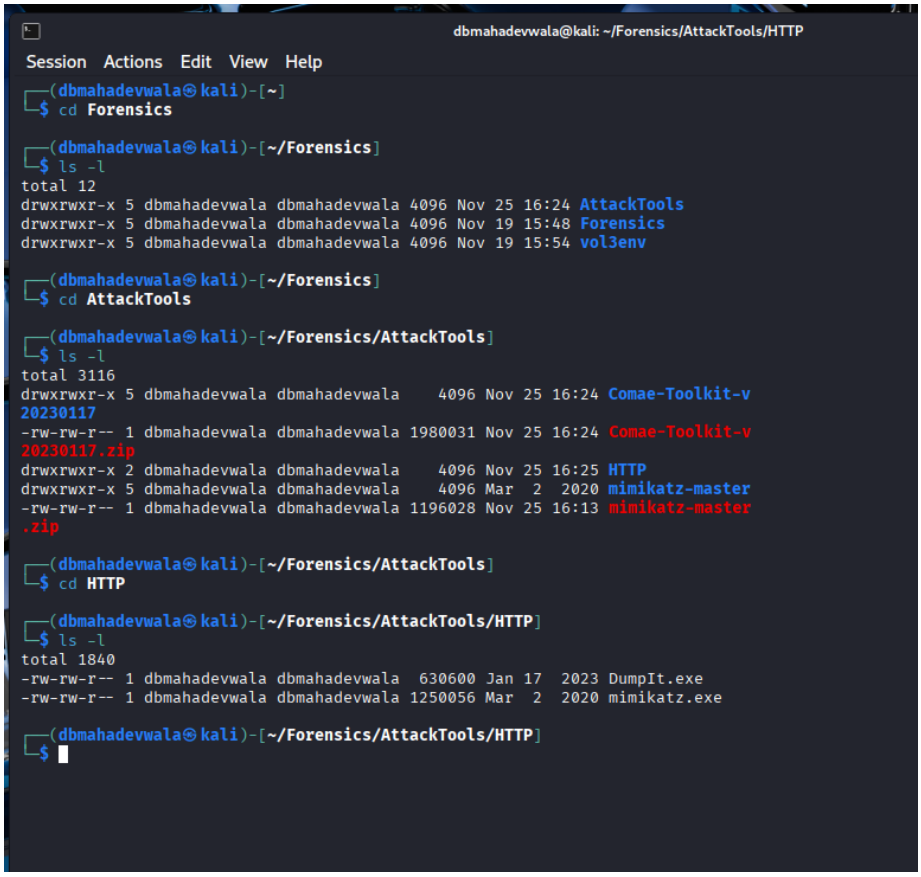
The attacker machine's IP configuration was confirmed to ensure stable communication with the Windows victim over the host-only interface.



This screenshot confirms the attacker's assigned IP address (192.168.72.128) and verifies the file system location from which the attack tools will operate.

2.1.2 Creating a Payload Staging Area

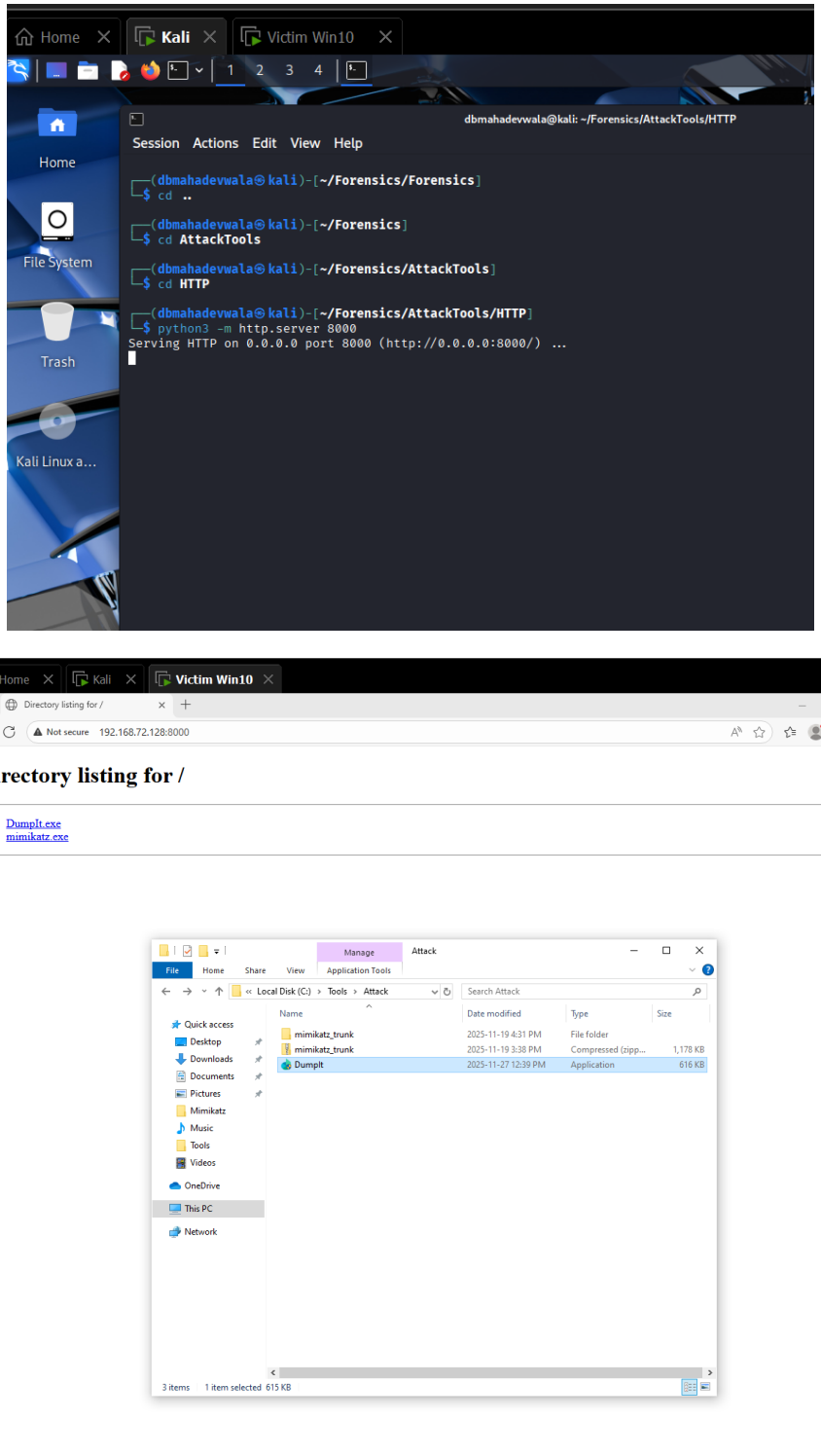
A dedicated folder was configured on Kali to host the attack tools and payloads used during exploitation.



This directory contains the malicious binaries, including DumpIt.exe and Mimikatz, ensuring clean separation between system files and attack materials.

2.1.3 Hosting an HTTP Server for Payload Delivery

To transfer malicious binaries to the Windows system, a lightweight Python HTTP server was launched from within the attack directory.

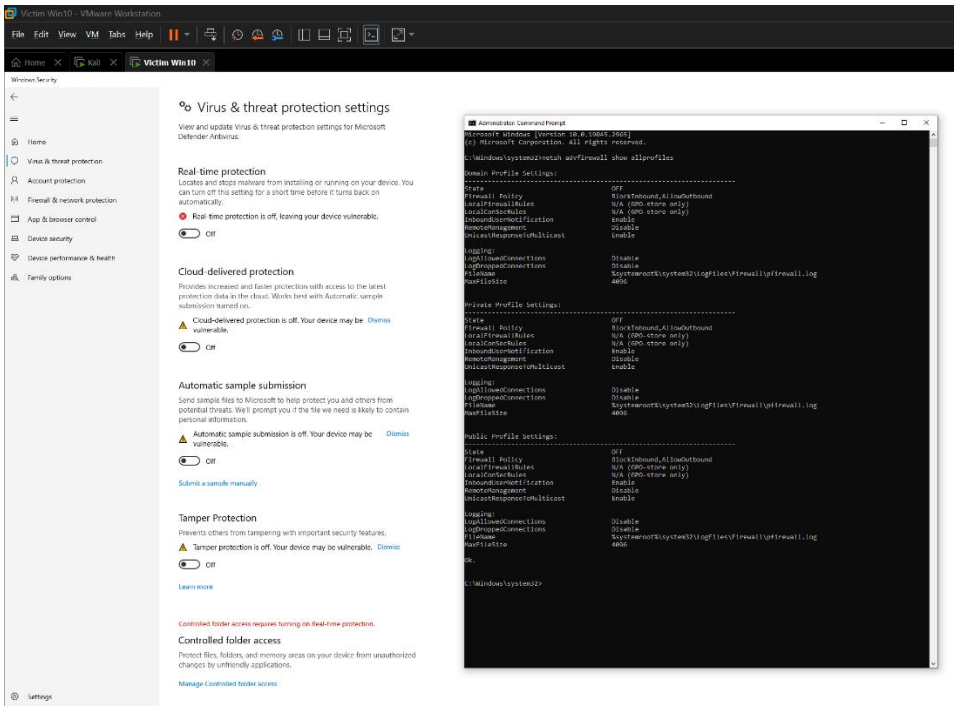


These screenshots illustrate how the attacker initiated the HTTP service and how the victim subsequently accessed it using a web browser to retrieve the DumpIt executable.

2.2 Preparation of the Victim Environment

2.2.1 Disabling Security Controls

To ensure the malicious payloads executed without interference, Windows Defender and firewall policies were manually disabled.



These configurations ensured the attacker-delivered binaries (DumpIt.exe, Mimikatz.exe) would run unimpeded.

2.2.2 Verifying Remote Accessibility

RDP access and folder permissions were confirmed to support attack phases including tool execution and file retrieval.

2.2.3 Preparing Tool Directories

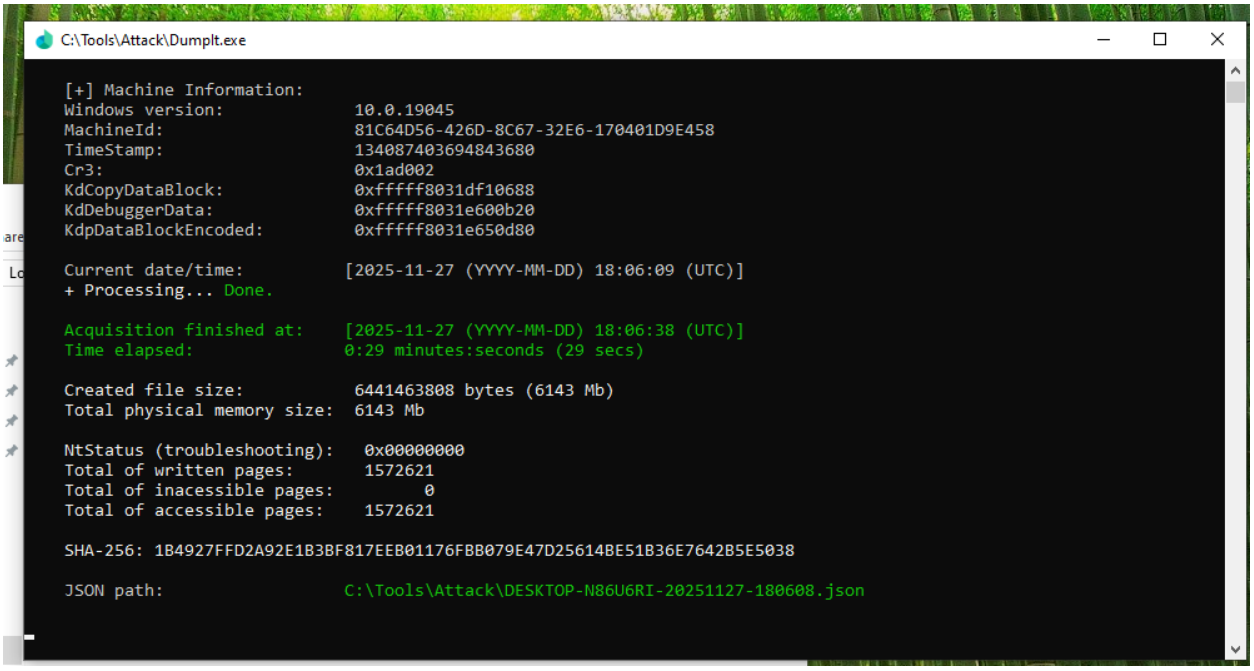
Dedicated folders were created within the Windows host to store the incoming malicious tools.

2.3 Pre-Attack Memory Capture

Prior to executing any malicious tools, a baseline memory snapshot was collected. This allows later comparison against the compromised memory state.

2.3.1 Running DumpIt.exe Before the Attack

DumpIt was executed from the victim's Windows host to generate the pre-attack memory image.

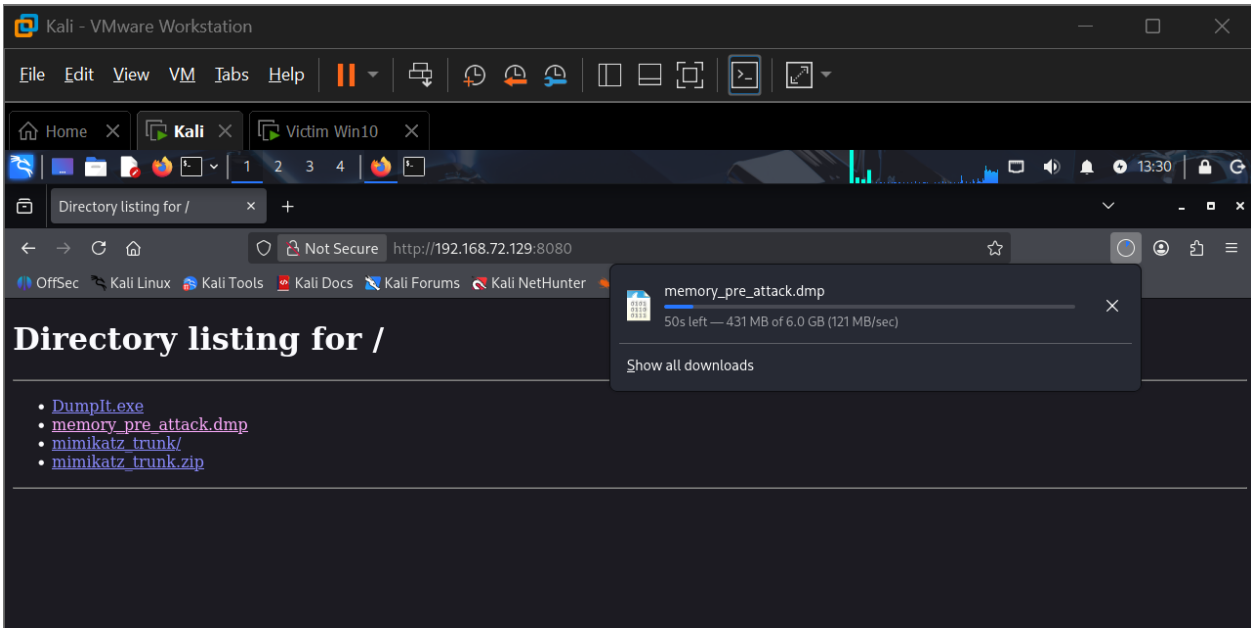


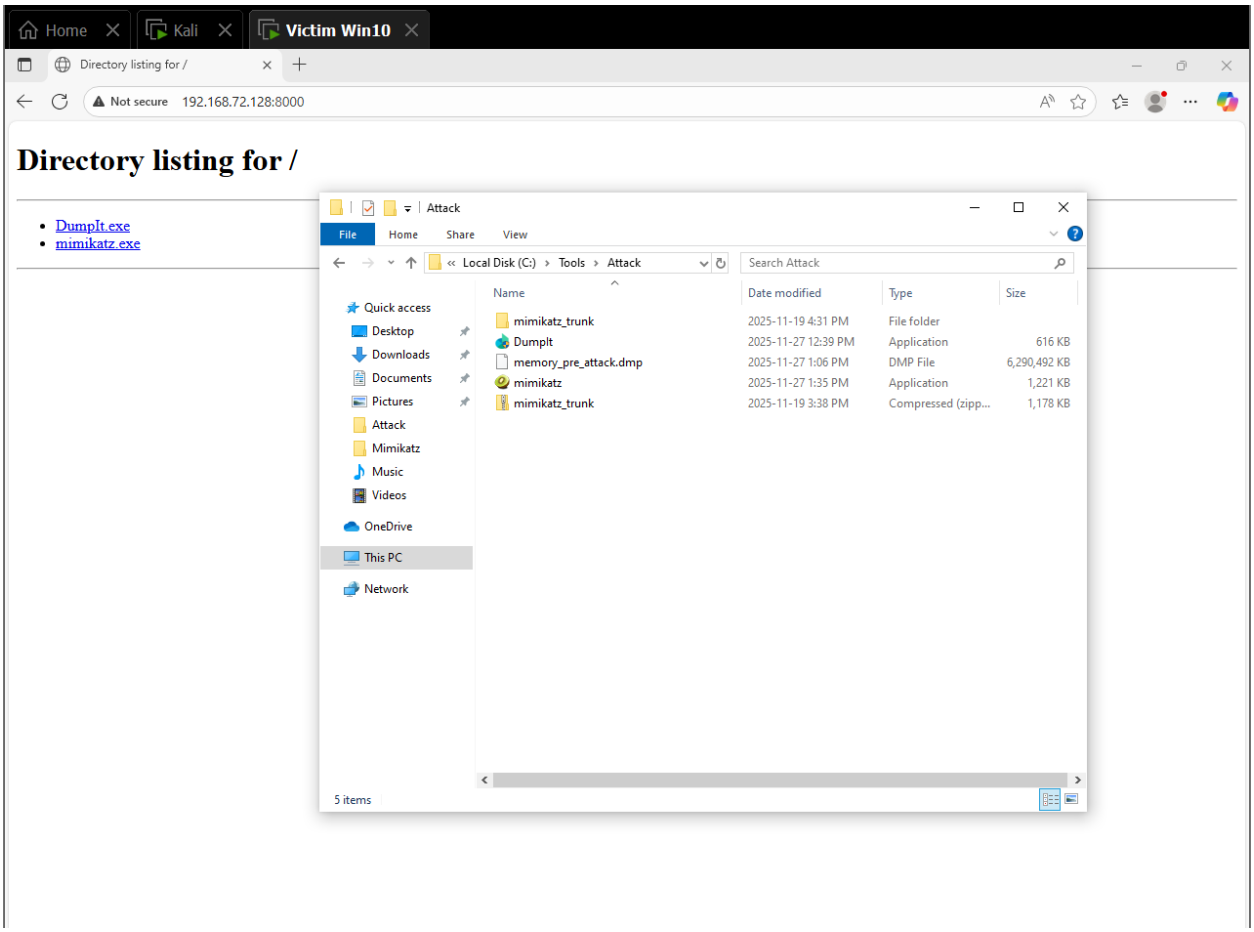
The resulting memory dump was transferred back to the Kali system for analysis in later phases.

2.4 Downloading and Executing Attack Tools

2.4.1 Downloading Mimikatz from the Attacker Server

Mimikatz was retrieved onto the victim system directly through the attacker-controlled HTTP server.

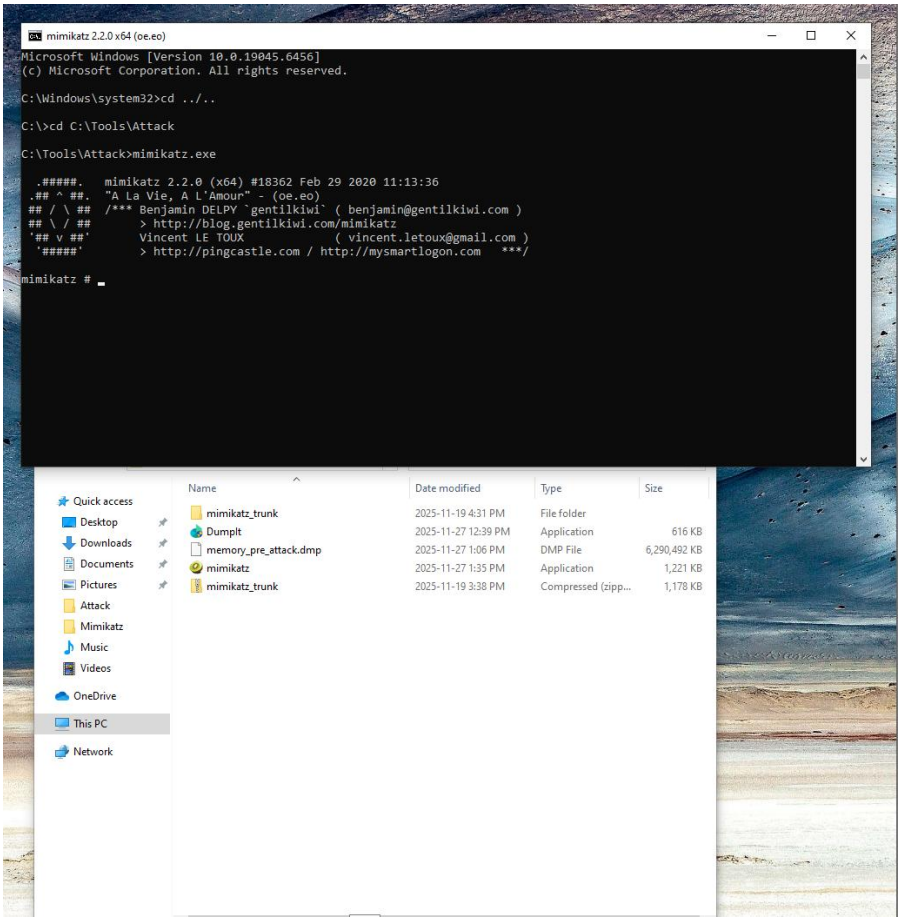




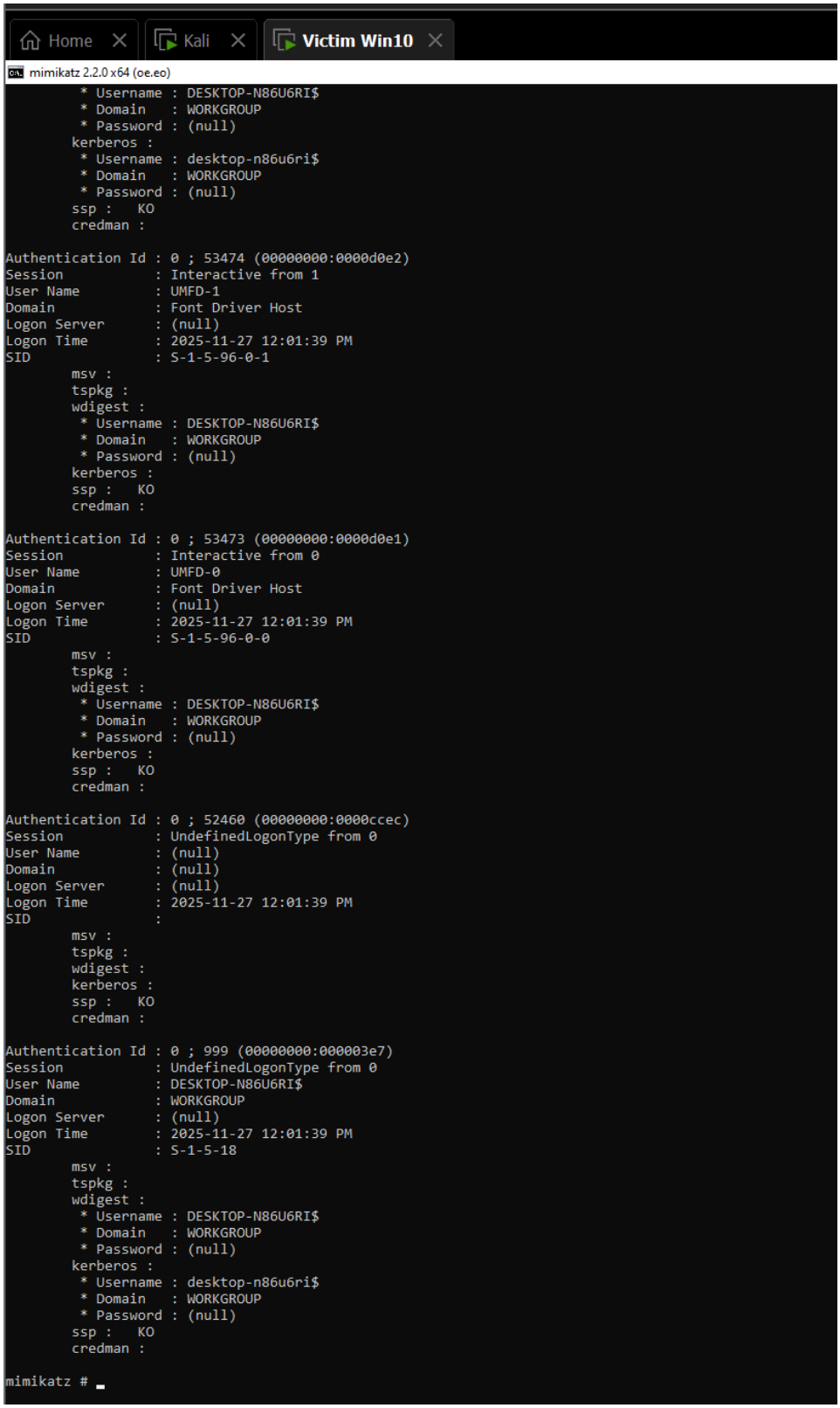
This confirms successful payload delivery from the attacker to the victim.

2.4.2 Executing Mimikatz on the Victim

Mimikatz was executed on the Windows host to extract sensitive credential material.




Victim Win10 - VMware Workstation
File Edit View VM Tabs Help
Home
Kali
Victim Win10
mimikatz 2.2.0 x64 (oe.eo)
mimikatz # privilege::debug
Privilege '20' OK
mimikatz # sekurlsa::logonpasswords
Authentication Id : 0 ; 2855980 (00000000:002b942c)
Session : Interactive from 1
User Name : dbvictim
Domain : DESKTOP-N86U6RI
Logon Server : DESKTOP-N86U6RI
Logon Time : 2025-11-27 12:11:49 PM
SID : S-1-5-21-527815699-547650139-4209623718-1001
msv :
[00000003] Primary
* Username : dbvictim
* Domain : DESKTOP-N86U6RI
* NTLM : c85da1a30ee68f95d635a55562290a2b
* SHA1 : 5fa5e93e76ac32aea0e22caff512ff9cdf891ef0
* DPAPI : 5fa5e93e76ac32aea0e22caff512ff9c
tspkg :
wdigest :
* Username : dbvictim
* Domain : DESKTOP-N86U6RI
* Password : (null)
kerberos :
* Username : dbvictim
* Domain : DESKTOP-N86U6RI
* Password : (null)
ssp : KO
credman :
Authentication Id : 0 ; 2855937 (00000000:002b9401)
Session : Interactive from 1
User Name : dbvictim
Domain : DESKTOP-N86U6RI
Logon Server : DESKTOP-N86U6RI
Logon Time : 2025-11-27 12:11:49 PM
SID : S-1-5-21-527815699-547650139-4209623718-1001
msv :
[00000003] Primary
* Username : dbvictim
* Domain : DESKTOP-N86U6RI
* NTLM : c85da1a30ee68f95d635a55562290a2b
* SHA1 : 5fa5e93e76ac32aea0e22caff512ff9cdf891ef0
* DPAPI : 5fa5e93e76ac32aea0e22caff512ff9c
tspkg :
wdigest :
* Username : dbvictim
* Domain : DESKTOP-N86U6RI
* Password : (null)
kerberos :
* Username : dbvictim
* Domain : DESKTOP-N86U6RI
* Password : (null)
ssp : KO
credman :
Authentication Id : 0 ; 81206 (00000000:00013d36)
Session : Interactive from 1
User Name : DWM-1
Domain : Window Manager
Logon Server : (null)
Logon Time : 2025-11-27 12:01:39 PM
SID : S-1-5-90-0-1
msv :
tspkg :
wdigest :
* Username : DESKTOP-N86U6RI$
* Domain : WORKGROUP
* Password : (null)
kerberos :
ssp : KO
credman :
Authentication Id : 0 ; 81177 (00000000:00013d19)
Session : Interactive from 1
User Name : DWM-1
Domain : Window Manager
Logon Server : (null)
Microsoft Store



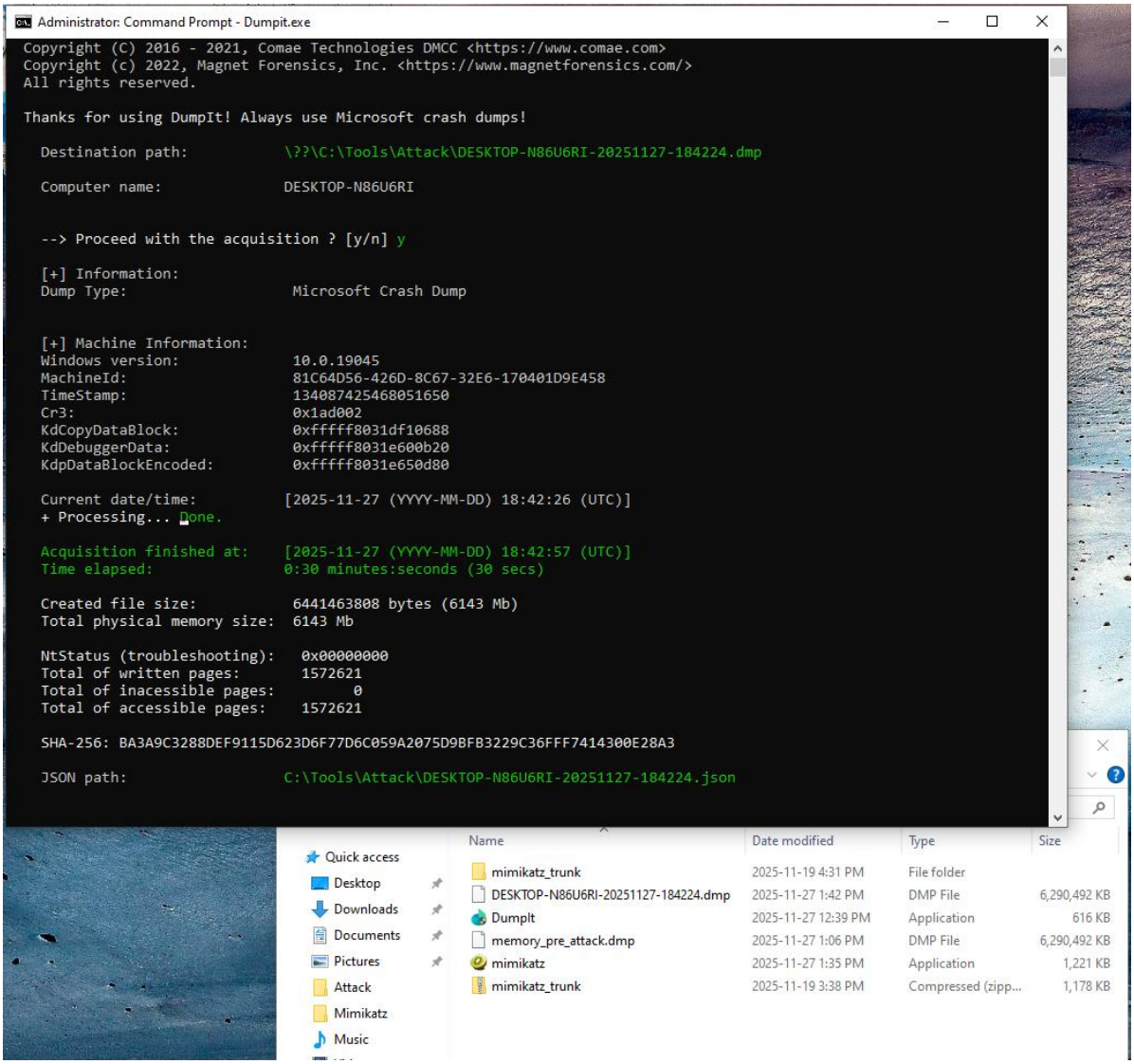
These illustrate the execution of sekurlsa::logonpasswords and other credential-theft modules.

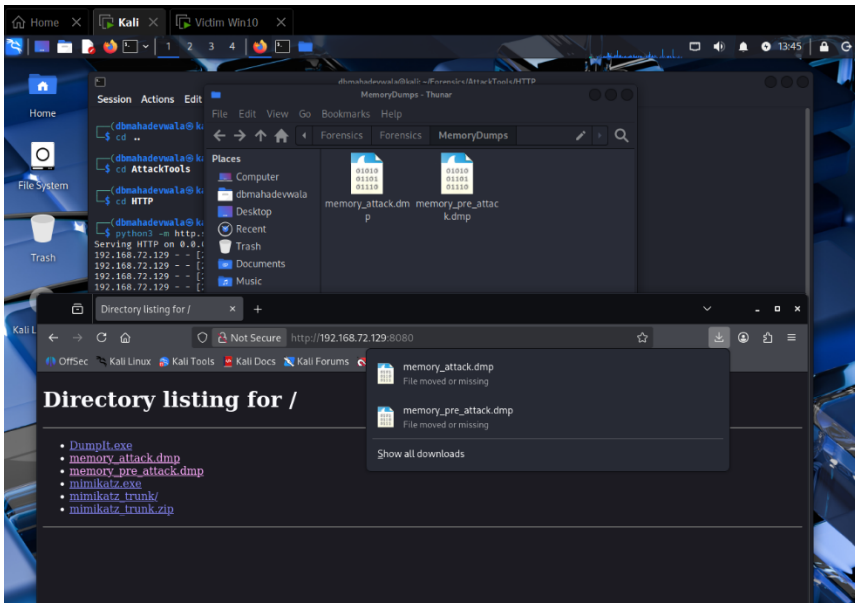
2.5 Memory Capture During the Attack

With Mimikatz running, a second memory dump was captured. This ensures the malicious activity—including active credential harvesting—is preserved for forensic analysis.

2.5.1 DumpIt Executed During Live Mimikatz Session

DumpIt was executed again to produce the “attack-state” memory image.

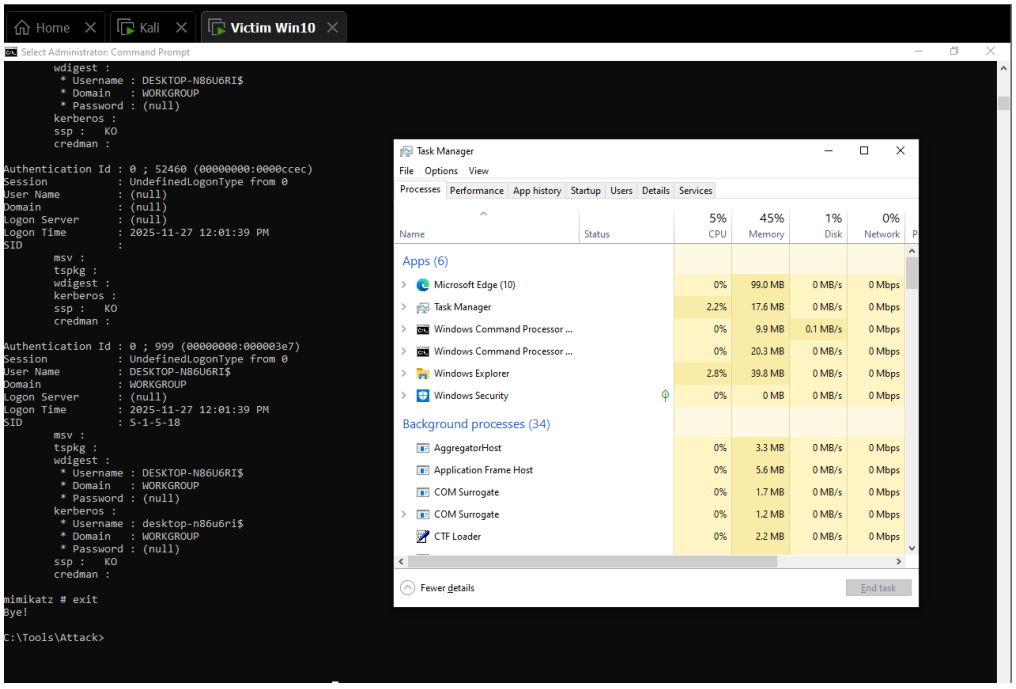




Both attacker and victim views confirm successful capture and transfer of the dump to the analysis environment.

2.5.2 Termination of Mimikatz

After memory acquisition, the Mimikatz process was terminated.



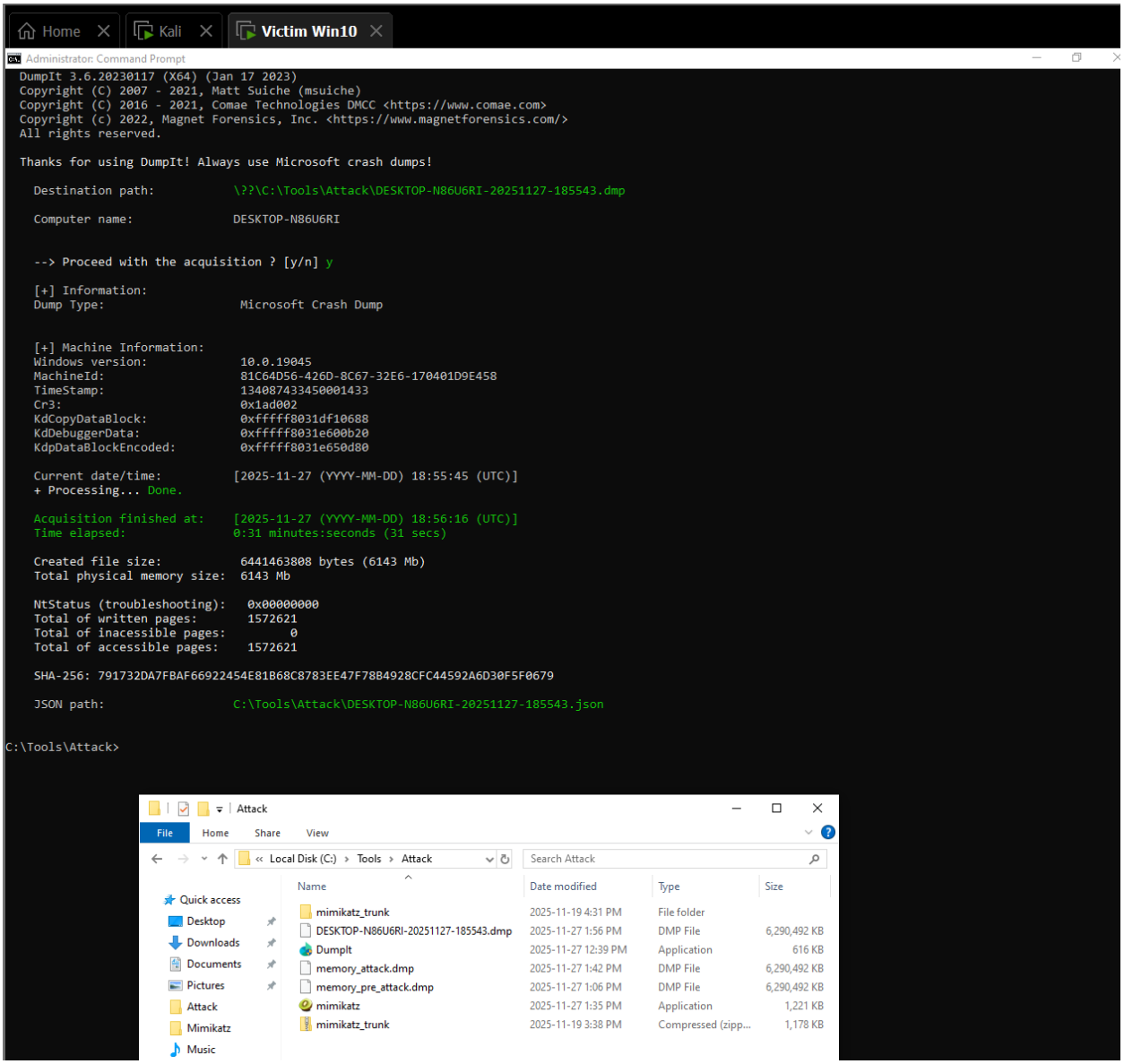
This marks the end of active compromise activity.

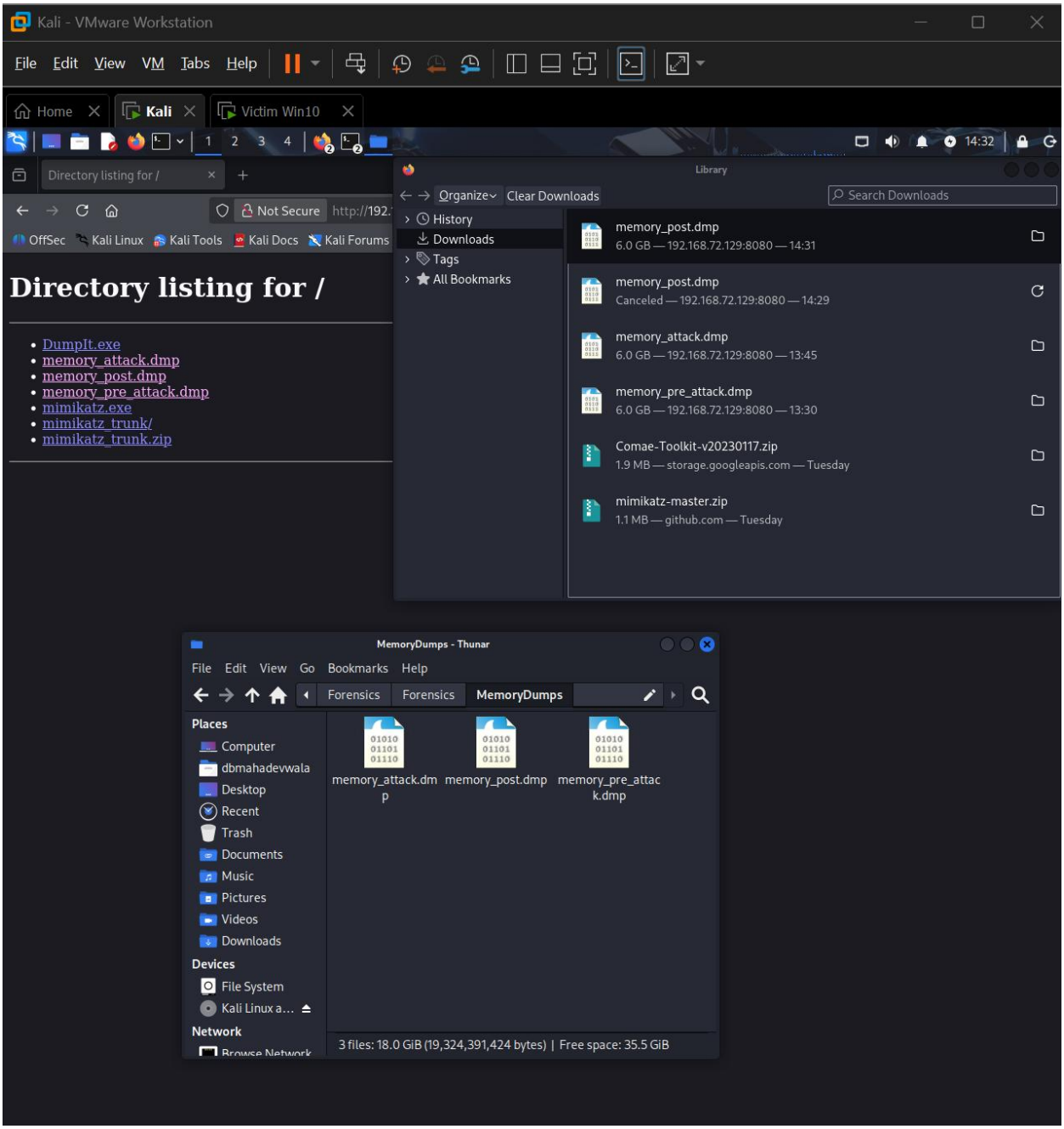
2.6 Post-Attack Memory Acquisition

To create a third comparison point, a final memory dump was taken after all malicious activity ceased.

2.6.1 Capturing the Post-Attack Dump

DumpIt was executed once more, collecting the memory state after the attack had been cleaned up.



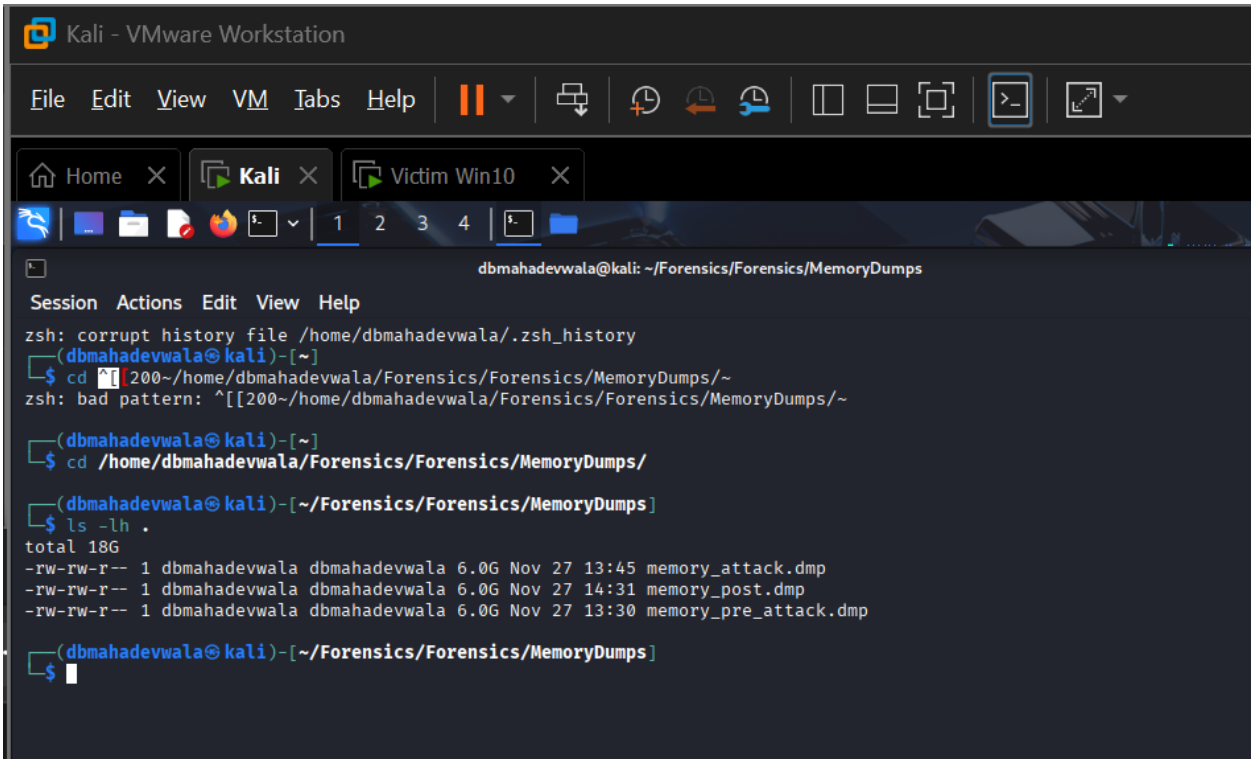


This dump allows examiners to differentiate between persistent artifacts and transient attack elements.

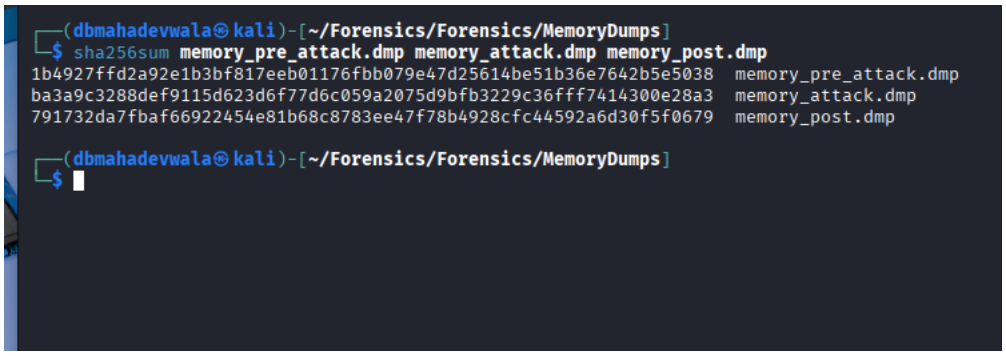
Phase 3 – Memory Forensics with Volatility 3 (Pre-, During- and Post-Attack)

This phase documents how the three memory images were analysed with Volatility 3 and how the results were used to reconstruct the attack and its impact. The analysis follows the incident-response timeline: baseline (pre-attack), live attack, and post-attack state, then compares them side-by-side.

3.1 Tool preparation and image integrity

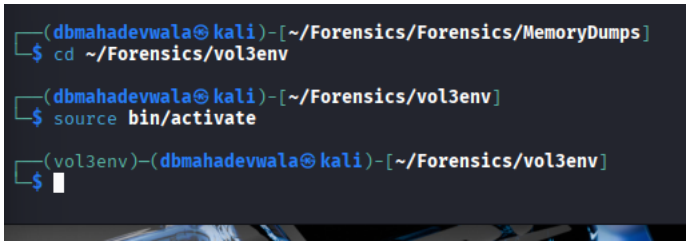


This screenshot lists the three memory images (memory_pre_attack.dmp, memory_attack.dmp, memory_post.dmp) in the MemoryDumps directory. It establishes that all later analysis is based on separate acquisition points and that the filenames clearly encode the capture time relative to the attack.



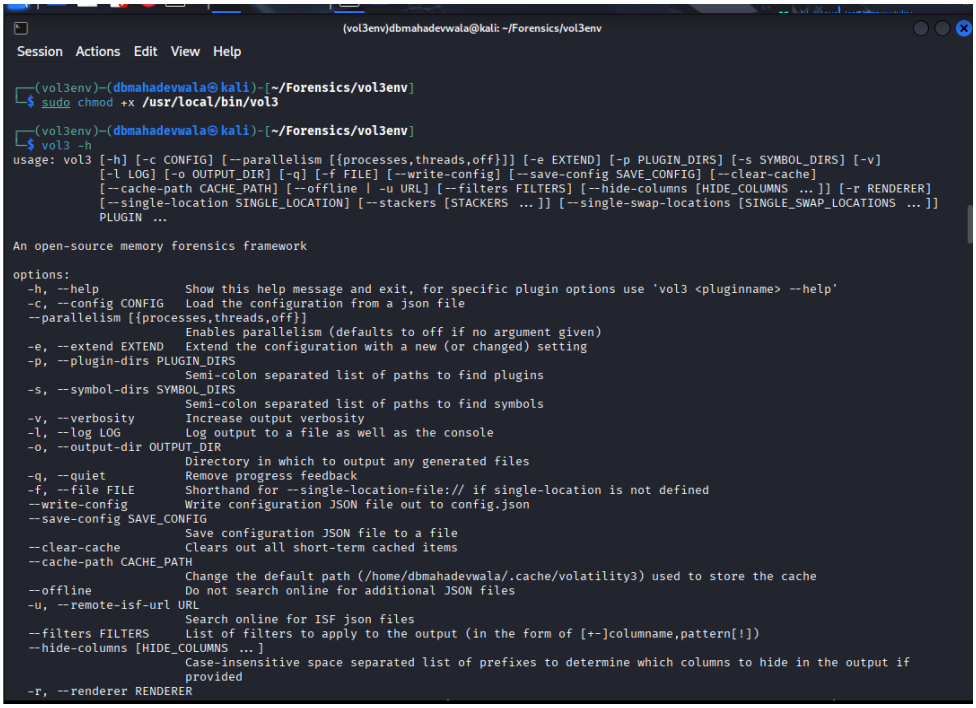
The terminal output shows distinct SHA-256 hashes for each dump. From a DFIR standpoint this proves:

- Each file is a different capture (no accidental copy-paste of the same image), and
- The hashes can be reused later for chain-of-custody and integrity verification in reports or case notes.

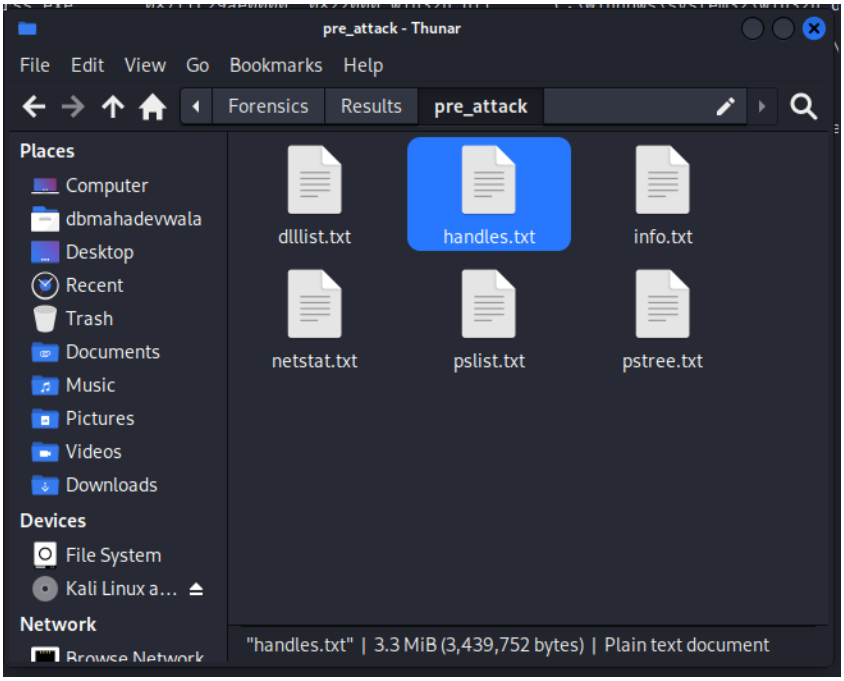


Here the Volatility 3 virtual environment (vol3env) is activated. This demonstrates that:

- Analysis was done in an isolated Python environment,
- Dependencies are pinned and won't be affected by system-wide package changes, and
- All subsequent vol3 commands were executed with consistent tooling.



This shows the vol3 -h output. It documents the exact Volatility 3 framework version and available plugins. Including this in the report shows the tool's capabilities at the time of analysis and supports reproducibility: another analyst can confirm they are using the same version and plugin set.



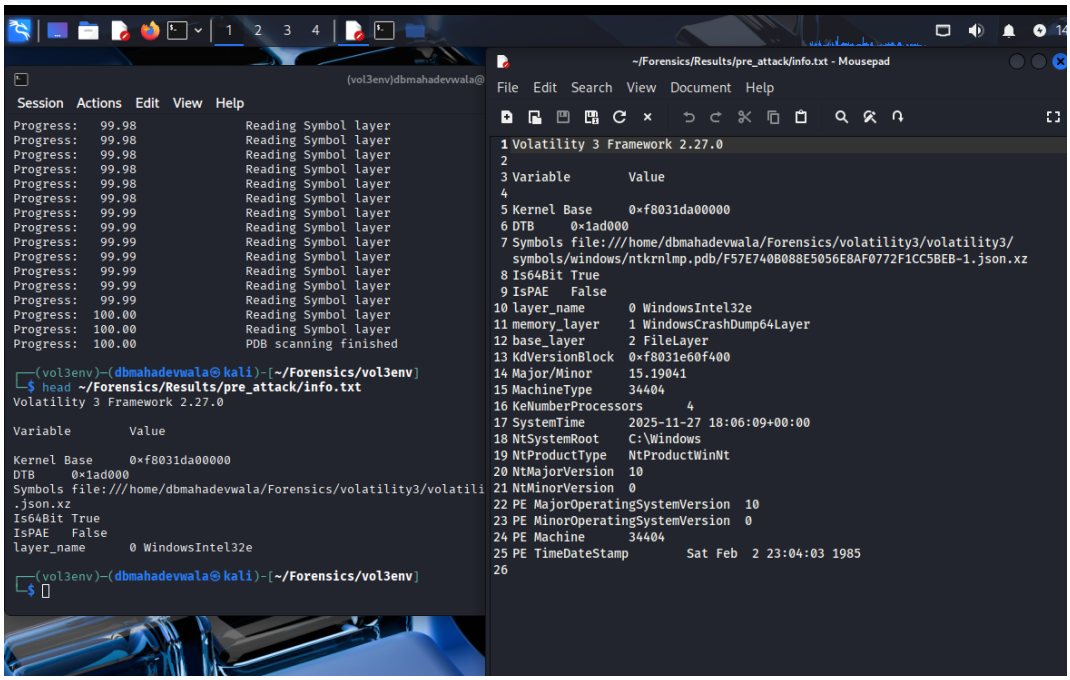
This screenshot captures Volatility downloading symbol files and building the Windows profile the first time the image is analysed. From a forensic perspective this is important because:

- It confirms Volatility successfully identified the OS build and kernel symbols,
- It reduces later ambiguity around plugin reliability, and
- It justifies trusting the subsequent plugin output (pslist, dlllist, handles, etc.) as structurally correct for this Windows build.

3.2 Baseline analysis – pre-attack memory image

The pre-attack image represents a clean but fully configured system. This baseline is critical for distinguishing normal activity from malicious artefacts in later dumps.

3.2.1 System profile and environment

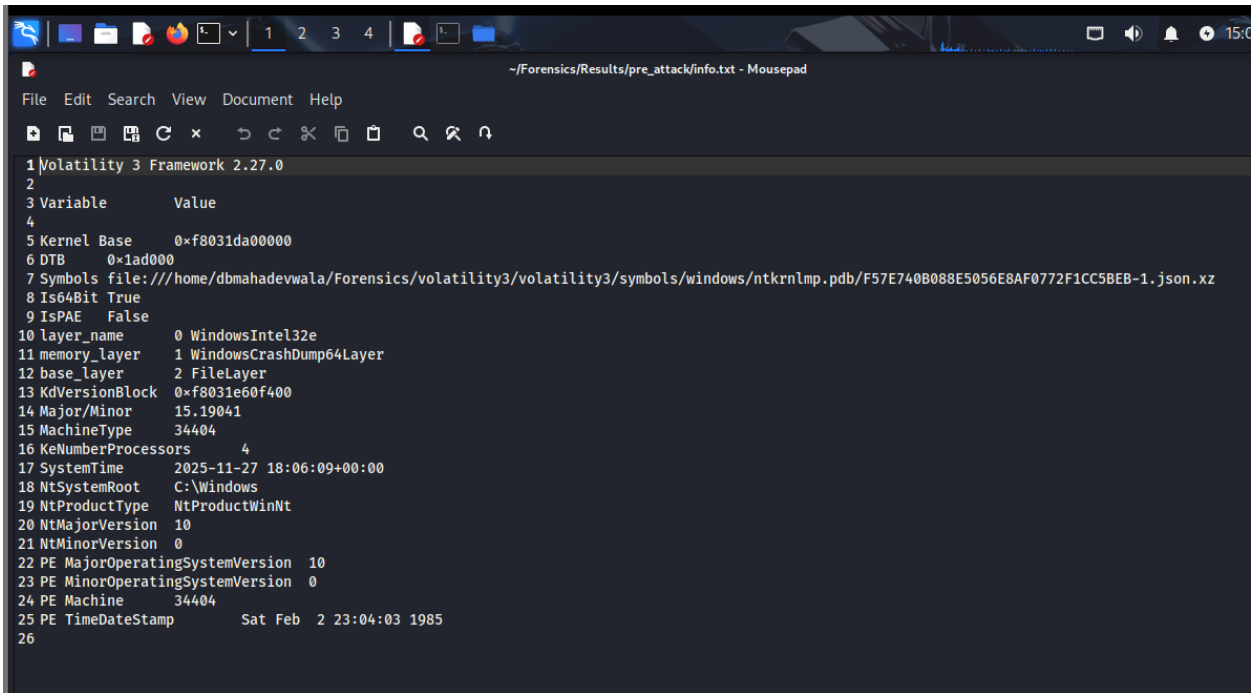


The windows.info.Info / banner output shows details such as:

- Windows version and build number,
- Kernel base address and DTB,
- Processor architecture (64-bit), number of cores,
- System time at the moment of acquisition.

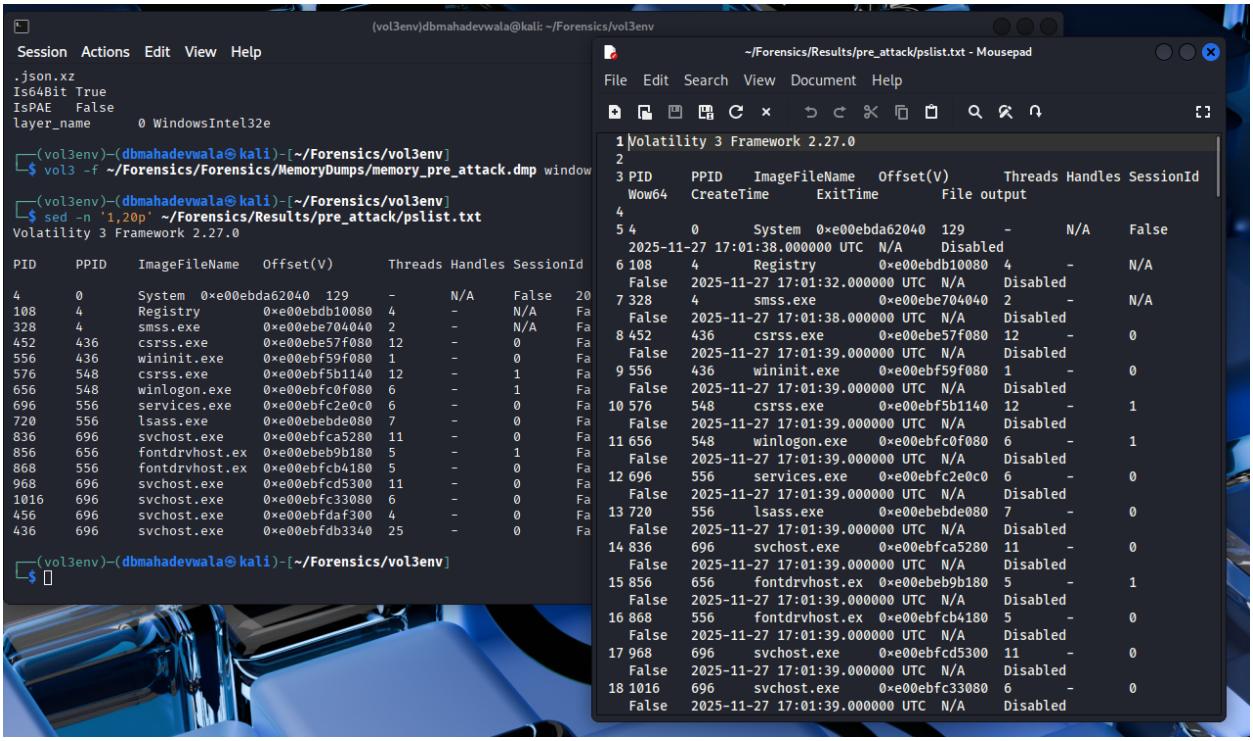
This establishes:

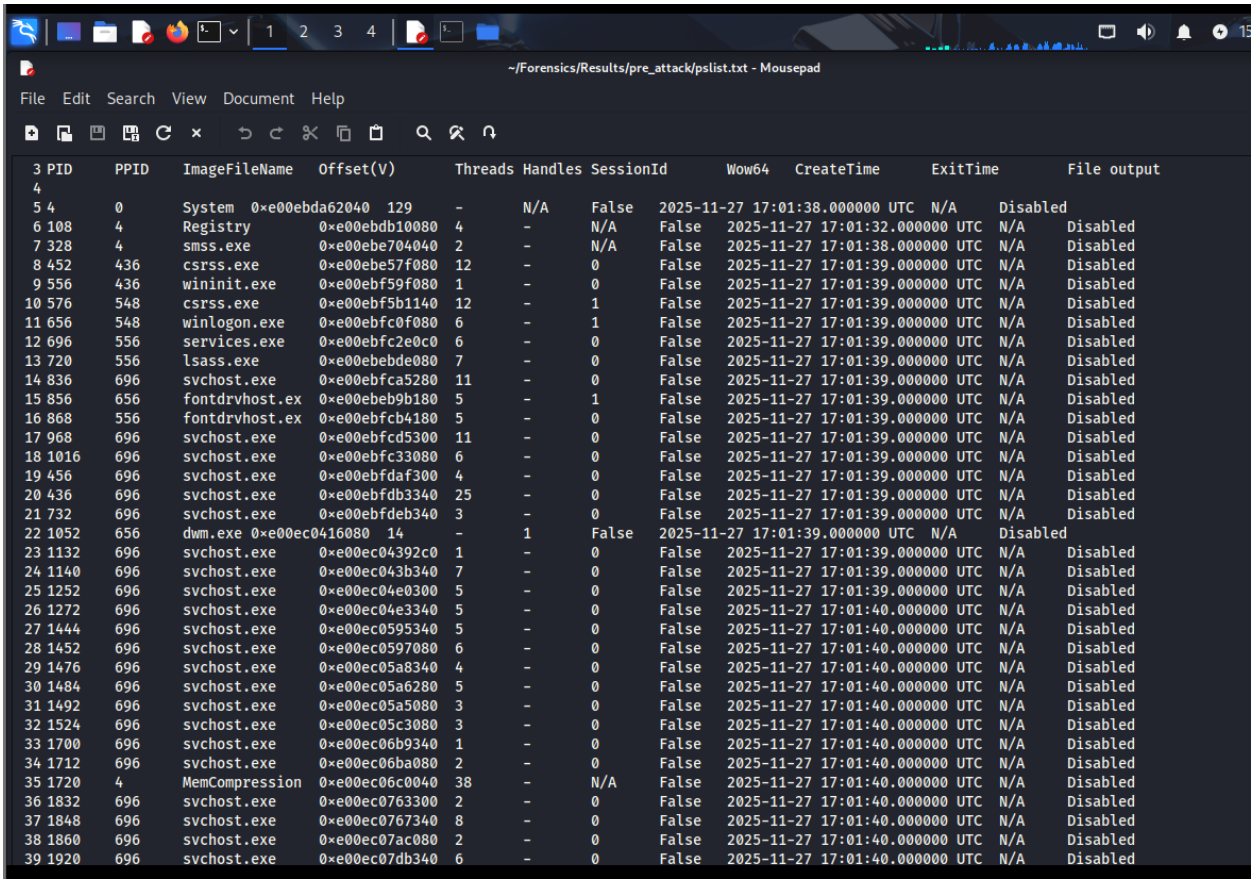
- The victim is a 64-bit Windows 10 host,
- The acquisition timestamp matches the expected “pre-attack” timeline, and
- There is no obvious sign of tampering with basic OS metadata (e.g., wrong time zone or impossible boot time).

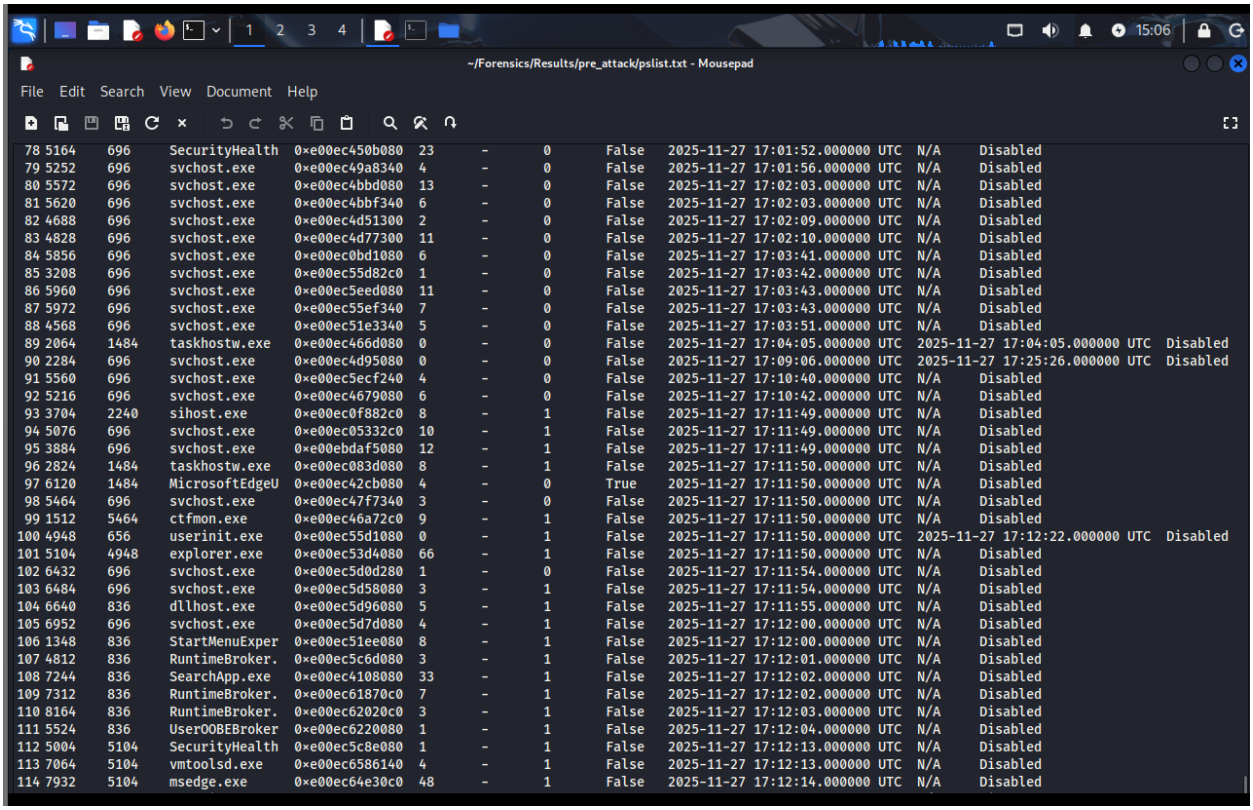


This is the same information exported to a text file under Results/pre_attack/info.txt. Documenting that the output was saved shows proper forensic workflow: analysis is not just done interactively but preserved for later review, peer validation or scripting.

3.2.2 Baseline processes



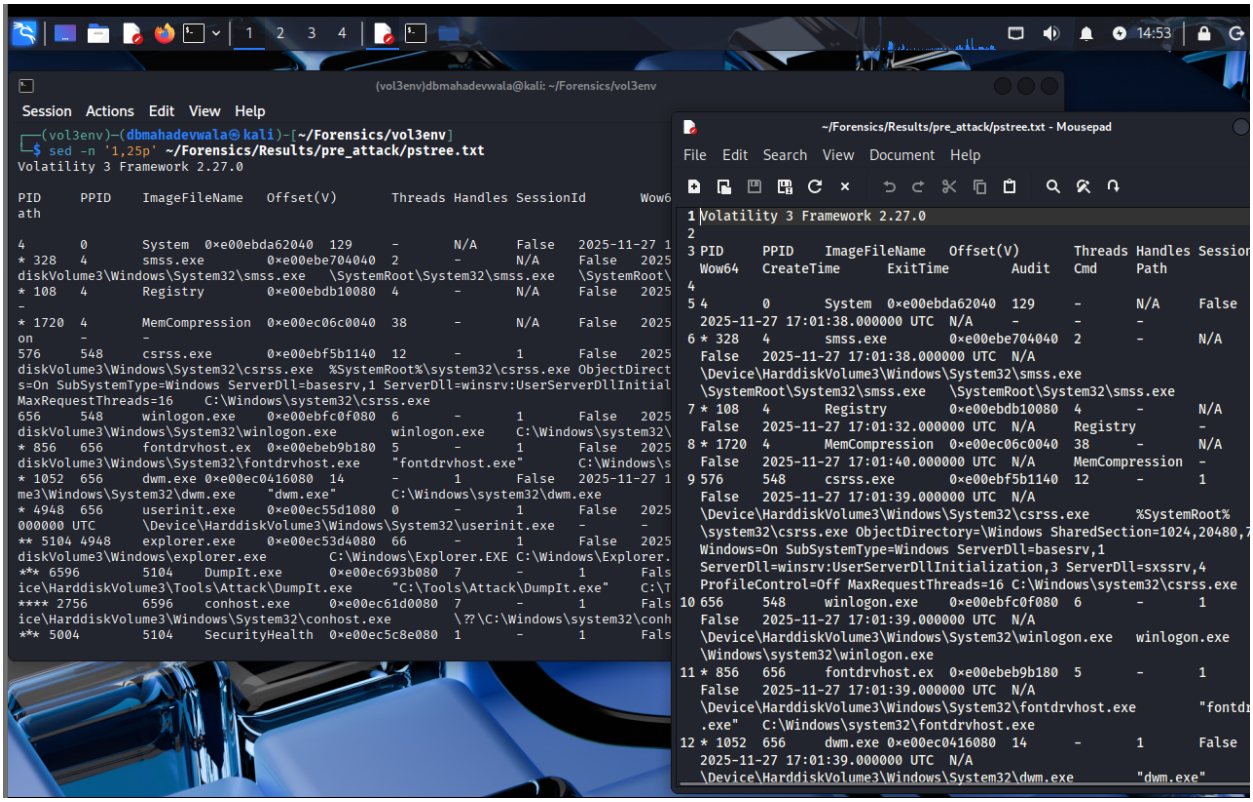


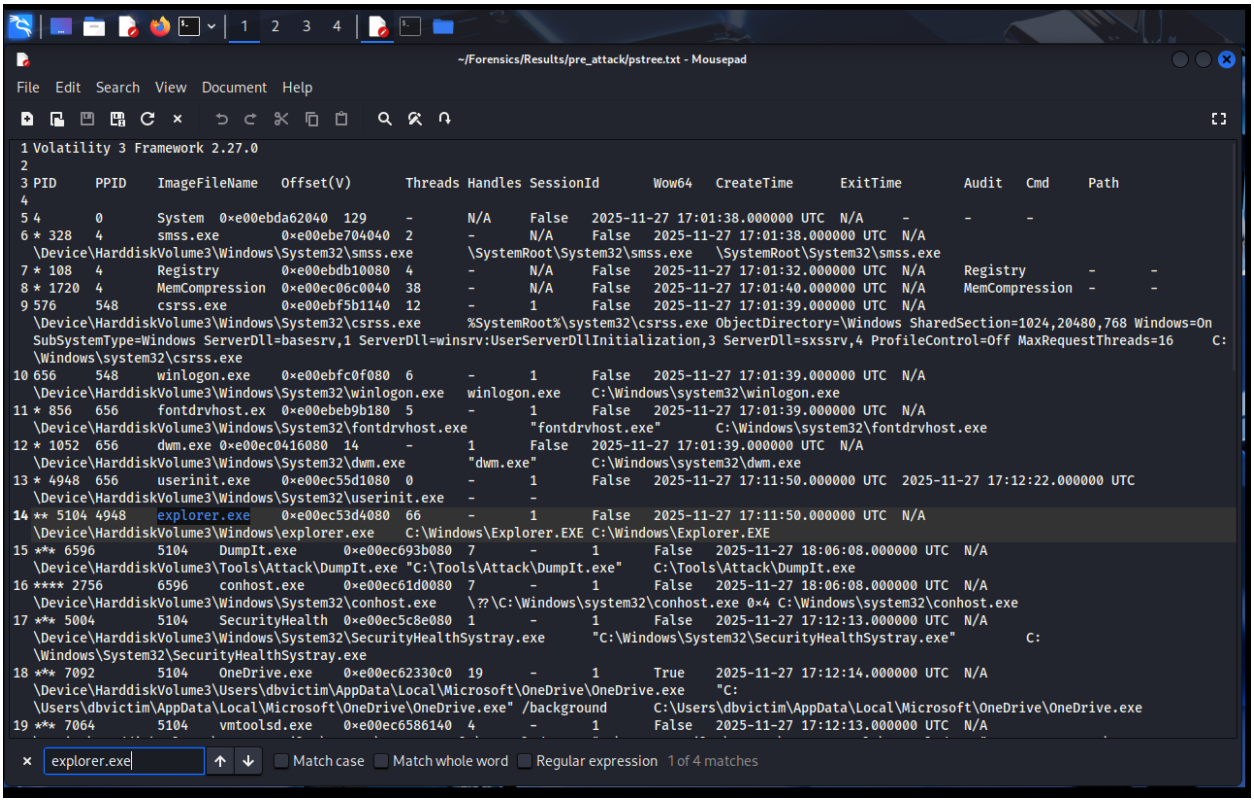


These screenshots show the process list produced by windows.pslist.PsList against the pre-attack dump. Key observations:

- Only core Windows processes and standard userland applications appear: System, smss.exe, csrss.exe, wininit.exe, services.exe, lsass.exe, multiple svchost.exe instances, explorer.exe, RuntimeBroker.exe, StartMenuExperienceHost.exe, Edge/OneDrive, etc.
- No instances of DumpIt.exe, mimikatz.exe, suspicious PowerShell, or unknown binaries.
- Timestamps (CreateTime) cluster around system boot and user login and precede the planned attack window.

This pslist output serves as the “known good” reference. Any additional processes that show up only in the later dumps immediately stand out as candidate malicious or investigation-relevant binaries.


(vol3env)dbmahadevwala@kali: ~/Forensics/vol3env
Session Actions Edit View Help
(vol3env)-(dbmahadevwala㉿kali)-[~/Forensics/vol3env]
$ sed -n '1,25p' ~/Forensics/Results/pre_attack/pstree.txt
Volatility 3 Framework 2.27.0
~/Forensics/Results/pre_attack/pstree.txt - Mousepad
File Edit Search View Document Help

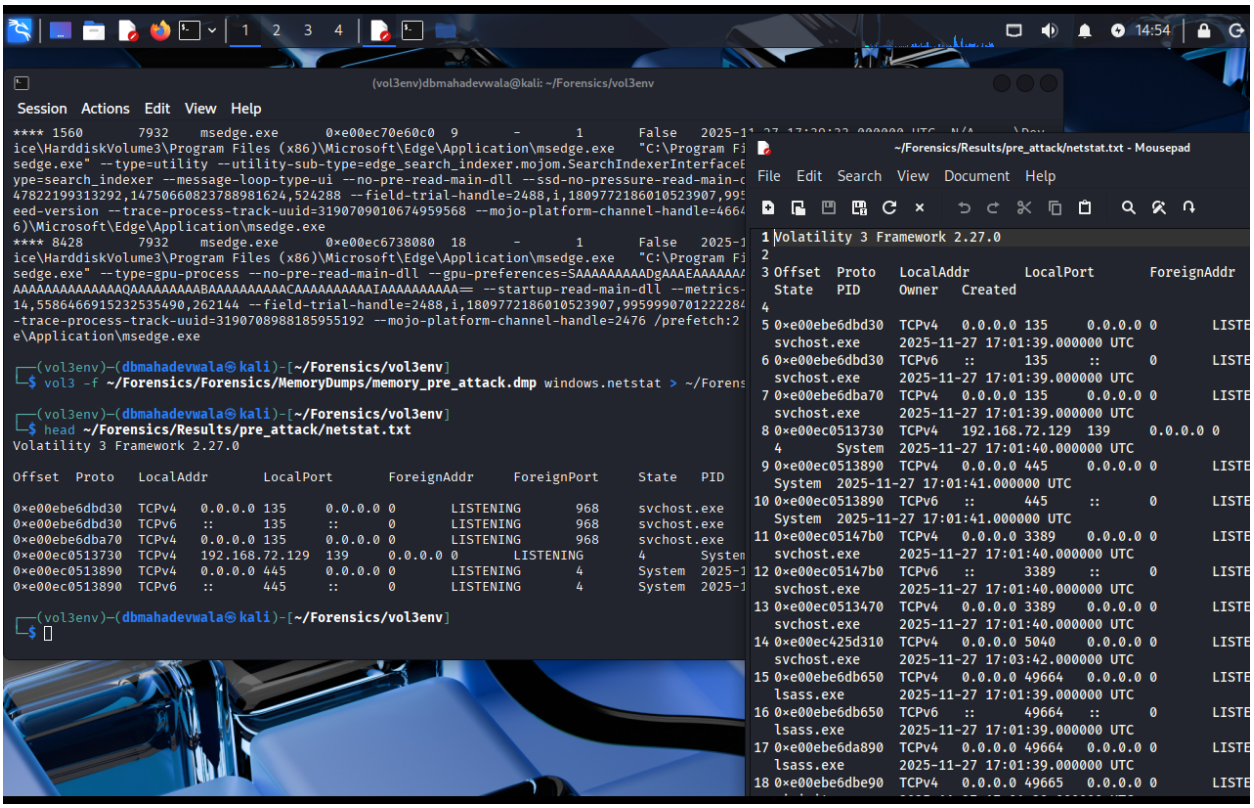

~/Forensics/Results/pre_attack/pstree.txt - Mousepad
File Edit Search View Document Help
explorer.exe
Match case
Match whole word
Regular expression
1 of 4 matches

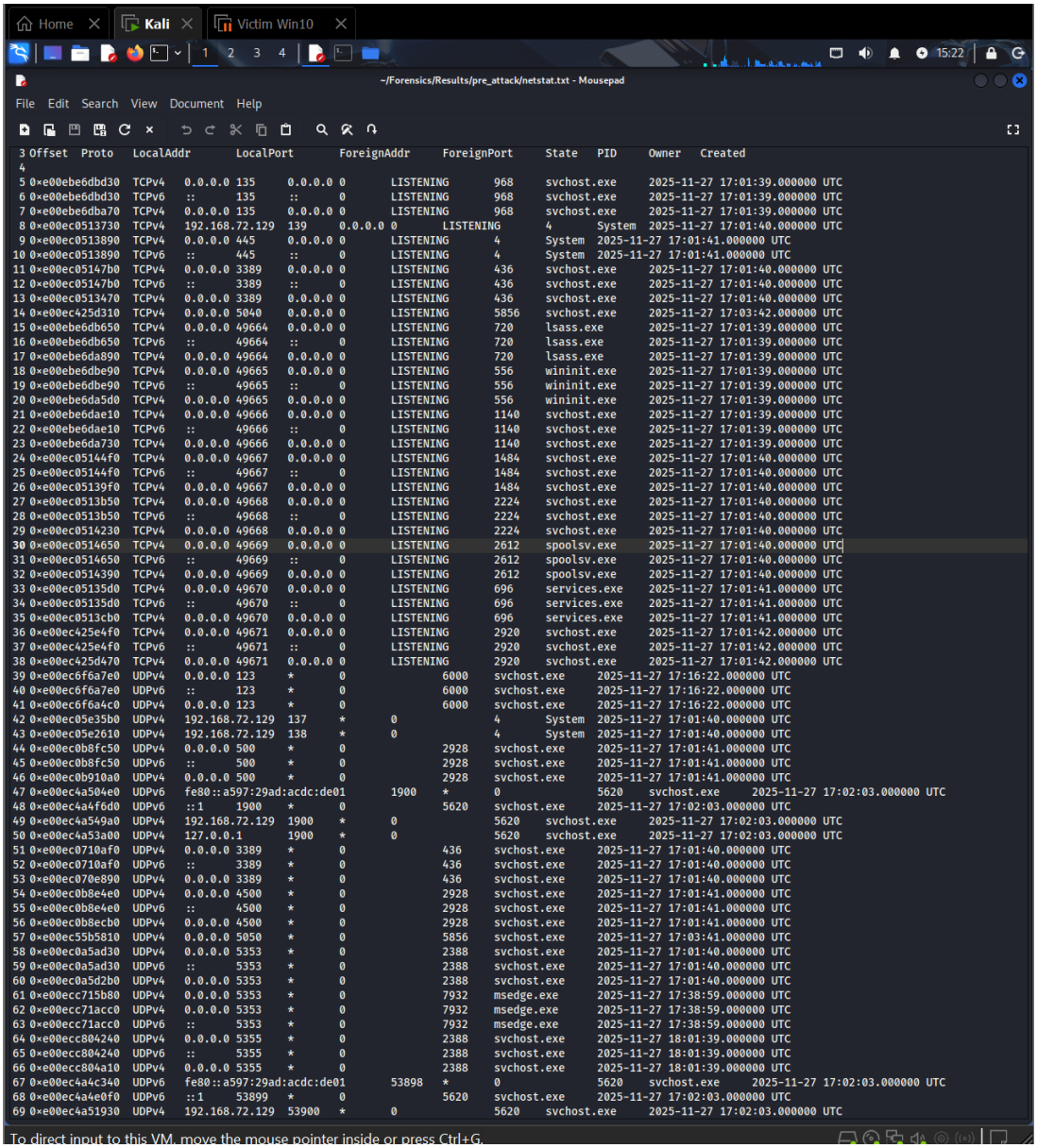
The windows.pstree.PsTree output visualises parent/child relations. In the baseline:

- explorer.exe is spawned by userinit.exe under winlogon.exe, reflecting a normal interactive logon.
- Downstream processes from explorer.exe are benign: standard shell components, OneDrive sync, Edge, etc.
- There is no abnormal command-line tool (e.g., cmd.exe, powershell.exe) chained from explorer.exe at this stage.

This hierarchical view will be essential later when we show cmd.exe → conhost.exe → DumpIt.exe / mimikatz.exe appearing only in the attack image.

3.2.3 Baseline network activity



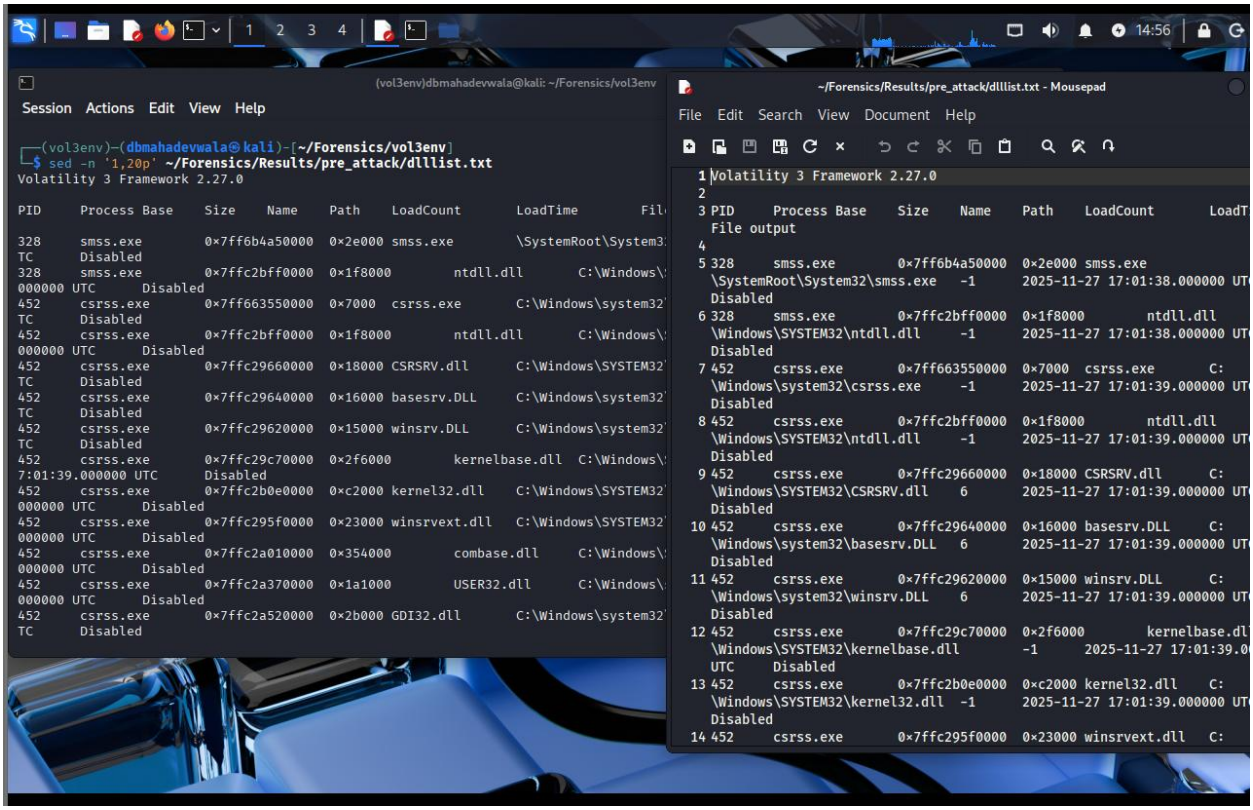


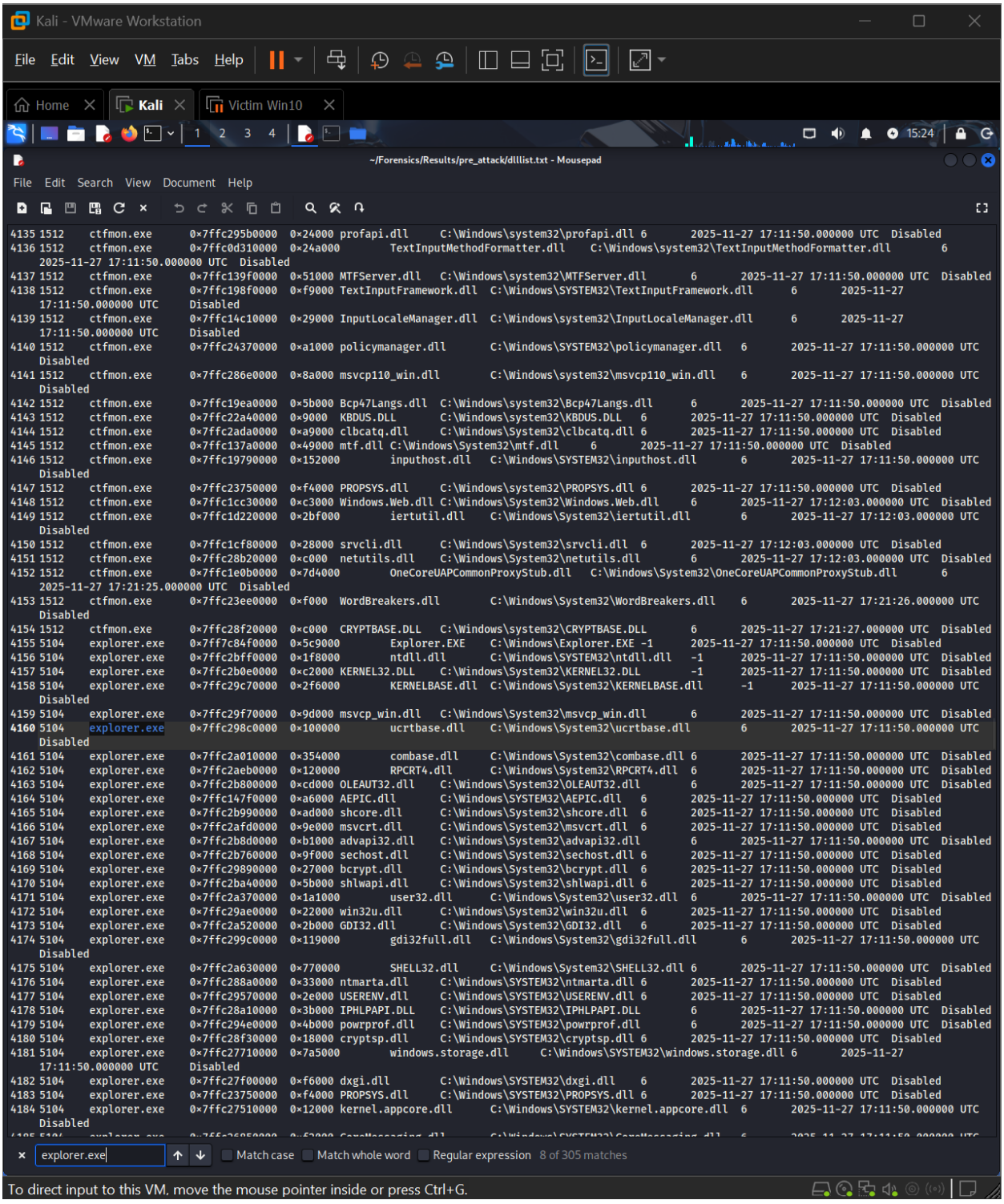
The windows.netstat.NetStat output lists TCP and UDP connections and listeners:

- Local ports primarily associated with svchost.exe (Windows services), lsass.exe, and system components.
- No outbound connections to suspicious external IPs; traffic is limited to the local network and routine Windows services.
- There is no established connection linked to attacker tools at this time.

Saving this output provides a reference for normal port usage. Later we will confirm that the attack phase does not introduce external C2 channels, supporting the conclusion that this was a local / hands-on intrusion scenario.

3.2.4 Baseline DLL inventory





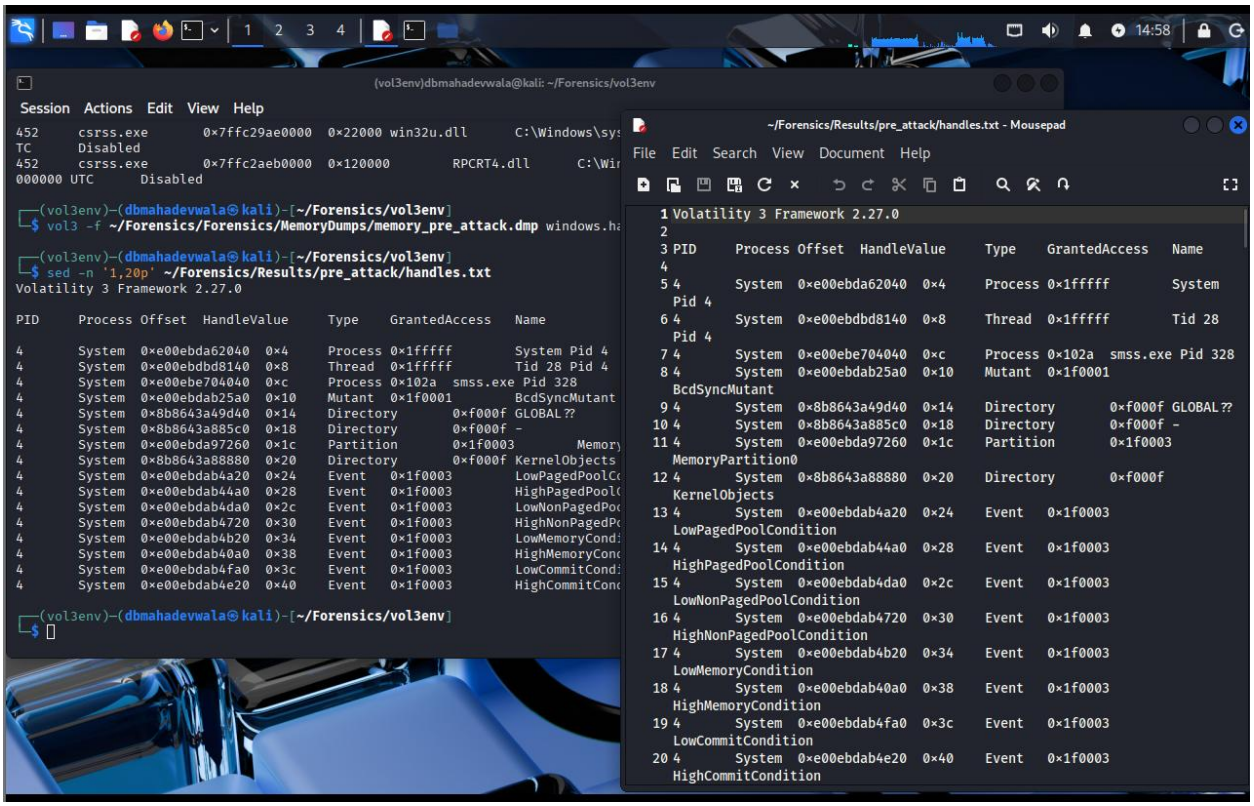
Using windows.dlllist.DllList, the DLLs loaded by explorer.exe are enumerated. The screenshot highlights:

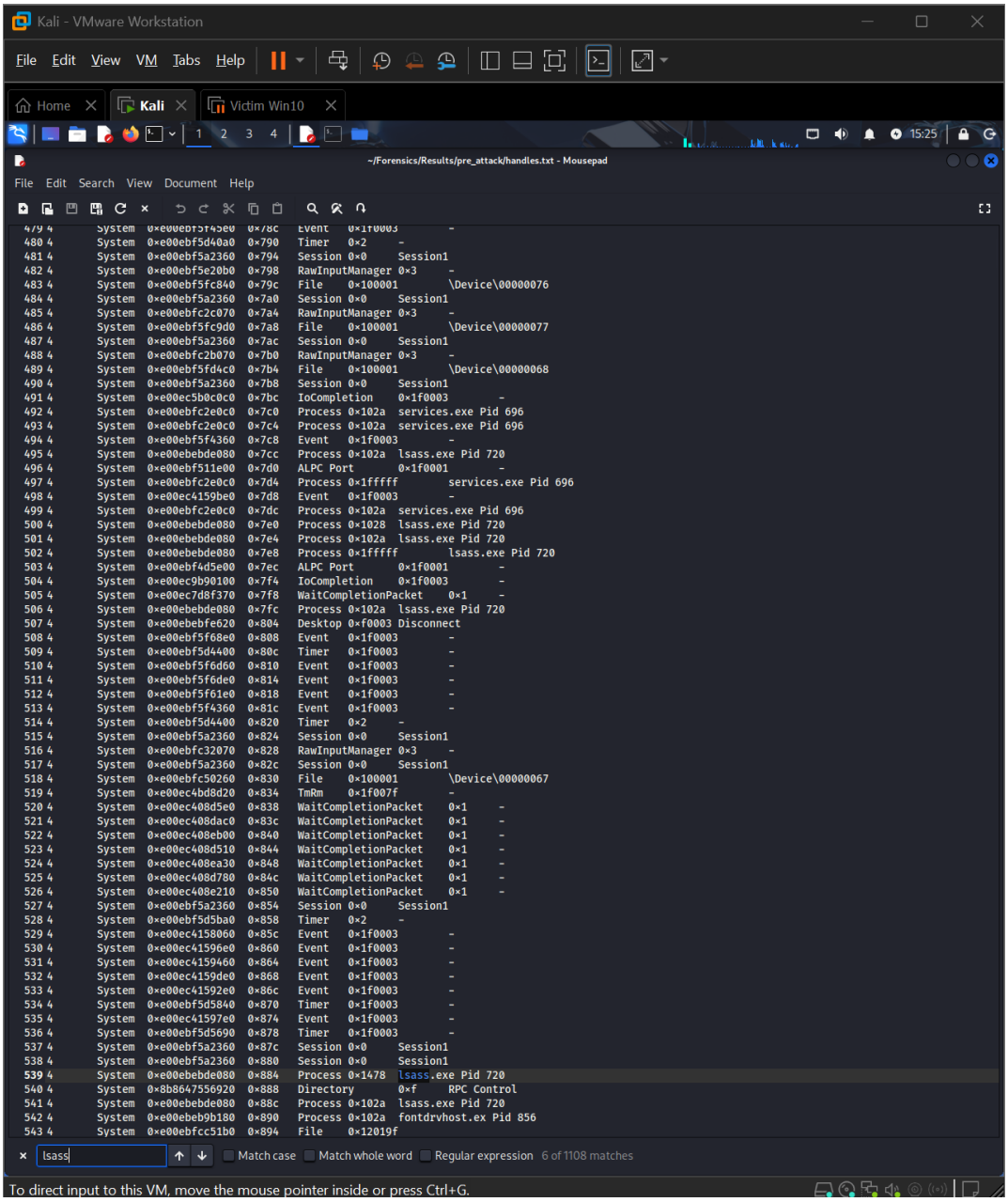
- Only signed, Microsoft-supplied modules located under C:\Windows\System32 and related OS folders.

- No modules mapped from user-writable locations such as C:\Users\<user>\AppData\... or from the attack tools directory.
- Timestamps and load addresses consistent with a stable, long-running explorer session.

This provides strong evidence that prior to the attack, the shell process was not yet interacting with or loading any binaries from the planned attack directory (C:\Tools\Attack / Mimikatz).

3.2.5 Baseline handle usage





The windows.handles.Handles output in the baseline is used to understand “normal” object access:

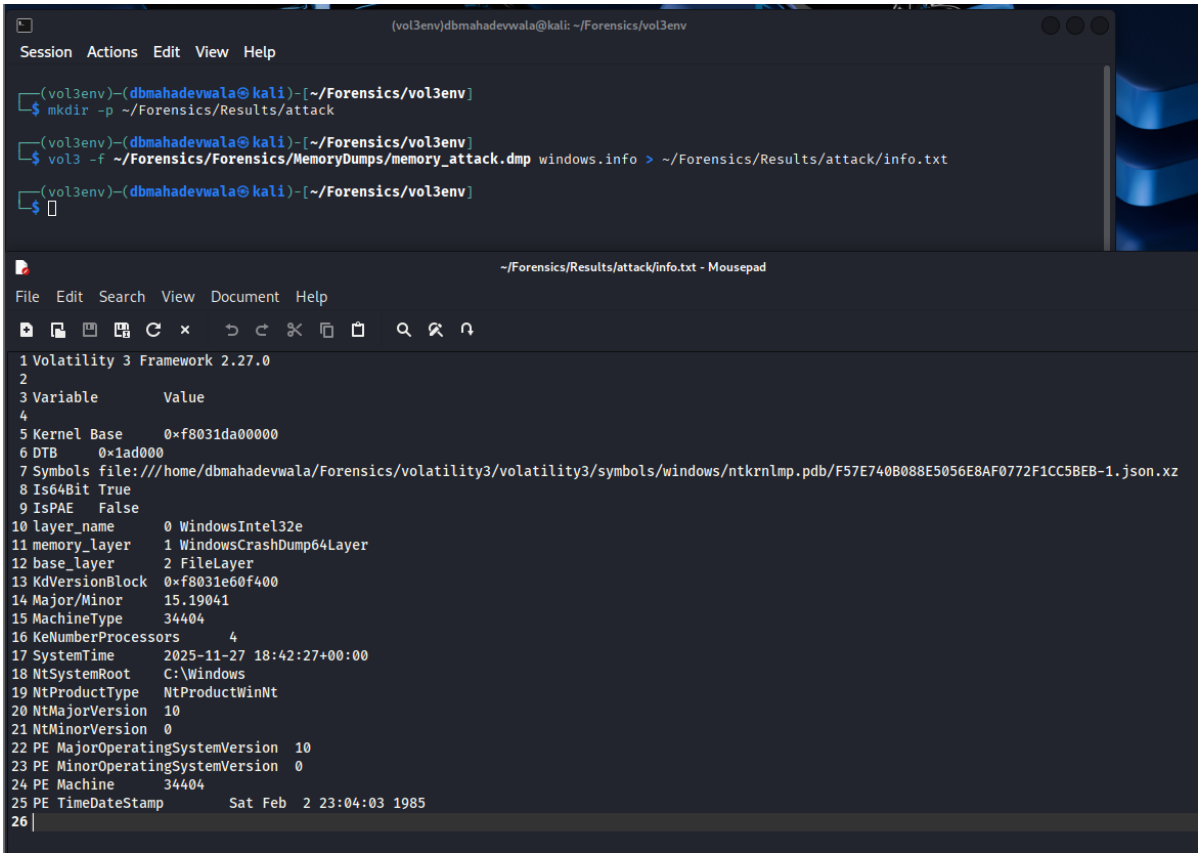
- lsass.exe holds token, key, and named pipe handles related only to its core authentication duties.
- There are no handles from unusual processes into lsass.exe or sensitive security objects.
- File and device handles generally reference system paths, registry keys, and benign services.

This clean handle landscape will later be contrasted against the attack dump, where mimikatz.exe opens sensitive handles into lsass.exe and where DumpIt.exe interacts with low-level devices such as \Device\KsecDD.

3.3 During-attack analysis – live compromise image

The second memory image captures the system while the attack is in progress: DumpIt.exe is acquiring memory and mimikatz.exe is harvesting credentials. This is where the bulk of forensic evidence appears.

3.3.1 System profile during the attack

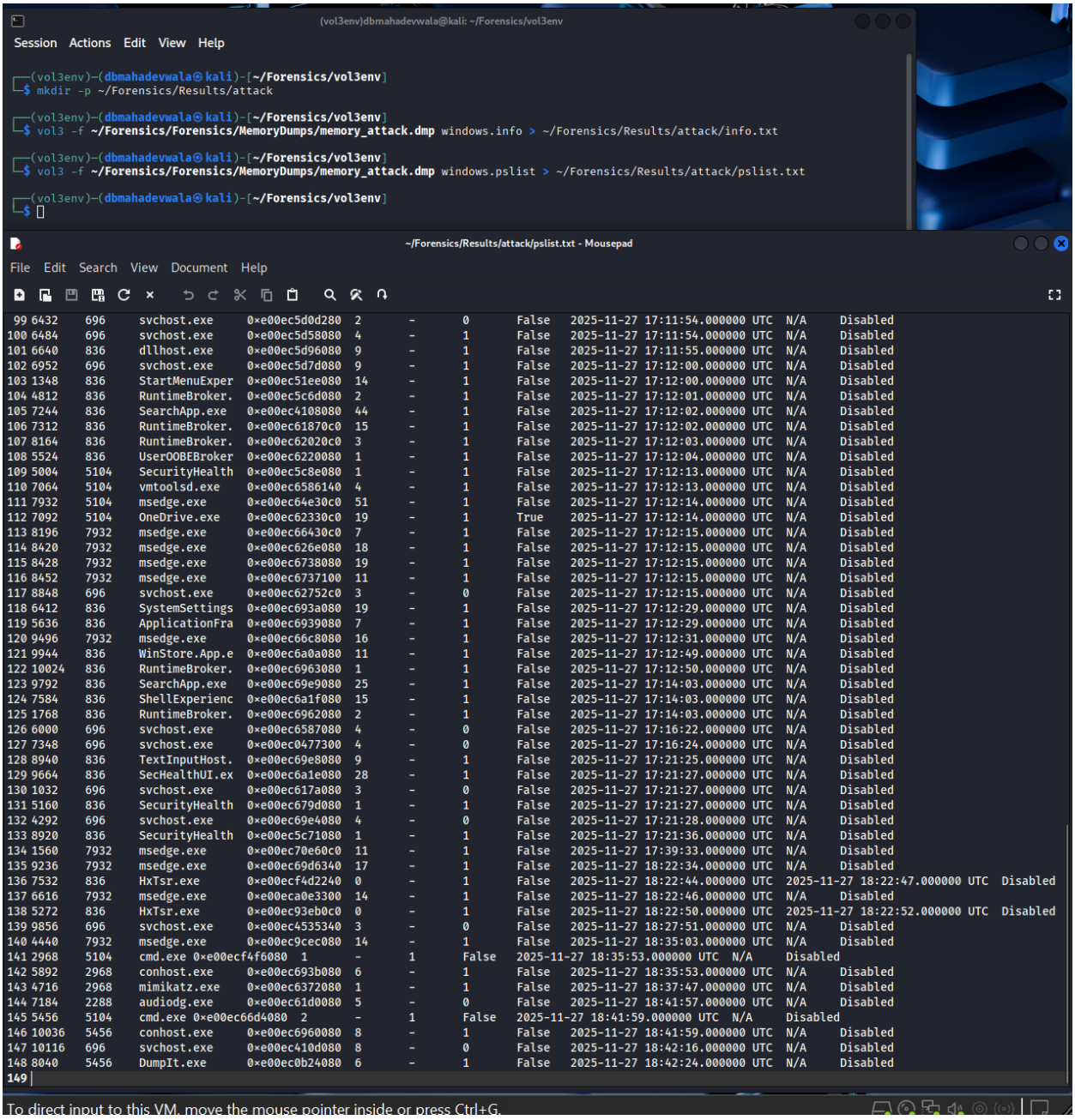


The windows.info output for memory_attack.dmp shows:

- Same OS version, kernel, and architecture as the baseline, confirming we are dealing with the same host and configuration.
- A later system time that aligns with the planned attack window.

This temporal anchor is important: it demonstrates that the malicious execution occurred after the baseline and before the post-attack snapshot.

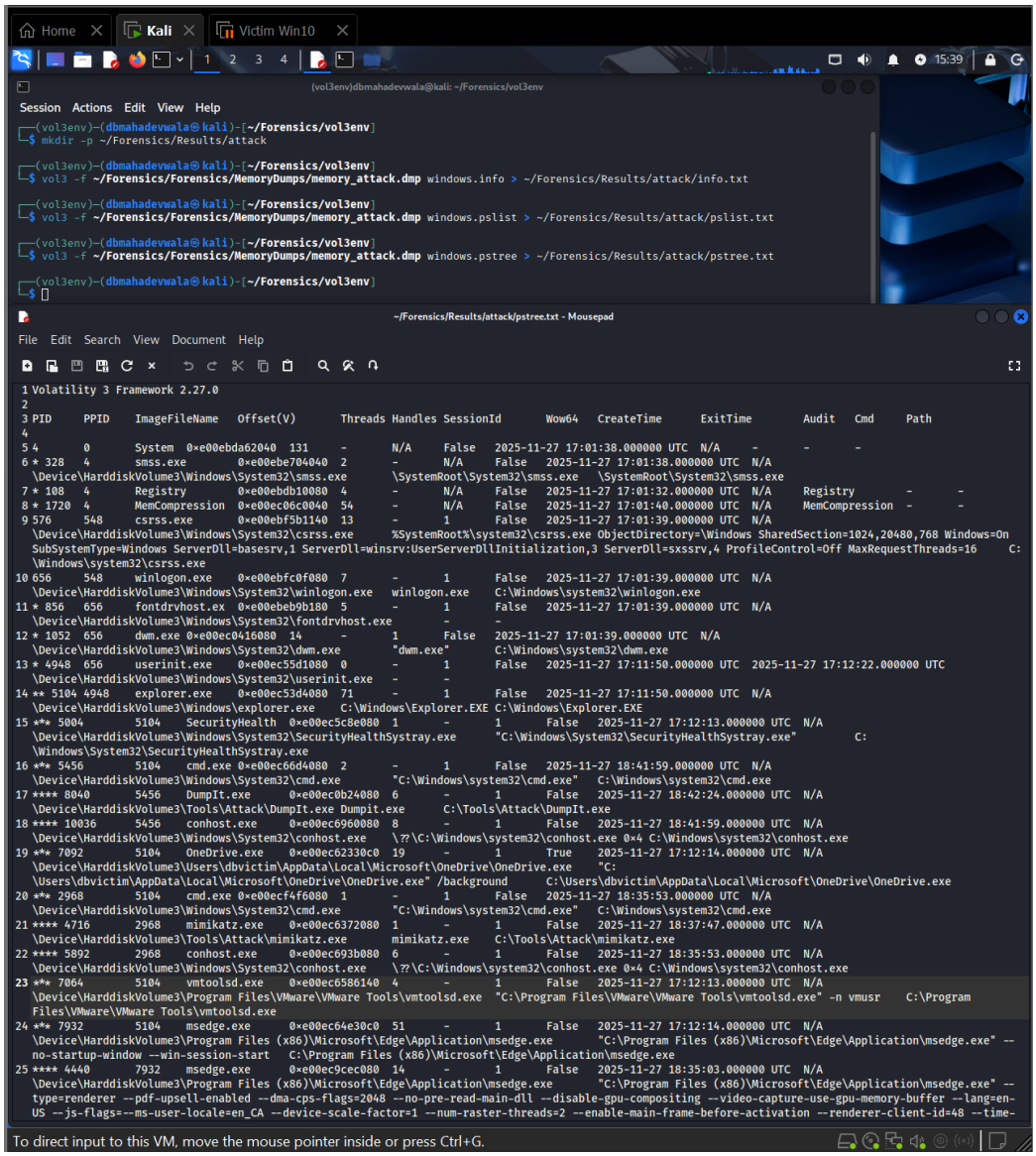
3.3.2 Malicious processes identified



The pslist output now includes new processes that were absent in the baseline:

- DumpIt.exe with a specific PID, typically a child of cmd.exe.
- An associated conhost.exe instance sharing the same console session.
- Depending on timing, mimikatz.exe may or may not still be running as a process at the moment of the dump; either way, subsequent artefacts confirm it executed.

Comparing this pslist to the pre-attack pslist immediately flags DumpIt.exe and the additional conhost.exe as suspicious. They do not belong to normal Windows operation and are directly linked to memory acquisition activity.



The pstree screenshot is one of the most important evidentiary artefacts:

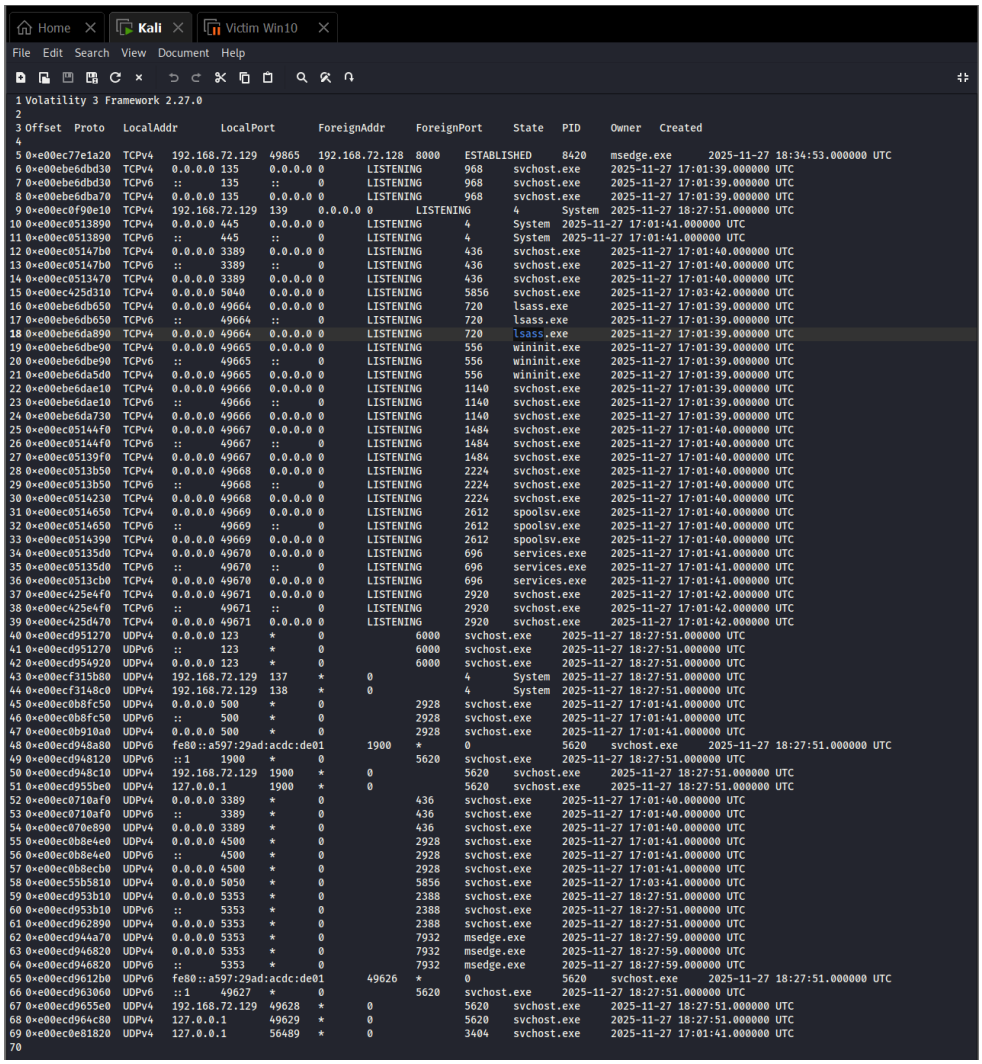
- explorer.exe spawns cmd.exe, which is consistent with the attacker launching a command prompt from the GUI session.
- cmd.exe then spawns conhost.exe (console host).
- From this cmd.exe / conhost.exe chain, both DumpIt.exe and mimikatz.exe appear as child processes.

This tree reconstructs the attacker's execution path:

User shell (explorer.exe) → command shell (cmd.exe) → console host (conhost.exe) → attacker tools (DumpIt.exe, mimikatz.exe)

From an IR perspective this proves that the malicious tooling was launched interactively under the logged-in user context, and not as a hidden service or scheduled task.

3.3.3 Network state during the attack



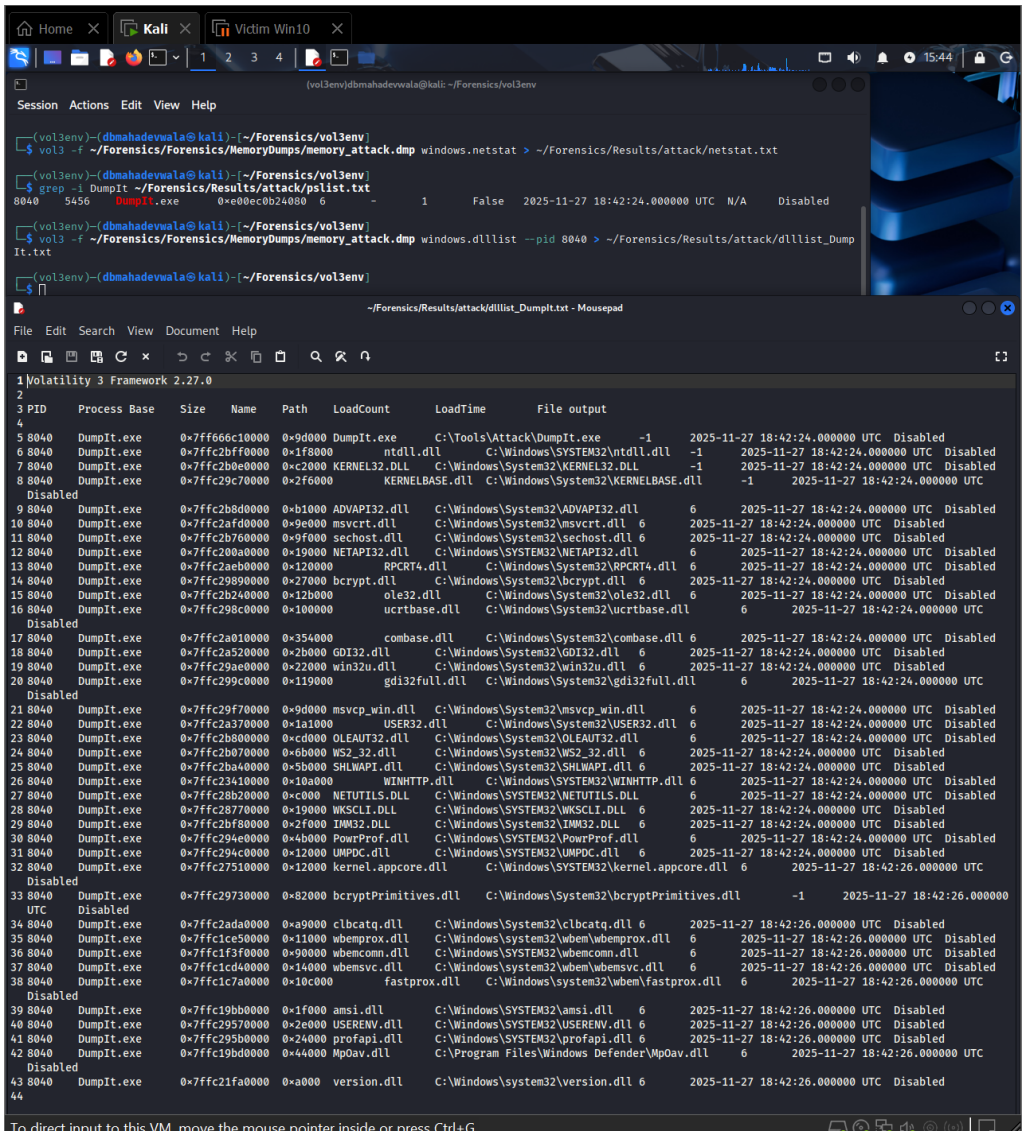
```
Home | Kali | Victim Win10

File  Edit  Search  View  Document  Help

Volatility 3 Framework 2.27.0

Offset  Proto  LocalAddr  LocalPort  ForeignAddr  ForeignPort  State  PID  Owner  Created

0x...  TCPv4  192.168.72.129  49865  192.168.72.128  8000  ESTABLISHED  8420  msedge.exe  2025-11-27 18:34:53.000000 UTC
0x...  TCPv4  0.0.0.0  135  0.0.0.0  0  LISTENING  968  svchost.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv6  ::  135  ::  0  LISTENING  968  svchost.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv4  0.0.0.0  135  0.0.0.0  0  LISTENING  968  svchost.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv4  192.168.72.129  139  0.0.0.0  0  LISTENING  4  System  2025-11-27 18:27:51.000000 UTC
0x...  TCPv4  0.0.0.0  445  0.0.0.0  0  LISTENING  4  System  2025-11-27 17:01:41.000000 UTC
0x...  TCPv6  ::  445  ::  0  LISTENING  4  System  2025-11-27 17:01:41.000000 UTC
0x...  TCPv4  0.0.0.0  3389  0.0.0.0  0  LISTENING  436  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv6  ::  3389  ::  0  LISTENING  436  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv4  0.0.0.0  3389  0.0.0.0  0  LISTENING  436  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv4  0.0.0.0  5040  0.0.0.0  0  LISTENING  5856  svchost.exe  2025-11-27 17:03:42.000000 UTC
0x...  TCPv4  0.0.0.0  49664  0.0.0.0  0  LISTENING  720  lsass.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv6  ::  49664  ::  0  LISTENING  720  lsass.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv4  0.0.0.0  49664  0.0.0.0  0  LISTENING  720  lsass.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv4  0.0.0.0  49665  0.0.0.0  0  LISTENING  556  wininit.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv6  ::  49665  ::  0  LISTENING  556  wininit.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv4  0.0.0.0  49665  0.0.0.0  0  LISTENING  556  wininit.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv4  0.0.0.0  49666  0.0.0.0  0  LISTENING  1140  svchost.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv6  ::  49666  ::  0  LISTENING  1140  svchost.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv4  0.0.0.0  49666  0.0.0.0  0  LISTENING  1140  svchost.exe  2025-11-27 17:01:39.000000 UTC
0x...  TCPv4  0.0.0.0  49667  0.0.0.0  0  LISTENING  1484  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv6  ::  49667  ::  0  LISTENING  1484  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv4  0.0.0.0  49667  0.0.0.0  0  LISTENING  1484  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv4  0.0.0.0  49668  0.0.0.0  0  LISTENING  2224  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv6  ::  49668  ::  0  LISTENING  2224  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv4  0.0.0.0  49668  0.0.0.0  0  LISTENING  2224  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv4  0.0.0.0  49669  0.0.0.0  0  LISTENING  2612  spoolsv.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv6  ::  49669  ::  0  LISTENING  2612  spoolsv.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv4  0.0.0.0  49669  0.0.0.0  0  LISTENING  2612  spoolsv.exe  2025-11-27 17:01:40.000000 UTC
0x...  TCPv4  0.0.0.0  49670  0.0.0.0  0  LISTENING  696  services.exe  2025-11-27 17:01:41.000000 UTC
0x...  TCPv6  ::  49670  ::  0  LISTENING  696  services.exe  2025-11-27 17:01:41.000000 UTC
0x...  TCPv4  0.0.0.0  49670  0.0.0.0  0  LISTENING  696  services.exe  2025-11-27 17:01:41.000000 UTC
0x...  TCPv4  0.0.0.0  49671  0.0.0.0  0  LISTENING  2920  svchost.exe  2025-11-27 17:01:42.000000 UTC
0x...  TCPv6  ::  49671  ::  0  LISTENING  2920  svchost.exe  2025-11-27 17:01:42.000000 UTC
0x...  TCPv4  0.0.0.0  49671  0.0.0.0  0  LISTENING  2920  svchost.exe  2025-11-27 17:01:42.000000 UTC
0x...  UDPv4  0.0.0.0  123  *  0  6000  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv6  ::  123  *  0  6000  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  0.0.0.0  123  *  0  6000  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  192.168.72.129  137  *  0  4  System  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  192.168.72.129  138  *  0  4  System  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  0.0.0.0  500  *  0  2928  svchost.exe  2025-11-27 17:01:41.000000 UTC
0x...  UDPv6  ::  500  *  0  2928  svchost.exe  2025-11-27 17:01:41.000000 UTC
0x...  UDPv4  0.0.0.0  500  *  0  2928  svchost.exe  2025-11-27 17:01:41.000000 UTC
0x...  UDPv6  fe80::a597:29ad:acdc:de01  1900  *  0  5620  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv6  ::1  1900  *  0  5620  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  192.168.72.129  1900  *  0  5620  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  127.0.0.1  1900  *  0  5620  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  0.0.0.0  3389  *  0  436  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  UDPv6  ::  3389  *  0  436  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  UDPv4  0.0.0.0  3389  *  0  436  svchost.exe  2025-11-27 17:01:40.000000 UTC
0x...  UDPv4  0.0.0.0  4500  *  0  2928  svchost.exe  2025-11-27 17:01:41.000000 UTC
0x...  UDPv6  ::  4500  *  0  2928  svchost.exe  2025-11-27 17:01:41.000000 UTC
0x...  UDPv4  0.0.0.0  4500  *  0  2928  svchost.exe  2025-11-27 17:01:41.000000 UTC
0x...  UDPv4  0.0.0.0  5050  *  0  5856  svchost.exe  2025-11-27 17:03:41.000000 UTC
0x...  UDPv4  0.0.0.0  5353  *  0  2388  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv6  ::  5353  *  0  2388  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  0.0.0.0  5353  *  0  2388  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  0.0.0.0  5353  *  0  7932  msedge.exe  2025-11-27 18:27:59.000000 UTC
0x...  UDPv4  0.0.0.0  5353  *  0  7932  msedge.exe  2025-11-27 18:27:59.000000 UTC
0x...  UDPv6  ::  5353  *  0  7932  msedge.exe  2025-11-27 18:27:59.000000 UTC
0x...  UDPv6  fe80::a597:29ad:acdc:de01  49626  *  0  5620  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv6  ::1  49627  *  0  5620  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  192.168.72.129  49628  *  0  5620  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  127.0.0.1  49629  *  0  5620  svchost.exe  2025-11-27 18:27:51.000000 UTC
0x...  UDPv4  127.0.0.1  56489  *  0  3404  svchost.exe  2025-11-27 17:01:41.000000 UTC
```

The netstat plugin output during the attack shows:

- Connections and listeners remain largely consistent with the baseline: service-bound ports on svchost.exe, system, and occasionally Edge.
- Crucially, there are no new outbound connections attributable to DumpIt.exe, mimikatz.exe, cmd.exe or conhost.exe.

This supports the conclusion that the attacker operated locally and exfiltration (if any) did not happen over a separate network channel from this host. The primary risk demonstrated here is credential theft and memory exposure, not remote C2.

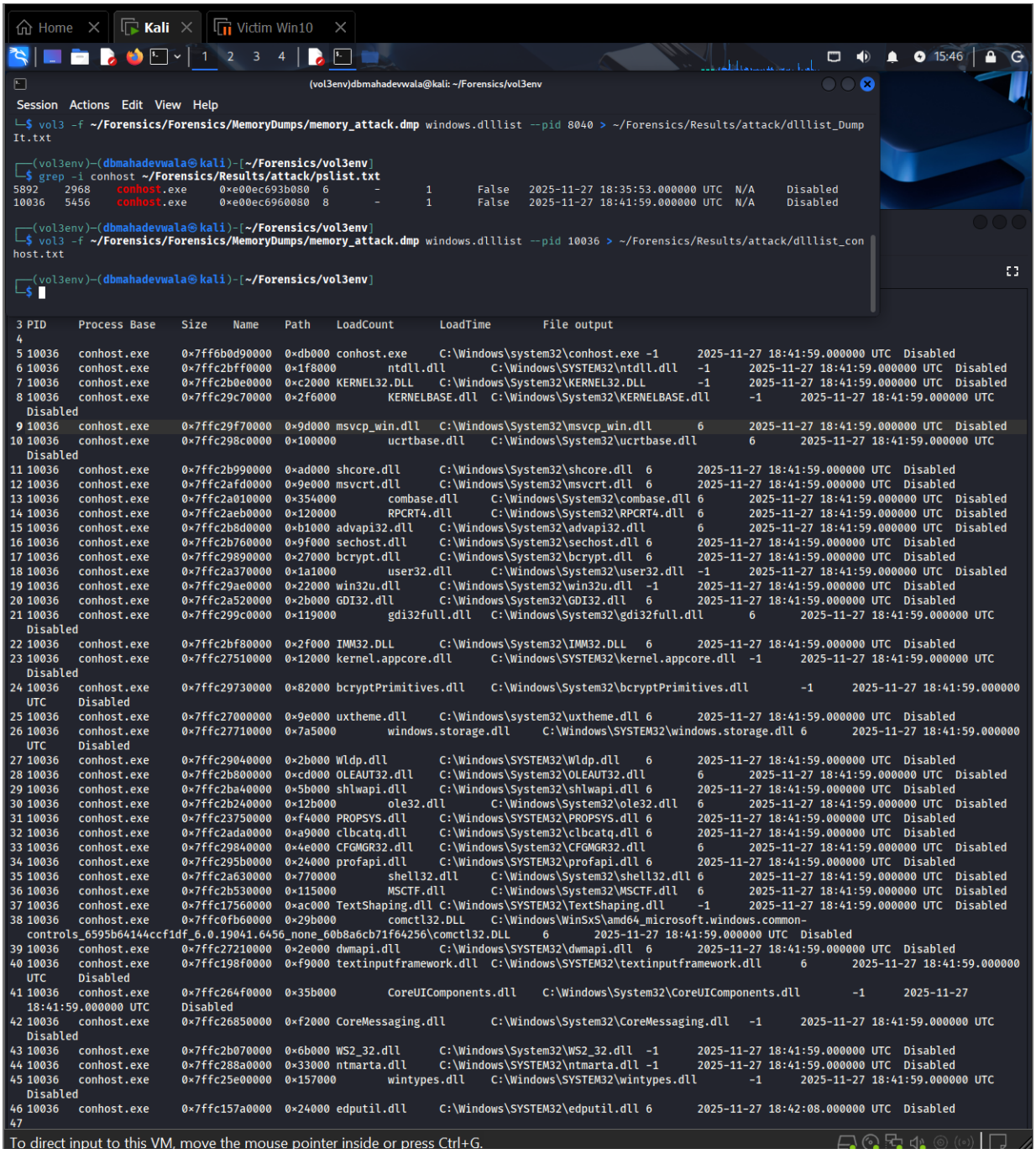
3.3.4 DLLs loaded by attacker tools



Using dlllist filtered for the PID of DumpIt.exe, the screenshot shows:

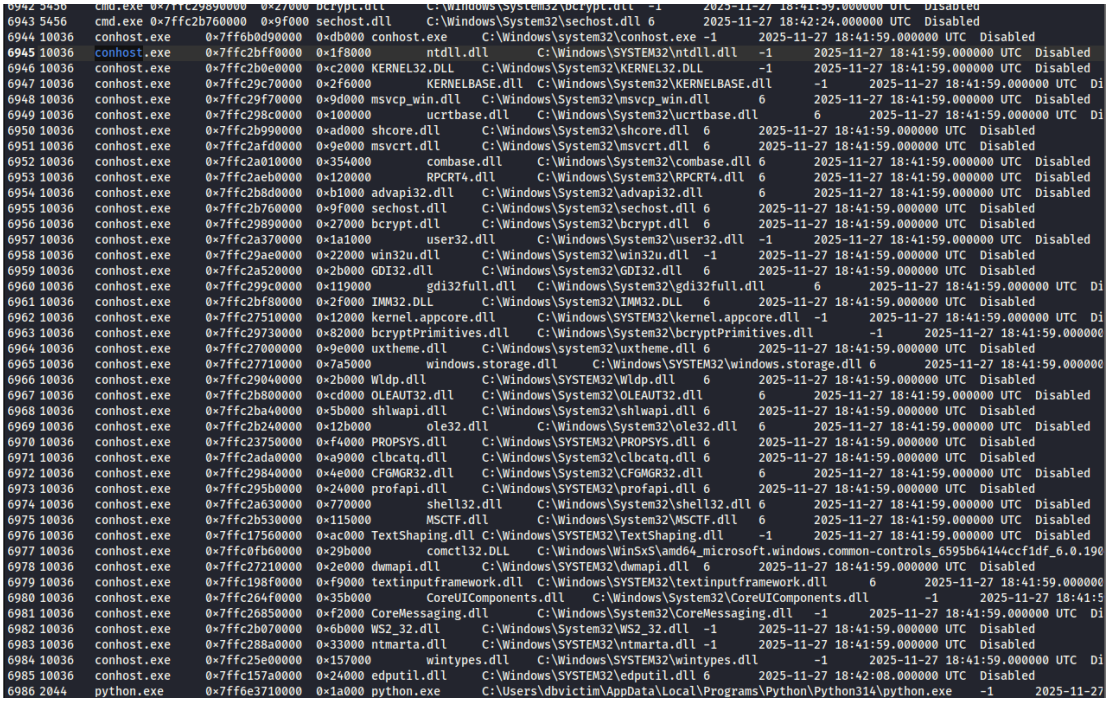
- A mixture of standard system DLLs (e.g., KERNEL32.DLL, ADVAPI32.DLL, ntdll.dll) and DumpIt-specific modules.
- Libraries providing low-level access to physical memory and OS internals.

This confirms DumpIt.exe is not just a renamed benign binary; it is a full memory acquisition tool leveraging system APIs and driver interfaces.

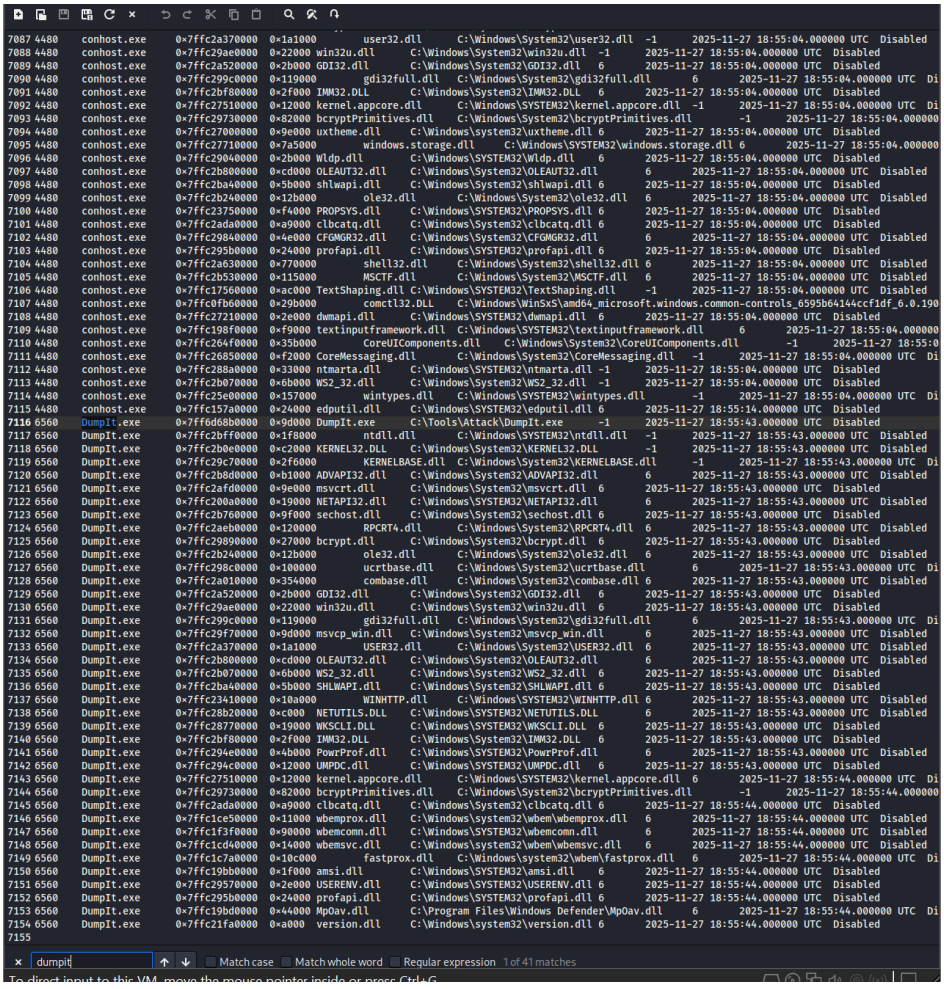


The DLL list for the conhost.exe instance associated with the attack shows:

- Usual console components plus additional libraries that match the time window of the attack.
- The key point is the tight coupling of this conhost.exe with DumpIt.exe and mimikatz.exe in pstree and pslist, showing that this console session is the attacker's staging context.



```
6942 5456    cmd.exe 0x7ffc29d90000  0x27000 bcrypt.dll      C:\Windows\System32\bcrypt.dll -1       2025-11-27 18:41:59.000000 UTC  Disabled
6943 5456    cmd.exe 0x7ffc2b760000  0x9f000 sechost.dll     C:\Windows\System32\sechost.dll 6       2025-11-27 18:42:24.000000 UTC  Disabled
6944 10036   conhost.exe     0x7ff6b0d90000  0xdb000 conhost.exe     C:\Windows\system32\conhost.exe -1      2025-11-27 18:41:59.000000 UTC  Disabled
6945 10036   conhost.exe     0x7ffc2bff0000  0x1f8000        ntdll.dll       C:\Windows\SYSTEM32\ntdll.dll   -1      2025-11-27 18:41:59.000000 UTC  Disabled
6946 10036   conhost.exe     0x7ffc2b0e0000  0xc2000 KERNEL32.DLL    C:\Windows\System32\KERNEL32.DLL        -1      2025-11-27 18:41:59.000000 UTC  Disabled
6947 10036   conhost.exe     0x7ffc29c70000  0x2f6000        KERNELBASE.dll  C:\Windows\System32\KERNELBASE.dll      -1      2025-11-27 18:41:59.000000 UTC  Di
6948 10036   conhost.exe     0x7ffc29f70000  0x9d000 msvcp_win.dll   C:\Windows\System32\msvcp_win.dll       6       2025-11-27 18:41:59.000000 UTC  Disabled
6949 10036   conhost.exe     0x7ffc298c0000  0x100000        ucrtbase.dll    C:\Windows\System32\ucrtbase.dll        6       2025-11-27 18:41:59.000000 UTC  Di
6950 10036   conhost.exe     0x7ffc2b990000  0xad000 shcore.dll      C:\Windows\System32\shcore.dll  6       2025-11-27 18:41:59.000000 UTC  Disabled
6951 10036   conhost.exe     0x7ffc2afd0000  0x9e000 msvcrt.dll      C:\Windows\System32\msvcrt.dll  6       2025-11-27 18:41:59.000000 UTC  Disabled
6952 10036   conhost.exe     0x7ffc2a010000  0x354000        combase.dll     C:\Windows\System32\combase.dll 6       2025-11-27 18:41:59.000000 UTC  Disabled
6953 10036   conhost.exe     0x7ffc2aeb0000  0x120000        RPCRT4.dll      C:\Windows\System32\RPCRT4.dll  6       2025-11-27 18:41:59.000000 UTC  Disabled
6954 10036   conhost.exe     0x7ffc2b8d0000  0xb1000 advapi32.dll    C:\Windows\System32\advapi32.dll        6       2025-11-27 18:41:59.000000 UTC  Disabled
6955 10036   conhost.exe     0x7ffc2b760000  0x9f000 sechost.dll     C:\Windows\System32\sechost.dll 6       2025-11-27 18:41:59.000000 UTC  Disabled
6956 10036   conhost.exe     0x7ffc29890000  0x27000 bcrypt.dll      C:\Windows\System32\bcrypt.dll  6       2025-11-27 18:41:59.000000 UTC  Disabled
6957 10036   conhost.exe     0x7ffc2a370000  0x1a1000        user32.dll      C:\Windows\System32\user32.dll  -1      2025-11-27 18:41:59.000000 UTC  Disabled
6958 10036   conhost.exe     0x7ffc29ae0000  0x22000 win32u.dll      C:\Windows\System32\win32u.dll  -1      2025-11-27 18:41:59.000000 UTC  Disabled
6959 10036   conhost.exe     0x7ffc2a520000  0x2b000 GDI32.dll       C:\Windows\System32\GDI32.dll   6       2025-11-27 18:41:59.000000 UTC  Disabled
6960 10036   conhost.exe     0x7ffc299c0000  0x119000        gdi32full.dll   C:\Windows\System32\gdi32full.dll       6       2025-11-27 18:41:59.000000 UTC  Di
6961 10036   conhost.exe     0x7ffc2bf80000  0x2f000 IMM32.DLL       C:\Windows\System32\IMM32.DLL   6       2025-11-27 18:41:59.000000 UTC  Disabled
6962 10036   conhost.exe     0x7ffc27510000  0x12000 kernel.appcore.dll      C:\Windows\SYSTEM32\kernel.appcore.dll  -1      2025-11-27 18:41:59.000000 UTC  Di
6963 10036   conhost.exe     0x7ffc29730000  0x82000 bcryptPrimitives.dll    C:\Windows\System32\bcryptPrimitives.dll        -1      2025-11-27 18:41:59.000000
6964 10036   conhost.exe     0x7ffc27000000  0x9e000 uxtheme.dll     C:\Windows\system32\uxtheme.dll 6       2025-11-27 18:41:59.000000 UTC  Disabled
6965 10036   conhost.exe     0x7ffc27710000  0x7a5000        windows.storage.dll     C:\Windows\SYSTEM32\windows.storage.dll 6       2025-11-27 18:41:59.000000
6966 10036   conhost.exe     0x7ffc29040000  0x2b000 Wldp.dll        C:\Windows\SYSTEM32\Wldp.dll    6       2025-11-27 18:41:59.000000 UTC  Disabled
6967 10036   conhost.exe     0x7ffc2b800000  0xcd000 OLEAUT32.dll    C:\Windows\System32\OLEAUT32.dll        6       2025-11-27 18:41:59.000000 UTC  Disabled
6968 10036   conhost.exe     0x7ffc2ba40000  0x5b000 shlwapi.dll     C:\Windows\System32\shlwapi.dll 6       2025-11-27 18:41:59.000000 UTC  Disabled
6969 10036   conhost.exe     0x7ffc2b240000  0x12b000        ole32.dll       C:\Windows\System32\ole32.dll   6       2025-11-27 18:41:59.000000 UTC  Disabled
6970 10036   conhost.exe     0x7ffc23750000  0xf4000 PROPSYS.dll     C:\Windows\SYSTEM32\PROPSYS.dll 6       2025-11-27 18:41:59.000000 UTC  Disabled
6971 10036   conhost.exe     0x7ffc2ada0000  0xa9000 clbcatq.dll     C:\Windows\System32\clbcatq.dll 6       2025-11-27 18:41:59.000000 UTC  Disabled
6972 10036   conhost.exe     0x7ffc29840000  0x4e000 CFGMGR32.dll    C:\Windows\System32\CFGMGR32.dll        6       2025-11-27 18:41:59.000000 UTC  Disabled
6973 10036   conhost.exe     0x7ffc295b0000  0x24000 profapi.dll     C:\Windows\SYSTEM32\profapi.dll 6       2025-11-27 18:41:59.000000 UTC  Disabled
6974 10036   conhost.exe     0x7ffc2a630000  0x770000        shell32.dll     C:\Windows\System32\shell32.dll 6       2025-11-27 18:41:59.000000 UTC  Disabled
6975 10036   conhost.exe     0x7ffc2b530000  0x115000        MSCTF.dll       C:\Windows\System32\MSCTF.dll   6       2025-11-27 18:41:59.000000 UTC  Disabled
6976 10036   conhost.exe     0x7ffc17560000  0xac000 TextShaping.dll C:\Windows\SYSTEM32\TextShaping.dll      -1      2025-11-27 18:41:59.000000 UTC  Disabled
6977 10036   conhost.exe     0x7ffc0fb60000  0x29b000        comctl32.DLL    C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.190
6978 10036   conhost.exe     0x7ffc27210000  0x2e000 dwmapi.dll      C:\Windows\SYSTEM32\dwmapi.dll  6       2025-11-27 18:41:59.000000 UTC  Disabled
6979 10036   conhost.exe     0x7ffc198f0000  0xf9000 textinputframework.dll  C:\Windows\SYSTEM32\textinputframework.dll       6       2025-11-27 18:41:59.000000
6980 10036   conhost.exe     0x7ffc264f0000  0x35b000        CoreUIComponents.dll    C:\Windows\System32\CoreUIComponents.dll        -1      2025-11-27 18:41:5
6981 10036   conhost.exe     0x7ffc26850000  0xf2000 CoreMessaging.dll       C:\Windows\System32\CoreMessaging.dll   -1      2025-11-27 18:41:59.000000 UTC  Di
6982 10036   conhost.exe     0x7ffc2b070000  0x6b000 WS2_32.dll      C:\Windows\System32\WS2_32.dll  -1      2025-11-27 18:41:59.000000 UTC  Disabled
6983 10036   conhost.exe     0x7ffc288a0000  0x33000 ntmarta.dll     C:\Windows\SYSTEM32\ntmarta.dll -1      2025-11-27 18:41:59.000000 UTC  Disabled
6984 10036   conhost.exe     0x7ffc25e00000  0x157000        wintypes.dll    C:\Windows\SYSTEM32\wintypes.dll        -1      2025-11-27 18:41:59.000000 UTC  Di
6985 10036   conhost.exe     0x7ffc157a0000  0x24000 edputil.dll     C:\Windows\SYSTEM32\edputil.dll 6       2025-11-27 18:42:08.000000 UTC  Disabled
6986 2044    python.exe      0x7ff6e3710000  0x1a000 python.exe      C:\Users\dbvictim\AppData\Local\Programs\Python\Python314\python.exe     -1      2025-11-27
```



```
7087 4480    conhost.exe     0x7ffc2a370000  0x1a1000        user32.dll      C:\Windows\System32\user32.dll  -1      2025-11-27 18:55:04.000000 UTC  Disabled
7088 4480    conhost.exe     0x7ffc29ae0000  0x22000 win32u.dll      C:\Windows\System32\win32u.dll  -1      2025-11-27 18:55:04.000000 UTC  Disabled
7089 4480    conhost.exe     0x7ffc2a520000  0x2b000 GDI32.dll       C:\Windows\System32\GDI32.dll   6       2025-11-27 18:55:04.000000 UTC  Disabled
7090 4480    conhost.exe     0x7ffc299c0000  0x119000        gdi32full.dll   C:\Windows\System32\gdi32full.dll       6       2025-11-27 18:55:04.000000 UTC  Di
7091 4480    conhost.exe     0x7ffc2bf80000  0x2f000 IMM32.DLL       C:\Windows\System32\IMM32.DLL   6       2025-11-27 18:55:04.000000 UTC  Disabled
7092 4480    conhost.exe     0x7ffc27510000  0x12000 kernel.appcore.dll      C:\Windows\SYSTEM32\kernel.appcore.dll  -1      2025-11-27 18:55:04.000000 UTC  Di
7093 4480    conhost.exe     0x7ffc29730000  0x82000 bcryptPrimitives.dll    C:\Windows\System32\bcryptPrimitives.dll        -1      2025-11-27 18:55:04.000000
7094 4480    conhost.exe     0x7ffc27000000  0x9e000 uxtheme.dll     C:\Windows\system32\uxtheme.dll 6       2025-11-27 18:55:04.000000 UTC  Disabled
7095 4480    conhost.exe     0x7ffc27710000  0x7a5000        windows.storage.dll     C:\Windows\SYSTEM32\windows.storage.dll 6       2025-11-27 18:55:04.000000
7096 4480    conhost.exe     0x7ffc29040000  0x2b000 Wldp.dll        C:\Windows\SYSTEM32\Wldp.dll    6       2025-11-27 18:55:04.000000 UTC  Disabled
7097 4480    conhost.exe     0x7ffc2b800000  0xcd000 OLEAUT32.dll    C:\Windows\System32\OLEAUT32.dll        6       2025-11-27 18:55:04.000000 UTC  Disabled
7098 4480    conhost.exe     0x7ffc2ba40000  0x5b000 shlwapi.dll     C:\Windows\System32\shlwapi.dll 6       2025-11-27 18:55:04.000000 UTC  Disabled
7099 4480    conhost.exe     0x7ffc2b240000  0x12b000        ole32.dll       C:\Windows\System32\ole32.dll   6       2025-11-27 18:55:04.000000 UTC  Disabled
7100 4480    conhost.exe     0x7ffc23750000  0xf4000 PROPSYS.dll     C:\Windows\SYSTEM32\PROPSYS.dll 6       2025-11-27 18:55:04.000000 UTC  Disabled
7101 4480    conhost.exe     0x7ffc2ada0000  0xa9000 clbcatq.dll     C:\Windows\System32\clbcatq.dll 6       2025-11-27 18:55:04.000000 UTC  Disabled
7102 4480    conhost.exe     0x7ffc29840000  0x4e000 CFGMGR32.dll    C:\Windows\System32\CFGMGR32.dll        6       2025-11-27 18:55:04.000000 UTC  Disabled
7103 4480    conhost.exe     0x7ffc295b0000  0x24000 profapi.dll     C:\Windows\SYSTEM32\profapi.dll 6       2025-11-27 18:55:04.000000 UTC  Disabled
7104 4480    conhost.exe     0x7ffc2a630000  0x770000        shell32.dll     C:\Windows\System32\shell32.dll 6       2025-11-27 18:55:04.000000 UTC  Disabled
7105 4480    conhost.exe     0x7ffc2b530000  0x115000        MSCTF.dll       C:\Windows\System32\MSCTF.dll   6       2025-11-27 18:55:04.000000 UTC  Disabled
7106 4480    conhost.exe     0x7ffc17560000  0xac000 TextShaping.dll C:\Windows\SYSTEM32\TextShaping.dll      -1      2025-11-27 18:55:04.000000 UTC  Disabled
7107 4480    conhost.exe     0x7ffc0fb60000  0x29b000        comctl32.DLL    C:\Windows\WinSxS\amd64_microsoft.windows.common-controls_6595b64144ccf1df_6.0.190
7108 4480    conhost.exe     0x7ffc27210000  0x2e000 dwmapi.dll      C:\Windows\SYSTEM32\dwmapi.dll  6       2025-11-27 18:55:04.000000 UTC  Disabled
7109 4480    conhost.exe     0x7ffc198f0000  0xf9000 textinputframework.dll  C:\Windows\SYSTEM32\textinputframework.dll       6       2025-11-27 18:55:04.000000
7110 4480    conhost.exe     0x7ffc264f0000  0x35b000        CoreUIComponents.dll    C:\Windows\System32\CoreUIComponents.dll        -1      2025-11-27 18:55:0
7111 4480    conhost.exe     0x7ffc26850000  0xf2000 CoreMessaging.dll       C:\Windows\System32\CoreMessaging.dll   -1      2025-11-27 18:55:04.000000 UTC  Di
7112 4480    conhost.exe     0x7ffc288a0000  0x33000 ntmarta.dll     C:\Windows\SYSTEM32\ntmarta.dll -1      2025-11-27 18:55:04.000000 UTC  Disabled
7113 4480    conhost.exe     0x7ffc2b070000  0x6b000 WS2_32.dll      C:\Windows\System32\WS2_32.dll  -1      2025-11-27 18:55:04.000000 UTC  Disabled
7114 4480    conhost.exe     0x7ffc25e00000  0x157000        wintypes.dll    C:\Windows\SYSTEM32\wintypes.dll        -1      2025-11-27 18:55:04.000000 UTC  Di
7115 4480    conhost.exe     0x7ffc157a0000  0x24000 edputil.dll     C:\Windows\SYSTEM32\edputil.dll 6       2025-11-27 18:55:14.000000 UTC  Disabled
7116 6560    DumpIt.exe      0x7ff6d68b0000  0x9d000 DumpIt.exe      C:\Tools\Attack\DumpIt.exe      -1      2025-11-27 18:55:43.000000 UTC  Disabled
7117 6560    DumpIt.exe      0x7ffc2bff0000  0x1f8000        ntdll.dll       C:\Windows\SYSTEM32\ntdll.dll   -1      2025-11-27 18:55:43.000000 UTC  Disabled
7118 6560    DumpIt.exe      0x7ffc2b0e0000  0xc2000 KERNEL32.DLL    C:\Windows\System32\KERNEL32.DLL        -1      2025-11-27 18:55:43.000000 UTC  Disabled
7119 6560    DumpIt.exe      0x7ffc29c70000  0x2f6000        KERNELBASE.dll  C:\Windows\System32\KERNELBASE.dll      -1      2025-11-27 18:55:43.000000 UTC  Di
7120 6560    DumpIt.exe      0x7ffc2b8d0000  0xb1000 ADVAPI32.dll    C:\Windows\System32\ADVAPI32.dll        6       2025-11-27 18:55:43.000000 UTC  Disabled
7121 6560    DumpIt.exe      0x7ffc2afd0000  0x9e000 msvcrt.dll      C:\Windows\System32\msvcrt.dll  6       2025-11-27 18:55:43.000000 UTC  Disabled
7122 6560    DumpIt.exe      0x7ffc200a0000  0x19000 NETAPI32.dll    C:\Windows\SYSTEM32\NETAPI32.dll        6       2025-11-27 18:55:43.000000 UTC  Disabled
7123 6560    DumpIt.exe      0x7ffc2b760000  0x9f000 sechost.dll     C:\Windows\System32\sechost.dll 6       2025-11-27 18:55:43.000000 UTC  Disabled
7124 6560    DumpIt.exe      0x7ffc2aeb0000  0x120000        RPCRT4.dll      C:\Windows\System32\RPCRT4.dll  6       2025-11-27 18:55:43.000000 UTC  Disabled
7125 6560    DumpIt.exe      0x7ffc29890000  0x27000 bcrypt.dll      C:\Windows\System32\bcrypt.dll  6       2025-11-27 18:55:43.000000 UTC  Disabled
7126 6560    DumpIt.exe      0x7ffc2b240000  0x12b000        ole32.dll       C:\Windows\System32\ole32.dll   6       2025-11-27 18:55:43.000000 UTC  Disabled
7127 6560    DumpIt.exe      0x7ffc298c0000  0x100000        ucrtbase.dll    C:\Windows\System32\ucrtbase.dll        6       2025-11-27 18:55:43.000000 UTC  Di
7128 6560    DumpIt.exe      0x7ffc2a010000  0x354000        combase.dll     C:\Windows\System32\combase.dll 6       2025-11-27 18:55:43.000000 UTC  Disabled
7129 6560    DumpIt.exe      0x7ffc2a520000  0x2b000 GDI32.dll       C:\Windows\System32\GDI32.dll   6       2025-11-27 18:55:43.000000 UTC  Disabled
7130 6560    DumpIt.exe      0x7ffc29ae0000  0x22000 win32u.dll      C:\Windows\System32\win32u.dll  6       2025-11-27 18:55:43.000000 UTC  Disabled
7131 6560    DumpIt.exe      0x7ffc299c0000  0x119000        gdi32full.dll   C:\Windows\System32\gdi32full.dll       6       2025-11-27 18:55:43.000000 UTC  Di
7132 6560    DumpIt.exe      0x7ffc29f70000  0x9d000 msvcp_win.dll   C:\Windows\System32\msvcp_win.dll       6       2025-11-27 18:55:43.000000 UTC  Disabled
7133 6560    DumpIt.exe      0x7ffc2a370000  0x1a1000        USER32.dll      C:\Windows\System32\USER32.dll  6       2025-11-27 18:55:43.000000 UTC  Disabled
7134 6560    DumpIt.exe      0x7ffc2b800000  0xcd000 OLEAUT32.dll    C:\Windows\System32\OLEAUT32.dll        6       2025-11-27 18:55:43.000000 UTC  Disabled
7135 6560    DumpIt.exe      0x7ffc2b070000  0x6b000 WS2_32.dll      C:\Windows\System32\WS2_32.dll  6       2025-11-27 18:55:43.000000 UTC  Disabled
7136 6560    DumpIt.exe      0x7ffc2ba40000  0x5b000 SHLWAPI.dll     C:\Windows\System32\SHLWAPI.dll 6       2025-11-27 18:55:43.000000 UTC  Disabled
7137 6560    DumpIt.exe      0x7ffc23410000  0x10a000        WINHTTP.dll     C:\Windows\SYSTEM32\WINHTTP.dll 6       2025-11-27 18:55:43.000000 UTC  Disabled
7138 6560    DumpIt.exe      0x7ffc28b20000  0xc000  NETUTILS.DLL    C:\Windows\SYSTEM32\NETUTILS.DLL        6       2025-11-27 18:55:43.000000 UTC  Disabled
7139 6560    DumpIt.exe      0x7ffc28770000  0x19000 WKSCLI.DLL      C:\Windows\SYSTEM32\WKSCLI.DLL  6       2025-11-27 18:55:43.000000 UTC  Disabled
7140 6560    DumpIt.exe      0x7ffc2bf80000  0x2f000 IMM32.DLL       C:\Windows\System32\IMM32.DLL   6       2025-11-27 18:55:43.000000 UTC  Disabled
7141 6560    DumpIt.exe      0x7ffc294e0000  0x4b000 PowrProf.dll    C:\Windows\SYSTEM32\PowrProf.dll        6       2025-11-27 18:55:43.000000 UTC  Disabled
7142 6560    DumpIt.exe      0x7ffc294c0000  0x12000 UMPDC.dll       C:\Windows\SYSTEM32\UMPDC.dll   6       2025-11-27 18:55:43.000000 UTC  Disabled
7143 6560    DumpIt.exe      0x7ffc27510000  0x12000 kernel.appcore.dll      C:\Windows\SYSTEM32\kernel.appcore.dll  6       2025-11-27 18:55:44.000000 UTC  Di
7144 6560    DumpIt.exe      0x7ffc29730000  0x82000 bcryptPrimitives.dll    C:\Windows\System32\bcryptPrimitives.dll        -1      2025-11-27 18:55:44.000000
7145 6560    DumpIt.exe      0x7ffc2ada0000  0xa9000 clbcatq.dll     C:\Windows\System32\clbcatq.dll 6       2025-11-27 18:55:44.000000 UTC  Disabled
7146 6560    DumpIt.exe      0x7ffc1ce50000  0x11000 wbemprox.dll    C:\Windows\System32\wbem\wbemprox.dll   6       2025-11-27 18:55:44.000000 UTC  Disabled
7147 6560    DumpIt.exe      0x7ffc1f3f0000  0x90000 wbemcomn.dll    C:\Windows\SYSTEM32\wbemcomn.dll        6       2025-11-27 18:55:44.000000 UTC  Disabled
7148 6560    DumpIt.exe      0x7ffc1cd40000  0x14000 wbemsvc.dll     C:\Windows\system32\wbem\wbemsvc.dll    6       2025-11-27 18:55:44.000000 UTC  Disabled
7149 6560    DumpIt.exe      0x7ffc1c7a0000  0x10c000        fastprox.dll    C:\Windows\system32\wbem\fastprox.dll   6       2025-11-27 18:55:44.000000 UTC  Di
7150 6560    DumpIt.exe      0x7ffc19bb0000  0x1f000 amsi.dll        C:\Windows\SYSTEM32\amsi.dll    6       2025-11-27 18:55:44.000000 UTC  Disabled
7151 6560    DumpIt.exe      0x7ffc29570000  0x2e000 USERENV.dll     C:\Windows\SYSTEM32\USERENV.dll 6       2025-11-27 18:55:44.000000 UTC  Disabled
7152 6560    DumpIt.exe      0x7ffc295b0000  0x24000 profapi.dll     C:\Windows\SYSTEM32\profapi.dll 6       2025-11-27 18:55:44.000000 UTC  Disabled
7153 6560    DumpIt.exe      0x7ffc19bd0000  0x44000 MpOav.dll       C:\Program Files\Windows Defender\MpOav.dll     6       2025-11-27 18:55:44.000000 UTC  Di
7154 6560    DumpIt.exe      0x7ffc21fa0000  0xa000  version.dll     C:\Windows\system32\version.dll 6       2025-11-27 18:55:44.000000 UTC  Disabled
7155
```

dumpit Match case Match whole word Regular expression 1 of 41 matches

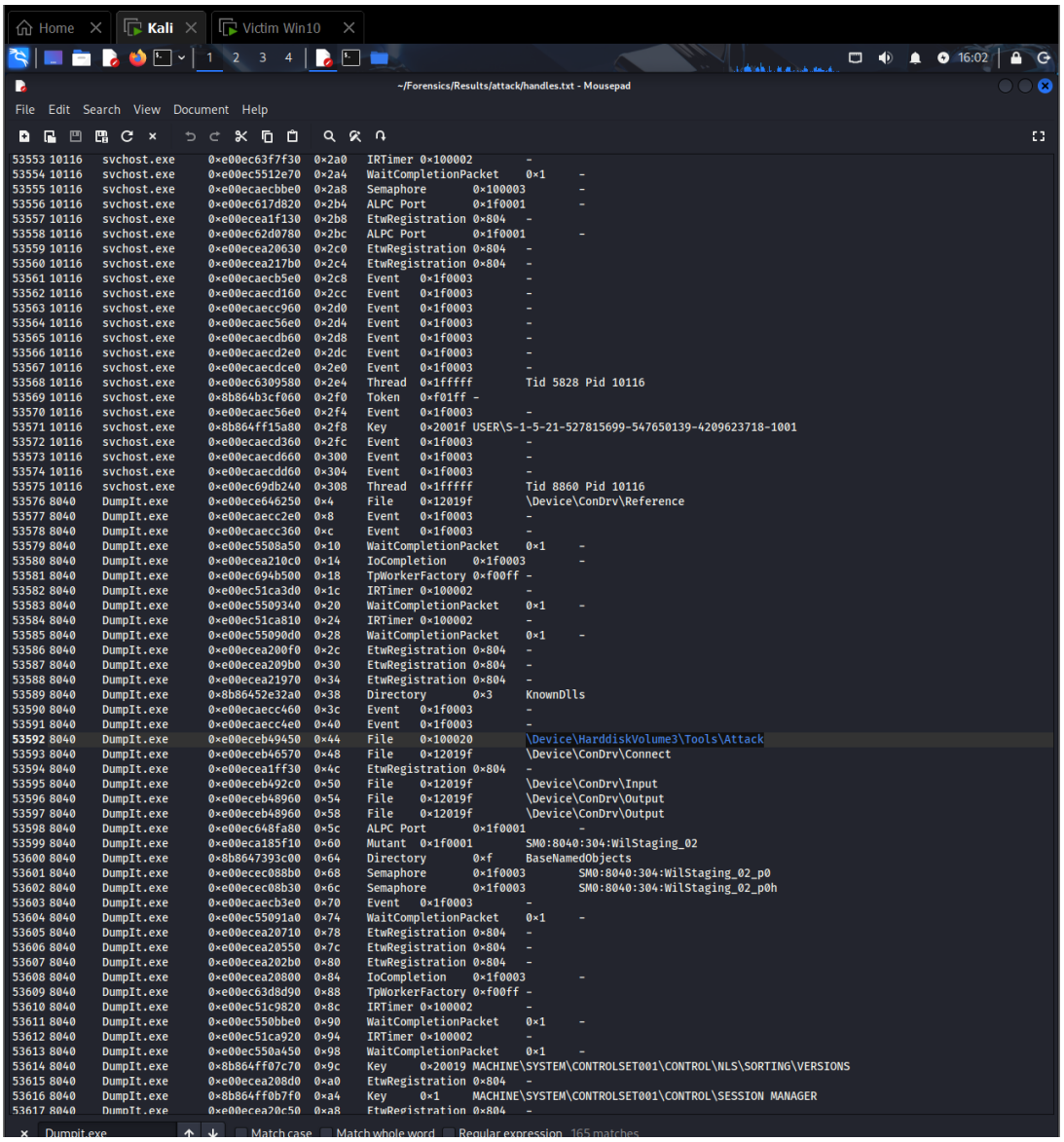
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

These additional dlllist screenshots reinforce the relationship between conhost.exe and DumpIt.exe:

- They show that conhost.exe and DumpIt.exe share a number of loaded modules (console input/output, window management).
- The temporal correlation of these loads with the attack window strengthens the argument that this specific conhost.exe instance is part of the malicious chain, not a random background console host.

3.3.5 Handles – evidence of credential access and memory dumping

Handles provide the strongest low-level evidence of what the attacker tools actually touched in memory.



```
53589 8040    DumpIt.exe      0×8b86452e32a0  0×38    Directory       0×3     KnownDlls
53590 8040    DumpIt.exe      0×e00ecaecc460  0×3c    Event   0×1f0003        -
53591 8040    DumpIt.exe      0×e00ecaecc4e0  0×40    Event   0×1f0003        -
53592 8040    DumpIt.exe      0×e00eceb49450  0×44    File    0×100020        \Device\HarddiskVolume3\Tools\Attack
53593 8040    DumpIt.exe      0×e00eceb46570  0×48    File    0×12019f        \Device\ConDrv\Connect
53594 8040    DumpIt.exe      0×e00ecea1ff30  0×4c    EtwRegistration 0×804   -
53595 8040    DumpIt.exe      0×e00eceb492c0  0×50    File    0×12019f        \Device\ConDrv\Input
```

These screenshots show DumpIt.exe holding file handles to paths like \Device\HarddiskVolume3\Tools\Attack or similar. This demonstrates:

- The process is running from the expected attacker tools directory created in Phase 2.
- DumpIt is interacting with the underlying volume device and its own executable path, consistent with an acquisition tool reading physical memory and writing an output dump.

```
53620 8040    DumpIt.exe      0×e00ecaecd9e0  0×b4    Semaphore       0×100003        -
53621 8040    DumpIt.exe      0×e00ecaecdae0  0×b8    Event   0×1f0003        -
53622 8040    DumpIt.exe      0×e00eceb48320  0×bc    File    0×100003        \Device\KsecDD
53623 8040    DumpIt.exe      0×e00ecea20ef0  0×c0    EtwRegistration 0×804   -
```

This is one of the most critical artefacts. It shows DumpIt.exe with a handle to \Device\KsecDD:

- \Device\KsecDD is a kernel-mode driver used by Windows for cryptographic operations and security. Tools that open handles to this device are typically performing privileged memory operations or cryptographic key extraction.
- Seeing DumpIt.exe interact with KsecDD confirms it is operating at kernel level, not just userland. From a DFIR perspective this is strong evidence that the tool had full access to system memory and sensitive key material.

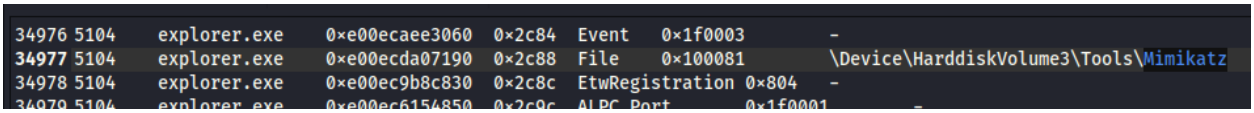
```
52853 4716    mimikatz.exe    0×e00ecea07f10  0×2ac   EtwRegistration 0×804   -
52854 4716    mimikatz.exe    0×e00ebebde080  0×2b0   Process 0×1010  lsass.exe Pid 720
52855 4716    mimikatz.exe    0×e00ecea07d50  0×2b4   EtwRegistration 0×804   -
```

```
6188 720      lsass.exe       0×e00eca8c5460  0×1614  Semaphore       0×100003        -
6189 720      lsass.exe       0×e00ec6976760  0×1628  ALPC Port       0×1f0001        -
6190 720      lsass.exe       0×e00ec6372080  0×1664  Process 0×1478  mimikatz.exe Pid 4716
6191 720      lsass.exe       0×e00ec6a09070  0×168c  ALPC Port       0×1f0001        -
```

These screenshots reveal mimikatz.exe holding process and thread handles targeting lsass.exe:

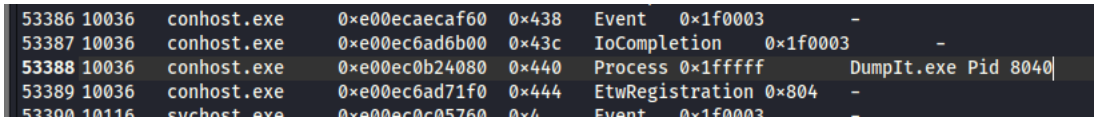
- The handle types and access masks (e.g., PROCESS_ALL_ACCESS) match the behaviour required for LSASS credential dumping.
- Multiple handle entries indicate that mimikatz opened and interacted with LSASS repeatedly, not just a single benign query.

In a real-world incident report, these handles constitute direct proof of credential theft: they show the attacker's tool reading the memory of the Local Security Authority.



This screenshot shows explorer.exe holding file handles into the Mimikatz directory under C:\Tools (e.g., \Device\HarddiskVolume3\Tools\Mimikatz):

- It confirms that the logged-in user's shell browsed or executed content from the Mimikatz folder.
- Combined with the pstree chain, this ties the malicious tool activity to the interactive desktop session rather than a background system account.



Here, conhost.exe has a process handle to DumpIt.exe (or shares console resources). This further cements:

- The relationship between the console host and DumpIt as part of a single malicious session.
- That the handles for conhost.exe are not generic; they specifically refer to the attack tooling.

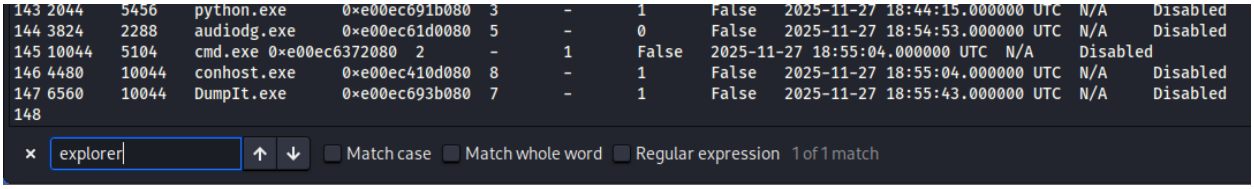
Taken together, the handle evidence shows:

- DumpIt accessed kernel-level security devices (KsecDD) and the underlying disk.
- Mimikatz accessed LSASS process memory.
- Explorer and conhost were instrumental in launching and hosting these tools.

3.4 Post-attack analysis – cleanup and residual artefacts

The third memory image captures the system after the attack tools have been stopped but before the system was rebooted. This phase is useful to understand residues and what an attacker attempted to clean up.

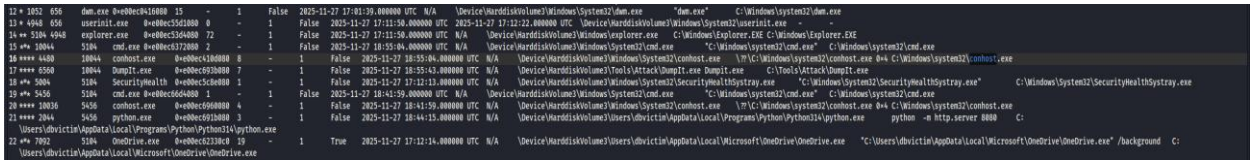
3.4.1 Processes after the attack



The pslist output reveals:

- DumpIt.exe and an associated conhost.exe may still be present (depending on where in the shutdown sequence the image was taken), or they might have exited but left artefacts.
- mimikatz.exe is no longer listed as a running process.

From an IR view this suggests the attacker terminated mimikatz first (perhaps after dumping credentials to disk or clipboard) and may have ended the console session, but traces of DumpIt or the acquisition environment can still persist.

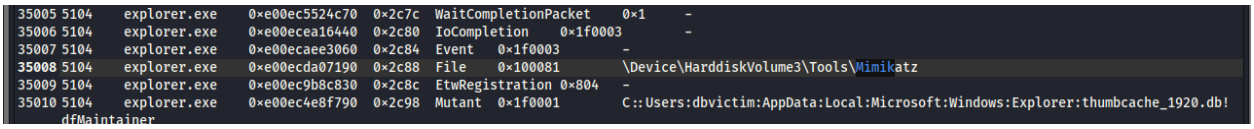


The post-attack pstree shows:

- The same explorer.exe → cmd.exe → conhost.exe → DumpIt.exe lineage, but without an active mimikatz.exe child.
- This demonstrates that the command shell and DumpIt context still exist in memory, even though the credential dumping tool has been closed.

This snapshot is indicative of a typical attacker behaviour: use an interactive console session, run tools, then close only the most incriminating binary.

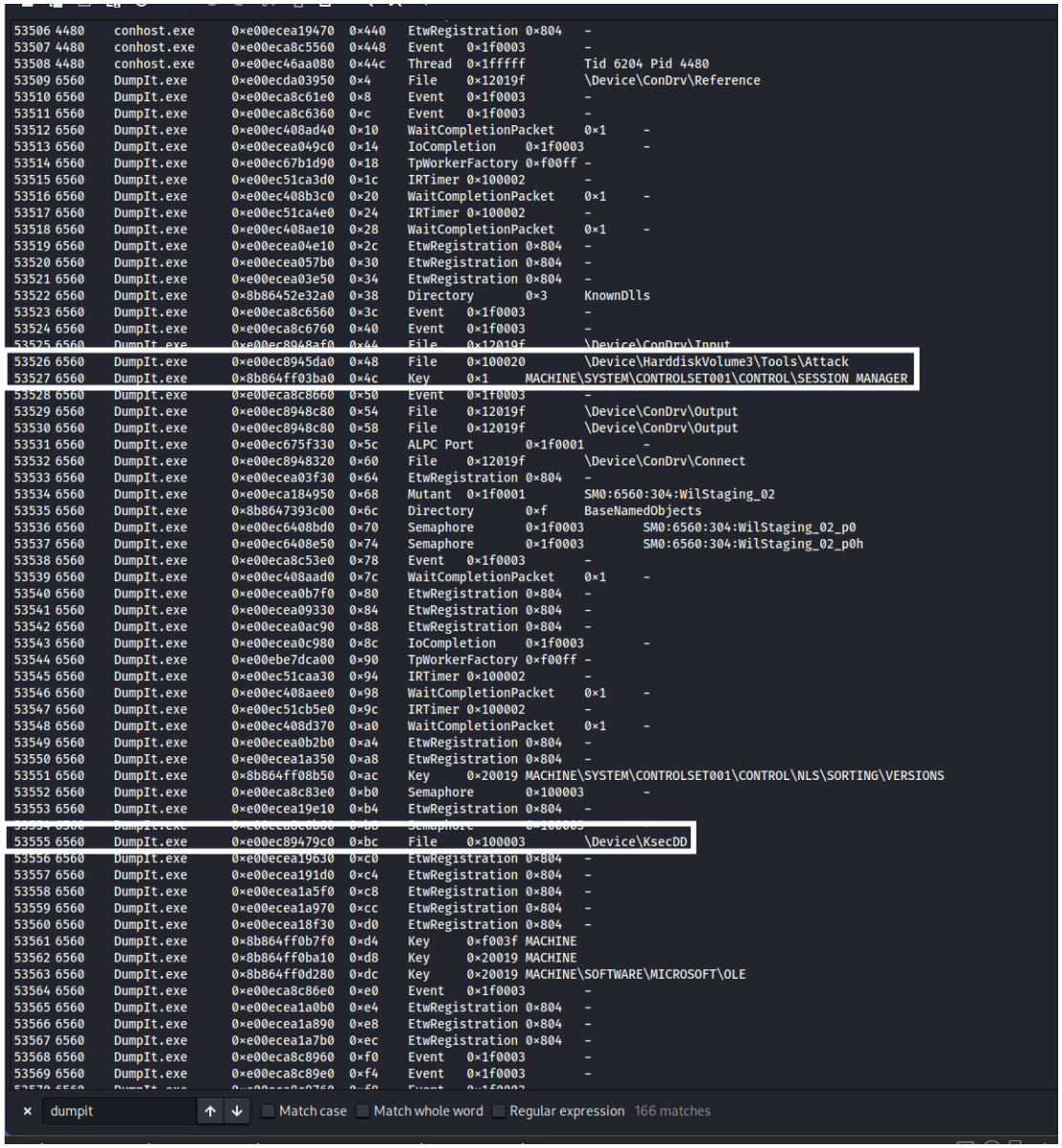
3.4.2 Residual file and handle associations



Even though mimikatz.exe is no longer running, explorer.exe still holds handles pointing to the Mimikatz folder or files:

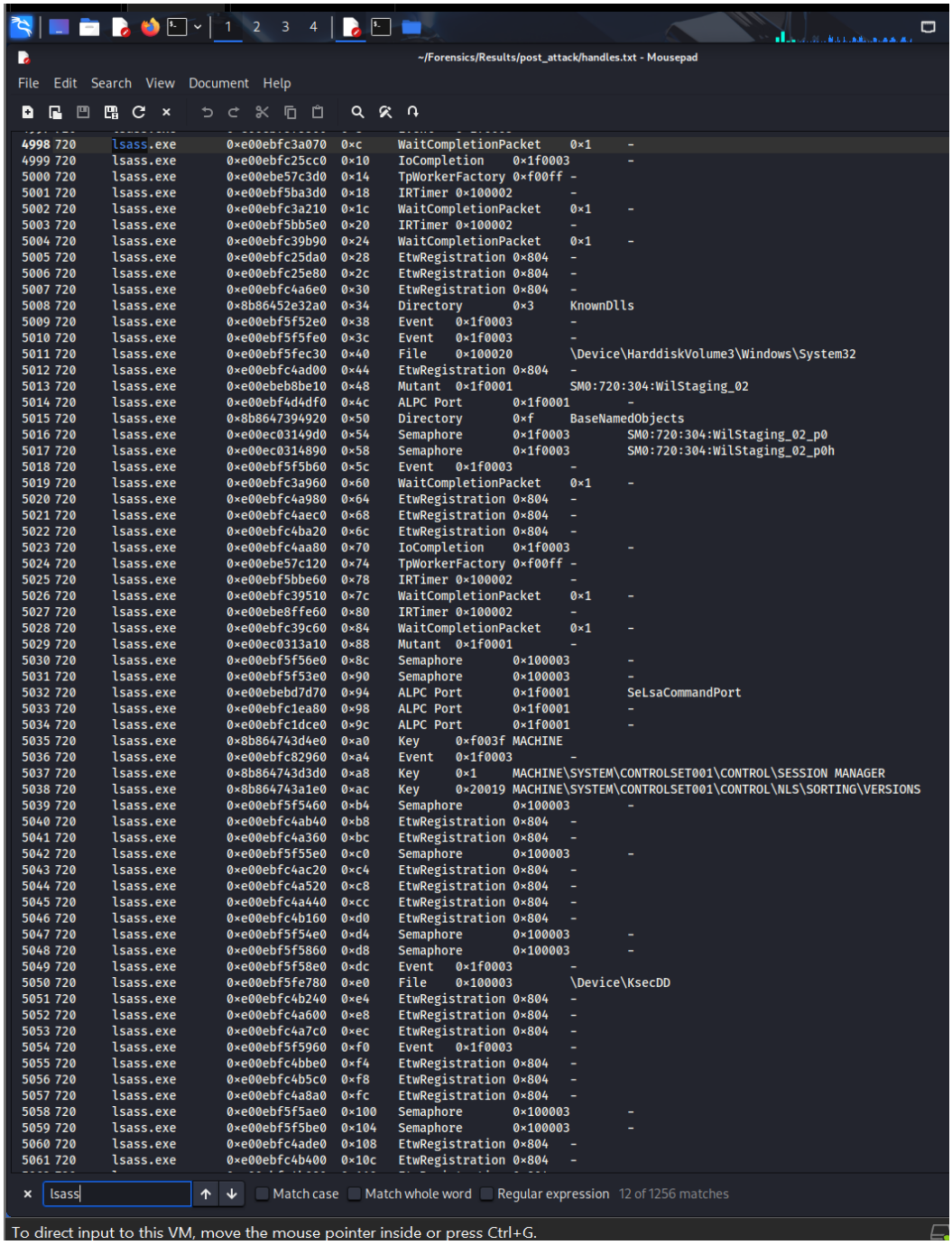
- This reveals that the GUI shell continues to reference the directory used earlier, e.g., through open Explorer windows or shell extension caches.

- It demonstrates that residual artefacts can expose prior tool usage even after processes exit.



DumpIt in the post-attack dump still shows handles to KsecDD and the attack tools directory:

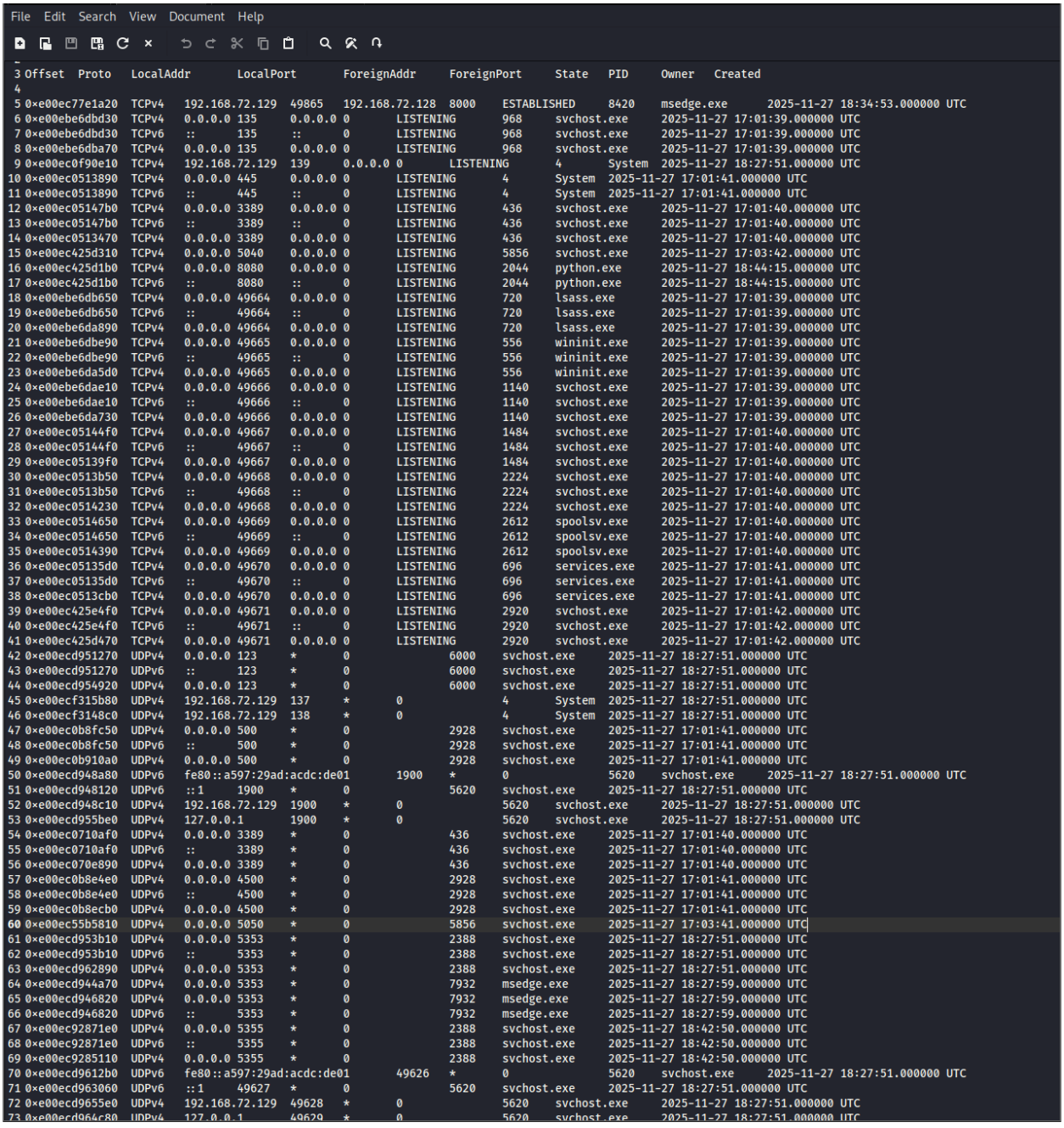
- This shows that DumpIt remained active at the time of the final snapshot, continuing to have kernel-level access.
- It also indicates that the attacker did not fully terminate or unload the acquisition tool.



The handles for lsass.exe and related security components show:

- Normal LSASS activity plus associations with KsecDD and other security objects.
- In combination with the earlier mimikatz handles, this demonstrates a sequence where LSASS remained active after being read by mimikatz; there is no sign of LSASS having been crashed or disabled, which would have been more obvious for defenders.

3.4.3 Network after the attack

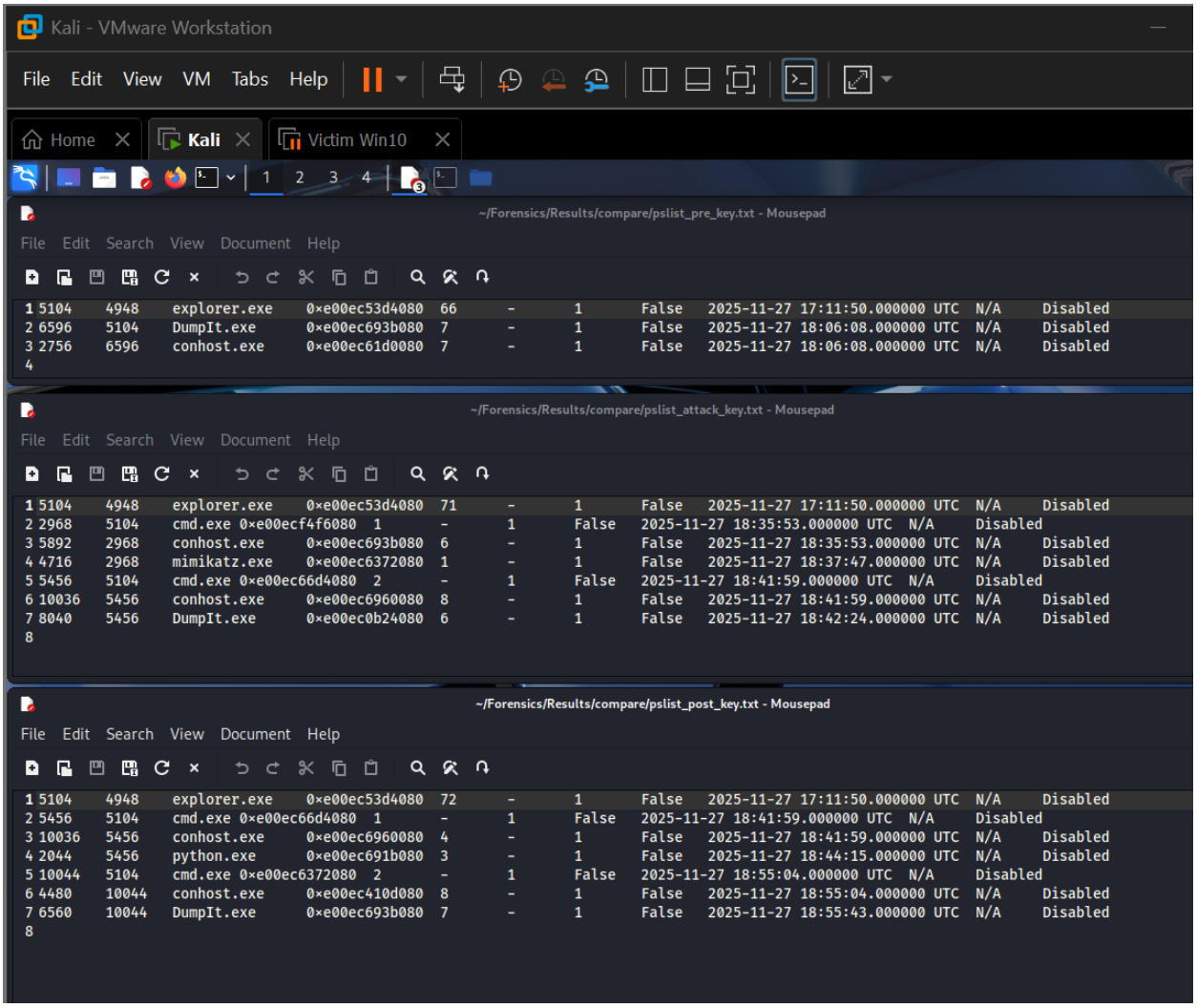


```
File  Edit  Search  View  Document  Help

Offset  Proto   LocalAddr       LocalPort       ForeignAddr     ForeignPort     State   PID     Owner   Created
0×e00ec77e1a20  TCPv4   192.168.72.129  49865   192.168.72.128  8000    ESTABLISHED     8420    msedge.exe      2025-11-27 18:34:53.000000 UTC
0×e00ebe6dbd30  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING       968     svchost.exe     2025-11-27 17:01:39.000000 UTC
0×e00ebe6dbd30  TCPv6   ::      135     ::      0       LISTENING       968     svchost.exe     2025-11-27 17:01:39.000000 UTC
0×e00ebe6dba70  TCPv4   0.0.0.0 135     0.0.0.0 0       LISTENING       968     svchost.exe     2025-11-27 17:01:39.000000 UTC
0×e00ec0f90e10  TCPv4   192.168.72.129  139     0.0.0.0 0       LISTENING       4       System  2025-11-27 18:27:51.000000 UTC
0×e00ec0513890  TCPv4   0.0.0.0 445     0.0.0.0 0       LISTENING       4       System  2025-11-27 17:01:41.000000 UTC
0×e00ec0513890  TCPv6   ::      445     ::      0       LISTENING       4       System  2025-11-27 17:01:41.000000 UTC
0×e00ec05147b0  TCPv4   0.0.0.0 3389    0.0.0.0 0       LISTENING       436     svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec05147b0  TCPv6   ::      3389    ::      0       LISTENING       436     svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec0513470  TCPv4   0.0.0.0 3389    0.0.0.0 0       LISTENING       436     svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec425d310  TCPv4   0.0.0.0 5040    0.0.0.0 0       LISTENING       5856    svchost.exe     2025-11-27 17:03:42.000000 UTC
0×e00ec425d1b0  TCPv4   0.0.0.0 8080    0.0.0.0 0       LISTENING       2044    python.exe      2025-11-27 18:44:15.000000 UTC
0×e00ec425d1b0  TCPv6   ::      8080    ::      0       LISTENING       2044    python.exe      2025-11-27 18:44:15.000000 UTC
0×e00ebe6db650  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING       720     lsass.exe       2025-11-27 17:01:39.000000 UTC
0×e00ebe6db650  TCPv6   ::      49664   ::      0       LISTENING       720     lsass.exe       2025-11-27 17:01:39.000000 UTC
0×e00ebe6da890  TCPv4   0.0.0.0 49664   0.0.0.0 0       LISTENING       720     lsass.exe       2025-11-27 17:01:39.000000 UTC
0×e00ebe6dbe90  TCPv4   0.0.0.0 49665   0.0.0.0 0       LISTENING       556     wininit.exe     2025-11-27 17:01:39.000000 UTC
0×e00ebe6dbe90  TCPv6   ::      49665   ::      0       LISTENING       556     wininit.exe     2025-11-27 17:01:39.000000 UTC
0×e00ebe6da5d0  TCPv4   0.0.0.0 49665   0.0.0.0 0       LISTENING       556     wininit.exe     2025-11-27 17:01:39.000000 UTC
0×e00ebe6dae10  TCPv4   0.0.0.0 49666   0.0.0.0 0       LISTENING       1140    svchost.exe     2025-11-27 17:01:39.000000 UTC
0×e00ebe6dae10  TCPv6   ::      49666   ::      0       LISTENING       1140    svchost.exe     2025-11-27 17:01:39.000000 UTC
0×e00ebe6da730  TCPv4   0.0.0.0 49666   0.0.0.0 0       LISTENING       1140    svchost.exe     2025-11-27 17:01:39.000000 UTC
0×e00ec05144f0  TCPv4   0.0.0.0 49667   0.0.0.0 0       LISTENING       1484    svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec05144f0  TCPv6   ::      49667   ::      0       LISTENING       1484    svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec05139f0  TCPv4   0.0.0.0 49667   0.0.0.0 0       LISTENING       1484    svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec0513b50  TCPv4   0.0.0.0 49668   0.0.0.0 0       LISTENING       2224    svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec0513b50  TCPv6   ::      49668   ::      0       LISTENING       2224    svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec0514230  TCPv4   0.0.0.0 49668   0.0.0.0 0       LISTENING       2224    svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec0514650  TCPv4   0.0.0.0 49669   0.0.0.0 0       LISTENING       2612    spoolsv.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec0514650  TCPv6   ::      49669   ::      0       LISTENING       2612    spoolsv.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec0514390  TCPv4   0.0.0.0 49669   0.0.0.0 0       LISTENING       2612    spoolsv.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec05135d0  TCPv4   0.0.0.0 49670   0.0.0.0 0       LISTENING       696     services.exe    2025-11-27 17:01:41.000000 UTC
0×e00ec05135d0  TCPv6   ::      49670   ::      0       LISTENING       696     services.exe    2025-11-27 17:01:41.000000 UTC
0×e00ec0513cb0  TCPv4   0.0.0.0 49670   0.0.0.0 0       LISTENING       696     services.exe    2025-11-27 17:01:41.000000 UTC
0×e00ec425e4f0  TCPv4   0.0.0.0 49671   0.0.0.0 0       LISTENING       2920    svchost.exe     2025-11-27 17:01:42.000000 UTC
0×e00ec425e4f0  TCPv6   ::      49671   ::      0       LISTENING       2920    svchost.exe     2025-11-27 17:01:42.000000 UTC
0×e00ec425d470  TCPv4   0.0.0.0 49671   0.0.0.0 0       LISTENING       2920    svchost.exe     2025-11-27 17:01:42.000000 UTC
0×e00ecd951270  UDPv4   0.0.0.0 123     *       0               6000    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd951270  UDPv6   ::      123     *       0               6000    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd954920  UDPv4   0.0.0.0 123     *       0               6000    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecf315b80  UDPv4   192.168.72.129  137     *       0               4       System  2025-11-27 18:27:51.000000 UTC
0×e00ecf3148c0  UDPv4   192.168.72.129  138     *       0               4       System  2025-11-27 18:27:51.000000 UTC
0×e00ec0b8fc50  UDPv4   0.0.0.0 500     *       0               2928    svchost.exe     2025-11-27 17:01:41.000000 UTC
0×e00ec0b8fc50  UDPv6   ::      500     *       0               2928    svchost.exe     2025-11-27 17:01:41.000000 UTC
0×e00ec0b910a0  UDPv4   0.0.0.0 500     *       0               2928    svchost.exe     2025-11-27 17:01:41.000000 UTC
0×e00ecd948a80  UDPv6   fe80::a597:29ad:acdc:de01       1900    *       0               5620    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd948120  UDPv6   ::1     1900    *       0               5620    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd948c10  UDPv4   192.168.72.129  1900    *       0               5620    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd955be0  UDPv4   127.0.0.1       1900    *       0               5620    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ec0710af0  UDPv4   0.0.0.0 3389    *       0               436     svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec0710af0  UDPv6   ::      3389    *       0               436     svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec070e890  UDPv4   0.0.0.0 3389    *       0               436     svchost.exe     2025-11-27 17:01:40.000000 UTC
0×e00ec0b8e4e0  UDPv4   0.0.0.0 4500    *       0               2928    svchost.exe     2025-11-27 17:01:41.000000 UTC
0×e00ec0b8e4e0  UDPv6   ::      4500    *       0               2928    svchost.exe     2025-11-27 17:01:41.000000 UTC
0×e00ec0b8ecb0  UDPv4   0.0.0.0 4500    *       0               2928    svchost.exe     2025-11-27 17:01:41.000000 UTC
0×e00ec55b5810  UDPv4   0.0.0.0 5050    *       0               5856    svchost.exe     2025-11-27 17:03:41.000000 UTC
0×e00ecd953b10  UDPv4   0.0.0.0 5353    *       0               2388    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd953b10  UDPv6   ::      5353    *       0               2388    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd962890  UDPv4   0.0.0.0 5353    *       0               2388    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd944a70  UDPv4   0.0.0.0 5353    *       0               7932    msedge.exe      2025-11-27 18:27:59.000000 UTC
0×e00ecd946820  UDPv4   0.0.0.0 5353    *       0               7932    msedge.exe      2025-11-27 18:27:59.000000 UTC
0×e00ecd946820  UDPv6   ::      5353    *       0               7932    msedge.exe      2025-11-27 18:27:59.000000 UTC
0×e00ec92871e0  UDPv4   0.0.0.0 5355    *       0               2388    svchost.exe     2025-11-27 18:42:50.000000 UTC
0×e00ec92871e0  UDPv6   ::      5355    *       0               2388    svchost.exe     2025-11-27 18:42:50.000000 UTC
0×e00ec9285110  UDPv4   0.0.0.0 5355    *       0               2388    svchost.exe     2025-11-27 18:42:50.000000 UTC
0×e00ecd9612b0  UDPv6   fe80::a597:29ad:acdc:de01       49626   *       0               5620    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd963060  UDPv6   ::1     49627   *       0               5620    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd9655e0  UDPv4   192.168.72.129  49628   *       0               5620    svchost.exe     2025-11-27 18:27:51.000000 UTC
0×e00ecd964c80  UDPv4   127.0.0.1       49629   *       0               5620    svchost.exe     2025-11-27 18:27:51.000000 UTC
```

The netstat output for the post-attack dump remains similar:

- No additional suspicious outbound connections appear.
- Existing service ports are consistent with the baseline, reinforcing the assessment that the primary malicious activity was local credential access and memory dumping, not network-based exfiltration.

3.5 Cross-dump comparison of key processes



This screenshot contrasts selected high-value processes (explorer.exe, cmd.exe, conhost.exe, DumpIt.exe, mimikatz.exe, lsass.exe) across the three dumps. The following table summarises the comparison in report form (you can reconstruct it from pslist or the screenshot):

Process	Pre-attack dump	During-attack dump	Post-attack dump
explorer.exe	Present, parent userinit, no link to attack tools	Present; parent of cmd.exe that launches tools	Present; still holds handles to Mimikatz directory
cmd.exe	Not present or only short-lived baseline activity	Present as child of explorer.exe; parent of conhost.exe and attacker tools	May still be present with conhost.exe and DumpIt chain
conhost.exe	Only generic console hosts (if any)	Specific instance associated with cmd.exe, child for DumpIt.exe / mimikatz.exe	Associated with cmd.exe / DumpIt.exe, but no active mimikatz.exe child
DumpIt.exe	Absent	Present; child of cmd.exe; holds KsecDD and disk handles	Present or recently exited; still has kernel-level and tool directory handles
mimikatz.exe	Absent	Present; child of cmd.exe / conhost.exe; opens LSASS handles	No longer listed as a running process, but handles and explorer references remain
lsass.exe	Present with normal handles and DLLs	Present; targeted by mimikatz.exe handles and possibly by DumpIt via full memory access	Present; shows no crash but continues normal activity after being accessed

3.5.1 Pre- vs During-attack

Comparing the baseline and attack images shows a clear escalation:

- **New processes**: cmd.exe, a dedicated conhost.exe, DumpIt.exe and mimikatz.exe appear only in the attack dump.
- **Process hierarchy**: The pstree shifts from a pure user-interface tree to one that includes a command shell hosting offensive tooling.
- **Handles and DLLs**: Previously clean processes now load additional DLLs and hold sensitive handles:
    - DumpIt.exe opens \Device\KsecDD and volume devices, giving it full memory dumping capability.

- mimikatz.exe acquires full-access handles to lsass.exe, enabling credential theft.
- explorer.exe gains file handles into the attack tools directory, tying the GUI session to the malicious executables.

From a DFIR perspective, this delta is sufficient to attribute the attack to interactive activity by the logged-in user or an attacker using that session.

3.5.2 During- vs Post-attack

Comparing the attack and post-attack images reveals the attacker's cleanup behaviour:

- **Process termination**: mimikatz.exe is no longer running in the post-attack snapshot, while DumpIt and the console chain may remain. This suggests the attacker deliberately closed the more obviously malicious binary first.
- **Residual artefacts**:
  - explorer.exe still references the Mimikatz directory.
  - Handles from DumpIt to KsecDD and attack paths persist.
- **No new network activity**: The netstat output remains broadly unchanged, indicating no further exfiltration or lateral movement after the primary credential theft.

This phase is typical of an attacker trying to reduce their footprint while leaving the system running.

3.5.3 Combined view – Pre vs During vs Post

Looking at all three images together:

1. **Baseline** – A clean user session with standard Windows processes, normal LSASS activity, no interaction with attack tools, and benign network connections.
2. **Attack phase** –
   - Introduction of offensive tooling (DumpIt.exe, mimikatz.exe) launched via a user interactive shell (explorer.exe → cmd.exe → conhost.exe).
   - Kernel-level interaction (DumpIt.exe → \Device\KsecDD) confirming full memory access.
   - Direct credential theft activity (mimikatz.exe handles into lsass.exe).

    - No evidence of remote C2, suggesting local or “hands-on-keyboard” compromise.

3. **Post-attack** –
    - Termination of mimikatz.exe, but ongoing presence of DumpIt and console context.
    - Residual file and handle references from explorer and security components, which provide forensic proof of prior malicious activity even after the tools are closed.
    - System remains otherwise stable, with LSASS and core services running normally.

From an incident-response standpoint, this three-stage memory analysis provides:

- **Attribution of actions** – which processes executed, in what order, and under which parent process.
- **Scope of impact** – LSASS was accessed; full memory was dumped; however, no evidence of sustained network-based exfiltration or persistence mechanisms appears in the analysed artefacts.
- **Residual evidence** – even after attempted cleanup, handles and file references persist that clearly demonstrate the presence and use of DumpIt and mimikatz.

This completes the Phase 3 memory-forensic narrative and sets up Phase 4, where these technical findings can be translated into higher-level conclusions, recommendations, and mitigations.

Errors Faced During the Investigation

Throughout the project, several technical challenges were encountered. These issues mirror realistic obstacles that occur during actual incident-response operations.

1. Volatility 2 Compatibility Issues

One of the major challenges involved the incompatibility between the memory dump format generated by DumpIt (.dmp) and the Volatility 2 framework. Some memory captures lacked the required profile support, causing Volatility 2 to fail or produce partial results. This is a common issue when working with modern Windows OS versions that rely on newer kernel structures not fully supported in Volatility 2.

2. Insufficient Disk Space in the Kali VM

The memory dumps were large (multiple gigabytes), and downloading them from the Windows VM to Kali often failed due to insufficient allocated virtual disk space. This required resizing the virtual disk, removing snapshots, and reconfiguring storage—an effort that consumed time and required careful handling to avoid corruption of the forensic workstation.

3. Download Failures and Incomplete Transfers

While attempting to transfer memory images via the Kali-hosted HTTP server, several attempts failed due to storage limitations and temporary network misconfigurations. This resulted in corrupted or incomplete downloads, requiring validation of hashes and re-transfer to ensure forensic integrity.

4. Mimikatz Output Inconsistencies

Mimikatz execution did not always yield plaintext credentials, which can happen when certain Windows security components block credential caching or when LSASS protection is partially enabled. This required validation of whether credential theft actually occurred and whether memory artifacts still substantiated the attack.

5. Environment Isolation Conflicts

Switching between NAT and Host-Only networking modes occasionally caused connectivity loss between attacker and victim VMs. Isolated forensic environments often require careful network reconfiguration to maintain reachability without exposing attack tools to the external network.

Mitigation Steps and How Each Issue Was Resolved

1. Volatility 2 Compatibility

To overcome Volatility 2's limitations, Volatility 3 was installed and used as the primary analysis engine because of its superior plugin coverage, modern Windows compatibility, and reduced dependency on OS profiles. This ensured full extraction of processes, handles, DLLs, and LSASS-related artifacts.

2. Expanding Disk Space on Kali

VM snapshots were removed to unlock disk-extension features. The Kali VM's storage was expanded using VMware's disk management interface, followed by filesystem resizing from within the VM. This allowed proper storage of large memory dumps and ensured uninterrupted forensic analysis.

3. Ensuring Reliable Memory Transfers

After disk expansion, memory dumps were transferred again using the embedded Python HTTP server method. The integrity of each file was then validated using sha256sum to confirm that no corruption occurred during transfer.

4. Addressing Mimikatz Output Variability

Even though plaintext passwords were not retrieved, memory dumps still showed LSASS access attempts, Mimikatz process handles, relevant DLLs, and unlocked security tokens. These forensic traces were sufficient to prove credential dumping activity and reconstruct the attack workflow.

5. Stabilizing the Network Environment

Host-Only networking was reinstated for attack execution to ensure the victim could not reach the external internet. Connectivity was tested via reciprocal ICMP checks before proceeding with the attack. This preserved the isolation necessary for a legitimate forensic laboratory environment.

Recommendations

1. Always Acquire Multiple Memory Captures

Capturing *pre-attack*, *during attack*, and *post-attack* memory states dramatically enhances the accuracy of forensic timelines. This methodology should be adopted in real-world investigations wherever feasible.

2. Validate Memory Image Integrity

Memory dumps should always be hash-verified (MD5/SHA-256) before analysis to ensure forensic soundness. Integrity validation is critical in legal or enterprise breach-response contexts.

3. Prefer Volatility 3 for Modern Windows Systems

Volatility 3's plugin architecture, OS-agnostic approach, and improved support for modern Windows versions make it the recommended tool for contemporary incident response work.

4. Enable Secure Logging During Investigations

Maintaining logs of all commands executed, file transfer methods used, and intermediate results is essential for reproducibility. This practice is a standard requirement for forensic chain-of-custody.

5. Maintain Strict Environment Isolation

All malware testing and exploitation activity must occur in an isolated network (Host-Only). This limits the risk of accidental propagation and ensures forensic containment.

6. Use Automated Forensic Pipelines

In enterprise environments, memory acquisition and triage tools such as Velociraptor, KAPE, and GRR Rapid Response can significantly accelerate analysis, especially when dealing with larger incident scopes.

7. Perform Differential Analysis Across Dumps

Cross-dump comparison is one of the most powerful techniques in memory forensics. It exposes privilege escalation, process injection, unauthorized credential access, and persistence artifacts that may not be visible in a single snapshot.

8. Document Findings in a Structured IR Format

Reports should always include:

- Executive summary
- Scope and objectives
- Methodology
- Evidence collected
- Analysis
- Comparisons
- Conclusion
- Recommendations

This ensures the document aligns with industry reporting standards used by IR teams such as Mandiant, CrowdStrike, and Deloitte Cyber.

Conclusion

This investigation successfully replicated a full credential-theft attack chain and demonstrated how memory forensics can be used to reconstruct adversary behavior with high precision. Through three strategically timed memory captures, it was possible to map the evolution of system state from normal operation to active compromise and finally to the aftermath of the attack.

Volatility 3 provided extensive visibility into process execution, child–parent process relationships, LSASS interaction, loaded modules, handle objects, and other volatile artifacts that exposed the actions of Mimikatz even when the executable itself was no longer running. Differential analysis revealed critical behavioral shifts between baseline, active compromise, and post-attack states, thus providing a comprehensive timeline of malicious activity.

The methodologies, tools, and findings in this report reflect real-world digital forensics and incident-response workflows. The evidence collected and analyzed not only fulfills academic requirements but also demonstrates practical, industry-relevant forensic skills. This project is suitable for inclusion in a technical portfolio, GitHub repository, or professional resume.

References

CrowdStrike. *What Is Mimikatz?*. CrowdStrike, https://www.crowdstrike.com/en-us/blog/credential-theft-mimikatz-techniques/

Gentilkiwi, Benjamin Delpy. *Mimikatz Project*. GitHub, github.com/gentilkiwi/mimikatz.

Magnet Forensics. *DumpIt for Windows: Memory Acquisition Tool*. Magnet Forensics, http://www.magnetforensics.com.

Volatility Foundation. *Volatility 3 Framework Documentation*. Volatility Foundation, volatilityfoundation.org.

Moore, Harlan Carvey. *Windows Forensic Analysis Toolkit*. Syngress, 2018.

Ligh, Michael Hale, et al. *The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory*. Wiley, 2014.

Microsoft. *Credential Theft Mitigation Strategies*. Microsoft Docs, learn.microsoft.com.