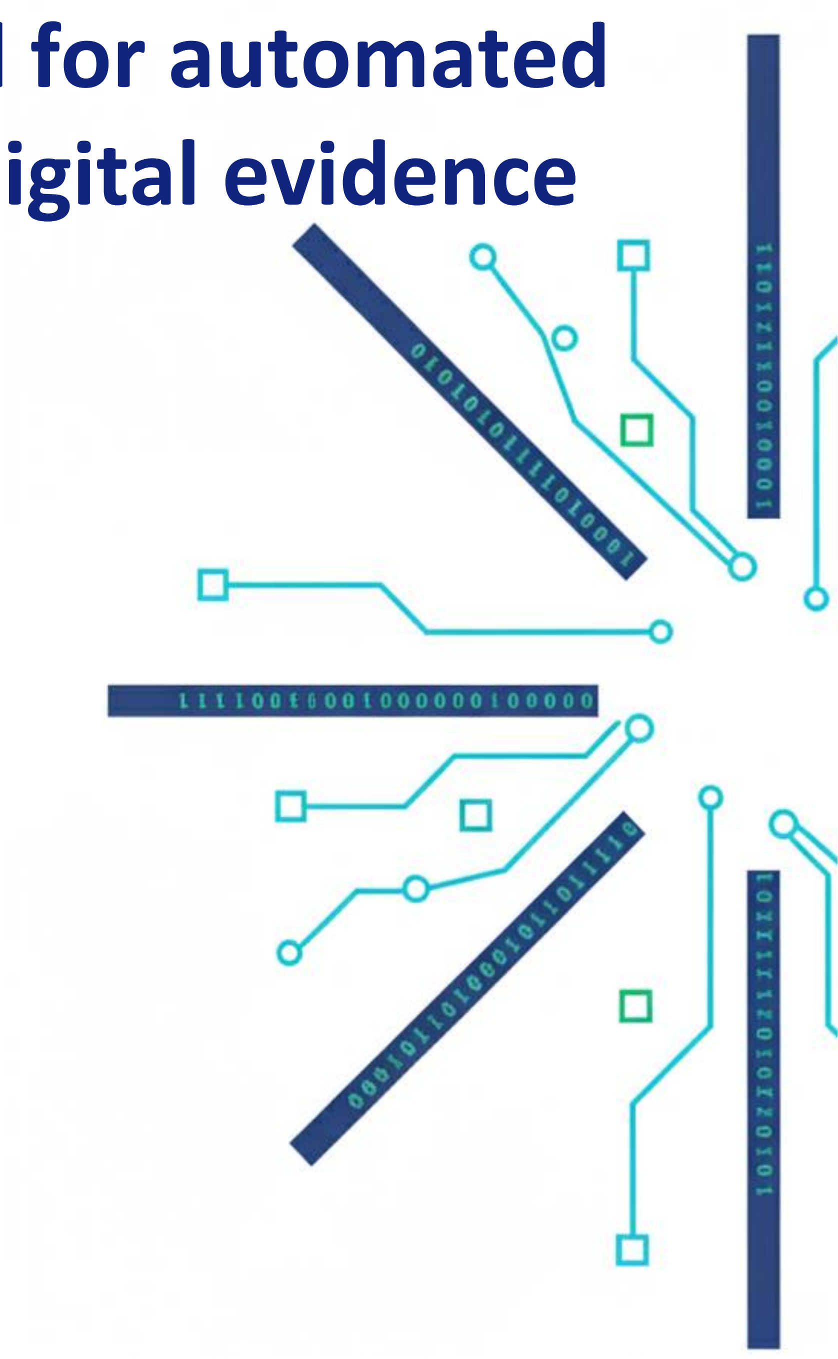
D12 – Method for automated collection of digital evidence



The project Automatization of digital forensics and incident response (ADFIR) funded by the European Union – Next GenerationEU through the Recovery and Resilience Plan of the Slovak Republic under project No. č. 09-I05-03-V02-00079.

CONTENTS

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU
VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA
VÝSKUMNÁ AGENTÚRA

1 Project description

The project **Automatization of digital forensics and incident response** (hereinafter referred to as **"ADFIR"**) is funded by the **European Union – Next GenerationEU through the Recovery and Resilience Plan of the Slovak Republic** under project No. č. 09-I05-03-V02-00079. This project addresses one of the key challenges in cybersecurity and information security – how to process the massive volume of digital evidence generated during cybersecurity incidents or forensic investigations. Currently, this process is highly demanding in terms of human resources and time. Therefore, automation using machine learning methods can significantly **improve the quality of digital forensic analysis** and reduce the time required to perform it. Overall, this enables security teams to respond more effectively to cyber threats. Main benefits of this project are:

- **Accelerated Resolution of Cybersecurity Incidents**. The project ADFIR introduces automated approaches to collecting, processing, and analyzing digital traces. As a result, security teams can identify the causes of incidents more quickly and adopt effective measures to address them.
- **Reduced Workload for Forensic Analysts**. Routine and time-consuming tasks involved in processing digital traces will be replaced by automated methods. This will allow analysts to focus on more complex cases and strategic decision-making.
- **Higher Quality and Consistency of Outputs**. The use of unified methodologies and tools ensures that the processed digital traces will be more accurate, consistent, and easily verifiable. This significantly reduces the risk of errors caused by human factors.
- **Potential Use in Criminal Proceedings.** The project outputs will be developed in compliance with legal requirements and standards, allowing the digital traces to be accepted as relevant evidence for investigations and court proceedings.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

2 Introduction

The work packages (KPB2 – KPB3) that we will describe reflect part of the lifecycle of digital evidence processing and analysis, starting with the identification of digital evidence sources and digital evidence collection (KPB2), continuing with the extraction of digital evidence/digital artefacts important for further analysis and representation of the extracted information (KPB3). The data processed in this way is suitable for further analysis by users (digital forensic analysts).

As part of the initial phase of the ADFIR project, we primarily focused on the conceptualization and detailed design of the methodology for the creation of the dataset and the collection of forensic data. Since the quality of the input data is critical for the subsequent training of machine learning models, the main goal of this period was to define robust and repeatable practices that will ensure the forensic integrity and consistency of future datasets. The methodology envisages a hybrid approach, which will combine the collection of data from simulated attacks in a controlled environment with anonymized artifacts from real practice in the implementation phase.

For the process of data collection and processing, we design and approve a technology stack that combines established "industry-standard" solutions with IstroSec's internal tools. The collection methodology will explicitly define the use of tools from Eric Zimmerman (parsers for forensic artifacts in operating system Windows) for the phase of rapid triage and extraction of standard artifacts. However, the key added value of the proposed methodology is the integration of the proprietary **Gryphon** and **Athena tools**. The Gryphon tool is designed in the methodology to capture telemetry and behavioral patterns at the process level, while the Athena tool was incorporated into the process for the needs of specific normalization and data correlation, thus paving the way for the processing of formats that are not covered in sufficient depth by conventional tools.

As part of the preparation of the methodology, we also carry out an in-depth analysis and subsequent selection of forensic artifacts, which will form the core of the future dataset. This selection is based on mapping attackers' techniques to the MITRE ATT&CK framework. The methodology strictly determines that the subject of collection will be primarily execution artifacts (ShimCache, Amcache, Prefetch), file system (MFT, USN Journal) and persistence. These artifacts were selected because, according to our analyses, they provide the highest information value for the planned anomaly detection algorithms and incident timeline reconstruction.

The output of this phase will be comprehensively prepared and verified procedures. The goal is to define exactly **how** the data will be collected, **what** tools will be used to ensure its integrity, and **why** we will focus on specific types of digital traces. This will create the necessary prerequisite for the start of the actual collection and generation of data in the next stage of the project.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3 Task KPB2.1

Task KPB2.1 – Identification of forensic objectives and their sources of artifacts that fulfil forensic objectives for the purposes of confirming or refuting a forensic hypothesis (IstroSec).

In this task, we define the goals of forensic analysis and determine the sources of artifacts that are most useful for achieving the goals of the analysis. Sources should be sufficiently robust and carry such records that will allow to confirm or refute the forensic hypothesis.

3.1Forensic Analysis Process

Forensic analysis as a process consists of several steps. Before proceeding to the description of a forensic tool, it is necessary to clarify what steps the digital footprint goes through during the analytical process.

The most well-known standard that describes the forensic analysis process is **NIST SP 800-86 (2006).** It defines **the 4 main phases** that we list in Tab. 1.

Phase	Key activities included in this phase
Collection	Identify, label, record and retrieve data from relevant sources while maintaining integrity
Examination	Forensic processing of acquired data (decompression, search, extraction of hidden data) using automated and manual tools
Analysis	Analysis of the results of the review → answers to incident questions (what, when, how, who)
Reporting	Presentation of results (what was found, how it was found, significance to the case)

Tab. 1 -Forensic analysis process phases according to NIST SP 800-86

NIST explicitly says that these phases can take place iteratively and in parallel. The project focuses mainly on the second and third phases of the forensic analysis process: pre-processing of traces, extraction of relevant data from artifacts, and processing of this data into a dataset used in the next phase of the project. However, as part of the data collection, the project also extends to the first phase of the NIST methodology.

3.2The main categories of forensic artifacts (Windows) and their sources

Category	Typical artifacts and resources	Example of meaning
Program Execution	Prefetch, Amcache, Shimcache, UserAssist, Jump Lists, LNK	Which executable files were triggered, when, how many times

Financované Európskou úniou NextGenerationEU
PLÁN OBNOVY
ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU
VAIA
VÝSKUMNÁ AGENTÚRA

	súbory, SRUM, Event Logs (4688/4689)	
Persistence	Run/RunOnce keys, Services, Scheduled Tasks, WMI Event Subscriptions, AppInit_DLLs, Startup folder	How the attacker/malware maintains itself on the system after a reboot
File / Folder Access	Shellbags, LNK + CustomDestinations, OpenSaveMRU, LastVisitedMRU, RecentDocs, Windows 10 Timeline, Thumbcache	Which folders and files were opened by the user
USB / External Devices	setupapi.dev.log, registry USBSTOR/USB, Amcache, MountedDevices, Event ID 4663 + 4656 v Security logu	Which USB thumbdrives were connected and when
User Activity	Windows Timeline, SRUDB.dat, TypedPaths, RunMRU, Browser history/cache	What the user did, searched for, wrote
Authentication / Logon	Security Event Log (4624, 4625, 4648, 4672), Netlogon.log, LSA secrets	Who logged in, from where, in what way, successfully/unsuccessfully
Network Activity	DNS cache, BITS logs, PowerShell Operational, Sysmon Event ID 3, Event ID 3 (Microsoft-Windows-NetworkProfile)	Communication with the network, C2 servers
Browser Artifacts	Edge/Chrome/Firefox – History, Cookies, Downloads, WebCacheV01.dat, Login Data	Browsing, site logins, downloads
System Configuration	Registry HKLM\SYSTEM\CurrentControlSet, keys BCD, Drivers, Services	How the system is set up, which drivers are there etc.
Anti-Forensics / Timestomping	$MFT $STANDARD_INFORMATION vs $FILE_NAME, Event Log clearing (ID 1102 in Security.evtx), $RecycleBin	Attempts to cover up traces of the attacker's presence
Memory / Volatile	RAM image → Volatility/WinPMEM tool (hivelist, pslist, netscan, malfind)	Running malware, injections, encryption keys
Data Exfiltration	Most of the above can help identify data exfiltration to some extent. This also includes various logs of programs used for data sharing, etc., which are not	What executable files/programs were in the system, what were they running, which applications transmitted data over the network, under which accounts

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

	discussed in detail in the document.	the activities took place, whether any larger archive files were created, which may indicate data staggering before their exfiltration.
File Presence	A significant intersection with the above categories.	It testifies to the presence of a file on the system, whether it provides an accurate timestamp or not.

Tab. 2 – The main categories of forensic artifacts (operating system Windows)

Based on the previous lines, we can derive the minimum set of data that the tool must be able to secure and process:

1. All *.evtx logs
2. Prefetch Folder
3. Amcache.hve
4. SRUDB.dat
5. All NTUSER. DAT + UsrClass.dat
6. HKLM kľúče (SYSTEM, SOFTWARE, SAM, SECURITY)
7. %SYSTEMROOT%\appcompat\pca* (Program Compatibility Assistant)
8. Scheduled Tasks + WMI repository
9. $MFT, $LogFile, $UsnJrnl:$J (if possible)
10. Memory dump (if the system is running)

These sources will most likely cover most common forensic questions (who, what, when, how it got into the system, what it did, how it persists).

Naturally, a tool developed for commercial purposes would provide a much wider portfolio of data, such as data on email clients and mailbox content, artifacts left by remote access tools (TeamViewer and a number of alternatives) or data sharing in the cloud (OneDrive, Google Cloud, etc.). In the context of our project, however, it is essential to define a set of data enabling the development and testing of an analytical model. The created procedure should then be expandable with other sources of data (artefacts), their collection, processing and involvement in the evaluation.

3.3Summary

These categories and artifacts form a comprehensive set of digital footprints on Windows. For each artifact, it is necessary that the forensic tool:

- **Collected** a given file or key (e.g. copied NTUSER. DAT, MFT, etc. from live or image).
- **Parsed** evidence of various formats (whether it is a text log, binary format, database) and extracted forensically significant information (timestamps, paths, names, values).

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A | VÝSKUMNÁ AGENTÚRA

- **Correlated** the retrieved information into categories: e.g. derive "Execution" from Prefetch and Shimcache, "File Use" from LNK and Shellbag, "Login/Remote Access" from Event logs, etc.

Such a tool would allow experts to quickly get answers to the key questions of the investigation: What was executed on the system? What files did the user open? When and where did he connect remotely? What persistences did he have? What did he do on the Internet? Who did he communicate with? etc., which would greatly streamline the digital forensic analysis of Windows systems.

In the next chapter, we will use the abbreviations (Tab. 3) when listing the paths to the artifacts on the disk.

Abbreviation	Meaning
%SYSTEMROOT%	corresponds to the location of C:\Windows
%SYSTEMDRIVE%	system disk, C:\
%USERPROFILE%	the user's profile directory, C:\Users\<username>
%APPDATA%	%USERPROFILE%\AppData\User Roaming folder
HKLM	HKEY_LOCAL_MACHINE registry hive; on an offline (shut down) system, its contents are reflected in the SAM, SYSTEM, SECURITY support files, stored in the %SYSTEMROOT%\System32\config folder
HKCU	HKEY_CURRENT_USER, the registry hive for the currently logged in user; offline, its contents are retrieved from the NTUSER.DAT support file from the user profile, as well as from the USRCLASS.DAT support file (since this contains data from the key HKEY_CURRENT_USER\Software\CLASSES)
HKU	HKEY_USERS, registry hives for all users on the system, in offline state their content is reflected by NTUSER.DAT support files in user profiles;
Live and offline	Live acquisition means gathering data from a running, switched on device. Offline, or post-mortem, is a term that we will use to refer to the securing of traces or artifacts from a switched off device or disk image.

Tab. 3 –List of abbreviations

3.4Artifacts of program execution and their presence in the system

3.4.1 Amcache

3.4.1.1 Description

Application Compatibility database, which records details about the programs that are running, especially when they were first launched. It appears on systems from OS version Windows 8. (In Win7, the RecentFileCache.bcf file had a similar function.) Amcache contains

Financované Európskou úniou NextGenerationEU · PLÁN OBNOVY · ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU · VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA · VÝSKUMNÁ AGENTÚRA

executable file names, versions, SHA-1 hashes, and a timestamp that has been interpreted as the first execution date in the past. The launch date and time is not 100% reliable. The artifact still proves the presence of the program in the system, but the timestamp cannot be taken as definitive proof that the program has started. Correlation with other artifacts is appropriate if we want to prove if and when the program executed.

3.4.1.2 Location

%SYSTEMROOT%\appcompat\programs\Amcache*

3.4.1.3 Method of acquisition

Live and offline.

3.4.1.4 Primary category

Execution

3.4.1.5 Other categories

File presence

3.4.2 Background Activity Monitor (BAM/DAM)

3.4.2.1 Description

BAM (and similar DAM) from Win10 1709 onwards monitors running applications at the OS background tasks level. Under the user-specific key (SID), it lists the path of the executable file and the last execution date/time.

3.4.2.2 Location

Live: HKLM\SYSTEM\CurrentControlSet\Services\bam\State\UserSettings\<SID>
Offline: SYSTEM\ControlSet*\Services\bam\State\UserSettings\<SID>

3.4.2.3 Method of acquisition

Live and offline.

3.4.2.4 Primary category

Execution

3.4.2.5 Other categories

User Activity

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.4.3 MUICache

3.4.3.1 *Description*

When the GUI launches the application via Windows Explorer, Windows stores the localized name of the application in the so-called MUICache. The presence of a record indicates that the program in question was run by that user (albeit without a timestamp).

3.4.3.2 *Location*

HKCU\Software\Classes\LocalSettings\Software\Microsoft\Windows\Shell\MuiCache (in UsrClass.DAT registry of each user)

3.4.3.3 *Method of acquisition*

Live and offline.

3.4.3.4 *Primary category*

Execution

3.4.3.5 *Other categories*

File presence, User activity

3.4.4 Prefetch

3.4.4.1 *Description*

Windows Prefetch files are created to optimize the execution of an application on Windows. These files contain the number of runs for each app, from one to eight timestamps of the last runs, and a record of all files opened during a set period of time after the app was launched.

Windows creates *.pf files when executables are executed. As mentioned above, PF files contain timestamps of the last up to 8 program executions and a list of loaded DLLs/files. Prefetch serves as a strong proof that the program is running (if the prefetcher is not turned off, e.g. on servers or SSD systems it may be off by default).

3.4.4.2 *Location*

%SYSTEMROOT%\Prefetch\*.pf

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.4.4.3 Method of acquisition

Live and offline.

3.4.4.4 Primary category

Execution

3.4.4.5 Other categories

-

3.4.5 RecentApps

3.4.5.1 Description

It records the most recently launched apps (especially UWP and modern apps). Each record confirms the launch of the program by the given user.

3.4.5.2 Location

HKCU\Software\Microsoft\Windows\CurrentVersion\Search\RecentApps (Win10+)

3.4.5.3 Method of acquisition

Live and offline.

3.4.5.4 Primary category

Execution

3.4.5.5 Other categories

User Activity

3.4.6 Recentfilecache

3.4.6.1 Description

The RecentFileCache.bcf binary is a cache of recently run executable files (especially on systems where there is no Prefetch or for installation files). It contains a list of paths to running files and basic metadata. For forensic analysis, it is useful in determining whether and which binaries have been triggered, especially if Prefetch is not available or is disabled.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

Type/Attributes: Binary BCF format. Each record contains the path to the executable file and possibly a timestamp (the time the file was last modified). For example, the record might look like: C:\Temp\malware.exe – timestamp 2025-06-01.... Notably, if a file appears in this list, it has been run (either manually or as part of an installation).

3.4.6.2 Location

%SYSTEMROOT%\AppCompat\Programs\RecentFileCache.bcf (other than Windows 7)

3.4.6.3 Method of acquisition

Copy from disk. Live and offline.
Tools: *A RecentFileCache Parser* (e.g., from *Eric Zimmerman* tools or *FireEye's RecentFileCacheParser*) can decode .bcf into a readable CSV. Also *KAPE* (target **EvidenceOfExecution** contains it).

3.4.6.4 Primary category

Execution

3.4.6.5 Other categories

File Presence

3.4.7 RunMRU

3.4.7.1 Description

Records of commands run via the Win+R (Run) dialog. It stores the history of commands/exes that the user ran manually via "Run".

3.4.7.2 Location

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU

3.4.7.3 Method of acquisition

Live and offline.

3.4.7.4 Primary category

Execution

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.4.7.5 Other categories

User Activity

3.4.8 Shimcache

3.4.8.1 Description

Application compatibility cache stored in **System** hive. It contains a list of 1024 recently executed **or accessed** executables with their path and timestamp of the last modification of the executable at the time of its execution (that is, not the startup time itself, and since Windows 10 it does not even indicate whether the file was run at all![1]). **Attention – it is written to the registry only when the OS is turned off and does not have to mean *the actual* startup** (maybe user just opened the folder with EXE).

3.4.8.2 Location

HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\AppCompatCache\AppCompatCache

3.4.8.3 Method of acquisition

Live and offline. Live ideally from RAM, because changes are recorded in the registry only when the device is rebooted, until then they are only in memory.

3.4.8.4 Primary category

Historically, it is classified as Execution, in fact it corresponds more to File Presence.

3.4.8.5 Other categories

-

3.4.9 SRUM: System Resource Usage Monitor

3.4.9.1 Description

A database that stores information about the use of network system. For example, the Task Manager tool draws data from it.

[1] https://artefacts.help/windows_shimcache.html

The SRUM DATABASE (ESE DB) was introduced in Windows 8 and tracks various statistics about the running of applications – e.g. when and for how long the processes were running, the volumes of data transferred over the network per process, battery consumption, etc. It can be used to find out the timeline of application execution (at 60-minute intervals) and the network activity of processes. It can also contain a record of the user under whose account the application was running.

- Application Resource Usage

A table in the System Resource Usage database that stores statistics on the use of system resources by individual applications.

- Network Connectivity Usage

A table in the System Resource Usage database that stores statistics related to network connections. Specifically, it contains the startup time and duration of connections to each network interface.

- Network Data Usage

A table in the System Resource Usage database that stores statistics regarding the use of network data to run applications. It includes the application path, network interface, bytes sent, and bytes received.

3.4.9.2 Location

%SYSTEMROOT%\System32\sru\SRUDB.dat

3.4.9.3 Method of acquisition

Copy the database file, ideally with the entire folder. Live and offline.

3.4.9.4 Primary category

Execution

3.4.9.5 Other categories

Data Exfiltration, User Activity

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.4.10 UserAssist

3.4.10.1 Description

The UserAssist key monitors the launch of GUI applications by the user. The names are ROT13-coded; UserAssist contains a run counter and the last start time for each program run via the GUI. It provides clues about what the user executed and when. Counter can be reset after a major system update.

3.4.10.2 Location

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{GUID}\Count
On the switched off system and for non-logged-in users, we replace HKEY_CURRENT_USER registers with NTUSER.DAT.

3.4.10.3 Method of acquisition

Live and offline.

3.4.10.4 Primary category

Execution

3.4.10.5 Other categories

User Activity

3.4.11 Windows Activities – W10 Timeline

3.4.11.1 Description

Windows Timeline (ActivitiesCache.db) is a database introduced in Windows 10 v. 1803 (Timeline feature). When the timeline is enabled, it records the user's activities, potentially across multiple devices: opening a document, a web page, launching an app, including the start and end times of the activity. Each activity has an AppID (e.g., Word, Excel, Edge) and often a filename or URL. It is a goldmine for forensic analysis – it can be read that, for example, on May 10, 2023 at 2:30 p.m., a user opened the C:\Docs\Tajný.xlsx file in Excel and closed it at 2:55 p.m. It keeps records for a maximum of the last 30 days.
Note: in Windows 11, it will no longer contain significant data.

3.4.11.2 Location

%USERPROFILE%\AppData\Local\ConnectedDevicesPlatform\L.*\ActivitiesCache.db*

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.4.11.3 *Method of acquisition*

Live and offline.

3.4.11.4 *Primary category*

Execution

3.4.11.5 *Other categories*

User Activity, File Open, Web browsing

3.5Artifacts of persistence

3.5.1 DNS Hosts File

3.5.1.1 *Description*

Records from the etc\hosts file. Modifying this file can help the attacker to ensure persistence.

3.5.1.2 *Location*

%SYSTEMROOT%\System32\drivers\etc\hosts

3.5.1.3 *Method of acquisition*

Live and offline.

3.5.1.4 *Primary category*

Persistence

3.5.1.5 *Other categories*

-

3.5.2 Registry Persistence

3.5.2.1 *Description*

A collection of registry keys that can be used to ensure malware persistence.

Run keys - HKLM\Software\Microsoft\Windows\CurrentVersion\Run (and RunOnce, ...) and similarly HKCU\Software\Microsoft\Windows\CurrentVersion\Run - "Run" keys contain a list of programs to be automatically started when the system starts (HKLM for all users) or when

Financované Európskou úniou NextGenerationEU · PLÁN OBNOVY · ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU · VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA · VÝSKUMNÁ AGENTÚRA

a specific user logs on (HKCU). A value is the path to a .exe or script. If the forensic tool finds an entry in Run, it means that the program in question was running persistently at startup.

AppInit_DLLs – Path (in the registry): HKLM\Software\Microsoft\Windows NT\CurrentVersion\Windows\LoadAppInit_DLLs and AppInit_DLLs value – If enabled, the DLLs listed in AppInit_DLLs will be loaded into each process using the User32.dll. This is an older persistence mechanism (less used today, default off or limited by signature in newer Windows). Still, if there is a value, it indicates an effort to inject code into processes.

Winlogon Shell / Userinit – Path (in the registry): HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell and Userinit – Normally, Shell has *a value of explorer.exe*. If an attacker sets the Shell to another executable file, it will run instead of Explorer after logging in (**or together with Explorer** if they add "Explorer.exe, malware.exe"). Similarly, Userinit default %SYSTEMROOT%\system32\userinit.exe, if it is modified (e.g. adds a path to malware), it will run at the logon.

3.5.2.2 Location

HKEY_USERS*\Software\Microsoft\Windows\CurrentVersion\Run\*
HKEY_USERS*\Software\Microsoft\Windows\CurrentVersion\RunOnce\*
HKEY_USERS*\Software\Microsoft\Windows\CurrentVersion\RunServices\*
HKEY_USERS*\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run\*
HKEY_USERS*\Software\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software\Microsoft\Windows\CurrentVersion\Run\*
HKEY_USERS*\Software\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\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\RunServices\*
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run\*
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software\Microsoft\Windows\CurrentVersion\Run\*
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Terminal Server\Install\Software\Microsoft\Windows\CurrentVersion\RunOnce\*
HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components**
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Session Manager\BootExecute

3.5.2.3 Method of acquisition

Reg save for live, offline capture is of course also possible.

3.5.2.4 Primary category

Persistence

3.5.2.5 Other categories

Execution, User Activity

3.5.3 Scheduled Tasks

3.5.3.1 Description

Scheduled tasks are used to run Windows programs or scripts at specified intervals. Scheduled tasks are a persistence mechanism – they allow you to run a script/program at startup, login, or at scheduled intervals. Malware often creates its own Scheduled Task to persist after a reboot. Analysis of the contents of Tasks\ reveals the name of the task, time triggers and especially **the path to the program or command being run**. This artifact is crucial when looking for persistent *backdoors* and automated activities. For example, a task called "Windows Update Service" running *C:\Temp\evil.exe* at login points to attacker persistence.

- **Path:** XML files in %SYSTEMROOT%\System32\Tasks\ – Each Scheduled Task is represented by an XML file. The important ones are those that execute some .exe or a suspicious script on a certain trigger (at startup, at login, or periodically). In addition, the task settings are also in the registry under HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\. Forensic analysis reveals the name of the task, the command/program being executed, the conditions of execution, and the most recently executed times. Attackers often create hidden tasks for persistence.
- **Scheduled Tasks Events** – In the **Microsoft-Windows-TaskScheduler/Operational log,** there are events **ID 200 and 201**, which indicate the start and end of the scheduler task. If malware or admin has created a Scheduled Task to run .exe, the following events in the Security.evtx log will reveal when and what was executed.
  - **4698:** A scheduled task was created.
  - **4699:** A scheduled task was deleted.
  - **4700:** A scheduled task was enabled.
  - **4701:** A scheduled task was disabled.
  - **4702:** A scheduled task was updated (modified)

- **Tools:** *KAPE* target **ScheduledTasks** (collects both .job and .xml). Parsing .job files is possible with *the jf.exe* tool (Job File parser) or *TZworks silentrunner*. XML tasks can be read directly (they contain the definition of triggers and actions in XML tags). Also,

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

Sysinternals **Autoruns** will display most of the scheduled tasks.

Type/attributes of scheduled tasks:

- *.job* (binary): contains the job name, schedule (start date/time), and command + parameters.
- *.xml*: contains the <Actions> section (with the command e.g. <Exec><Command>C:\path\program.exe</Command>), the <Triggers> section (when it is triggered – e.g. AtLogon) and <Principals> (under which user). **Key attributes:** *TaskName*, *TriggerType* (Startup/Logon/Daily/etc.), *ActionPath* (path to exe or script), *Arguments*. From this data, you can identify a malicious task and what exactly triggers it.

3.5.3.2 Location

- %SYSTEMROOT%\Tasks\*.job - legacy .job format
- %SYSTEMROOT%\System32\Tasks\* - XML files with detailed settings
- HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Schedule\TaskCache
- Event Logs: Microsoft-Windows-TaskScheduler/Operational (ID 106, 200–201)

3.5.3.3 Method of acquisition

Live and offline. We attach examples of how PowerShell can be used for this purpose.

```
Get-ScheduledTask
gci -path C:\windows\system32\tasks -recurse | Select-String Command
| FL Filename, Line
gci -path C:\windows\system32\tasks -recurse | Select-String
"<Command>",Argument | FT Filename,Command,Line
gci -path C:\windows\system32\tasks -recurse | Select-String Command
| ? {$_. Line -match "MALICIOUSNAME"} | FL Filename, Line
(gci -path C:\windows\system32\tasks -recurse | Select-String
"<Command>" | select -exp
Line).replace("<Command>","").trim("</Command>").replace("`"","").tr
im();
gci 'REGISTRY::HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Schedule\TaskCache\Tree' -rec -force | ? {$_.
Property -notcontains 'SD'}
gci 'REGISTRY::HKLM\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Schedule\TaskCache\Tree' -rec -force | Get-
ItemProperty | ? {$_. SD.length -lt 100}
```

3.5.3.4 Primary category

Persistence

3.5.3.5 *Other categories*

Execution, File Presence

3.5.4 Services

3.5.4.1 *Description*

Services are Windows applications that can be run without user interaction – kind of like daemons in UNIX/Linux OS. Services can be used as a method of malware persistence.

- **Path (in the registry):** HKLM\SYSTEM\CurrentControlSet\Services\<ServiceName> - All system services are in this key. If an attacker adds a malware service, it will appear here. Important fields: **ImagePath** (path to the .exe or .dll to be started), **Start** type (whether it starts automatically). Alternatively, the event log **System** (ID 7045) indicates the installation of a new service. For each service, it can be recorded here when it ran (Log events 7035/7036).

3.5.4.2 *Location*

HLM\SYSTEM\CurrentControlSet\Services\*

3.5.4.3 *Method of acquisition*

Live and offline.

3.5.4.4 *Primary category*

Persistence

3.5.4.5 *Other categories*

Execution

3.5.5 Shim Databases

3.5.5.1 *Description*

Databases used by the application compatibility infrastructure to apply "shims" to executable files for backward compatibility. These databases can be used to inject malicious code into legitimate processes and maintain persistence on the endpoint.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.5.5.2 *Location*

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\InstalledSDB\*

3.5.5.3 *Method of acquisition*

Live and offline.

3.5.5.4 *Primary category*

Persistence

3.5.5.5 *Other categories*

Execution

3.5.6 Windows Drivers files

3.5.6.1 *Description*

Suitable for cross-reference with the Services registry key. **Drivers are** also services (type=1 in the registry). Malicious rootkits can add drivers. Therefore, the artifact for drivers in the registry is the same as for Windows Services - Services keys. If there are suspicious .sys files in the %SYSTEMROOT%\System32\drivers\ directory and are registered with Services, this is kernel-level persistence.

3.5.6.2 *Location*

%SYSTEMROOT%\System32\drivers\*.sys

3.5.6.3 *Method of acquisition*

Live and offline.

3.5.6.4 *Primary category*

Persistence

3.5.6.5 *Other categories*

-

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.5.7 WMI Repository

3.5.7.1 Description

A list of WMI EventConsumers and any EventFilters that are bound to them via FilterToConsumerBinding. WMI EventConsumers can be used as a fileless malware persistence method. We also acquire WMI objects (.mof files) and WMI event log.

- **WMI Event Subscription** – Windows Management Instrumentation can be abused for persistence through so-called Event Subscriptions. Artifacts are stored in a WMI repository (MOF database). Using tools (e.g. PowerShell Get-WmiObject -Namespace root\subscription) you can find **the Consumer** and **Filter** settings that run commands on certain events (e.g. login). This is not directly visible in the registry, but the WMI-Activity Operational log (Microsoft-Windows-WMI-Activity/Operational) may contain events if WMI triggers have been executed.
- **WMI Persistence (MOF files)**
  Path: %SYSTEMROOT%\System32\wbem\mof\*.mof (or subfolder *...\wbem\mof\en-US* for localization)

  Persistence via WMI uses the so-called Persistence via WMI. **Permanent Event Subscriptions** – settings in WMI that can run code at events (e.g. at system startup). These settings can also be written via . MOF (Managed Object Format) files – if they are stored in the wbem\mof\ folder, WMI will automatically load and register them. Attackers thus create persistent *WMI Event Consumer* and *Filter*, which, for example, always run the command at boot. For forensic analysis, the detection of unknown .mof files or unexpected WMI Event Consumers is critical.

  The MOF files themselves are text (they can be deleted after loading).

  KAPE (target **MOF** if any) can copy *.mof. Furthermore, *PowerShell directly (Get-WMIObject __EventFilter)* or **Autoruns** (WMI item). There are also specialized scripts, e.g. *WMIPersist vbs* script to list WMI permanent events.

  Type/attributes: .mof script – text defining WMI classes. Typically, it contains instances of __EventFilter, CommandLineEventConsumer, and __FilterToConsumerBinding. The key is the **CommandLineTemplate attribute** in *EventConsumer*, which reveals what command/program will be executed. Also *Name* and *Query* in *EventFilter* (e.g. filter of type SELECT * FROM __InstanceModificationEvent ... triggered at launch). When analyzing WMI persistence, we look for commands, paths to executed files or scripts, and non-standard consumer names.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.5.7.2 Location

%SYSTEMROOT%\System32\wbem\Repository\OBJECTS. DATA
%SYSTEMROOT%\System32\wbem\Repository\FS\OBJECTS. DATA
%SYSTEMROOT%\System32\winevt\Logs\Microsoft-Windows-WMI-Activity%4Operational.evtx
%SYSTEMROOT%\System32\wbem\mof\*.mof (or subfolder *...\wbem\mof\en-US* or other, for localization)

3.5.7.3 Method of acquisition

Live or offline

3.5.7.4 Primary category

Persistence

3.5.7.5 Other categories

Execution

3.6User activity artifacts

3.6.1 Browser History

3.6.1.1 Description

Browser history with Chrome, Edge, Firefox, and Internet Explorer. Due to variability, we will adjust the format of this section.

Artifacts associated with web browsing – history of site visits, downloaded files, cookies, etc. Each browser has its own repositories, often in the form of databases.

3.6.1.2 Internet Explorer / Edge Legacy (IE10-11, Legacy Edge)

- History, cookies and cache are stored in a common ESE database **WebCacheV01.dat**: **Path:** %USERPROFILE%\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat. This database (Extensible Storage Engine) contains tables **Container_#** for visited URLs, History, Cookies, Cache, etc. Each history record contains a URL, page name, time of visit (in the so-called FILETIME/GUID format in local time). The cookies table contains cookies with attributes. **Downloads** are also recorded in the history.

    - Forensic: WebCacheV01.dat detects **all the websites that the user visited in IE or the old Edge**, with time and even, if the MS account is set to roaming, it can also contain history from other devices. Cookies can reveal login sessions (e.g. bank cookie).
- **Session Recovery**: %USERPROFILE%\AppData\Local\Microsoft\Internet Explorer\Recovery\ and for Edge Legacy in the MicrosoftEdge\User\Default\Recovery\Active\ - Contains snapshots of open tabs (URLs) for crash recovery.
- **IE TypedURLs**: HKCU\Software\Microsoft\Internet Explorer\TypedURLs – a list of URLs that the user has manually entered in the IE address bar (top 25).

3.6.1.3 Microsoft Edge (Chromium, v79+)

- The new Edge (as well as Google Chrome) uses chromium core, so the same data structure. **Profiles path:** %USERPROFILE%\AppData\Local\Microsoft\Edge\User Data\Default\ (or another profile). Here it is:
    - History – A SQLite database with a urls table (contains URL, title, number of visits, time of last visit in Unix epoch microseconds).
    - Cookies – SQLite DB (or from a certain version .sqlite or .ldb files), with host, name, value, last access.
    - Cache – cache files on the disk in the Cache\ folder.
    - Downloads – records of downloaded files (in History DB or separate History-journal).
    - Session Storage & Local Storage – in the Default\Local Storage\leveldb\ or Default\Network subfolder etc (see below in **WebStorage**).
    - **WebStorage (Local Storage & IndexedDB):** Modern browsers (Chrome, Edge) use WebStorage – locally stored data of web applications. These are in the form of files (LevelDB databases) in the profile under ...\Default\Local Storage\ and IndexedDB\. For example, chat web apps (Slack web) can store history in LocalStorage. SANS research highlights that **WebStorage stores a huge amount of data** (up to GB) about activity on web services, often more than a traditional cache.
    - *Forensics:* From the Edge/Chrome history, it is possible to reconstruct which pages the user visited and when (e.g. a specific URL to cloud storage, internet banking, social network). Cookies help identify logins to web services (e.g. if there is a Gmail session cookie). LocalStorage can contain chats (e.g. messages from Slack or Teams, if they ran in a web environment).

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.6.1.4 Google Chrome

- **Profile path:** %USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\ – the structure is analogous to Edge (History, Cookies, SQLite, etc.). **The history** file is a SQLite DB with a URL overview.
- **Downloads:** in the History DB downloads table or in a separate DownloadMetadata file.
- **Login Data:** SQLite DB with saved passwords (DPAPI encrypted).
- *It should be noted* that Chrome also uses WebStorage in the same way a MS Edge.

3.6.1.5 Mozilla Firefox

- **Profile path:** %USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\<profile>\ – The main DB is places.sqlite (SQLite) – contains **history** (tables moz_places and moz_historyvisits) and **bookmarks**. places.sqlite stores the URL of visited pages with a timestamp (unix epoch in microseconds) and the number of visits.
- **Downloads:** Firefox also writes downloaded files to places.sqlite (moz_annos table) and downloads.sqlite (in older versions).
- **Cookies:** cookies.sqlite – SQLite DB s cookies (name, value, host, lastAccessed).
- **Session restore:** sessionstore.jsonlz4 – compressed JSON with information about open tabs (URLs) if the user had something open before closing the browser.

3.6.1.6 Other

- **Safari for Windows** (no longer supported, but older installations): used SQLite databases History.db etc in %APPDATA%\Apple Computer\Safari\. Probably not necessary, a marginal case.
- **WebCache / Index.dat (IE <10)** – For the sake of completeness, if Windows 7 had IE8/9: the old IE used hidden index.dat files in the profile (History, Cookies, Cache). These can be extracted by parsers. However, on Windows 7, IE11 already uses WebCacheV01.dat.
- **Applications built on Chrome (Electron)** – **Important:** Modern desktop applications such as Skype for Desktop, Slack, WhatsApp Desktop, Signal, Discord, Teams, etc. are often built on Electron (Chromium) and **store data similarly to a browser**. That is, in their AppData folders we can find **Local Storage** (LevelDB) and possibly history if it is an integrated browser.
    - E.g. **Slack** (desktop) uses Chrome WebStorage to cache messages (the same goes for Teams). Therefore, in the forensic analysis of web storage, it is not only necessary to look for browser domains, but also the data of these applications (e.g. https://app.slack.com local storage can also be in the browser or in the Electron folder of Slack).

For the purposes of the automation tool, let's also mention that due to the facts described above, it should be able to parse the main formats – SQLite DB (Chrome, Firefox), ESE DB (WebCacheV01.dat), LevelDB (LocalStorage), JSONLZ4 (Firefox session), and extract URL history, cookies, downloads, etc.

3.6.1.7 *Primary category*

User activity

3.6.1.8 *Other categories*

Execution, file presence, data exfiltration

3.6.2 ComDlg32/CidSizeMRU

3.6.2.1 *Description*

A registry key that contains a list of recently launched applications. The application uses the Common Dialog Box function, i.e. it allows the user to open or save a file using the Windows Explorer dialog. An example is office application that allow you to use File... button to access various such functions.

3.6.2.2 *Location*

HKEY_USERS*\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\CIDSize MRU

3.6.2.3 *Method of acquisition*

Live and offline.

3.6.2.4 *Primary category*

User Activity

3.6.2.5 *Other categories*

Execution

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.6.3 ComDlg32/LastVisitedPidlMRU

3.6.3.1 Description

A registry key containing a list of applications and folder paths that are associated with recently opened files in the user's OpenSavePidlMRU key.

3.6.3.2 Location

HKEY_USERS*\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU
HKEY_USERS*\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRULegacy

3.6.3.3 Method of acquisition

Live and offline.

3.6.3.4 Primary category

User Activity

3.6.3.5 Other categories

Execution

3.6.4 ComDlg32/OpenSavePidlMRU

3.6.4.1 Description

A registry key that contains a list of recently opened and saved files for a given user account.
Open/Save MRU (ComDlg32) – **Path (in the registry):**
HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU – Saves the file and folder paths that the user selected in the standard "Open/Save As" dialogs. Also, LastVisitedPidlMRU keeps the last visited folders in dialogs. These keys indicate which files the user has worked with through the applications (e.g. the last opened file in the Player, the last saved document in Office, etc.).

3.6.4.2 Location

HKEY_USERS*\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePidlMRU

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.6.4.3 *Method of acquisition*

Live and offline

3.6.4.4 *Primary category*

User Activity - File Open

3.6.4.5 *Other categories*

-

3.6.5 EventTranscript.db

3.6.5.1 *Description*

This artifact is not created on the system by default! But if it is on the system, it can contain a huge amount of data about the activity on the device.
Present since Win 10, 01/2021.

- EventTranscript.db, the SQLite database is located at:
  C:\ProgramData\Microsoft\Diagnosis\EventTranscript\EventTranscript.db

It contains 7 tables within the SQLite database:

- categories
- event_categories
- event_tags
- events_persisted
- producers
- provider_groups
- tag_descriptions

It records six (6) different types of events with the following designations:

- Browsing history: Records of your web browsing history when using the features of an app or cloud service, stored in the service or app.
- Device connection and configuration: data that describes the connections and configuration of devices connected to the service and network, including device identifiers (e.g., IP addresses), configuration, settings, and performance.
- Inking and speech: Recording input data provided to end users through an interaction or action method, such as handwriting, typing, speech, or gestures.
- Product and Service Performance: Data collected about the measurement, performance, and operation capabilities of a product or service. This data represents

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

information about the capability and its use with a focus on providing the capabilities of the product or service.

- Use of Products and Services: Data provided or captured about an end user's interaction with the service or products by a cloud service provider. The recorded data includes records of the end-user's preferences and settings for the options, the options used, and the commands provided to the capabilities.
- Software setup and inventory: Data that describes the installation, setup, and update of software.

3.6.5.2 Location

- C:\ProgramData\Microsoft\Diagnosis\EventTranscript\EventTranscript.db

3.6.5.3 Method of acquisition

Live and offline

3.6.5.4 Primary category

User Activity

3.6.5.5 Other categories

-

3.6.6 JumpLists

Since Windows 7, Jump Lists have been used in the Windows OS: these are *.automaticDestinations-ms *.customDestinations-ms files that store a list of the most recently opened files in the application and/or a list of recent actions performed with the application for each application (for example, opening an RDP session using the Remote Desktop client, or opening a web browser window in private mode).

E.g. JumpList for MS Word will contain a list of recently opened .docx. These records indicate that specific files have **been seen or opened by** the user and with that application.
For forensic analysis, they provide evidence that specific files have been opened by users (e.g. the last N documents of MS Word, PDF in Adobe Reader, etc.) even if the files themselves no longer exist.

Type/Attributes: Binary format (.ms files). Each item contains the path to the recently opened file and the date/time (e.g. LastModified in the jump list).

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

Attributes: AppID (identifies the application, e.g. identification string in the file name of the jump list), FilePath of the open file, Create/LastAccess/ModifiedTime. This information allows you to tell which file was opened in which program and when it was last opened.

Tools: KAPE target LNKFilesAndJumpLists (collects both .automaticDestinations-ms and .customDestinations-ms). JumpList Explorer (JLECmd) by Eric Zimmerman or tools such as ShellBags & JumpLists Parser are used for parsing. These extract a list of files, last opened time, and app ID.

3.6.6.1 Method of acquisition

Both live and offline, Eric Zimmerman's JumpListExplorer and JLECmd tools can be used for parsing.

3.6.6.2 Location

AutomaticDestinations:
%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations\<APP_ID>.automaticDestinations-ms
CustomDestinations:
%USERPROFILE%\AppData\Roaming\Microsoft\Windows\Recent\CustomDestinations\<APP_ID>.customDestinations-ms

3.6.6.3 Primary category

User Activity

3.6.6.4 Other categories

Execution

3.6.7 RecentDocs

3.6.7.1 Description

It contains a list of recently opened files of various types (broken down by extension). For each file, it stores the name and the last open time. For example, the .docx subkey contains the most recently opened Word documents.

3.6.7.2 Location

HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA | VÝSKUMNÁ AGENTÚRA

3.6.7.3 Method of acquisition

Live and offline

3.6.7.4 Primary category

User Activity

3.6.7.5 Other categories

File Presence/Interaction

3.6.8 ShellBags

3.6.8.1 Description

Registry keys that record the user's layout preferences for each folder that the user interacts with.

- **Path (in the registry, XP-7):** HKCU\Software\Microsoft\Windows\Shell\BagMRU (+ \Bags) in NTUSER. DAT; **Win Vista+ (USRCLASS. DAT):** HKCU\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU (+ \Bags). Shellbags record folder settings in Explorer (for example, whether a user viewed files in a folder as a list, or as icons or previews), but indirectly reveal the **history of the folders viewed** (whether on local drives, network shares, or USB). For forensic purposes, they are valuable because they reveal which folder (path) the user has opened and even deleted or detached folders.
- *ShellBags* maintain folder settings in Explorer and thus **logs of folders that the user has opened**. They contain a hierarchy of most recently used directories (MRUs) with folder names and sometimes recent accesses. For forensic analysis, we can find out what directories the user has browsed (including, for example, via external drives or network shares), and whether the names of these folders indicate, for example, the storage of illegal data. ShellBags complement the timeline, especially when the classic "LastAccess" timestamps on files are disabled.
- **Tools:** *ShellBags Explorer* (Eric Zimmerman) or *RegRipper plugin shellbags.pl* can extract the contents of ShellBag keys and turn it into a list of folders with the last view and their settings. *The KAPE* module **ShellBags** (if any) would require obtaining the NTUSER/UsrClass registers in question (e.g. via **the RegistryHives** target).
- **Type/Attributes:** Data in binary form inside the registry.
    - Important fields: **Path (folder name/path)** – can be absolute or relative, **LastWrite key time** – indicates the time of the last change (often corresponds to the last opening of the folder), and view settings (icons, window size).

Shellbags sometimes also contain the GUID of devices (for external drives) and thus can identify, for example, the drive letter to which the folder belonged.

3.6.8.2 Location

HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\Shell\Bags
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\Shell\BagMRU
HKEY_CURRENT_USER\SOFTWARE\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags
HKEY_CURRENT_USER\SOFTWARE\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU

3.6.8.3 Method of acquisition

Live and offline

3.6.8.4 Primary category

User Activity: File (Folder) presence, Folder interaction

3.6.8.5 Other categories

-

3.6.9 SUM (UAL)

3.6.9.1 Description

User Access Logging (UAL) is a feature introduced and enabled by default in Windows Server 2012 that consolidates client activity data. Among other information, user access to specific Windows Server roles (for example, Active Directory Domain Services on a domain controller) is recorded in UAL. The specific activity that triggers the record to be recorded for that role isn't documented.

Thus, on domain controllers, UAL provides information about opening sessions on domain-joined computers (if the domain controller has been contacted for authentication or group policy retrieval).

Information is stored locally in a maximum of five (5) Extensible Storage Engine (ESE) database files (.mdb):

1. A Current.mdb file that contains data for the last 24 hours.
2. Up to three <GUID>.mdb files that contain data for the entire year (first to last day) from 2 years. The data in the Current.mdb database is copied to the corresponding database (<GUID>.mdb) for the current year every day.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3. Systemidentity.mdb that contains metadata on the on-premises server, including GUID mapping and role names.

Thus, historical data from 2 years ago can be retrieved in UAL database files.

The CLIENTS table of the Current.mdb and <GUID>.mdb database files contains several interesting information:

- GUID and description of the Windows Server role that was accessed. Among others, the following roles can be encountered:
  - Active Directory Domain Services (GUID: ad495fc3-0eaa-413d-ba7d-8b13fa7ec598).
  - File server (GUID: 10a9226f-50ee-49d8-a393-9a501d47ce04).
  - Active Directory Certificate Service (GUID: c50fcc83-bc8d-4df5-8a3d-89d7f80f074b).
- Client domain and username.
- The total number of hits.
- Timestamps of the first, last and daily access.
- The IPv4 or IPv6 address of the client. Domain controllers can retrieve the host name associated with an IP address at a given time, because the computer accounts of domain-joined computers are also authenticated to AD DS.

Each entry in the CLIENTS table consists of a unique set of Windows Server roles, client domain/username, and source IP address.

The DNS database file table contains information about DNS resolutions: hostname, assigned IP address, and timestamp of the last DNS resolution.

3.6.9.2 Location

Files in the %SYSTEMROOT%\System32\Logfiles\SUM\ directory:

- Current.mdb (data for the last 24 hours).
- Up to 3 "<GUID>.mdb" files (current year and history max 2 years).
- Systemidentity.mdb (mapping GUIDs of roles and their names).

3.6.9.3 Method of acquisition

Live and offline. When capturing live, beware of possible database inconsistencies, we recommend capturing the entire folder in which the DB is located.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA | VÝSKUMNÁ AGENTÚRA

3.6.9.4 Primary category

User Activity

3.6.9.5 Other categories

-

3.6.10 Windows Index Search (Windows.edb)

3.6.10.1 Description

The Windows Search database provides an index of Windows Search features to increase search speed by indexing content. The Windows Search index is used to search through the Windows taskbar, Windows Explorer, and some Universal Windows Platform (UWP) apps (such as Outlook, OneDrive, etc.).

By default, only a subset of folders and files are indexed (to reduce Windows Search database size and CPU usage). You can find the scanned folders and the number of indexed items in the "Windows Search Settings" menu.

From Windows Vista / Windows Server 2008 to Windows 10 / Windows Server 2019, Windows Search used the Extensible Storage Engine (ESE) database (Windows.edb). Starting with Windows 11 / Windows Server 2022, Windows Search switched to two SQLite databases (Windows.db and Windows-gather.db).

By default, only items from the following sources are scanned and indexed:

- Files and folders from the %USERPROFILE% folders (excluding the AppData directories) and C:\ProgramData\Microsoft\Windows\Start Menu\Programs\* (which include startup LNK files).
    - Available data: file name, path, size, attributes, MAC timestamps. In the case of a small file, part of the file's content may also be indexed.
- Outlook mail data (with the time stamp of receipt, possible mail content).
- The name of your OneNote notes.
- Internet Explorer history (URL, timestamp of the last visit).

3.6.10.2 Location

From Windows 11:
%SYSTEMDRIVE%\ProgramData\Microsoft\Search\Data\Applications\Windows\Windows.db

%SYSTEMDRIVE%\ProgramData\Microsoft\Search\Data\Applications\Windows\Windows-gather.db

Windows 7 to Windows 10:
%SYSTEMDRIVE%\ProgramData\Microsoft\Search\Data\Applications\Windows\Windows.edb

Windows XP:
%SYSTEMDRIVE%\Documents and Settings\All user\Application Data\Microsoft\Search\Data\Application\Windows\Windows.edb

3.6.10.3 Method of acquisition

Live and offline.

3.6.10.4 Primary category

User Activity

3.6.10.5 Other categories

File Presence

3.6.11 INetCache/CryptnetUrlCache

3.6.11.1 Description

The browser stores cached browsing data here. It pairs between web browser artifacts. In computer forensics, **INETCache** and **CryptnetUrlCache** are key artifacts in the context of browsing and internet activity. These artifacts can provide important information regarding a user's use of the web and internet, including website visits, cached data, and even encrypted communication activities. Below is a breakdown of the importance of each artifact, its content, and how an analyst can interpret the data.

*3.6.11.2 **INETCache (Internet Cache)***

INETCache caches copies of web pages, images, scripts, and other resources that the browser downloads from the Internet. This cache is created to speed up web browsing by storing copies of frequently visited web pages. When a user visits a website, instead of re-downloading all the elements, their browser can load them from the local cache, reducing the loading time.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA | VÝSKUMNÁ AGENTÚRA

3.6.11.2.1 Location

%USERPROFILE%\AppData\Local\Microsoft\Windows\INetCache
On different versions of Windows, the path may be slightly different.
Browser-specific locations: Internet Explorer and Microsoft Edge use **INETCache** , and other browsers, such as Chrome and Firefox, can cache data in their own cache directories.

3.6.11.2.2 Contents

HTML Pages: Cached versions of the web pages visited (including HTML content).
Images, Scripts, and Media Files: Common media files such as images (JPG, PNG, GIF) or other assets that are downloaded during a visit.
Web Forms and Cookies: Information that was part of a web form or session data was sometimes used to remember login statuses or preferences.
URLs: Cached URLs and other browsing data (i.e., history or metadata about pages visited).

3.6.11.2.3 Significance for Forensic Analysts:

- **Investigating user activity**: Analysts can determine which websites a user has visited, even if the history has been cleared from the browser. They can retrieve data about the pages visited and the specific elements that have been loaded.
- **Browser fingerprinting**: Cached files can give analysts an idea of the user's browser and device usage patterns.
- **Timestamps**: Analysts can restore timestamps about when websites were visited and which specific resources were accessed.
- **Proof of access**: Even if a user tries to clear their browsing history, the cached data may still be available and may provide a trace of their internet activity.

*3.6.11.3 **CryptnetUrlCache***

CryptnetUrlCache is a cache specifically related to **cryptographic operations** in Microsoft Windows and stores information associated with URLs that is used to access certificate-related data from the Internet. This artifact is primarily used in relation to secure internet activities that include certificates such as SSL/TLS connections, digital certificates, or secure websites.

3.6.11.3.1 Location

It is usually located in the following location:
%USERPROFILE%\AppData\Local\Microsoft\CryptnetUrlCache

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.6.11.3.2 Contents

- **Certificate Authority URLs:** The cached CAs URLs that were used to verify the authenticity of SSL/TLS certificates.
- **SSL/TLS metadata**: Cached data related to secure communication or obtaining public keys and certificates over HTTPS.
- **Certificate Revocation Lists (CRLs):** Information about whether a certificate has been revoked or whether it is still valid.
- **SSL Session Data**: Information about active or recently used SSL/TLS sessions, including, where applicable, encrypted communications.

3.6.11.3.3 Importance for Forensic Analysts

- **Investigating secure communications**: Analysts can analyze CryptnetUrlCache to determine which secure sites (HTTPS) the user has accessed and track the URLs involved in certificate validation.
- **Cryptographic proof**: Cryptographic URLs and certificates used to verify secure web connections can provide proof of the authenticity of a website or secure transaction.
- **Data correlation**: If a user has visited certain secure sites (such as banking or email services), this artifact can help confirm web activity and provide additional evidence of those visits.
- **Revealing sensitive data**: While this cache mainly stores cryptographic information, it can be a gateway to understanding which secure connections the user initiated, potentially revealing sensitive interactions that were otherwise encrypted.

3.6.11.3.4 Info an analyst can get

- **User Browsing History**: The INETCache artifact provides an overview of the websites the user has visited, and CryptnetUrlCache offers secure site information (HTTPS).
- **Site access timestamps**: Analysts can extract timestamps that detail when specific resources or sites were accessed, which is useful for creating a timeline of activity.
- **URLs and sources**: Extracting URLs and related resources can help the analyst understand what content the user has requested, even if the original website is no longer accessible.
- **Secure site visits**: CryptnetUrlCache detects secure connections that the user has made, such as those involving online banking or email, which can be crucial when investigating where encrypted traffic is involved.
- **Session information**: Both caches can contain session information to help analysts understand user activity patterns and potential data exfiltration.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.6.11.3.5 Key considerations

- **Data volatility**: Both caches can be volatile; they can be overwritten as the user continues browsing, especially if cache size limits are reached or if browser settings are configured to automatically clear the cache.
- **Cleaning potential**: Users may try to clear the cache or use privacy-focused browsers. However, forensic analysts may still be able to recover partially deleted or overwritten files using specialized recovery tools.
- **Correlation with other artifacts**: INETCache and CryptnetUrlCache artifacts often need to be correlated with other artifacts, such as browser history, event logs, or network logs, to render a complete picture of user actions.

In summary, **INETCache** and **CryptnetUrlCache** are valuable for forensic analysts investigating web browsing and secure internet activity. By examining these artifacts, analysts can identify visited sites, track secure communications, and potentially uncover key evidence related to a user's online behavior, even in the face of attempts to delete or obscure this information.

While **INETCache** in user directories (such as %USERPROFILE%\AppData\Local\Microsoft\Windows\INetCache) typically contains cached content related to the user's browsing activity, there is also **a system-level INETCache** that can be found in the **System32** directory, especially in relation to certain system processes and Windows components.

This system-level cache is associated with **Internet Explorer (IE)** or **Microsoft Edge** because these browsers and their system-level processes also cache data in system-managed locations.

3.6.11.4 *INETCache v System32*

INETCache at **the system level** can be found in:

- %SYSTEMROOT%\System32\inetsrv\INETCache

This location contains cached files that are used **by Internet Information Services (IIS)** and other Microsoft components, such as **Internet Explorer** or **Microsoft Edge,** when interacting with certain server-side features or system-level Internet operations.

3.6.11.4.1 Contents

- **System-level caching**: Cached resources related to Windows processes and system operations that involve Internet data.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

- **Server-related data**: When Windows communicates with Web sites or Web services through **Internet Information Services (IIS), it can cache files in that location.**
- **Content from updates/services**: System updates or services based on web data may also cache files in this folder.
- **Temporary Internet Files**: Files such as HTML pages, images, scripts, and other resources can be stored for faster access when the system interacts with certain web services.

3.6.11.4.2 Importance in Forensic Analysis

- **System-wide Internet Activities**: An analyst can examine this **INETCache** to examine system-level Internet activities, such as interactions with Windows Update, connections to remote servers, or system web traffic that may not appear in the user's browsing history.
- **Hidden or background activity**: Some system processes that interact with web services for software updates, certificates, or communication with other systems may cache data in this location. Even if the user does not have direct access to certain websites, this cache may contain evidence of indirect interactions.
- **Web service communication detection**: Forensic analysts can discover cached data related to corporate or system web services, which may include server communications or connection details.
- **Cross-references with other system artifacts**: Combining the data found in **System32\INETCache** with other system logs (such as event logs or network traffic) could reveal a wider range of the system's Internet activity that may not be immediately apparent.

3.6.11.4.3 Key considerations for analysts

- **System-level operations**: This cache is more likely to contain data related to system operations rather than user-specific browsing history. If the investigation is focused on user activity, this cache may not be as relevant as a user-specific cache, but it can still offer insight into system processes interacting with the internet.
- **Data persistence**: **The INETCache** in the **System32** directory may be more persistent than user-level cache data, depending on system settings or activity. However, it can still be deleted or overwritten by Windows maintenance processes.
- **Difference from User Cache**: An analyst must distinguish between user-level browsing data (located in **AppData**) and system-level cache (located in **System32**) as they provide different insights into system and user activities.

While most users are familiar with **INETCache** located in their user profile directories, **system-level INETCache,** located in the **System32 directory,** can provide valuable information about system-level interactions with web services, background processes, and updates. Forensic analysts should not overlook this location, especially when investigating system activities or looking for evidence related to Windows services, updates, or remote interactions with the server.

3.6.11.5 Summary

3.6.11.5.1 Location

%SYSTEMROOT%\System32\Config\SystemProfile\AppData\Local\Microsoft\Windows\INet Cache\IE\
%SYSTEMROOT%\System32\inetsrv\INETCache
%USERPROFILE%\AppData\Local\Microsoft\Windows\INetCache
%USERPROFILE%\AppData\Local\Microsoft\CryptnetUrlCache

3.6.11.5.2 Method of acquisition

Live and offline

3.6.11.5.3 Primary category

User Activity

3.6.11.5.4 Other categories

-

3.6.127-Zip Folder History

3.6.12.1 Description

A registry key containing a list of archive files that have been accessed using 7-Zip.

3.6.12.2 Location

Captured paths in registers:
HKEY_USERS*\Software\7-Zip\FM\FolderHistory

3.6.12.3 Method of acquisition

Live and offline

3.6.12.4 Primary category

User Activity

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.6.12.5 Other categories

Presence of a file, execution/presence of an executable file

3.7Other artifacts, falling into none or more categories

3.7.1 Event Logs

3.7.1.1 Description

A complete list of events found in Windows Event Log (.evtx) files.

3.7.1.2 Selected logs and events important for forensic analysis

- **Windows Security Event Log – Logon/Logoff** – The most important resource. With login audit enabled, it includes:
  - Event **4624** – Successful user login. Key fields: **Account Name**, **Logon Type** (2 = interactive console, 10 = remote/RDP, 3 = network e.g. SMB), **Source IP** (for network/RDP login).
  - Event **4625** – Failed login attempt (stating the reason – wrong password, locked account, etc.).
  - Event **4634** – Logout (Logon Type specified).
  - Event **4647** – The user initiated a logout (typically when logging out via the Start menu).
  - Event **4648** - A user successfully logged in to a computer using explicit credentials, even though they were already logged in as a different user.
  - Event **4672** – Login with assigned privileged rights (administrator always has, indicates privileged logon).
  - Event **4778/4779** – Reconnect to disconnected session / disconnect session (for RDP). These events create a chronology of login activity, including local logins, UAC elevations (type 11), network accesses (type 3 when accessing a shared folder, etc.).
- User creation – **Security.evtx**
  - Event **4720** - "A new account has been created" – contains the name and SID of the created account and the user who created the account

- **Windows System Event Log – Start/Stop PC** – Event **6005** (The Event log service was started) and **6006** (was stopped) in the System log indicate the boot and shutdown time. Also **6008** if the previous shutdown was unexpected. These help you find out in the timeline when interactive logins were possible.
- **User Profile – Times** – In the HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\<SID> registry, there is a **ProfileLoadTimeLow/High** value that can indicate when the profile was last loaded (i.e., when the user last logged

in). Also in NTUSER. The DAT of each profile LastWrite time of some keys (e.g. HKCU\Environment) is updated at the logo.

- **Other authentications** – If it is a membership in the domain: Event **4768/4769** in the Security log (they are not visible on the local WS, these are domain controller Kerberos events). But e.g. if the computer received a login through the domain in offline mode, the record is normally 4624 with Domain.
- **Wake-up from Lock/Unlock** – Event **4800** (workstation locked) and **4801** (unlocked) in the Security log, associated with the user ID. These can illustrate the user's activity within the login.
- **Windows RDP – Connection Events** – RDP (Remote Desktop Protocol) leaves multiple event logs:
    - **Security log: Event ID** 4624 (Logon Type 10 = RemoteInteractive) **is logged on a successful RDP login** – it contains the client's login name and IP address. On an unsuccessful attempt, Event **4625** (with Logon Type 10). An RDP session disconnection can be recorded as **4634/4647** and a special RDP reconnect as 4778/4779.
    - **Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational**: Event **1149** – indicates an authentication attempt via RDP (user name and client IP address). (1149 is logged when trying to log in – both successful and unsuccessful).
    - **Microsoft-Windows-TerminalServices-LocalSessionManager/Operational**: Event **21** (Session Login), **22** (Logout), **24** (Disconnect Session from Client), **25** (Session Reconnect) - Also contain user and IP.
    - **Microsoft-Windows-RDPCoreTS/Operational**: Event **131** – successful establishment of a TCP connection to the RDP service (states the IP address of the client), Event **98** – successful completion of the TLS handshak of the RDP connectionTogether, these logs provide a time track of who and from where remotely logged in via RDP.
- Process Creation – Security.evtx
    - **Process Tracking** – If process auditing is enabled, **the Security log** logs Event ID **4688** when each process is created (older OS ID 592). Log 4688 contains the process name, PID, user account, etc. (requires *Audit Process Creation to be turned on*).
- Installation of the service –System.evtx, Security.evtx
    - **The system log** registers event ID **7035** when the service starts or stops (Service Control Manager) – when the service starts, it can indicate the start of the corresponding exe as a service.
    - **System log** also records event ID **7045**, "A new service was installed in the system", is also used to track service installations, but may not always translate the user SID into a username.
    - Event ID **4697**, this time in **Security.evtx**, "A service was installed in the system", captures the creation of new services and provides details about the service and the user who installed it.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A | VÝSKUMNÁ AGENTÚRA

3.7.1.3 Location

%systemroot%\System32\winevt\Logs\*

3.7.1.4 Method of acquisition

Live and offline

3.7.1.5 Primary category

All

3.7.2 Master File Table

3.7.2.1 Description

Master File Table (MFT) is a database used by the NTFS file system to store file metadata, including timestamps, file name, and file size. This is the main table of the NTFS file system. It is a special hidden file in which each file or folder on the disk has one record (1024 B by default). The MFT record contains metadata: file name, physical location of the data, size, timestamps in 2 sets of 4 (created, modified, last accessed, etc.) and attribute information. Forensically, a complete list of files can be obtained from $MFT, including deleted files (if they have not already been overwritten, they are also present but marked as free) and timestamps that help determine when a file was created or modified.

3.7.2.2 Location

C:\$MFT, not just on the system disk, but on any mounted disk with the NTFS file system.

3.7.2.3 Method of acquisition

Both live and offline, although it is not accessible to the average user on a running system.

3.7.2.4 Primary category

File Presence

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.7.3 USN Journal

3.7.3.1 Description

Update Sequence Number Journal is an NTFS journal that logs all changes to files on the disk (creation, modification, deletion) along with timestamps. Since Windows Vista, USN Journal has been turned on by default and acts as a constantly scrolling log. The records contain the file ID (MFT reference), the type of change (flags – create, delete, rename, etc.) and the time in UTC. For forensic analysis, it is a key resource for the timeline of file system events – for example, proving that a certain file was deleted at a specific time.

3.7.3.2 Location

C:\$Extend\$USNJrnl:$J, not just on the system disk, but on any mounted NTFS file system disk.

3.7.3.3 Method of acquisition

Live and offline

3.7.3.4 Primary category

File presence

3.7.3.5 Other categories

Operations with files, as a rule, we focus on deleting them

3.7.4 Recycle Bin

3.7.4.1 Description

A folder used as temporary storage for deleted files before permanent deletion. When a file is deleted (if SHIFT+Delete is not used), system moves the file to the Recycle Bin. In the user's respectable SID folder, the \$I... and \$R...: $I contains metadata about the original name and path of the deleted file and the time of deletion, $R the contents of the file itself. Analysis of the Recycle Bin reveals what files the user has deleted and when (even if they have not emptied them completely).

3.7.4.2 Location

C:\$Recycle.Bin***

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.7.4.3 *Method of acquisition*

Live and offline

3.7.4.4 *Primary category*

User Activity

3.7.4.5 *Other categories*

File Presence, Deletion, Interaction

3.7.5 RAM

3.7.5.1 *Description*

In forensic analysis, RAM (Random Access Memory) is a critical source of volatile evidence. It can only be captured on a running system (although on virtual machines it is also written to a file on the hypervisor's disk, which simplifies the acquisition), for example through live review of memory vontent using tools such as Volatility or Rekall, or by creating a memory image using WinPMEM, DumpIt or FTK Imager.

RAM reveals real-time system states that are invisible on persistent storage. From the point of view of incident investigation, it can be a critical source of information about fileless malware, rootkits and other advanced threats.

RAM generally contains:

- Running processes and injected code
- Open network connections and ports
- Encryption keys
- Passwords/logins in plain text
- Data from the clipboard
- Chat messages
- Unsaved documents
- Traces of user activity
- Hidden malware artifacts

This allows for the reconstruction of the attacker's actions, the detection of transient threats, and the decryption of encrypted files.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

3.7.5.2 Method of acquisition

Live

3.7.5.3 Primary category

Defacto all of them

Financované Európskou úniou NextGenerationEU PLÁN OBNOVY ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA VÝSKUMNÁ AGENTÚRA

4 Task KPB2.2

4.1Assignment of the KPB2.2 task

Task KPB2.2 – **Identification of a suitable method for digital evidence collection** in running and switched off state for individual sources of artifacts. Simultaneously, the research team will focus in this task on the securing of digital evidence from multiple devices and sources (IstroSec).

KPB2.2 task – Identification of a suitable method of collecting digital traces on and from a device in a switched off state for individual artifact sources. At the same time, the research team in this role will focus on securing digital evidence from multiple devices and sources (IstroSec).

4.2Methodology for solving the KPB2.2 problem

After defining the artifacts, which we will collect in this phase of the project, we will focus on briefly describing the method of their extraction from the running and shutdown system.
When securing data from a running system, it is essential to ensure that the activity alters the system as little as possible. Standard activities that a technician usually cannot avoid include:

- Sign in or unlock the device by entering a password or by using other authentication method;
- Connecting a USB stick with ready-made tools for securing digital traces;
- Connecting a device (USB stick or external drive) to store the evidence obtained;
- Starting the process to secure digital evidence – the number and type varies depending on the volume and type of evidence being secured.

When securing digital evidence from a switched off device, the following aspects must be taken into account and the methodology of securing the digital evidence should be adapted based on them:

- What type of device carries the evidence?
    - server, workstation (desktop/tower), laptop, virtual server or virtual workstation, flash drive, external hard drive, ...
- What type of disk does the device contain?
    - HDD, SSD, various types of interfaces (SATA, NVMe, SCSI, USB, ...)

The first step, even before securing the data before processing by the tool, will be to create a bitwise copy of the source device. Although this is not the subject of the project, we considered it useful to mention these facts for the sake of completeness.

Before defining the requirements for the instrument, it is necessary to clarify what type of input we assume. For the purposes of this document, let's assume that a forensic analysis has

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

on its input an evidence in a form of a switched off device – more precisely, its forensic image in the E01, .raw or .dd format.

4.3Procedure for securing individual types of digital traces

4.3.1 Securing registers: live

In the context of computer forensic analysis, the collection of registers from **a running system** (live acquisition) is a sensitive process. System hives (like HKLM) are locked, so we can't just copy them. The safest and most used way is to use the built-in **reg save command**, which creates a binary copy of hive without changing the original.

This approach is recommended in forensic best practices (e.g., SANS) because it minimizes footprint and preserves integrity.

1. Preparation

- Run **the Command Prompt as Administrator** (right click on the Run as administrator cmd.exe →). Since it is impossible for a regular user account to access the registry, it is necessary to have access to the account of the local administrator of the device. This comes with some risks, especially if the attacker is still active on the device.
- Prepare a destination folder on external media (e.g. a USB drive), e.g. E:\Forensic\Registry\.
- After collection, calculate **the hash values** (e.g., MD5, SHA1, and SHA-256) of the exported files for the chain of custody.
- Document the time, commands, and hash values of secured registers (MD5, SHA1, SHA256).

2. Export HKLM (HKEY_LOCAL_MACHINE)
HKLM is made up of multiple sub-hives (SYSTEM, SOFTWARE, SAM, SECURITY, etc.). Export them individually:

```
reg save HKLM\SYSTEM E:\Forensic\Registry\SYSTEM.hive
reg save HKLM\SOFTWARE E:\Forensic\Registry\SOFTWARE.hive
reg save HKLM\SAM E:\Forensic\Registry\SAM.hive
reg save HKLM\SECURITY E:\Forensic\Registry\SECURITY.hive
```

- The command creates a binary .hive file (not .reg text export!).
- There is no direct way to export HKLM in its entirety: the sub-hives described cover most of the forensic artifacts and correspond to the hive support files located at %systemroot%\System32\config.

3. Export user hives (NTUSER. DAT and UsrClass.DAT for all users)
These are stored in user profiles and can be locked if the user is logged in.

For the currently logged in user (HKCU):

```
reg save HKCU E:\Forensic\Registry\HKCU_current.hive
```

For all users (even those who are not logged in):

- First, we identify where the profiles are located: dir C:\Users /b
- For each user, copy the physical files (if they are not locked):

```
copy "C:\Users\<username>\NTUSER. DAT" E:\Forensic\Registry\<username>_NTUSER. DAT
copy "C:\Users\<username>\AppData\Local\Microsoft\Windows\UsrClass.dat"
E:\Forensic\Registry\<username>_UsrClass.dat
```

- If they are locked (user logged in), use Volume Shadow Copy (VSS) or a tool like FTK Imager (see below).

Script automation (PowerShell as Admin):

```
PowerShell
$dest = "E:\Forensic\Registry"
New-Item -ItemType Directory -Path $dest -Force

# HKLM sub-hives
reg save HKLM\SYSTEM "$dest\SYSTEM.hive"
reg save HKLM\SOFTWARE "$dest\SOFTWARE.hive"
reg save HKLM\SAM "$dest\SAM.hive"
reg save HKLM\SECURITY "$dest\SECURITY.hive"

# All NTUSER. DAT and UsrClass.dat
Get-ChildItem C:\Users -Directory | ForEach-Object {
    $user = $_. Name
    $ntuser = "C:\Users\$user\NTUSER. DAT"
    $usrclass = "C:\Users\$user\AppData\Local\Microsoft\Windows\UsrClass.dat"

    if (Test-Path $ntuser) { copy $ntuser "$dest\$user`_NTUSER. DAT" }
    if (Test-Path $usrclass) { copy $usrclass "$dest\$user`_UsrClass.dat" }
}
```

4. Alternative methods for locked files – also applicable to artifacts described in the following subchapters:

- **Volume Shadow Copy Service (VSS):** Create a shadow copy and copy from it (a tool like vssadmin or DiskShadow).
- Forensic tools:
    - **FTK Imager** (free): Run the program → Create Disk Image → Choose "Obtain Protected Files" → Choose registry hives as needed.
    - **ProDiscover** or **EnCase**: Support for live acquisition.
    - **Belkasoft X** or **Magnet AXIOM:** Automated collection of registers from a live system.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA | VÝSKUMNÁ AGENTÚRA

- **Memory Dump + Volatility:** If you need volatile data, capture RAM (FTK Imager) and extract hives using Volatility (plugin hivelist, dumpfiles).

5. Important notices

- **Minimize changes:** Live collection always changes the system, so offline analysis is preferred when possible.
- **Integrity:** Always hash (Get-FileHash in PowerShell) and document.
- **Legal aspects:** Obtain consent or court order.
- After collecting data, analyze offline.

4.3.2 Securing registers: offline

- If a disk image has not yet been created, we start by creating an image copy!
- Export registers from an image – e.g.
    - mount images as Read-only and from the admin cmdline the analyst can also copy them
    - mount image and directly start parsing
    - It is essential not to neglect the cleaning of the registers: to combine the hive support files themselves with their transaction logs, as this will ensure that all changes are reflected, even those that have not yet been written to the support files on the dis.
- The tool must therefore allow copying the register from the evidence source, unifying transaction registers (for example, using E. Zimmerman's Registry Explorer tool or the cmdline tool rla.exe).
- After exporting, the files must be hashed (SHA1, SHA265, MD5).
- During the process, the tool should record what operation took place at what time and with what result. Thus, the analyst will know if, for any reason, one of the registers fails to be exported or unified with the transaction log (for example, if the transaction log or the register itself has been encrypted by ransomware).

4.3.3 Securing system files: live

The following article concerns securing files that are hidden or locked on the running system:

- $MFT, $USN:$J, $Recycle.Bin
- SRUM. DB, LNK files, JumpLists, PCA, ...

There are several ways and ready-made tools to get to similar artifacts.
There are a number of freely available tools available (such as those by Eric Zimmerman) that allow data extraction even from a running device. The primary method involves tzv. raw disk

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

access, or the Volume Shadow Copy Service (VSS). Both can be used to bypass restrictions - file locks. Of course, it is still true that access with local administrator privileges is required.

If we are interested in extracting artifacts for further processing, we can copy them from the system from the GUI of the FTK Imager tool (select a file, rightclick → Export files), or using the Kape tool. The details of these tools are beyond the scope of this document.

Similar to the previous cases, we will always create a hash of the secured files.

4.3.4 System file acquisition: offline

In offline scenario, we have a simplified task in the sense that there is no risk of altering the trace by extracting individual artifacts from it.

4.3.5 Securing Event logs: live

Since Event logs are not accessible to a regular user, we need permissions at the device administrator level. The logs can then be exported from the admin command line or PowerShell.

```
Copy-Item C:\Windows\System32\winevt\Logs\* -Destination D:\Work\Logs\
```

4.3.6 Securing Event logs: offline

From the connected and mounted disk, we can easily copy the files with Event logs. An alternative is to parse and process them directly from the location on the disk image, which, as we have already mentioned, we always mount in read-only mode.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

5 Task KPB2.3

5.1Assignment of the KPB2.3 task

Task KPB2.3 – **Verification of the suitability of the method for ensuring the integrity of digital evidence** – in this task we will focus on securing the integrity of digital evidence and possible detection of the necessary violation of digital evidence (IstroSec).

Within ADFIR, evidence is not static; it is a dynamic flow of data collected from live systems, often in real time, which is then parsed, normalized and transformed into structured formats (CSV, databases) for machine learning algorithms. This shift requires a rethinking of integrity concepts. We are no longer only interested in the physical integrity of the hard disk sectors or the fingerprint of the entire disk image; We need to verify the information integrity of abstract data structures. For example, if a project uses Formal Conceptual Analysis (FCA) to infer relationships between artifacts, the integrity mechanism must ensure that the generated lattice structures accurately reflect the underlying binary reality without introducing errors caused by the extraction process itself.

5.2Methodology for solving the KPB2.3 task

In this chapter, we will introduce the principles and standards relevant to solving the problem.

5.2.1 Daubert Standard and Scientific Validity

Although the ADFIR project is implemented in Slovakia/the EU, the principles of the Daubert Standard (used in the USA but globally influential in digital forensics) are relevant for scientific validity. The standard asks:

- Has the technique been tested?
- Has it been peer reviewed?
- Is the known or potential error rate acceptable?
- Are there standards to control the functioning of the technique?

For the ADFIR project, this means that any proprietary or "black box" integrity method is considered inappropriate. The project must use open, standardized algorithms (such as SHA-256 or BLAKE3), where error rates (collision probabilities) are mathematically proven and peer-reviewed.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

5.2.2 ISO/IEC 27037: Guidelines for Identification, Collection, Acquisition and Storage

This international standard provides a reference framework for the handling of digital evidence. It defines "integrity" as the property that ensures the accuracy and completeness of assets.

- **Relevance for ADFIR:** ISO 27037 explicitly states that a live acquisition involves changing digital evidence. It states that such actions are acceptable if they are necessary, proportionate and documented.
- **Verification criteria:** A method validated in this report is considered "appropriate" only if it generates the necessary metadata to meet the documentation requirements of ISO 27037. This includes recording the tool used, execution time, and user context.

5.2.3 Hash functions

The basic principle of digital evidence verification lies in cryptography. However, the ADFIR project's emphasis on automation and machine learning changes the criteria for selecting cryptographic primitives. We need to balance collision resistance (safety) with throughput (performance) and parallelizability (scalability).

The hash function maps data of any size to a fixed-size bit string. To be eligible for verification purposes in forensic analysis, a function must be:

1. **Deterministic:** The same input always leads to the same output.
2. **Pre-image Resistant:** It is impracticable to generate a report that provides a specific hash value.
3. **Collision Resistant:** It is impossible to find two different messages that have the same hash value.

MD5:

- MD5 produces a 128-bit digest. It is cryptographically cracked. Collision attacks can generate two different files with the same MD5 hash on modern hardware.
- Despite the breakthrough, MD5 remains the fastest of the traditional algorithms. In the context of "identifying relevant digital traces" (e.g., for the initial sorting or deduplication of massive, non-critical datasets, such as millions of common system files), MD5 offers a performance advantage.
- **However, it is not suitable** for primary integrity verification.

SHA256:

- SHA-256 produces a 256-bit digest. It is a current industry standard (e.g. NIST approved). There are no practical collision attacks.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

- Used for final evidence. In order to ensure 'the admissibility of the results ... in criminal proceedings", the final fingerprint of any evidence collected must be verified using at least SHA-256.

5.3Intrusion verification on Windows operating systems

Windows is the primary target of the ADFIR project. When an automated collection tool (developed by IstroSec) runs on a compromised Windows host, it affects specific artifacts.

System Memory and Prefetch

- **Mechanism:** When the acquisition process is running, the Windows kernel memory manager allocates pages in RAM. Windows Prefetcher (%SYSTEMROOT%\Prefetch) creates or updates a .pf file for this tool, recording, among other things, the time of its execution and the path to the tool's executable.

- **Integrity impact:** This overwrites older memory pages (potential evidence) and modifies the file system (creating a .pf file).

- **Validation of suitability:** An appropriate collection method must explicitly filter out memory pages containing the tool code and the tool-generated Prefetch. This "investigator subtraction" technique restores the logical integrity of the timeline.

Artefacts in registry (Shimcache / Amcache)

- **Mechanism:** Running the tool can activate the Application Compatibility Engine (Shimcache), updating the SYSTEM registry branch.

- **Integrity impact:** This changes the LastWriteTime of registry keys, which can potentially obscure the timeline of malware execution by an attacker if it occurred in the near term.

- **Mitigation:** The automated tool must ask for the current system time *before* any other action and log it. This will establish a "Time Zero" for the investigation. Any artifact with a timestamp *after* Time Zero is considered potentially generated by an investigator. It is important to note that you cannot rely on the time set on the device. We have to record the mentioned "time zero" as we see it on the device, but at the same time record the time from another, independent source (online clock, analyst's smartphone/smartwatch, etc.). If we find a deviation, we will take it into account, for example, if it is necessary to correlate the analysis outputs with the outputs of the analysis of other traces (endpoints, network devices, EDR, etc.).

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

Event Logs

- **Mechanism:** As the collection tool installs the kernel driver, Service Control Manager logs Event ID 7045 (Service Installed) in the system log.

- **Suitability:** This is actually *a beneficial disruption*. It creates an indelible audit trail of the acquisition. Verification of suitability requires that the tool *does not attempt* to delete or suppress this log, as this would constitute spoliation of evidence.

5.4Write-blocking mechanisms

Preventing the modification of evidence during an acquisition is a fundamental rule of forensic analysis. Software or hardware write blockers are used for this purpose. They are used to prevent unwanted writing to the secured media by ensuring that it is connected only in read mode.

5.4.1 Hardware Write Blockers (HWBs)

- **Mechanism:** Physical bridges that physically interrupt the "Write" command signal on the interface (SATA/USB).
- **Suitability:**
    - **Post-mortem:** Necessary for the analysis of secured hard disks (disk devices of end devices).
    - **Automation/Live Forensic Analysis:** Completely **inappropriate**. You can't plug a hardware write blocker into the RAM of a running server or remote cloud instance.

5.4.2 Software Write Blocker (SWB)

- **Mechanism:** Operating system configurations that instruct the kernel to reject write requests for a specific device.
- **Windows implementation:** HKLM\SYSTEM\CurrentControlSet\Control\StorageDevicePolicies\WriteProtect registry key.
- **Verification of suitability:**
    - **Risk:** Software blocking is potentially fallible. A kernel-level rootkit or a faulty driver can bypass it.
    - **ADFIR Verdict:** For "Digital Trace Collection" in live scenarios, software write blocking is **accepted as appropriate**, provided it is associated with:
        1. **Hash verification:** Hashing before and after analysis (if the device is static).

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VAIA | VÝSKUMNÁ AGENTÚRA

2. **Audit logs:** Logging of each executed order.

6 Task KPB2.4

6.1Assignment of the KPB2.4 task

Task KPB2.4 – **Creation of a tool for experimental verification of digital evidence securement** – creation of a tool for experimental verification of secured digital evidence (IstroSec).

This chapter defines the methodological framework for the automated acquisition of digital evidence using the "Athena" tool, which was developed by the partner IstroSec for the needs of this research.

6.2The concept of targeted acquisition

In the context of large-scale infrastructures, it is not always possible or efficient to create complete disk images for every suspicious device. Therefore, Athena implements the methodology of "forensic triage" and targeted acquisition, which IstroSec has been using for a long time in the Incident Response – IR - engagements. This approach focuses on the rapid identification and extraction of those artifacts that have the highest informative value for confirming or refuting the hypothesis of compromise.

The Athena methodology divides the collection process into two phases:

1. **Collection Phase:** Raw data is extracted from target systems and stored in a structured SQL database. This step ensures the persistence and integrity of the original values before any analytical transformation.
2. **Processing Phase:** Specific SQL scripts are run over the raw data, which perform cleaning, normalization, de-duplication and formatting of the data into output CSV files.

6.3SQL Database as a Central Forensic Repository

The decision to use a relational database (Microsoft SQL Server) as a central repository is supported by several technical and forensic, analytical and research benefits.

First, the relational SQL model allows you to maintain logical links between entities that would disappear if you exported directly to CSV. For example, the relationship between a parent's process and his offspring, or the relationship between a network connection and the process that initiated it, is naturally representable by foreign keys.

Financované Európskou úniou NextGenerationEU | PLÁN OBNOVY | ÚRAD PODPREDSEDU VLÁDY SLOVENSKEJ REPUBLIKY PRE PLÁN OBNOVY A ZNALOSTNÚ EKONOMIKU | VA/A VÝSKUMNÁ A INOVAČNÁ AUTORITA | VÝSKUMNÁ AGENTÚRA

Second, SQL databases provide a robust mechanism for ensuring data integrity through transactional processing (ACID properties) and journaling. This is crucial for demonstrating that the data has not been corrupted or modified by a write error during the collection process.

6.4Collection agent

The use of a dedicated agent is essential for forensic collection on the Windows platform primarily to maintain the integrity of evidence and the ability to retrieve locked system data. Unlike "agentless" servers (PowerShell/WinRM), which significantly change the contents of RAM, overwrite logs and leave a massive footprint by running scripts, an optimized agent operates with minimal impact on the system. A key advantage of the agent is the ability to access the physical disk directly, thanks to which it can extract even files that the operating system keeps locked for reading (e.g. $MFT, SAM/SYSTEM registries) without the need to rely on limited Windows APIs or create shadow copies.

From an operational point of view, the agent provides stability and scalability that scripting tools cannot guarantee. While WinRM sessions are prone to network outages and can overwhelm the infrastructure during bulk collection, the agent can collect data locally, buffer and send it asynchronously, and even with a stable connection. Moreover, a digitally signed binary agent is generally a legitimate part of a security policy. On the other hand, complex PowerShell scripts, necessary for deep exploration, are often blocked as suspicious activity by modern EDR/AV systems. Such a blockage can make it impossible to investigate at the most inopportune time of crisis, which cybersecurity incidents often are.

For the needs of the agent, we have defined the requirements (Tab. 4).

Parameter	Requirement	Justification for ADFIR
Minimal impact	The agent must run with minimal CPU/RAM consumption.	Prevention of disruption of production systems and overwriting of volatile artifacts in memory.
Integrita	Utilization of cryptographic hashes (SHA-256) on each record.	Securing the Chain of Custody for legal purposes.
Scalability	Ability to process 100,000+ records (e.g. MFT) in real-time.	Forensic analysis of modern disks with millions of files requires a robust backend (SQL).

Tab. 4 – Requirements for agent