

Complete Windows Forensics

Project Overview

This project is a **complete toolkit** and guide for **investigating Windows computers**. It helps investigators **collect and analyze memory, disk, and system artifacts** to understand **user activity and system behavior**. From capturing RAM and full disk images to analyzing **registry entries, system logs, and artifacts like MFT, USN Journal, and Prefetch**, the toolkit provides a full picture of what happened on a computer. It also includes **automated collection, evidence verification, and timeline creation** to make investigations faster, safer, and more organized.

Current Session: VM Acquisition & Triage Analysis :

Session Scope

In this practical session, we will cover:

- **Memory and disk acquisition** procedures within virtual environments
- **Alternative tools available** for physical machine acquisition
- **Mounting disk images** using Arsenal Image Mounter for analysis
- **Automated triage collection** using Kape for rapid evidence gathering
- **Registry examination** with Registry Explorer and bulk parsing techniques

Session Objectives - Key Questions to Answer:

- **Active accounts during the attack timeframe ?**
- **Which account(s) were created ?**
- **Which accounts are Administrator group members ?**
- **Which users have profiles on the system ?**

Note: This session covers the basic acquisition and triage phase. Advanced artifact analysis will be addressed in later sessions.

Tools Required :

Tool	Purpose
VirtualBox & VBoxManage	Virtual machine management and evidence acquisition
FTK Imager	Disk imaging and forensic analysis
Arsenal Image Mounter	Mounting disk images for analysis
Kape	Automated forensic triage and artifact collection
Registry Explorer	Windows registry analysis and exploration
RegRipper	Automated registry parsing and data extraction
Notepad++	Log analysis and text file examination

Investigation Documentation :

Field	Details
Case Number	CF-2025-001
Investigator	NIZAR ADERBAZ
Date	2025-11-11

Target System	Windows 10 Pro
Investigation Type	Complete Windows System Forensics

1) Memory Forensics: Acquisition

Memory forensics involves capturing and analyzing a computer's **RAM (Random Access Memory)** to extract **volatile evidence** that disappears when the system is powered off. This includes **running processes, network connections, open files, encryption keys, and malware activities.**

Memory Acquisition Methods :

To capture a memory image, we can use various public tools like **FTK Imager** for physical machines, or if we're in a **VirtualBox environment**, we can use the built-in **VBoxManage tool**.

Step 1: Identify Virtual Machine

First, we need to identify our **VMs** and obtain the **UUID** of the target machine:

Command :

⇒ `.\VBoxManage.exe list vms`

A screenshot of a Windows Command Prompt window titled "Select Administrator: Command Prompt". The command entered is "VBoxManage.exe list vms". The output shows five virtual machines: "Kali Linux", "REMinux v7", "Malware Machine", "DF Machine", and "Victim". The "DF Machine" UUID is highlighted with a red box. The Command Prompt path is "C:\Users\NIZAR\Documents\VirtualBox>". The system tray icons are visible at the bottom, and the system status bar shows "ENG US" and the date "11/11/2025" at the bottom right.

```
C:\Users\NIZAR\Documents\VirtualBox>VBoxManage.exe list vms
"Kali Linux" {b79a4107-7d8c-496d-b401-bf1693b6f071}
"REMinux v7" {e6bb663c2-8cb4-4027-a252-a3feefde9a3e}
"Malware Machine" {3fd237a6-8548-4c5d-b4ca-6ca650b81b8d}
"DF Machine" {cde83ad7-6c0f-4003-8749-6a9e518e64b3}
"Victim" {88d9ct8b-9a3b-4e/e-8036-3b294t/42c50}

C:\Users\NIZAR\Documents\VirtualBox>
```

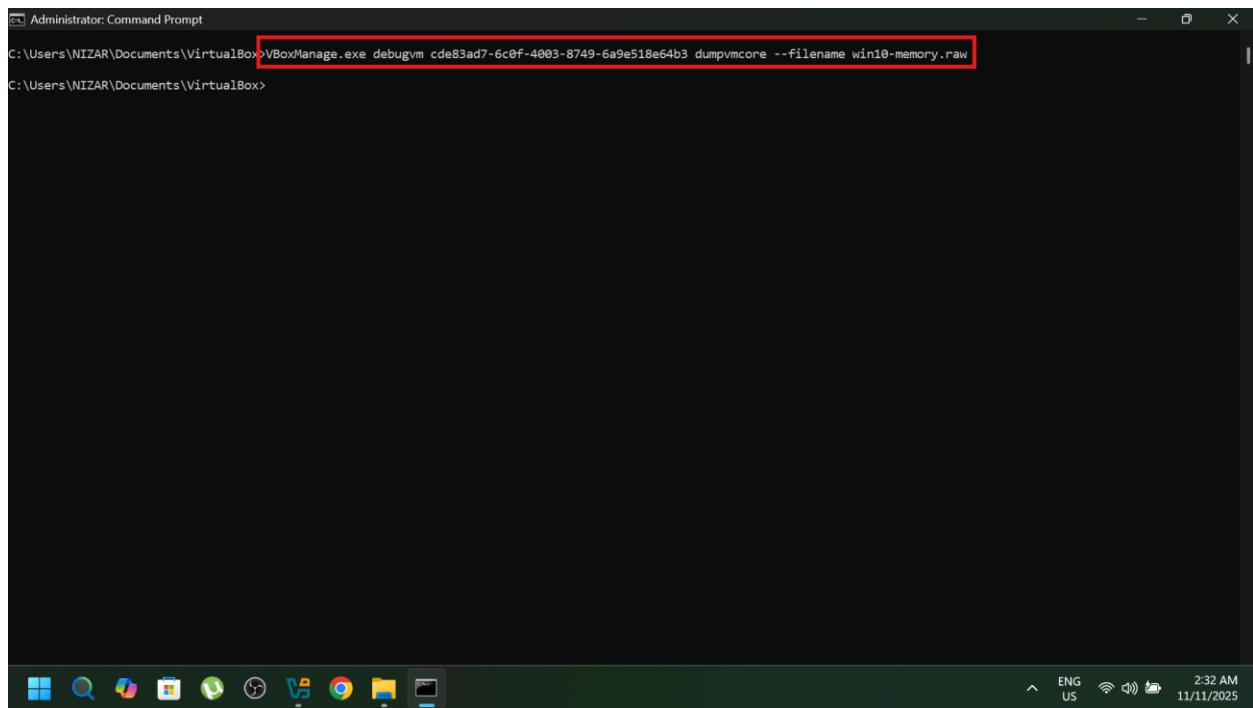
Output showing all available virtual machines with their UUIDs

Step 2: Capture Memory Image

Using the **UUID** from the previous step, we capture **the memory image**:

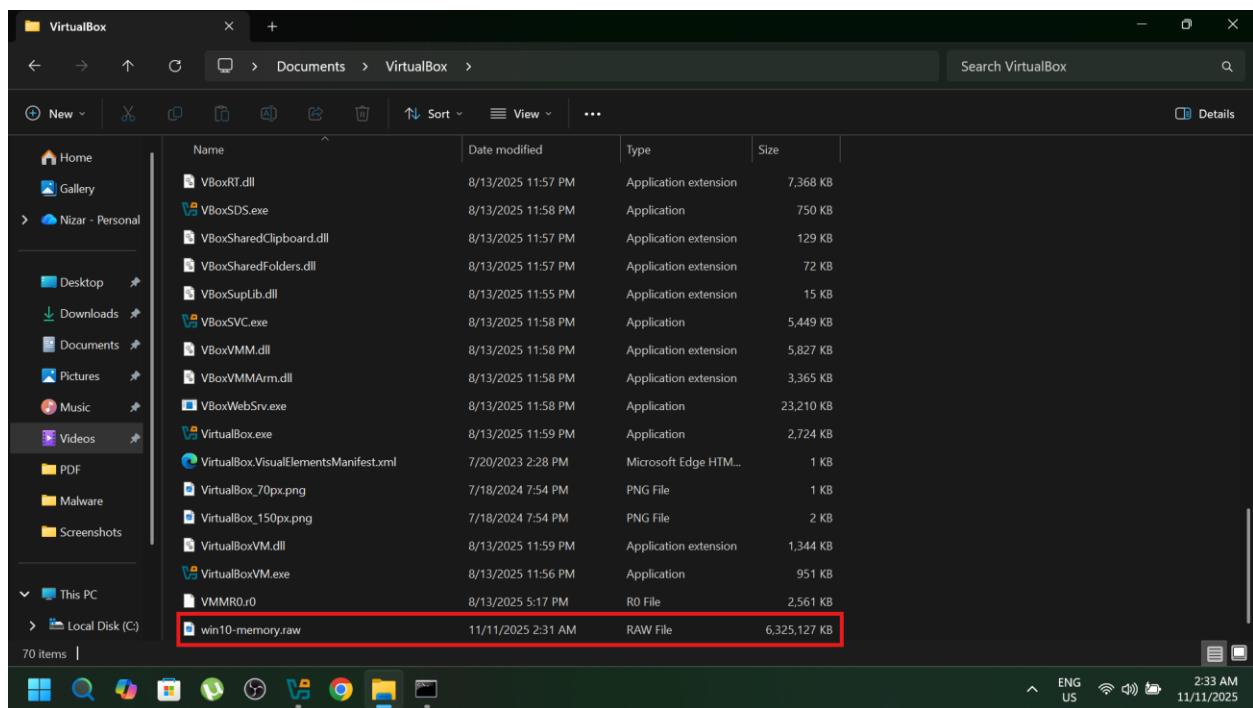
Command :

⇒ .\VBoxManage.exe debugvm cde83ad7-6c0f-4003-8749-6a9e518e64b3 dumpvmpcore --filename win10-memory.raw



```
Administrator: Command Prompt
C:\Users\NIZAR\Documents\VirtualBox>VBoxManage.exe debugvm cde83ad7-6c0f-4003-8749-6a9e518e64b3 dumpvmcore --filename win10-memory.raw
C:\Users\NIZAR\Documents\VirtualBox>
```

Executing `VBoxManage debugvm` command to capture memory dump from target virtual machine



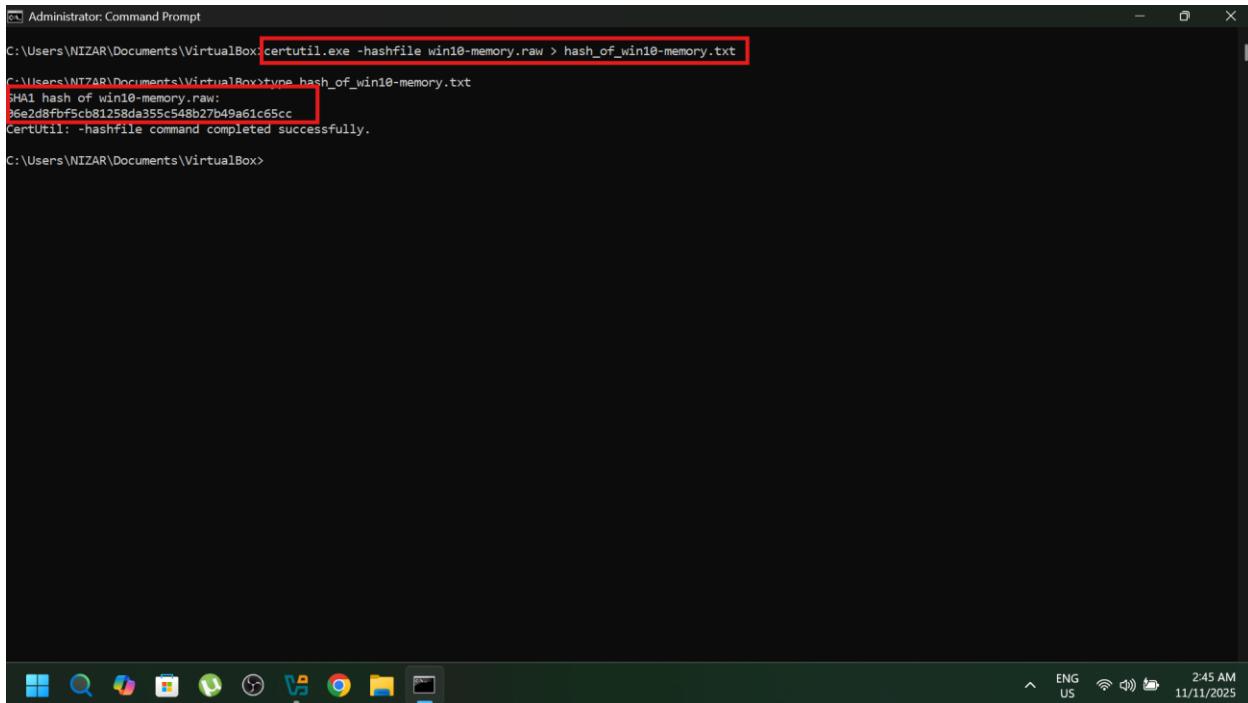
Memory dump file (`win10-memory.raw`) successfully created in the target folder

Step 3: Verify Integrity

After acquiring **the memory dump**, we calculate the **hash** for integrity verification:

Command :

⇒ `certutil.exe -hashfile win10-memory.raw > hash_of_win10-memory.txt`



The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The command `certutil.exe -hashfile win10-memory.raw > hash_of_win10-memory.txt` is entered and executed. The output shows the SHA-1 hash of the file: `SHA1 hash of win10-memory.raw: 96e2d8fbfcbb1258da355c548b27b49a61c65cc`. A red box highlights this hash value. The command concludes with `CertUtil: -hashfile command completed successfully.`. The system tray at the bottom right shows the date and time as 11/11/2025 and 2:45 AM.

SHA-1 hash verification of memory dump file using certutil, ensuring evidence integrity

2) Disk Forensics: Acquisition

Disk forensics involves **creating a forensic image of a computer's storage media (HDD, SSD)** to **preserve file systems, documents, applications, and system artifacts for analysis**. Unlike memory, **disk evidence persists after shutdown**.

Disk Acquisition Methods

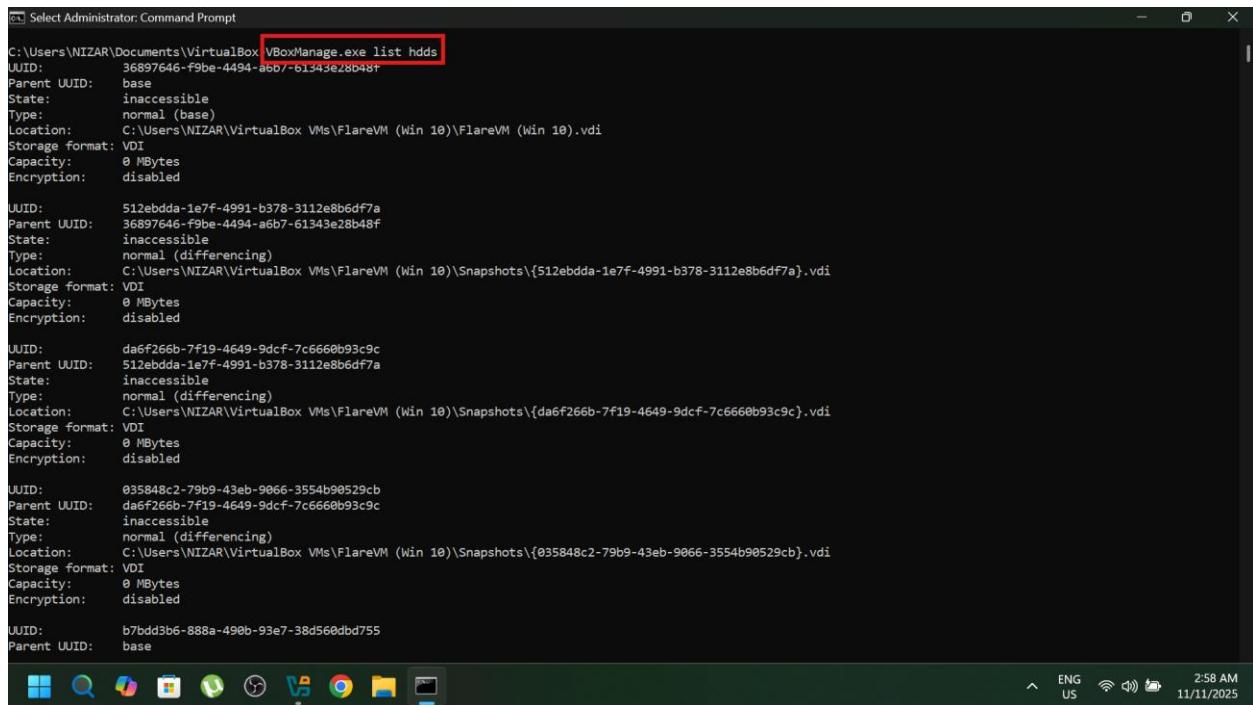
Now that **memory acquisition is complete**, we can safely shut down the target system and proceed with disk imaging using **VBoxManage**.

Step 1: Identify Disk Images

First, list all available disk images to identify the target:

Command :

⇒ .\VBoxManage.exe list hdds



```
C:\Users\NIZAR\Documents\VirtualBox>.\VBoxManage.exe list hdds
UUID: 36897646-f9be-4494-abd7-b1343e28b48f
Parent UUID: base
State: inaccessible
Type: normal (base)
Location: C:\Users\NIZAR\VirtualBox VMs\FlareVM (Win 10)\FlareVM (Win 10).vdi
Storage format: VDI
Capacity: 0 MBytes
Encryption: disabled

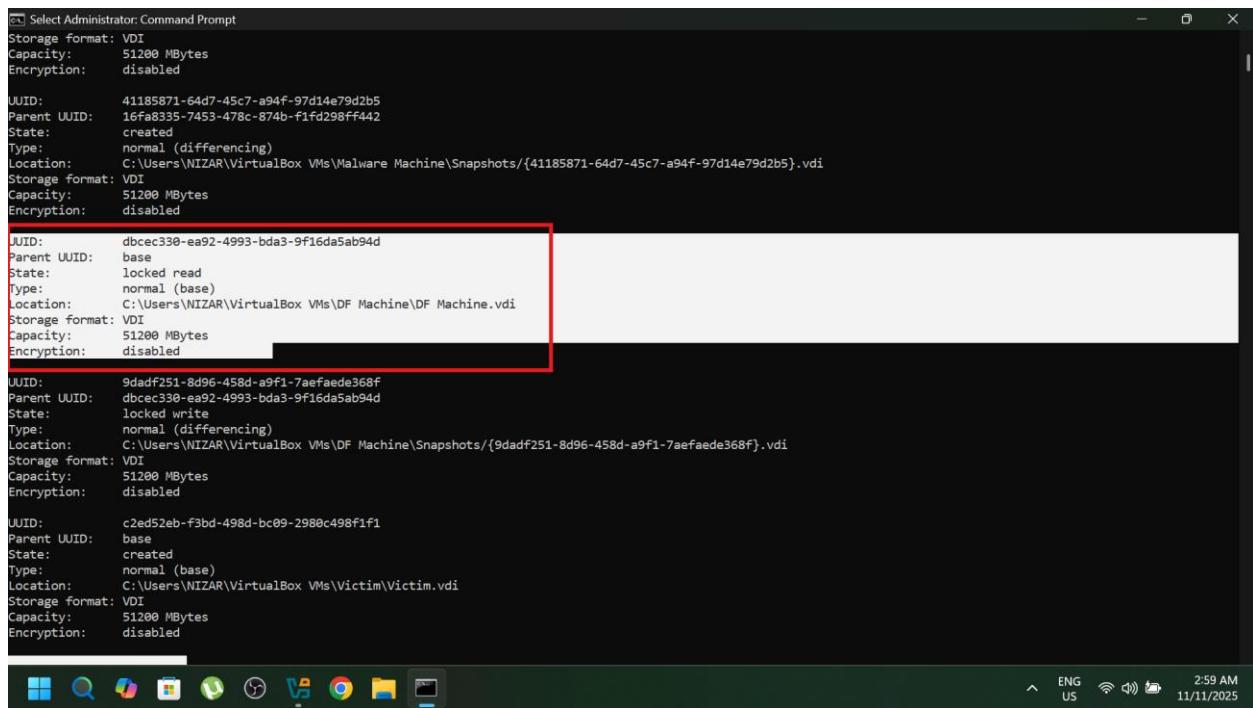
UUID: 512ebdda-1e7f-4991-b378-3112e8b6df7a
Parent UUID: 36897646-f9be-4494-abd7-b1343e28b48f
State: inaccessible
Type: normal (differencing)
Location: C:\Users\NIZAR\VirtualBox VMs\FlareVM (Win 10)\Snapshots\{512ebdda-1e7f-4991-b378-3112e8b6df7a}.vdi
Storage format: VDI
Capacity: 0 MBytes
Encryption: disabled

UUID: da6f266b-7f19-4649-9dcf-7c6660b93c9c
Parent UUID: 512ebdda-1e7f-4991-b378-3112e8b6df7a
State: inaccessible
Type: normal (differencing)
Location: C:\Users\NIZAR\VirtualBox VMs\FlareVM (Win 10)\Snapshots\{da6f266b-7f19-4649-9dcf-7c6660b93c9c}.vdi
Storage format: VDI
Capacity: 0 MBytes
Encryption: disabled

UUID: 035848c2-79b9-43eb-9066-3554b90529cb
Parent UUID: da6f266b-7f19-4649-9dcf-7c6660b93c9c
State: inaccessible
Type: normal (differencing)
Location: C:\Users\NIZAR\VirtualBox VMs\FlareVM (Win 10)\Snapshots\{035848c2-79b9-43eb-9066-3554b90529cb}.vdi
Storage format: VDI
Capacity: 0 MBytes
Encryption: disabled

UUID: b7dd3b6-888a-490b-93e7-38d560dbd755
Parent UUID: base
```

Listing all virtual disk images with VBoxManage to identify the target disk UUID for acquisition



```
Administrator: Command Prompt
Storage format: VDI
Capacity: 51200 MBytes
Encryption: disabled

UUID: 41185871-64d7-45c7-a94f-97d14e79d2b5
Parent UUID: 16fa8335-7453-478c-874b-f1fd298ff442
State: created
Type: normal (differencing)
Location: C:\Users\NIZAR\VirtualBox VMs\Malware Machine\Snapshots\{41185871-64d7-45c7-a94f-97d14e79d2b5}.vdi
Storage format: VDI
Capacity: 51200 MBytes
Encryption: disabled

JUID: dbcec330-ea92-4993-bda3-9f16da5ab94d
Parent UUID: base
State: locked read
Type: normal (base)
Location: C:\Users\NIZAR\VirtualBox VMs\DF Machine\DF Machine.vdi
Storage format: VDI
Capacity: 51200 MBytes
Encryption: disabled

UUID: 9dadf251-8d96-458d-a9f1-7aeefade368f
Parent UUID: dbcec330-ea92-4993-bda3-9f16da5ab94d
State: locked write
Type: normal (differencing)
Location: C:\Users\NIZAR\VirtualBox VMs\DF Machine\Snapshots\{9dadf251-8d96-458d-a9f1-7aeefade368f}.vdi
Storage format: VDI
Capacity: 51200 MBytes
Encryption: disabled

UUID: c2ed52eb-f3bd-498d-bc09-2980c498f1f1
Parent UUID: base
State: created
Type: normal (base)
Location: C:\Users\NIZAR\VirtualBox VMs\Victim\Victim.vdi
Storage format: VDI
Capacity: 51200 MBytes
Encryption: disabled
```

Identifying the victim's virtual disk from the list of available disk images

Step 2: Capture Disk Image

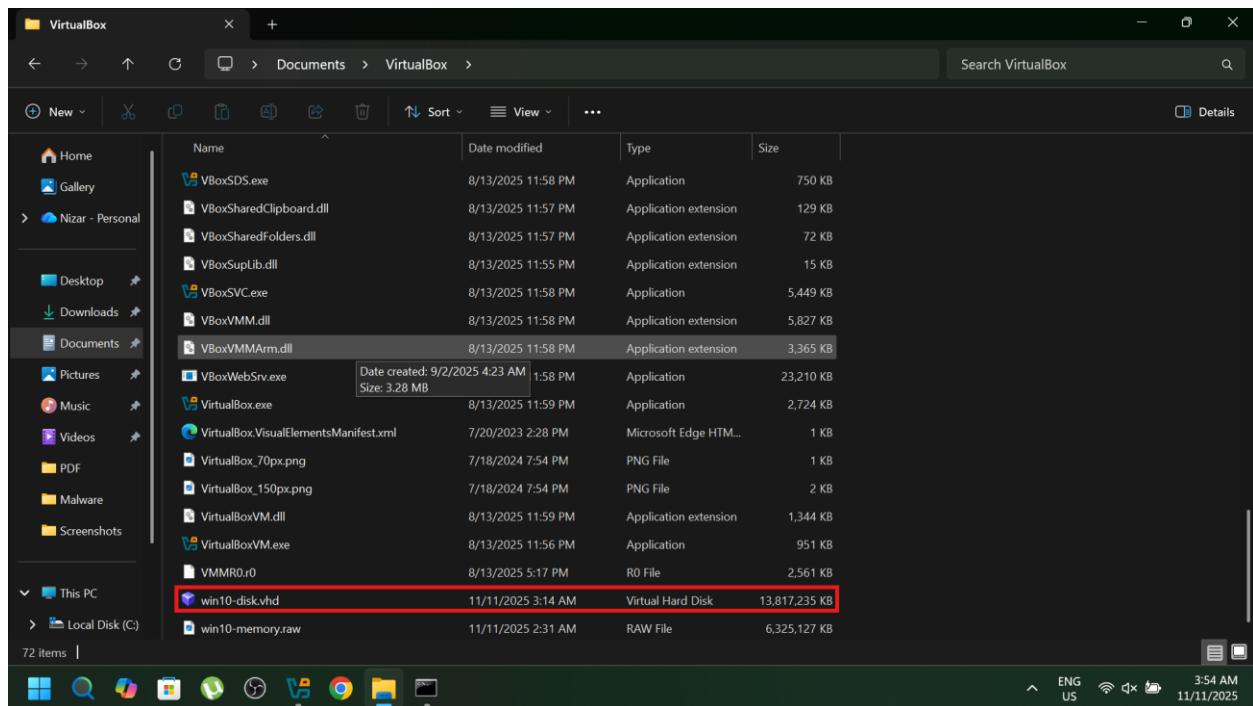
Using the disk UUID, create a forensic copy of the disk:

Command :

⇒ .\VBoxManage.exe clonemedium disk dbcec330-ea92-4993-bda3-9f16da5ab94d --format VHD win10-disk.vhd

```
C:\Users\NIZAR\Documents\VirtualBox>VBoxManage.exe clonemedium disk dbcec330-ea92-4993-bda3-9f16da5ab94d --format VHD win10-disk.vhd  
2%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%  
clone medium created in format 'VHD'. UUID: d8599c4f-9a63-483f-b948-fd30da7bac70  
C:\Users\NIZAR\Documents\VirtualBox>
```

Creating forensic image of victim's disk using VBoxManage clonemedium command



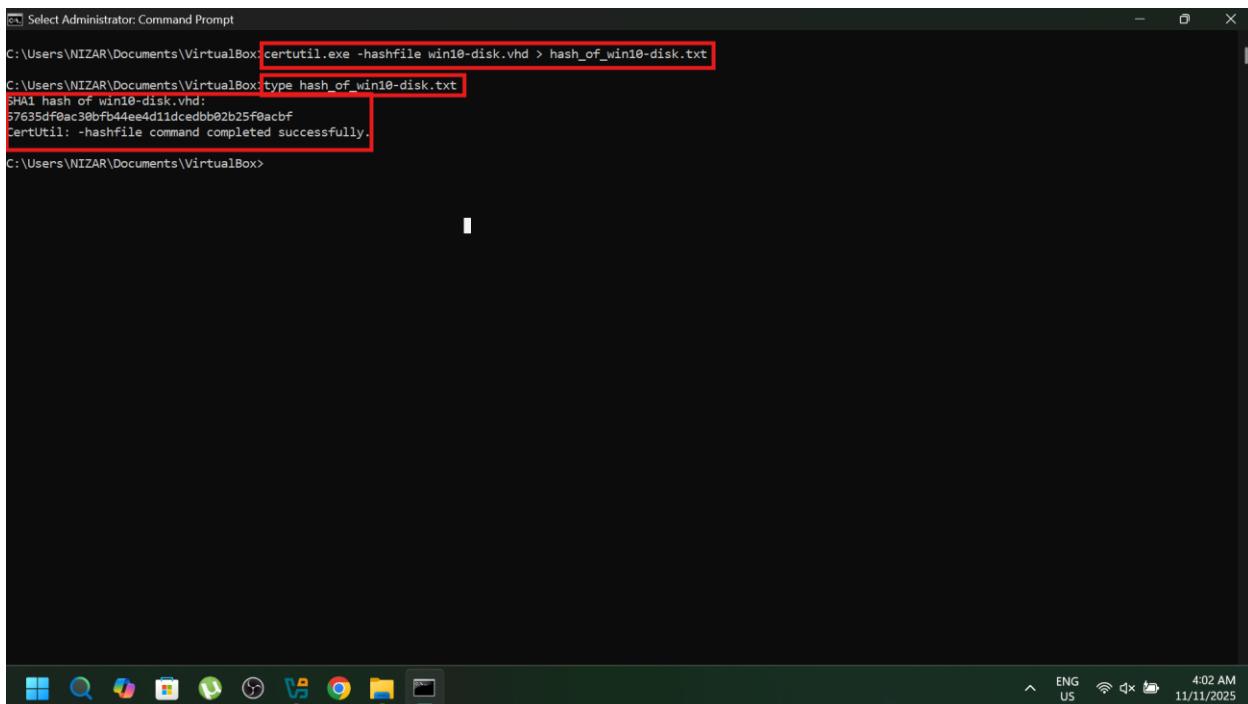
Victim's disk image (win10-disk.vhd) successfully created and visible in the evidence folder

Step 3: Verify Integrity

Generate hash verification for the disk image:

Command :

⇒ **certutil.exe -hashfile win10-disk.vhd > hash_of_win10-disk.txt**



The screenshot shows a Windows Command Prompt window titled "Select Administrator: Command Prompt". The command entered is "certutil.exe -hashfile win10-disk.vhd > hash_of_win10-disk.txt". The output shows the SHA-1 hash of the disk image: "SHA1 hash of win10-disk.vhd: 57635d0ac30bf044e4d11cedbb02b25f0acbfb". The command concludes with "CertUtil: -hashfile command completed successfully.". The taskbar at the bottom includes icons for File Explorer, Task View, Start, Search, Taskbar View, Task Manager, File Explorer, and Task View again. The system tray shows the date and time as 11/11/2025 and 4:02 AM.

Verifying integrity of victim's disk image using SHA-1 hash calculation

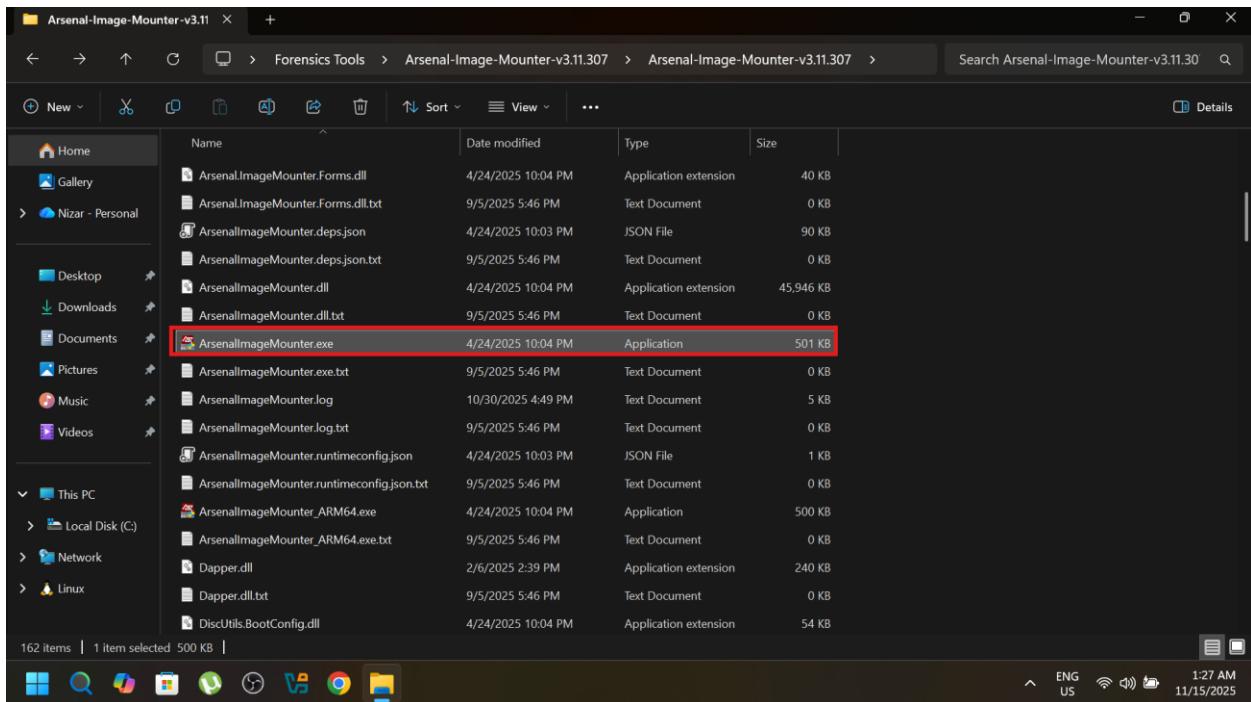
3) Disk Forensics: Mounting & Analysis

Disk mounting allows us to access and interact with forensic disk images as if they were physical drives, enabling file system exploration and analysis while maintaining evidence integrity.

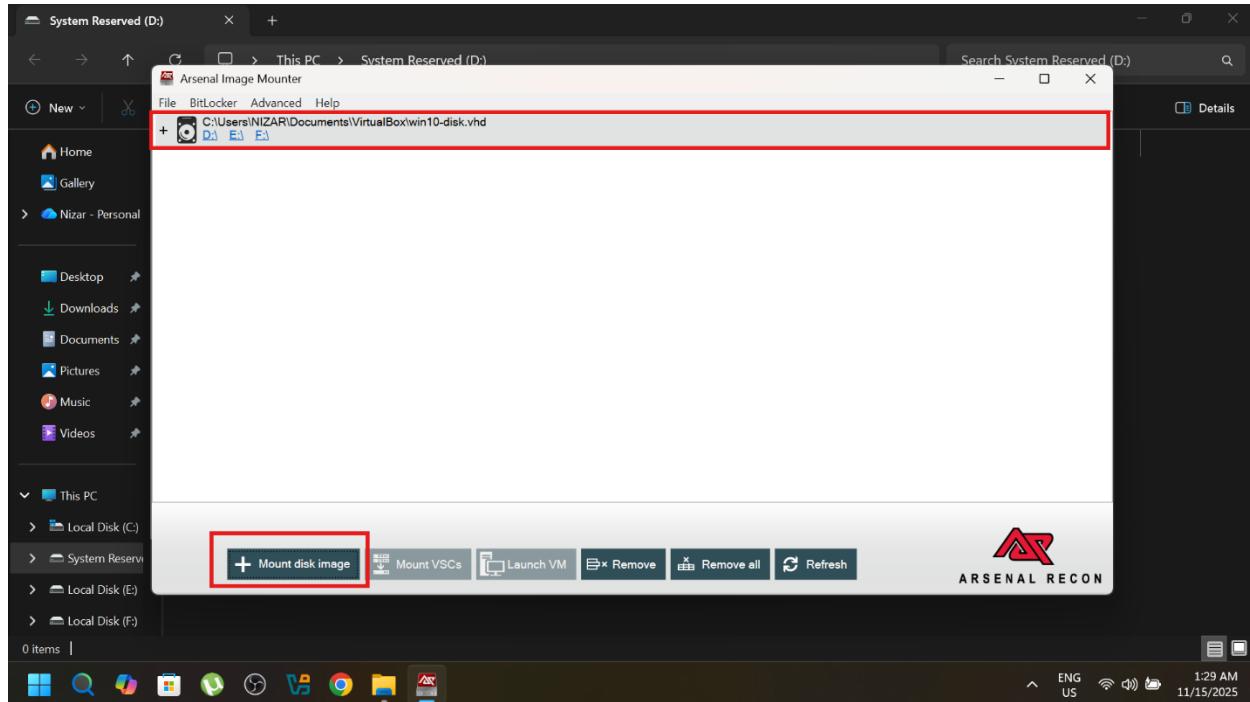
Now that we have our disk image, we'll use **Arsenal Image Mounter** to mount it for analysis.

Step 1: Open Arsenal Image Mounter

- Launch **Arsenal Image Mounter** application
- Click "Mount Image" and select our **win10-disk.vhd** file



Launching Arsenal Image Mounter application to begin the disk mounting process



Selecting "Mount Image" in Arsenal and choosing the victim's `win10-disk.vhd` file for analysis

Important Mounting Configuration !!!

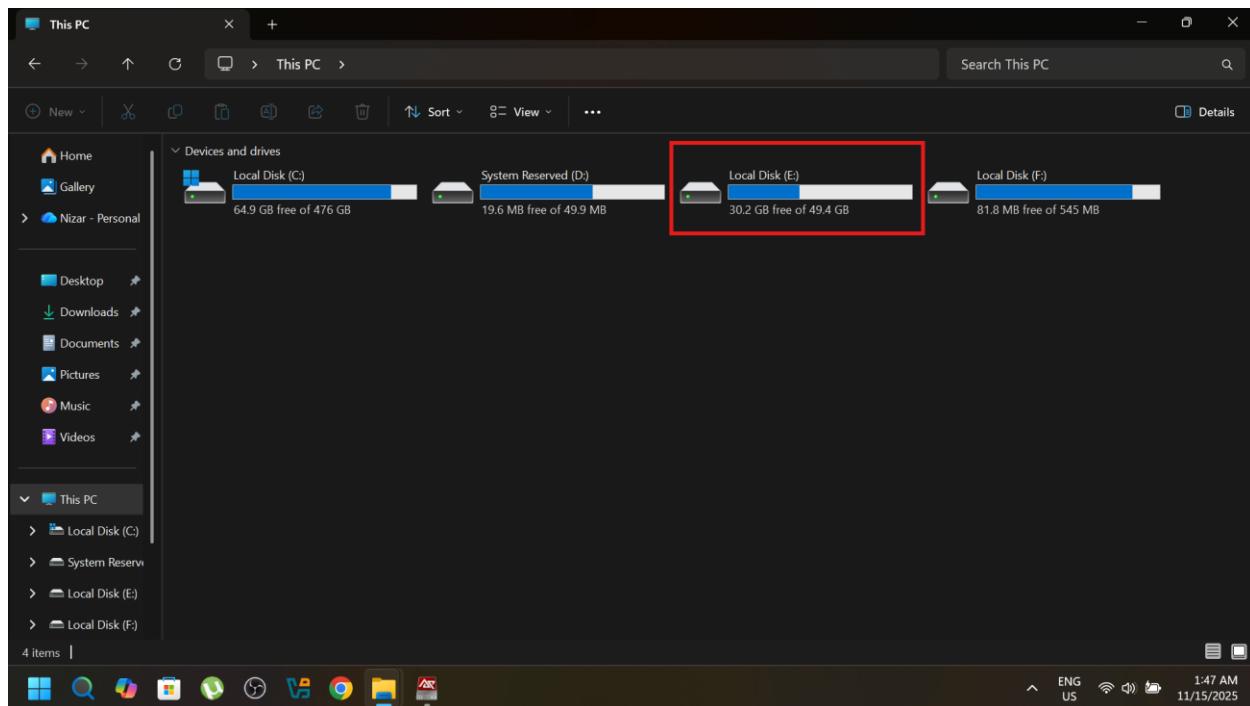
When Arsenal prompts for mounting options, ensure you select:

- **Mount on: Physical drive letter (e.g., E:)**
- **Access mode: Write temporary (creates temporary writeable copy)**
- **File system: Automatic detection**

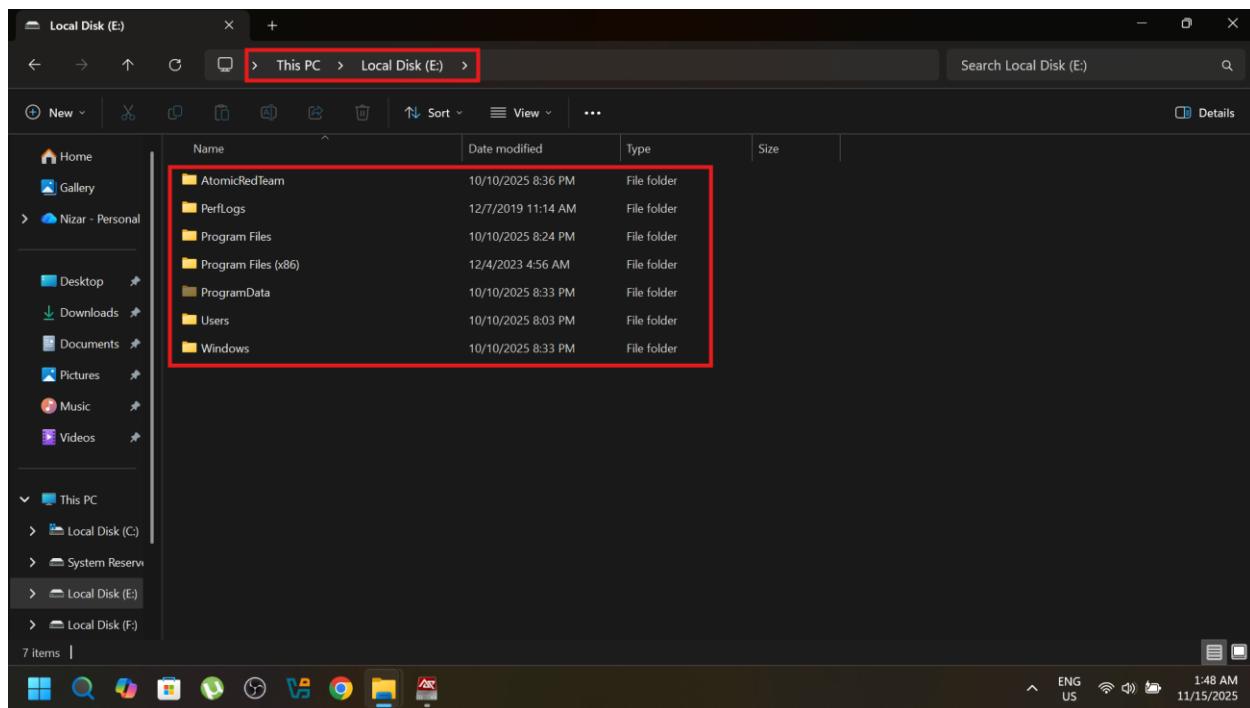
Note: The "Write temporary" option is crucial as it creates a temporary copy for analysis while preserving the original evidence integrity.

Step 2: Access Mounted Disk

Once mounted, the disk will appear as a new drive in Windows Explorer, but we'll use command line for complete access.



Victim's disk successfully mounted as drive E: in Windows Explorer - ready for command line analysis



Initial view inside the mounted victim's disk showing Windows system files and folders

Forensic Triage with Kape

Forensic triage involves **rapidly collecting and analyzing key digital artifacts** to quickly understand **system activity, user behavior, and potential security incidents**.

Now that our disk is mounted, we'll use **Kape** to automatically collect crucial forensic artifacts from the victim's system.

Step 1: Prepare Kape Command

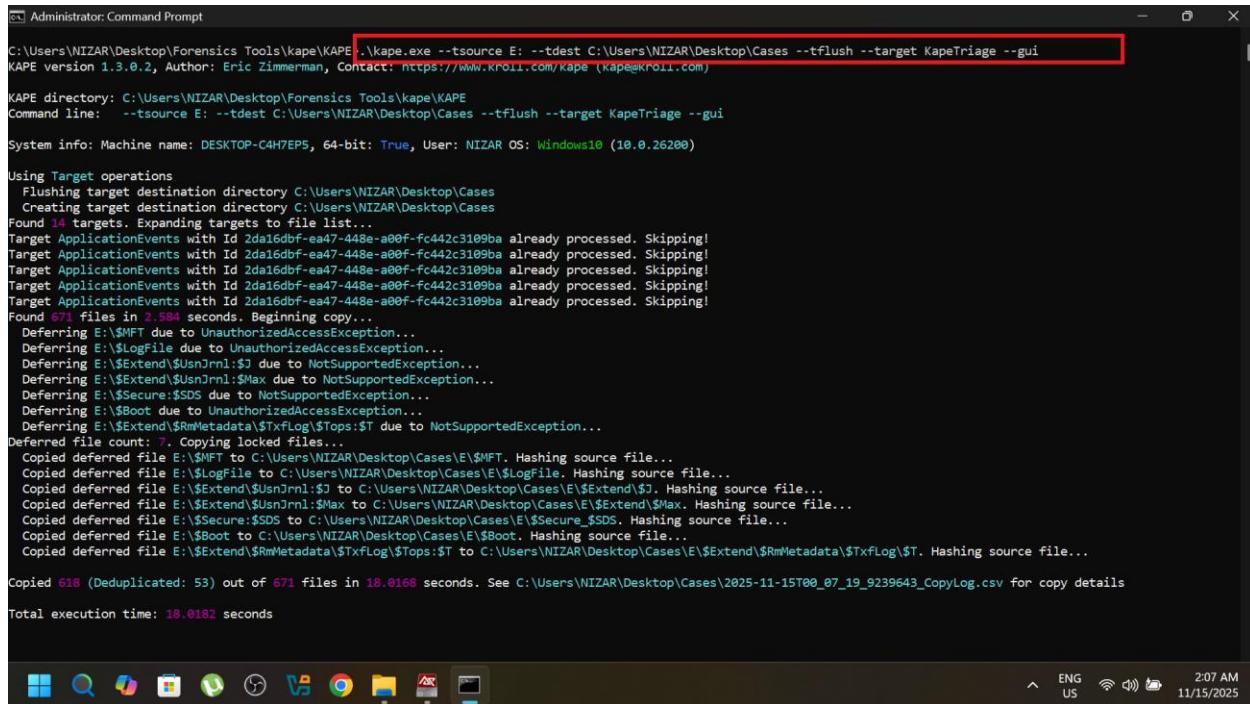
We'll use **Kape** to target our mounted **drive (E:)** and collect essential artifacts:

Command :

```
⇒ .\kape.exe --tsource E: --tdest C:\Users\NIZAR\Desktop\Cases --tflush --target  
KapeTriage --gui
```

Command Breakdown

- ❖ **--tsource E:** : Target source (our mounted victim disk)
- ❖ **--tdest C** : \Users\NIZAR\Desktop\Cases: Destination for collected artifacts
- ❖ **--tflush** : Clear destination before collection
- ❖ **--target KapeTriage**: Predefined collection targets
- ❖ **--gui** : Show graphical interface



```
C:\Users\NIZAR\Desktop\Forensics Tools\kape\KAPE> ./kape.exe --tsource E: --tdest C:\Users\NIZAR\Desktop\Cases --tflush --target KapeTriage --gui
KAPE version 1.3.0.2, Author: Eric Zimmerman, Contact: https://www.Kroll.com/kape (Kape@Kroll.com)

KAPE directory: C:\Users\NIZAR\Desktop\kape\KAPE
Command line: --tsource E: --tdest C:\Users\NIZAR\Desktop\Cases --tflush --target KapeTriage --gui

System info: Machine name: DESKTOP-C4H7EP5, 64-bit: True, User: NIZAR OS: Windows10 (10.0.26200)

Using Target operations
    Flushing target destination directory C:\Users\NIZAR\Desktop\Cases
    Creating target destination directory C:\Users\NIZAR\Desktop\Cases
Found 14 targets. Expanding targets to file list...
Target ApplicationEvents with Id 2da16dbf-ea47-448e-a00f-fc442c3109ba already processed. Skipping!
Found 671 files in 3.584 seconds. Beginning copy...
    Deferring E:\$MFT due to UnauthorizedAccessException...
    Deferring E:\$LogFile due to UnauthorizedAccessException...
    Deferring E:\$Extend\($UsnJrnL:$J due to NotSupportedException...
    Deferring E:\$Extend\($UsnJrnL:$Max due to NotSupportedException...
    Deferring E:\$Secure:$SDS due to NotSupportedException...
    Deferring E:\$Boot due to UnauthorizedAccessException...
    Deferring E:\$Extend\($RmMetadata\$TxLog\$Tops:$T due to NotSupportedException...
    Deferred file count: 7. Copying locked files...
Copied deferred file E:\$MFT to C:\Users\NIZAR\Desktop\Cases\E\$\MFT. Hashing source file...
Copied deferred file E:\$LogFile to C:\Users\NIZAR\Desktop\Cases\E\$\LogFile. Hashing source file...
Copied deferred file E:\$Extend\($UsnJrnL:$J to C:\Users\NIZAR\Desktop\Cases\E\$\Extend\($J. Hashing source file...
Copied deferred file E:\$Extend\($UsnJrnL:$Max to C:\Users\NIZAR\Desktop\Cases\E\$\Extend\($Max. Hashing source file...
Copied deferred file E:\$Secure:$SDS to C:\Users\NIZAR\Desktop\Cases\E\$\Secure:$SDS. Hashing source file...
Copied deferred file E:\$Boot to C:\Users\NIZAR\Desktop\Cases\E\$\Boot. Hashing source file...
Copied deferred file E:\$Extend\($RmMetadata\$TxLog\$Tops:$T to C:\Users\NIZAR\Desktop\Cases\E\$\Extend\($RmMetadata\$TxLog\$T. Hashing source file...

Copied 618 (Deduplicated: 53) out of 671 files in 18.0188 seconds. See C:\Users\NIZAR\Desktop\Cases\2025-11-15T00_07_19_9239643_CopyLog.csv for copy details
Total execution time: 18.0182 seconds
```

Kape command executing successfully, showing the triage collection process in progress

Registry Forensics Analysis :

The Windows Registry is a hierarchical database that stores system configuration, user settings, and application data. Registry forensics examines these artifacts to reconstruct user activities, system changes, and potential malicious behavior.

Now that we have our triage data, we'll analyze the registry using specialized forensic tools.

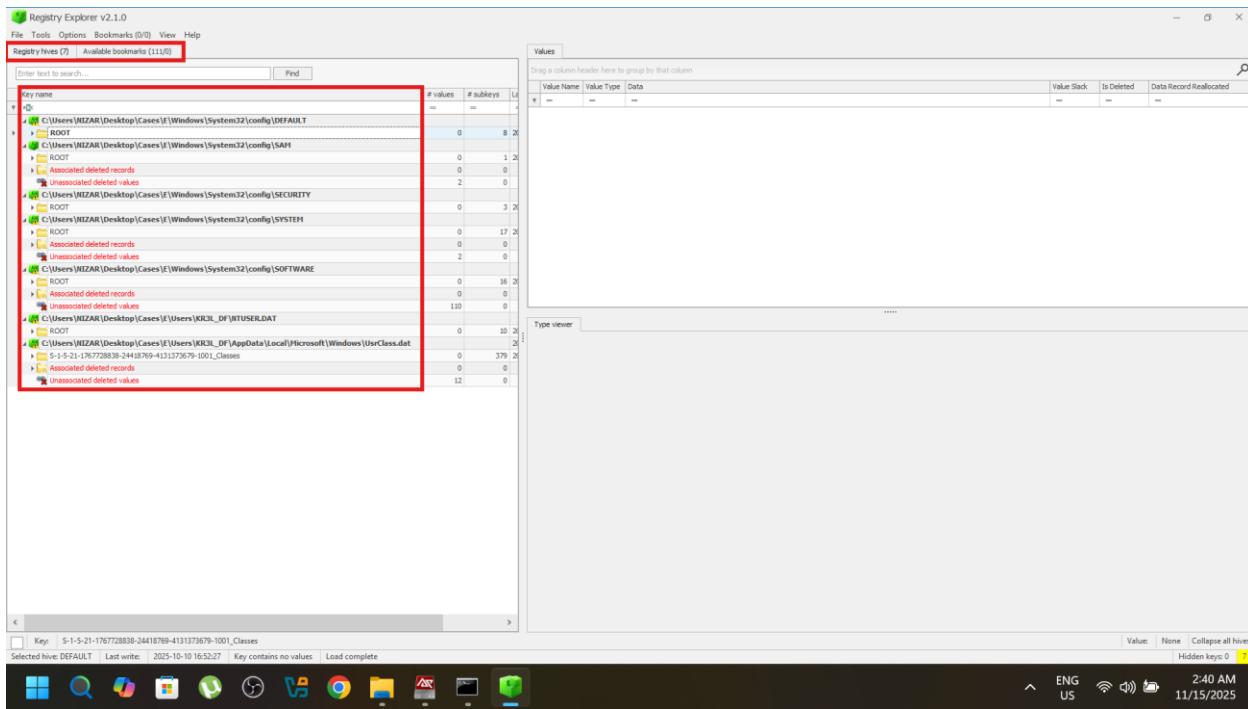
Step 1: Locate Registry Hives

Key registry hives are located at:

- ❖ **User Profiles:** E:\Users\Target\NTUSER.DAT
 - ❖ **User Classes:** E:\Users\Target\AppData\Local\Microsoft\Windows\UsrClass.dat
 - ❖ **System Hives:** E:\Windows\System32\config\ (SYSTEM, SOFTWARE, SAM, SECURITY)

Step 2: Analyze with Registry Explorer

We'll use **Registry Explorer** to load and examine the registry hives with **built-in bookmarks** for common forensic artifacts.



Using Registry Explorer to load victim's registry hives with built-in forensic bookmarks for artifact analysis

Step 3: Bulk Parsing with RegRipper

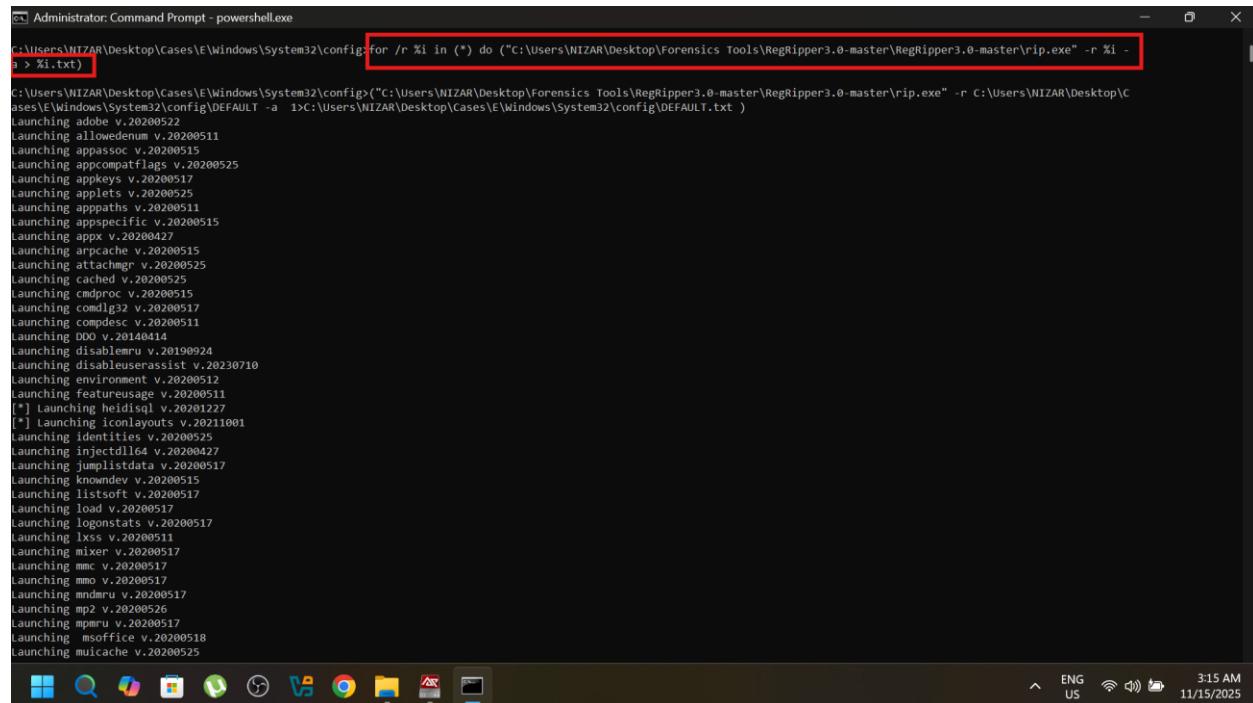
For comprehensive analysis, we'll use **RegRipper** to parse all registry hives automatically using bulk processing.

Command :

```
+> for /r %i in (*.dat) do ("C:\Users\NIZAR\Desktop\Tools\RegRipper3.0-master\RegRipper3.0-master\rip.exe" -r %i -a > %i.txt)
```

Command Breakdown :

- ❖ **for /r %i in (*.dat) do:** Loops through all **(.dat)** files in current directory
- ❖ **rip.exe -r %i -a:** Runs **RegRipper** on each file with automatic plugin selection
- ❖ **%i.txt:** Outputs results to text files



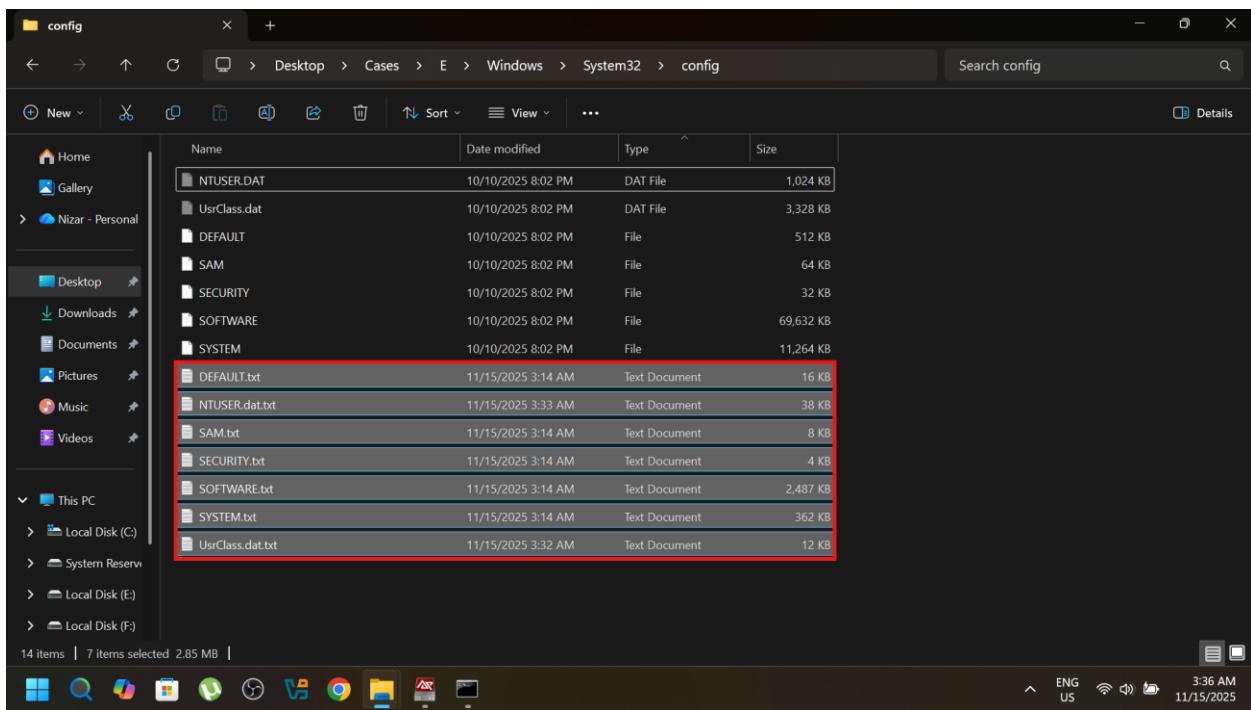
```
C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config>for /r %i in (*.dat) do ("C:\Users\NIZAR\Desktop\Tools\RegRipper3.0-master\RegRipper3.0-master\rip.exe" -r %i -a > %i.txt")
C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config>("C:\Users\NIZAR\Desktop\Tools\RegRipper3.0-master\RegRipper3.0-master\rip.exe" -r C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\DEFAULT -a >C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\DEFAULT.txt )
Launching adobe.v.20200511
Launching allowedenv.V.20200511
Launching appassoc.v.20200515
Launching appcompatflags.V.20200525
Launching appkeys.v.20200517
Launching applets.v.20200525
Launching apppaths.v.20200511
Launching appspecific.V.20200515
Launching appx.v.20200427
Launching arpcache.v.20200515
Launching attachmg.v.20200525
Launching cached.v.20200525
Launching cmdproc.v.20200515
Launching cmdlg32.v.20200517
Launching compdesc.v.20200511
Launching DDO.v.20140414
Launching disablemru.v.20190924
Launching disableuserassist.v.20230710
Launching environment.v.20200512
Launching featureusage.v.20200511
[*] Launching heidisql.v.20201227
[*] Launching iconlayouts.v.20211001
Launching identities.v.20200525
Launching injectcdll64.v.20200427
Launching jumplistdata.v.20200517
Launching knowndev.v.20200515
Launching listsoft.v.20200517
Launching load.v.20200517
Launching logonstats.v.20200517
Launching Ixss.v.20200511
Launching mixer.v.20200517
Launching mmc.v.20200517
Launching mmc.v.20200517
Launching mndmru.v.20200517
Launching mp2.v.20200526
Launching mpmmru.v.20200517
Launching msoffice.v.20200518
Launching mulcache.v.20200518
```

Bulk registry parsing command successfully completed, generating text files for all analyzed registry hives

After processing:

- Select all **.txt files** in the output directory
 - Open with **Notepad++**
 - Use **Ctrl+F** to search across all files for **specific artifacts** and **plugins**

This automated approach ensures we extract all available registry evidence efficiently.



Generated text files containing parsed registry data from all hives, ready for analysis in Notepad++

All registry text files opened in Notepad++ for comprehensive search and analysis across parsed data

Answering Investigation Questions :

Now that we have completed our **forensic acquisition**, **triage**, and **parsing**, we can answer the key investigation questions using the collected evidence.

Questions to Answer:

1. Active accounts during the attack timeframe?

To identify active accounts:

- Open all registry text files in **Notepad++**
 - Press **Ctrl + F** to open search
 - Search for: **Last Login**

C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SYSTEM.txt - Notepad++

```

1 appcompatdls v.20200427
2 ControlSet001\Session Manager\appCompatDlls not found.
3 =====
4 appcompatcache v.20220921
5 (System) Param files from System hive AppCompatCache
6 ControlSet001\Session Manager\appCompatCache
7 LastWrite Time: 2025-10-10 17:02:42Z
8 Signature: 0x34
9 C:\Windows\layer\m3\appcompatcache.exe 2023-12-04 02:48:43
10 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:51:09
11 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:49:29
12 00000000 03e4a610e40000 0000000000000000 8664 Microsoft.AAD.BrokerPlugin cw5nh2txyewy neutral
13 00000000 000a0004a10e34 000a0004a10e34 8664 Microsoft.Windows.OOBENetworkaptivePortal d
14 00000000 0000000000000000 000000027410000 8664 Microsoft.NET.Native.Runtime.2.2 Swekyb3db
15 00000000 03e4a610e40000 0000000000000000 8664 Microsoft.Windows.KernelAccessLocAmp cw5nh2txyewy
16 00000000 000a0004a10e34 000a0004a10e34 8664 Microsoft.Windows.PeopleExperienceHost cw5nh2txyewy
17 C:\Windows\system2\3ppExtComObj.exe 2023-12-04 02:49:12
18 C:\Windows\system2\ihost.exe 2023-12-04 02:48:11Z
19 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
20 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
21 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
22 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
23 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
24 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
25 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
26 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
27 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
28 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
29 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
30 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
31 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
32 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
33 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
34 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
35 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
36 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
37 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
38 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
39 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
40 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
41 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
42 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
43 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
44 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
45 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
46 C:\Users\KR3L_DF\AppData\Local\Microsoft\OneDrive\21.220.1024.0005\FileSyncConfig.exe 2025-10-10 17:00:33
47 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
48 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
49 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
50 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
51 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
52 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
53 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
54 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z
55 C:\Windows\system2\WindowsPowerShellV1\powershell.exe 2023-12-04 02:48:11Z

```

Find what: **Last Login**

Find Next Count Find All in Current Document Close Transparency On losing focus Always

Normal text file length: 369,762 lines: 7,198 Ln: 11 Col: 53 Sel: 14 | 1 Windows (CR LF) UTF-8 INS

Using Ctrl+F in Notepad++ to search "Last Login" across all registry files to identify active accounts during attack timeframe

C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SAM.txt - Notepad++

```

62 Pwd Reset Date : Sat Oct 11 02:47:26 2025 z
63 Pwd Fail Date : Never
64 Login Count : 0
65 --> Account Disabled
66 --> Normal user account
67
68 Username : KR3L_DF [1001]
69 SID : S-1-5-21-1767728838-24418769-4131373679-1001
70 Full Name :
71 User Comment :
72 Account Type :
73 Account Created : Fri Oct 10 16:56:42 2025 z
74 Security Questions:
75 Question 1 :
76 Question 2 :
77 Answer 2 :
78 Question 3 :
79 Answer 3 :
80 Name :
81 Last Login Date : Fri Oct 10 17:03:01 2025 z
82 Pwd Reset Date : Never
83 Pwd Fail Date : Never
84 Login Count : 0
85 --> Password does not expire
86 --> Password not required
87

```

Search results - (6 hits)

- Search "Last Login" (6 hits in 1 file of 7 searched) [Normal]
 - C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SAM.txt (6 hits)
 - Line 14: Last Login Date : Never
 - Line 29: Last Login Date : Never
 - Line 45: Last Login Date : Never
 - Line 61: Last Login Date : Never
 - Line 82: **Last Login Date : Fri Oct 10 17:03:01 2025 Z**
 - Line 97: Last Login Date : Never
- Search "last login" (6 hits in 1 file of 7 searched) [Normal]

Normal text file length: 7,728 lines: 217 Ln: 82 Col: 11 Sel: 10 | 1 Windows (CR LF) UTF-8 INS

Discovery of active user account "KR3L_DF" during the attack timeframe through registry analysis

The screenshot shows the Registry Explorer interface. On the left, the tree view displays the SAM hive under C:\Users\WIZARD\Desktop\Cases\1\Windows\System32\config. A red box highlights the 'Users' folder within the SAM hive. On the right, the 'User accounts' table is shown, also with a red box highlighting the 'Last Login Time' column. The table has one row for the account 'KR3I_DF', which was last logged in at 2025-10-10 17:03:01.

User	Total Login Count	Create	Last Login Time	User Name
KR3I_DF	0	2025-10-10 17:03:01		KR3I_DF

Identifying active user account "KR3I_DF" in Registry Explorer during the attack timeframe

So the answer is : KR3I_DF

2. Which account(s) were created?

As we investigated, we discovered a new suspicious account was created just 30 minutes after KR3I_DF logged in, indicating potential attacker persistence activity.

Maybe this was the attacker creating a backdoor account to maintain access to the system

```

80 Answer 3 :
81 Name : art-test
82 Last Login Date : Fri Oct 10 17:03:01 2025 Z
83 Pwd Reset Date : Never
84 Pwd Fail Date : Never
85 Login Count : 0
86 >-- Password does not expire
87 --> Password not required
88 --> Normal user account
89
90 Username : art-test [1002]
91 SID : S-1-5-21-1767728838-24418769-4131373679-1002
92 Full Name :
93 User Comment :
94 Account Type :
95 Account Created : Fri Oct 10 17:03:01 2025 Z
96 Home :
97 Last Login Date : Never
98 Pwd Reset Date : Fri Oct 10 17:37:12 2025 Z
99 Pwd Fail Date : Never
100 Login Count : 0
101 >-- Normal user account
102
103 Group Membership Information
104 -----
105

```

Search results - (6 hits)

- C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SAM.txt (6 hits)
 - Last Login Date : Never
 - Last Login Date : Fri Oct 10 17:03:01 2025 Z**
 - Last Login Date : Never
- Search "last login" (6 hits in 1 file of 7 searched) [Normal]

Newly created suspicious account "**art-test**" detected in registry just 30 minutes after **KR3I_DF** login

Val	Us	In	Total Login Count	Created On	Last Login Time	User Name
10	0	0	0	2025-10-10 16:56:42	2025-10-10 17:03:01	KR3I_DF
10	0	0	0	2025-10-10 16:56:42	2025-10-10 17:37:12	art-test

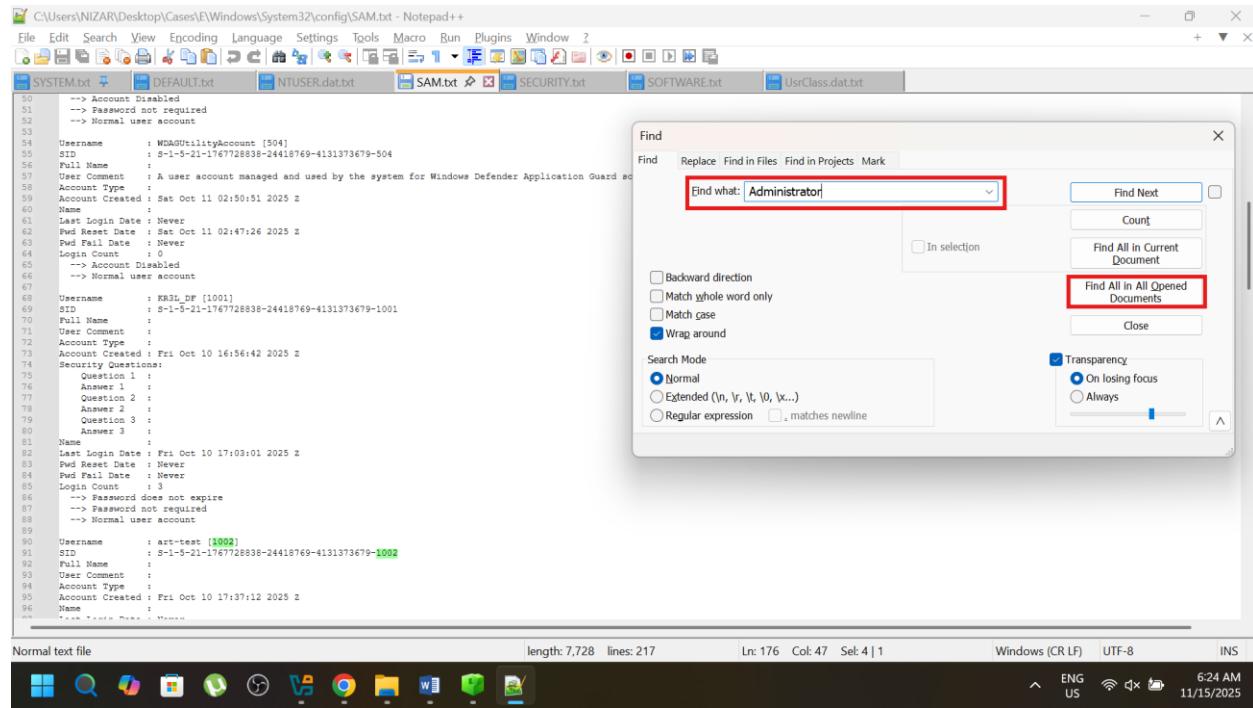
Recently created "**art-test**" account shown in Registry Explorer, timestamped 30 minutes after initial **KR3I_DF** compromise

So the answer is : art-test

3. Which accounts are Administrator group members?

By examining the **Administrators group** membership in the registry, we can identify all user SIDs with administrative privileges on the system

SIDs (Security Identifiers) are unique numbers that identify users, groups, and computers in Windows - like digital fingerprints for each account.



Searching for "Administrators" group in Notepad++ to identify all member SIDs with administrative privileges

C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SAM.txt - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window 2

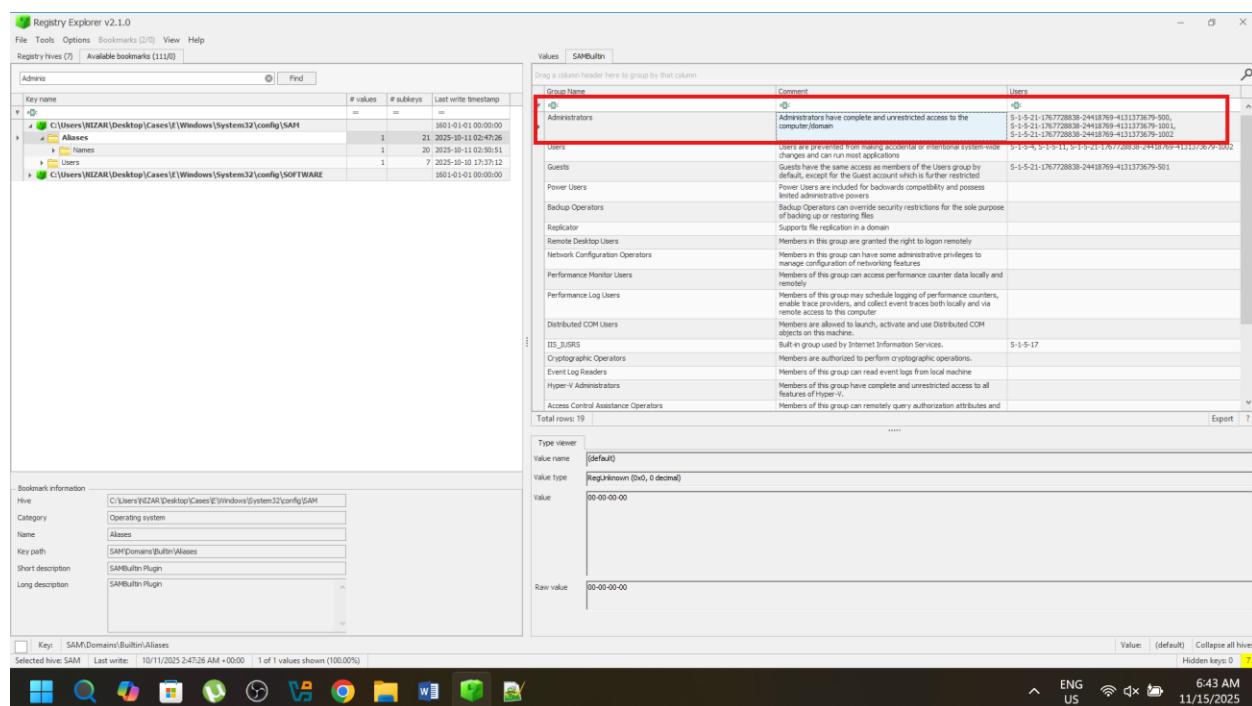
SYSTEM.txt DEFAULT.txt NTUSER.dat.txt SAM.txt SECURITY.txt SOFTWARE.txt UsrClass.dat.txt

161 Group Name : System Managed Accounts [1]
162 LastWrite : Sat Oct 11 22:47:16 2025 Z
163 Group Comment : Members of this group are managed by the system.
164 Users :
165 S-1-5-21-1767728838-24418769-4131373469-503
166
167 Group Name : Distributed COM Users [0]
168 LastWrite : Sat Oct 11 02:47:12 2025 Z
169 Group Comment : Members are allowed to launch, activate and use Distributed COM objects on this machine.
170 Users : None
171
172 Group Name : Administrators [3]
173 LastWrite : Fri Oct 10 17:37:12 2025 Z
174 Group Comment : Administrators have complete and unrestricted access to the computer/domain
175 Users :
176 S-1-5-21-1767728838-24418769-4131373469-1002
177 S-1-5-21-1767728838-24418769-4131373469-500
178 S-1-5-21-1767728838-24418769-4131373469-1001
179
180 Group Name : Power Users [0]
181 LastWrite : Sat Oct 11 02:47:26 2025 Z
182 Group Comment : Power Users are included for backwards compatibility and possess limited administrative powers
183 Users : None
184

Search results - (7 hits)

Search "Administrator" (7 hits in 2 files of 7 searched) [Normal]
C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SAM.txt (4 hits)
Line 1: Username : Administrator [500]
Line 13: Group Name : Hyper-V Administrators [0]
Line 17: Group Name : Administrators [3]
Line 174: Group Comment : Administrators have complete and unrestricted access to the computer/domain
C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SOFTWARE.txt (3 hits)
Line 41639: User Account Control: Run all administrators in Admin Approval Mode
Line 41649: FilterAdministratorToken value not found.
Line 41659: User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode
Search "Administrator" (7 hits in 2 files of 7 searched) [Normal]
Search "Administrator" (7 hits in 2 files of 7 searched) [Normal]
Search "Last Login" (6 hits in 1 file of 7 searched) [Normal]
Search "last login" (6 hits in 1 file of 7 searched) [Normal]

Viewing all SID members within the Administrators group, showing accounts with administrative privileges



Administrators group membership viewed in Registry Explorer, displaying all member SIDs with administrative rights

So the answer is :

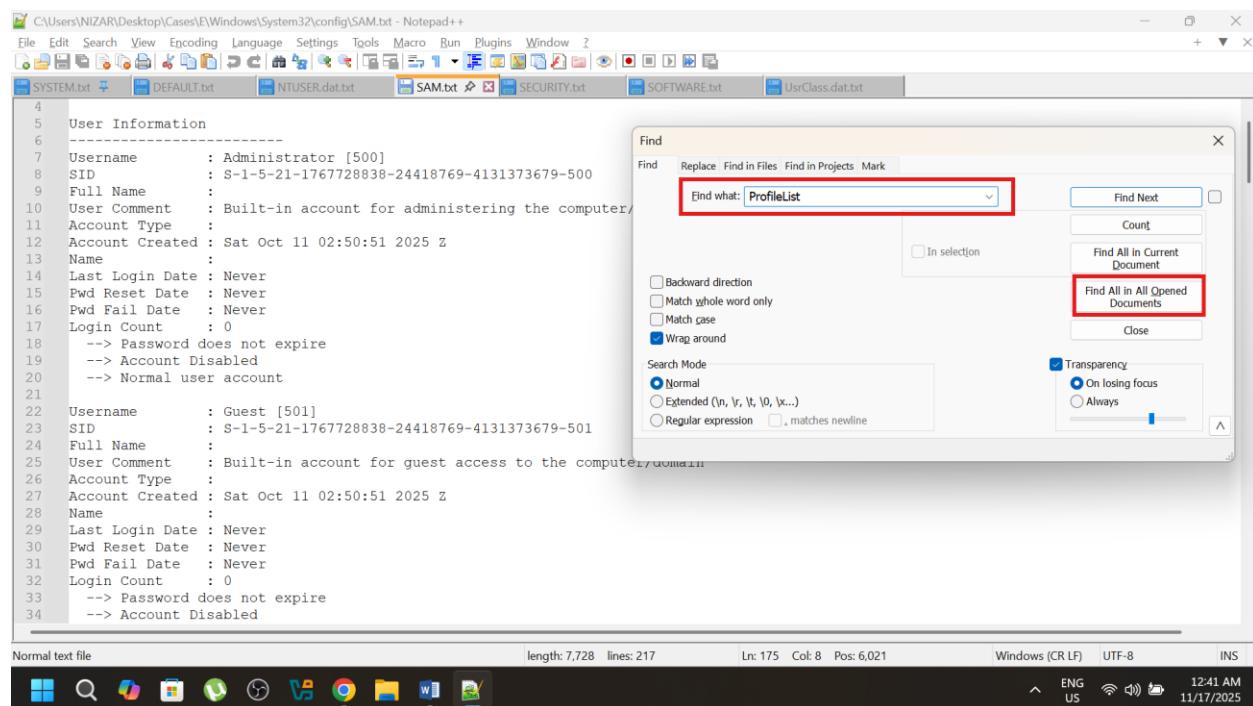
S-1-5-21-1767728838-24418769-4131373679-1002

S-1-5-21-1767728838-24418769-4131373679-500

S-1-5-21-1767728838-24418769-4131373679-1001

4. Which users have profiles?

A **user profile** is a collection of settings and data that Windows maintains for each user account - it contains their **desktop, documents, browser history, and personal configurations**.



Using Notepad++ to search for "ProfileList" across registry files to identify all user profiles on the system

```

C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SOFTWARE.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
SYSTEM.txt DEFAULT.txt NTUSER.dat.txt SAM.txt SECURITY.txt SOFTWARE.txt UsrClass.dat.txt
40407 -----
40408 profilelist v.20200518
40409 (Software) Get content of ProfileList key
40410 Microsoft\Windows NT\CurrentVersion\ProfileList
40411 Path : %systemroot%\system32\config\systemprofile
40412 SID : S-1-5-18
40413 LastWrite : 2019-12-07 09:17:27Z
40414
40415 Path : %systemroot%\ServiceProfiles\LocalService
40416 SID : S-1-5-19
40417 LastWrite : 2019-12-07 09:17:27Z
40418
40419 Path : %systemroot%\ServiceProfiles\NetworkService
40420 SID : S-1-5-20
40421 LastWrite : 2019-12-07 09:17:27Z
40422
40423 Path : C:\Users\KR3L_DF
40424 SID : S-1-5-21-1767728838-24418769-4131373679-1001
40425 LastWrite : 2025-10-10 17:03:01Z
40426
40427 Domain Accounts
40428
40429
40430 -----

```

Search results - (4 hits)

```

Search "ProfileList" (4 hits in 2 files of 7 searched) [Normal]
C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SAM.txt (1 hit)
Line 214: - Correlate the user SIDs to the output of the ProfileList plugin
C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SOFTWARE.txt (3 hits)
Line 40408: Profilelist v.20200518
Line 40409: (Software) Get content of ProfileList key
Line 40411: Microsoft\Windows NT\CurrentVersion\ProfileList

```

Normal text file length: 2,546,635 lines: 41,835 Ln: 40,408 Col: 12 Sel: 11 | 1 Windows (CR LF) UTF-8 INS

KR3L_DF user profile identified in ProfileList with corresponding SID and profile path

Timestamp	Key Name	Profile Image Path	Last Logon Time	Last Logoff Time
2019-12-07 09:17:27	S-1-5-18	%systemroot%\system32\config\systemprofile		
2019-12-07 09:17:27	S-1-5-19	%systemroot%\ServiceProfiles\LocalService		
2019-12-07 09:17:27	S-1-5-20	%systemroot%\ServiceProfiles\NetworkService		
2025-10-10 17:03:01	S-1-5-21-1767728838-24418769-4131373679-1001	C:\Users\KR3L_DF	2025-10-10 17:03:01	2025-10-10 17:02:39

Bookmark information

- Hive: C:\Users\NIZAR\Desktop\Cases\E\Windows\System32\config\SOFTWARE
- Category: User general
- Name: ProfileList
- Key path: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\ProfileList
- Show description: Show profile list including SIDs
- Long description:

Key: Microsoft\Windows NT\CurrentVersion\ProfileList
Selected hive: NTUSER.DAT | Last write: 10/10/2025 5:03:44 PM +00:00 | 4 of 4 values shown (100.00%)

KR3L_DF user profile visible in Registry Explorer under ProfileList with complete profile path details

So the answer is : **KR3L_DF**