Windows Forensics Overview

In this section, we will provide a concise overview of the key Windows artifacts and forensic procedures.

NTFS

NTFS (New Technology File System) is a proprietary file system developed by Microsoft as a part of its Windows NT operating system family. It was introduced with the release of Windows NT 3.1 in 1993, and it has since become the default and most widely used file system in modern Windows operating systems, including Windows XP, Windows 7, Windows 8, Windows 10, and their server counterparts.

NTFS was designed to address several limitations of its predecessor, the FAT (File Allocation Table) file system. It introduced numerous features and enhancements that improved the reliability, performance, security, and storage capabilities of the file system.

Here are some of the key forensic artifacts that digital investigators often analyze when working with NTFS file systems:

- File Metadata: NTFS stores extensive metadata for each file, including creation time, modification time, access time, and attribute information (such as read-only, hidden, or system file attributes). Analyzing these timestamps can help establish timelines and reconstruct user activities.
- MFT Entries: The Master File Table (MFT) is a crucial component of NTFS that stores metadata for all files and directories on a volume. Examining MFT entries provides insights into file names, sizes, timestamps, and data storage locations. When files are deleted, their MFT entries are marked as available, but the data may remain on the disk until overwritten.
- File Slack and Unallocated Space: Unallocated space on an NTFS volume may contain remnants of deleted files or fragments of data. File slack refers to the unused portion of a cluster that may contain data from a previous file. Digital forensic tools can help recover and analyze data from these areas.
- File Signatures: File headers and signatures can be useful in identifying file types even when file extensions have been changed or obscured. This information is critical for reconstructing the types of files present on a system.
- USN Journal: The Update Sequence Number (USN) Journal is a log maintained by NTFS to record changes made to files and directories. Forensic investigators can analyze the USN Journal to track file modifications, deletions, and renames.
- LNK Files: Windows shortcut files (LNK files) contain information about the target file or program, as well as timestamps and metadata. These files can provide insights into recently accessed files or executed programs.
- Prefetch Files: Prefetch files are generated by Windows to improve the startup performance of applications. These files can indicate which programs have been run on the system and when they were last executed.
- Registry Hives: While not directly related to the file system, Windows Registry hives contain important configuration and system information. Malicious activities or unauthorized changes can leave traces in the registry, which forensic investigators analyze to understand system modifications.
- Shellbags: Shellbags are registry entries that store folder view settings, such as window positions and sorting preferences. Analyzing shellbags can reveal user navigation patterns and potentially identify accessed folders.

- `Thumbnail Cache`: Thumbnail caches store miniature previews of images and documents. These caches can reveal files that were recently viewed, even if the original files have been deleted.
- `Recycle Bin`: The Recycle Bin contains files that have been deleted from the file system. Analyzing the Recycle Bin can help recover deleted files and provide insights into user actions.
- `Alternate Data Streams (ADS)`: ADS are additional streams of data associated with files. Malicious actors may use ADS to hide data, and forensic investigators need to examine these streams to ensure a comprehensive analysis.
- `Volume Shadow Copies`: NTFS supports Volume Shadow Copies, which are snapshots of the file system at different points in time. These copies can be valuable for data recovery and analysis of changes made over time.
- `Security Descriptors and ACLs`: Access Control Lists (ACLs) and security descriptors determine file and folder permissions. Analyzing these artifacts helps understand user access rights and potential security breaches.

Windows Event Logs

`Windows Event Logs` are an intrinsic part of the Windows Operating System, storing logs from different components of the system including the system itself, applications running on it, ETW providers, services, and others.

Windows event logging offers comprehensive logging capabilities for application errors, security events, and diagnostic information. As cybersecurity professionals, we leverage these logs extensively for analysis and intrusion detection.

Adversarial tactics from initial compromise using malware or other exploits, to credential accessing, privilege elevation and lateral movement using Windows operating system's internal tools are often captured via Windows event logs.

By viewing the available Windows event logs, investigators can get a good sense of what is being logged and even search for specific log entries. To access the logs directly for offline analysis, investigators should navigate to the default file path for log storage at `C:\Windows\System32\winevt\Logs`.

The analysis of Windows Event Logs has been addressed in the modules titled `Windows Event Logs & Finding Evil` and `YARA & Sigma for SOC Analysts`.

Execution Artifacts

`Windows execution artifacts` refer to the traces and evidence left behind on a Windows operating system when programs and processes are executed. These artifacts provide valuable insights into the execution of applications, scripts, and other software components, which can be crucial in digital forensics investigations, incident response, and cybersecurity analysis. By examining execution artifacts, investigators can reconstruct timelines, identify malicious activities, and establish patterns of behavior. Here are some common types of Windows execution artifacts:

- `Prefetch Files`: Windows maintains a prefetch folder that contains metadata about the execution of various applications. Prefetch files record information such as file paths, execution counts, and timestamps of when applications were run. Analyzing prefetch files can reveal a history of executed programs and the order in which they were run.
- `Shimcache`: Shimcache is a Windows mechanism that logs information about program execution to assist with compatibility and performance optimizations. It records details such as file paths, execution timestamps, and flags indicating whether a program was executed. Shimcache can help investigators identify recently executed programs and their associated files.
- `Amcache`: Amcache is a database introduced in Windows 8 that stores information about installed applications and executables. It includes details like file paths, sizes, digital signatures, and timestamps of when applications were last executed. Analyzing the Amcache can provide insights into program execution history and identify potentially suspicious or unauthorized software.

- `UserAssist`: UserAssist is a registry key that maintains information about programs executed by users. It records details such as application names, execution counts, and timestamps. Analyzing UserAssist artifacts can reveal a history of executed applications and user activity.
- `RunMRU Lists`: The RunMRU (Most Recently Used) lists in the Windows Registry store information about recently executed programs from various locations, such as the `Run` and `RunOnce` keys. These lists can indicate which programs were run, when they were executed, and potentially reveal user activity.
- `Jump Lists`: Jump Lists store information about recently accessed files, folders, and tasks associated with specific applications. They can provide insights into user activities and recently used files.
- `Shortcut (LNK) Files`: Shortcut files can contain information about the target executable, file paths, timestamps, and user interactions. Analyzing LNK files can reveal details about executed programs and the context in which they were run.
- `Recent Items`: The Recent Items folder maintains a list of recently opened files. It can provide information about recently accessed documents and user activity.
- `Windows Event Logs`: Various Windows event logs, such as the Security, Application, and System logs, record events related to program execution, including process creation and termination, application crashes, and more.

Artifact	Location/Registry Key	Data Stored
Prefetch Files	C:\Windows\Prefetch	Metadata about executed applications (file paths, timestamps, execution count)
Shimcache	Registry: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\AppCompatCache	Program execution details (file paths, timestamps, flags)
Amcache	C:\Windows\AppCompat\Programs\Amcache.hve (Binary Registry Hive)	Application details (file paths, sizes, digital signatures, timestamps)
UserAssist	Registry: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist	Executed program details (application names, execution counts, timestamps)
RunMRU Lists	Registry: HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU	Recently executed programs and their command lines
Jump Lists	User-specific folders (e.g., %AppData%\Microsoft\Windows\Recent)	Recently accessed files, folders, and tasks associated with applications
Shortcut (LNK) Files	Various locations (e.g., Desktop, Start Menu)	Target executable, file paths, timestamps, user interactions
Recent Items	User-specific folders (e.g., %AppData%\Microsoft\Windows\Recent)	Recently accessed files
Windows Event Logs	C:\Windows\System32\winevt\Logs	Various event logs containing process creation, termination, and other events

Windows Persistence Artifacts

Windows persistence refers to the techniques and mechanisms used by attackers to ensure their unauthorized presence and control over a compromised system, allowing them to maintain access and control even after initial intrusion. These persistence methods exploit various system components, such as registry keys, startup processes, scheduled tasks, and services, enabling malicious actors to withstand reboots and security measures while continuing

to carry out their objectives undetected.

Registry

The Windows `Registry` acts as a crucial database, storing critical system settings for the Windows OS. This encompasses configurations for devices, security, services, and even the storage of user account security configurations in the Security Accounts Manager (`SAM`). Given its significance, it's no surprise that adversaries often target the Windows Registry for establishing persistence. Therefore, it's essential to routinely inspect Registry autorun keys.

Example of `Autorun` keys used for persistence:

- `Run/RunOnce Keys`
  - `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run`
  - `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce`
  - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run`
  - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce`
  - `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\`
- `Keys used by WinLogon Process`
  - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon`
  - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell`
- `Startup Keys`
  - `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders`
  - `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders`
  - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders`
  - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\User`

Schtasks

Windows provides a feature allowing programs to schedule specific tasks. These tasks reside in `C:\Windows\System32\Tasks`, with each one saved as an XML file. This file details the creator, the task's timing or trigger, and the path to the command or program set to run. To scrutinize scheduled tasks, we should navigate to `C:\Windows\System32\Tasks` and examine the XML files' content.

Services

`Services` in Windows are pivotal for maintaining processes on a system, enabling software components to operate in the background without user intervention. Malicious actors often tamper with or craft rogue services to ensure persistence and retain unauthorized access. The registry location to keep an eye on is: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services`.

Web Browser Forensics

Diving into web browser forensics, it's a discipline centered on analyzing remnants left by web browsers. These remnants can shed light on user actions, online engagements, and potentially harmful behaviors. Some of the pivotal browser forensic artifacts include:

- `Browsing History`: Records of websites visited, including URLs, titles, timestamps, and visit frequency.
- `Cookies`: Small data files stored by websites on a user's device, containing information such as session details, preferences, and authentication tokens.
- `Cache`: Cached copies of web pages, images, and other content visited by the user. Can reveal websites accessed even if the history is cleared.
- `Bookmarks/Favorites`: Saved links to frequently visited websites or pages of interest.
- `Download History`: Records of downloaded files, including source URLs, filenames, and timestamps.

timestamps.

- Autofill Data: Information automatically entered into forms, such as names, addresses, and passwords.
- Search History: Queries entered into search engines, along with search terms and timestamps.
- Session Data: Information about active browsing sessions, tabs, and windows.
- Typed URLs: URLs entered directly into the address bar.
- Form Data: Information entered into web forms, such as login credentials and search queries.
- Passwords: Saved or autofilled passwords for websites.
- Web Storage: Local storage data used by websites for various purposes.
- Favicons: Small icons associated with websites, which can reveal visited sites.
- Tab Recovery Data: Information about open tabs and sessions that can be restored after a browser crash.
- Extensions and Add-ons: Installed browser extensions and their configurations.

SRUM

Switching gears to SRUM (System Resource Usage Monitor), it's a feature introduced in Windows 8 and subsequent versions. SRUM meticulously tracks resource utilization and application usage patterns. The data is housed in a database file named sru.db found in the C:\Windows\System32\sru directory. This SQLite formatted database allows for structured data storage and efficient data retrieval. SRUM's records, organized by time intervals, can help reconstruct application and resource usage over specific durations.

Key facets of SRUM forensics encompass:

- Application Profiling: SRUM can provide a comprehensive view of the applications and processes that have been executed on a Windows system. It records details such as executable names, file paths, timestamps, and resource usage metrics. This information is crucial for understanding the software landscape on a system, identifying potentially malicious or unauthorized applications, and reconstructing user activities.
- Resource Consumption: SRUM captures data on CPU time, network usage, and memory consumption for each application and process. This data is invaluable for investigating resource-intensive activities, identifying unusual patterns of resource consumption, and detecting potential performance issues caused by specific applications.
- Timeline Reconstruction: By analyzing SRUM data, digital forensics experts can create timelines of application and process execution, resource usage, and system activities. This timeline reconstruction is instrumental in understanding the sequence of events, identifying suspicious behaviors, and establishing a clear picture of user interactions and actions.
- User and System Context: SRUM data includes user identifiers, which helps in attributing activities to specific users. This can aid in user behavior analysis and determining whether certain actions were performed by legitimate users or potential threat actors.
- Malware Analysis and Detection: SRUM data can be used to identify unusual or unauthorized applications that may be indicative of malware or malicious activities. Sudden spikes in resource usage, abnormal application patterns, or unauthorized software installations can all be detected through SRUM analysis.
- Incident Response: During incident response, SRUM can provide rapid insights into recent application and process activities, enabling analysts to quickly identify potential threats and respond effectively.