

Forensic Analysis of Unauthorized Access on Windows VM

Introduction

This project involves investigating potential unauthorized access to a Windows VM using Windows Registry forensics. With tools like RegistryExplorer and other programs from Eric Zimmerman's toolkit, we dig into registry artifacts to see if someone else has been on this machine, find any extra user accounts, and uncover details about network drives and USB devices that might've been connected.

Objectives

The goal here is to confirm (or disprove) the theory that someone unauthorized accessed this system. Specifically, we'll:

- See if there are multiple user accounts (there should only be one).
- Figure out when specific files, like Changelog.txt, were last accessed.
- Locate the path of a Python script that was run to keep access open.
- Identify details about USB devices that were plugged in.

Lab Environment

This lab runs on a Windows VM provided through TryHackMe. Once logged in, the Desktop has two main folders: **triage** (with the KAPE-collected registry data) and **EZtools** (where Eric Zimmerman's tools, like RegistryExplorer and AppCompatCacheParser, are ready to go).

Tools and Technologies Used

Here's what we used to dig into the registry:

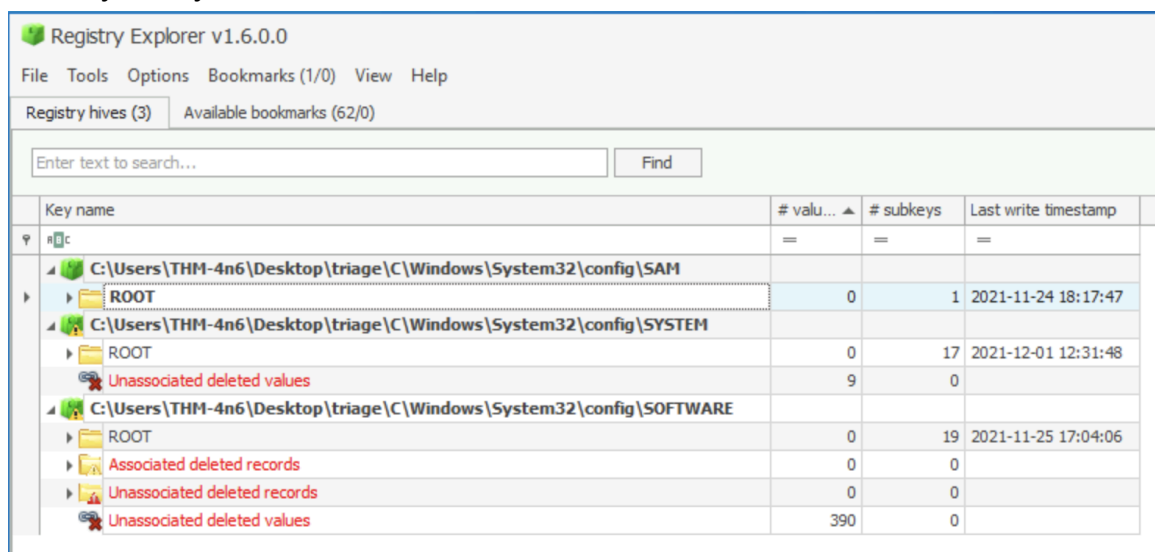
- **RegistryExplorer**: For navigating the registry and working with hives.
- **EZViewer**: Helpful for viewing logs and files.
- **AppCompatCacheParser**: Used to parse application compatibility cache data.

- **KAPE:** Already collected the registry data, which we'll analyze. These tools let us dig through Windows registry hives like SAM, SYSTEM, and SOFTWARE to track down our artifacts and see what was happening on this machine.

Step-by-Step Process

Step one

To initiate the investigation, I loaded critical registry hives located in the Windows System32 configuration folder. These hives contained vital artifacts for analyzing potential unauthorized access to the device. Using **RegistryExplorer**—a specialized tool in Eric Zimmerman's suite for navigating and analyzing the contents of the Windows registry—I accessed the SAM, SYSTEM, and SOFTWARE hives. As these files were flagged as "dirty," RegistryExplorer prompted me to load corresponding transaction logs (.LOG1 and .LOG2) to reconstruct a 'clean' version of each hive. This process ensured that my analysis was based on consistent and error-free data.



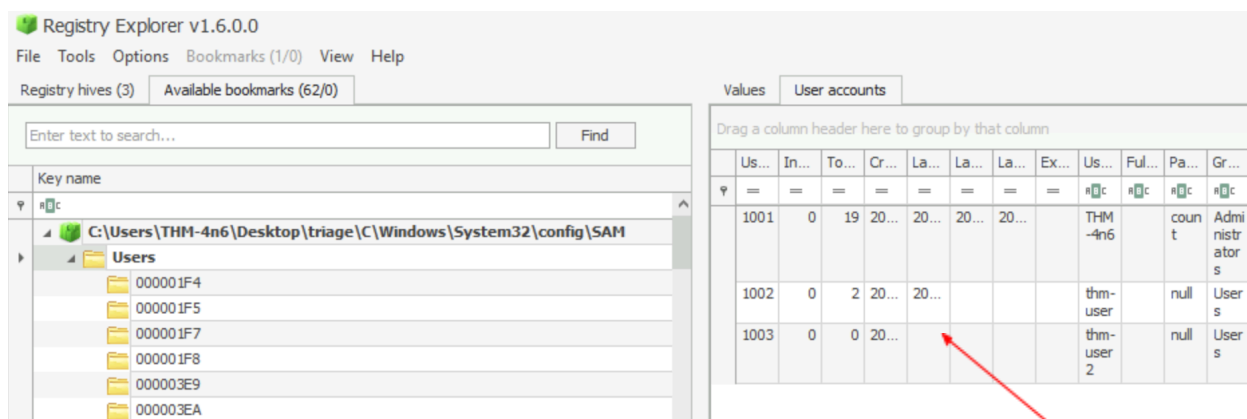
The screenshot shows the Registry Explorer v1.6.0.0 interface. The left pane displays the registry tree with the following structure:

- C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SAM
 - ROOT
- C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SYSTEM
 - ROOT
 - Unassociated deleted values
- C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SOFTWARE
 - ROOT
 - Associated deleted records
 - Unassociated deleted records
 - Unassociated deleted values

The right pane displays a table with the following columns: Key name, # valu..., # subkeys, and Last write timestamp.

Key name	# valu...	# subkeys	Last write timestamp
C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SAM\ROOT	0	1	2021-11-24 18:17:47
C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SYSTEM\ROOT	0	17	2021-12-01 12:31:48
C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SYSTEM\Unassociated deleted values	9	0	
C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SOFTWARE\ROOT	0	19	2021-11-25 17:04:06
C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SOFTWARE\Associated deleted records	0	0	
C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SOFTWARE\Unassociated deleted records	0	0	
C:\Users\THM-4n6\Desktop\trriage\C\Windows\System32\config\SOFTWARE\Unassociated deleted values	390	0	

The initial investigation focused on identifying the user accounts present on the system, as the organization suspected multiple unauthorized accounts. The **SAM (Security Accounts Manager)** hive was key to this analysis. Navigating through the SAM file, I reviewed subkeys related to user profiles, which recorded critical information about account creation dates, last logon timestamps, and associated security identifiers (SIDs). Through this, I identified three user accounts on the system—all named with a "THM" prefix.



Notably, the “THM-User2” account showed no recorded last logon date, indicating that it was created but never actively used. This account appears suspicious, as it could potentially serve as a backdoor entry.

Step Two

The next objective was to track recent access to the Changelog.txt file, which was flagged as related to malicious activity. For this, I used **RegistryExplorer** to search for the **RecentDocs** registry key, which logs metadata for recently accessed documents.



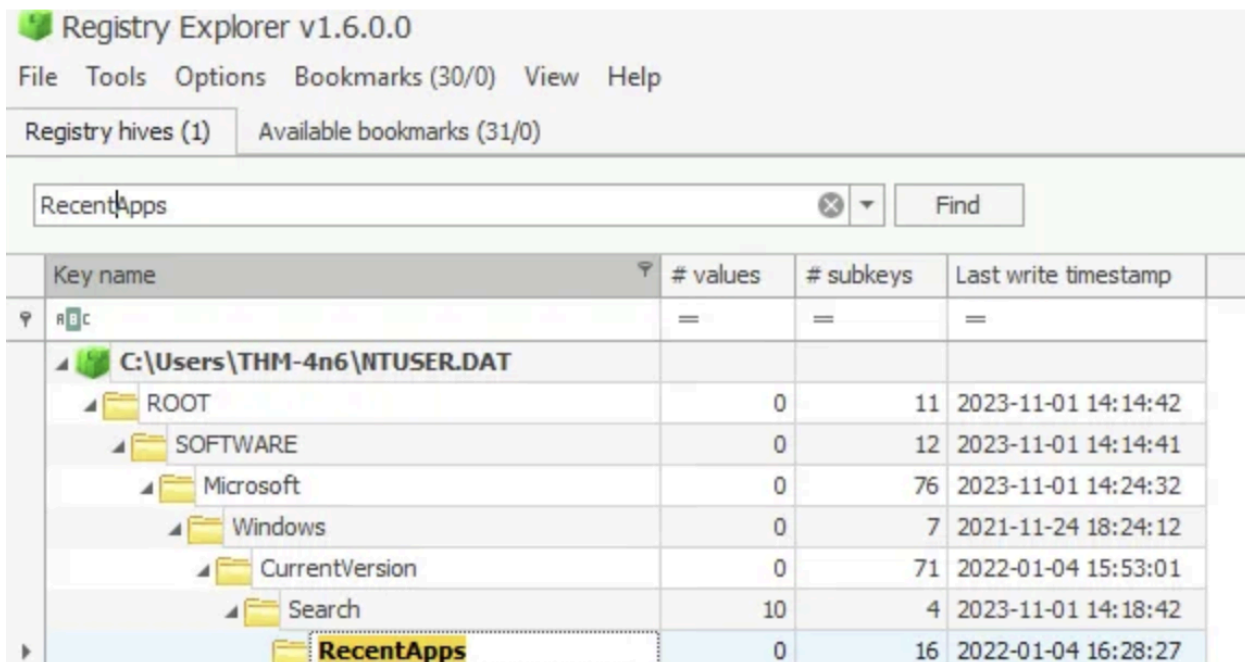
Drag a column header here to group by that column						
Extension	Value Name	Target Name	Lnk Name	Mrn Position	Opened On	Extension Last Opened
RecentDocs	7	EZtools	EZtools.lnk	0	2021-12-01 13:00:34	
RecentDocs	6	Settings	Settings.lnk	1		2021-11-30 10:56:23
RecentDocs	5	WallpaperSettings.xml	WallpaperSettings.lnk	2		2021-11-30 10:56:21
RecentDocs	4	System and Security	System and Security.lnk	3		
RecentDocs	3	::(BB06C0E4-D293-4F75-8A90-CB05B6477EEE)	System.lnk	4		
RecentDocs	1	KAPE	KAPE.lnk	5		
RecentDocs	0	Get-KAPEUpdate.ps1	Get-KAPEUpdate.lnk	6		2021-11-24 18:18:48
RecentDocs	2	Changelog.txt	Changelog.lnk	7		2021-11-24 18:18:48

Within this key, I found an entry for Changelog.txt, revealing that the file was last accessed on **November 24, 2021**. This timestamp provided crucial

information in building a timeline of the attack, confirming that the file was opened shortly before the organization detected signs of unauthorized activity.

Step Three

It was suspected that a Python script had been executed on the system to maintain persistence for unauthorized access. My task was to find the full path of this script. I conducted a search within the **RecentApps** registry key for any recent application launches, looking for logs that would likely contain execution records.



The screenshot shows the Registry Explorer v1.6.0.0 interface. The 'RecentApps' registry key is selected in the left pane. The right pane displays a table of registry keys and their values.

Key name	# values	# subkeys	Last write timestamp
HKEY_CURRENT_USER	=	=	=
C:\Users\THM-4n6\NTUSER.DAT			
ROOT	0	11	2023-11-01 14:14:42
SOFTWARE	0	12	2023-11-01 14:14:41
Microsoft	0	76	2023-11-01 14:24:32
Windows	0	7	2021-11-24 18:24:12
CurrentVersion	0	71	2022-01-04 15:53:01
Search	10	4	2023-11-01 14:18:42
RecentApps	0	16	2022-01-04 16:28:27

Key Name	App Id	App Path	Last Accessed
{273DE518-2B44-4F53-A7E0-4AE1A5C CAF79}	{1AC14E77-02E7-4E5D-B744-2EB1AE5198 B7}\WindowsPowerShell\v1.0\powershell.exe	C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe	2021-11-25 03:37:24
{2CD6F947-E9F7-4593-A6C2-139DF15 D3CFF}	Microsoft.Windows.RemoteDesktop	C:\Windows\system32\mstsc.exe	2021-11-25 03:59:55
{48ED5797-44C9-4AC8-B676-A15C6C3 E7026}	D:\setup64.exe		2021-11-25 03:21:31
{56C40D12-B31F-4EEA-8538-3865D28 B0517}	Z:\setups\python-3.8.2.exe		2021-11-25 03:32:00
{6E5F6115-4C59-4A36-9BEC-F6FD501 E3A7B}	C:\Users\THM-4n6\Desktop\KAL-APE\gkape.exe		2021-11-25 03:44:02
{75D4CE35-5CA0-4B3B-B042-652A6B1 ACAB1}	C:\Users\THM-4n6\Downloads\Windows Upgrade9252.exe		2021-11-24 15:24:43
{7EEDBC8F-5583-4A9D-901B-C45DA1C BD930}	Z:\setups\7z1900-x64.exe		2021-11-25 03:26:39
{9A9AB565-D561-4293-8FC6-FA72866 F3BFA}	C:\Users\THM-4n6\Downloads\MicrosoftEdgeSetup.exe		2021-11-25 04:12:45
{AF13CF4C-7867-449D-A074-AF1C605 3D339}	Microsoft.Windows.Explorer	::{52205FD8-5DFB-447D-801A-D0B52F2E 83E1}	2021-11-24 15:24:35
{C2CDA021-A597-4B51-80E8-71ED7A2}	OperaSoftware.OperaWebBrowser.16378	C:\Users\THM-4n6\AppData\Local\Progra	2021-11-25 04:10:02

This search identified a Python script located at **Z:\setups**. The file path allowed us to pinpoint the exact location of the script on the system and provided actionable information for removing the script and mitigating unauthorized persistence.

Step Four

The organization's notes indicated that the malicious script was introduced via a USB drive labeled "USB." The final objective was to determine the last connection date of this USB device to the system.

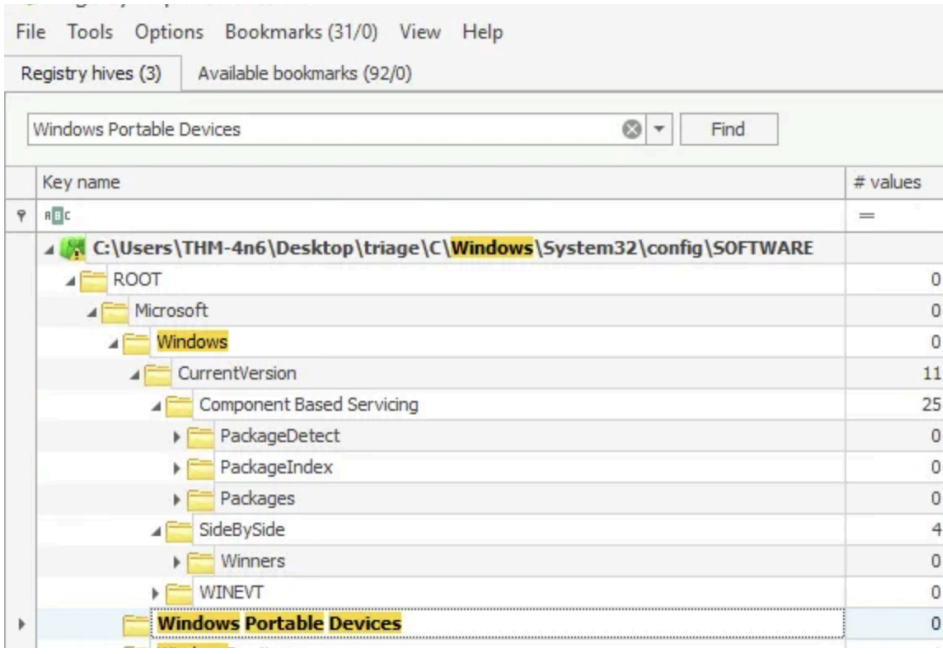
Device identification:

The following locations keep track of USB keys plugged into a system. These locations store the vendor id, product id, and version of the USB device plugged in and can be used to identify unique devices. These locations also store the time the devices were plugged into the system.

SYSTEM\CurrentControlSet\Enum\USBSTOR

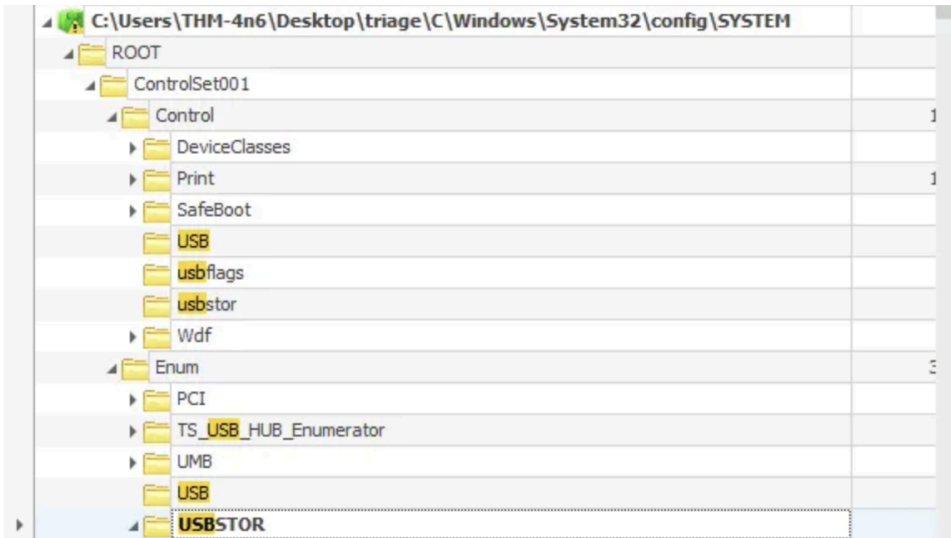
SYSTEM\CurrentControlSet\Enum\USB

First, I navigated to the **Windows Portable Devices** registry section, which lists the history of connected devices and provides identifiers such as GUIDs (Globally Unique Identifiers) and friendly names.



Timestamp	Device	Serial Number	Guid	Friendly Name
=	*c:	*c:	{E251921F-4DA2-11EC-A783-001A7DDA7110}	USB
2021-11-25 07:16:54			{F529A9D6-4D9E-11EC-A782-001A7DDA7110}	New V

I identified the USB drive under its friendly name, “USB,” confirming its presence on the system. To determine the last connection timestamp, I then accessed the **USBSTOR** registry key, which records connection details for external storage devices.



Timestamp	Manufacturer	Title	Version	Disk Id	Serial Number	Device Name	Installed	First Installed	Last Connected
=	*c:	*c:	*c:	*c:	*c:	*c:	=	=	=
2021-11-24 18:25:15	Ven_Kingston	Prod_DataTraveler_2.0	Rev_PMAP	{e251921f-4da2-11ec-a783-001a7dda7110}	1C6F654E59A3B0C179D366AE80	Kingston DataTraveler 2.0 USB Device	2021-11-24 18:25:15	2021-11-24 18:25:15	2021-11-24 18:40:06
2021-11-24 18:27:02	Ven_USB3.0	Prod_External_Device	Rev_SDM1	{f529a9d6-4d9e-11ec-a782-001a7dda7110}	0123456789ABCDE80	USB3.0 External Device USB Device	2021-11-24 18:27:02	2021-11-24 18:27:02	2021-11-24 18:27:02

This part took me a bit to understand, because the log information only provided me with a GUID and a friendly name, of which is not shown in the USBSTOR logs. Upon a close look, the GUID and Disk ID matched, and research showed me that a Disk ID for a GPT disk is the GUID, or Globally Unique Identifier. So as shown on the right hand side of the screen shot, the last time the USB device with the nickname "USB" was in 2021, November 24th.

Results and Analysis

Through this process, I was able to confirm the existence of an unused user account (THM-User2) and that the Changelog.txt file and the USB drive were both last accessed on the same date, which was a likely timeframe for the unauthorized access. The identified path for the Python script (Z:\setups) gave us a clear target to remove any persistence mechanisms left by the intruder.

Challenges Faced

One tricky part of this investigation was piecing together GUIDs and device IDs to match with the right USB connection logs. The naming conventions in the logs aren't always intuitive, so it took some time to confirm the USB drive's last connection date. Additionally, loading the registry hives with the right transaction logs was a bit finicky at first, as I had to understand how RegistryExplorer integrates .LOG1 and .LOG2 files.

Conclusion

This project demonstrated how registry artifacts can reveal a lot about potential unauthorized access. By analyzing the SAM hive, I confirmed extra accounts, and with RecentDocs and USBSTOR, I could pinpoint when specific files and devices were accessed. With tools like RegistryExplorer, I could dig deeper and reconstruct what might have happened, building a strong case for investigating potential backdoors and other persistence mechanisms.

References

1. TryHackMe. "THM-4n6 Lab Environment." URL: <https://tryhackme.com/>
2. Zimmerman, Eric. "EZtools for Registry and Forensic Analysis." URL: <https://ericzimmerman.github.io/>
3. Microsoft Documentation. "Windows Registry Reference." URL: <https://docs.microsoft.com/en-us/windows/win32/sysinfo/registry>
4. Forensics Wiki. "Windows Registry Forensics." URL: https://forensicswiki.org/wiki/Windows_Registry