

Jo

I am a forensic investigator and I found a file on a Windows 10 machine. How can I find out if the file has been executed by a particular user?

Si

As a forensic investigator, you can use various methods and tools to determine if a file has been executed by a particular user on a Windows 10 machine. Here are some ways to investigate this:

1. Check Windows Event Logs:
Windows Event Logs store information about application, security, and system events. To check if a file has been executed, you can look for Event ID 4688 in the Security Event Logs. This event is created when a new process is launched, and it contains details about the process and the user who executed it.

To access Windows Event Logs:

- a. Press **Win + R** and type **eventvwr.msc**, then press **Enter**.
- b. In the Event Viewer, navigate to **Windows Logs** > **Security**.
- c. Use the 'Filter Current Log' option to search for Event ID 4688.
- d. Review the events to see if the file in question was executed by the user.

2. Examine Prefetch Files:

Windows uses Prefetch files to store information about executed programs to speed up subsequent launches. You can analyze these files to find traces of the file's execution.

Prefetch files are located in **C:\Windows\Prefetch**. You can use a third-party tool like WinPrefetchView (https://www.nirsoft.net/utils/win_prefetch_view.html) to analyze these files and check if the file in question was executed.

3. Analyze UserAssist Registry Keys:
UserAssist is a Windows feature that tracks the execution of applications for various purposes, such as populating the Start menu's frequently used programs list. The UserAssist data is stored in the Windows Registry.

You can analyze the UserAssist Registry keys by navigating to the following location in the Registry Editor (**regedit.exe**):

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentV
```

To analyze the data, you can use third-party tools like UserAssistView (<https://www.nirsoft.net/utils/userassistview.html>), which can display the execution count, last execution time, and other relevant information.

4. Investigate NTFS file system metadata:
The NTFS file system maintains metadata, such as the last access time, which could provide a clue about whether the file was executed. You can view this metadata using the Windows File Explorer or a third-party tool like FTK Imager (<https://accessdata.com/product-download/ftk-imager-version-4-5-0>).

Keep in mind that this method may not be conclusive, as the last access time can be updated for various reasons other than execution.

Please remember that some of these methods might not provide conclusive evidence, so it's important to corroborate the findings from multiple sources when possible. Additionally, a skilled attacker might attempt to erase or manipulate these traces, so always consider the possibility of tampering when evaluating the evidence.

Jo

Are there any other artefacts that I should consider?

Si

Yes, there are additional artifacts that can provide evidence of file execution or user activity. Here are some other sources of information to consider during your forensic investigation:

1. Windows Task Scheduler:
Check the Task Scheduler to see if there are any tasks configured to execute the file in question. You can access the Task Scheduler by typing **taskschd.msc** in the Run dialog (Win + R).
2. Shortcut Files (LNK):
Windows creates shortcut files (LNK) when a user opens a file or application. These files can contain information about the target file, such as its file path, creation time, and last access time. You can find shortcut files in various locations, like the user's Desktop, Start menu, or Recent Items folder. Analyzing these files may provide evidence of file execution.
3. Shellbags:
Shellbags are a Windows Registry artifact that records folder view settings and the display of folders within Windows Explorer. They can provide information about user activities, such as accessing or creating directories. You can analyze Shellbags using third-party tools like ShellBags Explorer (<https://ericzimmerman.github.io/#index.md>).
4. Windows PowerShell History:
If the user executed the file using PowerShell, there might be traces in the PowerShell history. You can check the user's PowerShell console history file, typically located at:

```
shell
%UserProfile%\Documents\WindowsPowerShell\Consoles\Host
```
5. Windows Search Index:
Windows Search Index keeps track of files and their contents for quick searching. You can use third-party tools like Windows Search Index Analyzer (<https://github.com/CyberShadow/WindowsSearchIndexAnalyzer>) to analyze the index and potentially find evidence of the file's execution or access.
6. SRUM (System Resource Usage Monitor) database:
The SRUM database tracks resource usage by applications and users on a Windows system. You can analyze the SRUM database using third-party tools like SRUM-DUMP (<https://github.com/MarkBaggett/srum-dump>) to find evidence of the file's execution.
7. Browser History and Cache:
If the file was downloaded or accessed via a web browser, you might find evidence in the browser history, cache, or cookies. You can use tools like NirSoft's BrowsingHistoryView (https://www.nirsoft.net/utils/browsing_history_view.html) or Magnet AXIOM (<https://www.magnetforensics.com/magnet-axiom>) to analyze browser artifacts.
8. Log files:
Review log files generated by the operating system, applications, or security software, as they might contain evidence of file execution or related activities.

Remember that the availability and reliability of these artifacts depend on various factors, such as the system configuration, user behavior, and potential tampering. It is essential to corroborate your findings from multiple sources and consider the possibility of anti-forensic techniques employed by skilled attackers.

Jo

Are there any other artefacts that I should consider?

Si

Yes, there are even more artifacts that can provide additional information during your forensic investigation. Some of these artifacts include:

1. Amcache:
Amcache is a Windows Registry hive that stores information about executed programs, including file path, last modification time, and SHA-1 hash. You can find the Amcache hive at the following location:

```
makefile
C:\Windows\AppCompat\Programs\Amcache.hve
```
2. Shimcache:
Shimcache (also known as the Application Compatibility Cache) is another source of information about executed applications. Shimcache data is stored in the Windows Registry under the following key:

```
mathematica
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LS
```

To analyze Shimcache, you can use tools like ShimcacheParser (<https://github.com/mandiant/ShimCacheParser>) or Redline from FireEye (<https://www.fireeye.com/services/freeware/redline.html>).

3. User Activity:
Windows 10 introduced the User Activity feature, which tracks user activities and stores them as ActivityCmds. You can find these files at:

```
php
%UserProfile%\AppData\Local\ConnectedDevicesPlatform\
```

Analyzing these files using a tool like WindowsTimelineParser (<https://github.com/EricZimmerman/WindowsTimelineParser>) can provide insights into user activities, including potential file execution.

4. Jump Lists:
Jump Lists are another Windows feature that tracks recently or frequently accessed files and applications. They are stored in the AutomaticDestinations and CustomDestinations folders located at:

```
shell
%AppData%\Microsoft\Windows\Recent\AutomaticDestinations
%AppData%\Microsoft\Windows\Recent\CustomDestinations
```

You can analyze Jump Lists using tools like JLECmd (<https://github.com/EricZimmerman/JLECmd>) to gather evidence of file access or execution.

5. Network Artifacts:
Investigate network artifacts, such as DNS cache, proxy logs, or firewall logs, for connections related to the file in question. These artifacts can provide insights into the file's origin, download, or communication with external servers.
6. Memory Forensics:
Analyzing the system's volatile memory (RAM) using memory forensics tools like Volatility (<https://www.volatilityfoundation.org>) or Rekall (<https://www.rekall-forensic.com>) can reveal evidence of executed processes, network connections, and other relevant activities.
7. External Devices:
Examine the Windows Registry and Event Logs for evidence of external devices, such as USB drives, that may have been used to transfer or execute the file. Some relevant Registry keys include:

```
sql
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\I
```
8. File System Journaling:
File system journaling, such as the NTFS \$UsnJrnl or \$LogFile, can provide information about file and directory on an NTFS volume. Analyzing the MFT can provide information about file creation, modification, and access. You can analyze these artifacts using tools like NTFS Logfile Parser (<https://github.com/jschicht/LogfileParser>) or USN Journal Parser (<https://github.com/jschicht/UsnJrnl2Csv>).

It's essential to consider the context of your investigation and the specific details of the case when deciding which artifacts to analyze. The reliability and availability of these artifacts can be influenced by system configuration, user behavior, and potential tampering. Always corroborate your findings from multiple sources and consider the possibility of anti-

Jo

Are there any other artefacts that I should consider?

Si

Certainly, there are even more artifacts that can be considered during a forensic investigation, depending on the specifics of your case. Some additional artifacts include:

1. Prefetch Folder - Superfetch (SysMain) Database:
The Superfetch (formerly known as SysMain) feature in Windows maintains a database of frequently executed programs to optimize their performance. The database is stored in the following location:

```
arduino
C:\Windows\System32\config\systemprofile\AppData\Loca
```

You can analyze this database using tools like SuperfetchDumper (<https://github.com/EricZimmerman/SuperfetchDumper>).

2. MFT (Master File Table):
The MFT is an essential part of the NTFS file system and contains metadata about every file and directory on an NTFS volume. Analyzing the MFT can provide information about file creation, modification, and access. You can use tools like MFTAnalyzer (<https://github.com/dkovar/analyzeMFT>) or FTK Imager (<https://accessdata.com/product-download/ftk-imager-version-4-5-0>) to analyze the MFT.
3. ESE databases:
Windows uses the Extensible Storage Engine (ESE) to store various types of data, such as the Windows.edb search index and the SRUM database. Analyzing ESE databases using tools like ESEDatabaseView (https://www.nirsoft.net/utils/esedb_database_view.html) or libesedb (<https://github.com/libyal/libesedb>) can reveal valuable information about user activities and file execution.
4. WMI (Windows Management Instrumentation) Logs:
WMI logs can provide information about system events, including software installations and executions. You can use tools like WMIParser (<https://github.com/EricZimmerman/WMIParser>) to analyze WMI logs for evidence of file execution.
5. Scheduled Tasks XML files:
In addition to the Task Scheduler GUI, you can examine the XML files associated with scheduled tasks for evidence of file execution. These files are typically stored in the following location:

```
mathematica
C:\Windows\System32\Tasks\
```
6. COM+ Catalog:
The Component Object Model (COM+) catalog contains information about installed and registered components, including executables. You can use tools like COM+ Explorer (<https://github.com/woanware/ComPlusExplorer>) to analyze the COM+ catalog for evidence of file execution.
7. Windows Management Instrumentation (WMI) Repository:
The WMI Repository stores information about system configuration, including installed software and services. You can analyze the WMI Repository using tools like WMIExplorer (<https://github.com/vinaypamnani/wmie2>) or PowerShell cmdlets like **Get-WmiObject**.
8. Application Compatibility Shim Databases:
Windows uses shim databases (SDB files) to store compatibility fixes for legacy applications. These databases can contain information about executed applications, including their file paths and command-line arguments. You can analyze shim databases using tools like ShimDBExtractor (<https://github.com/EricZimmerman/ShimDBExtractor>).

The context of your investigation and the specific details of the case will determine which artifacts are most relevant. The reliability and availability of these artifacts can be influenced by system configuration, user behavior, and potential tampering. As always, corroborate your findings from multiple sources and consider the possibility of anti-forensic techniques employed by skilled attackers.