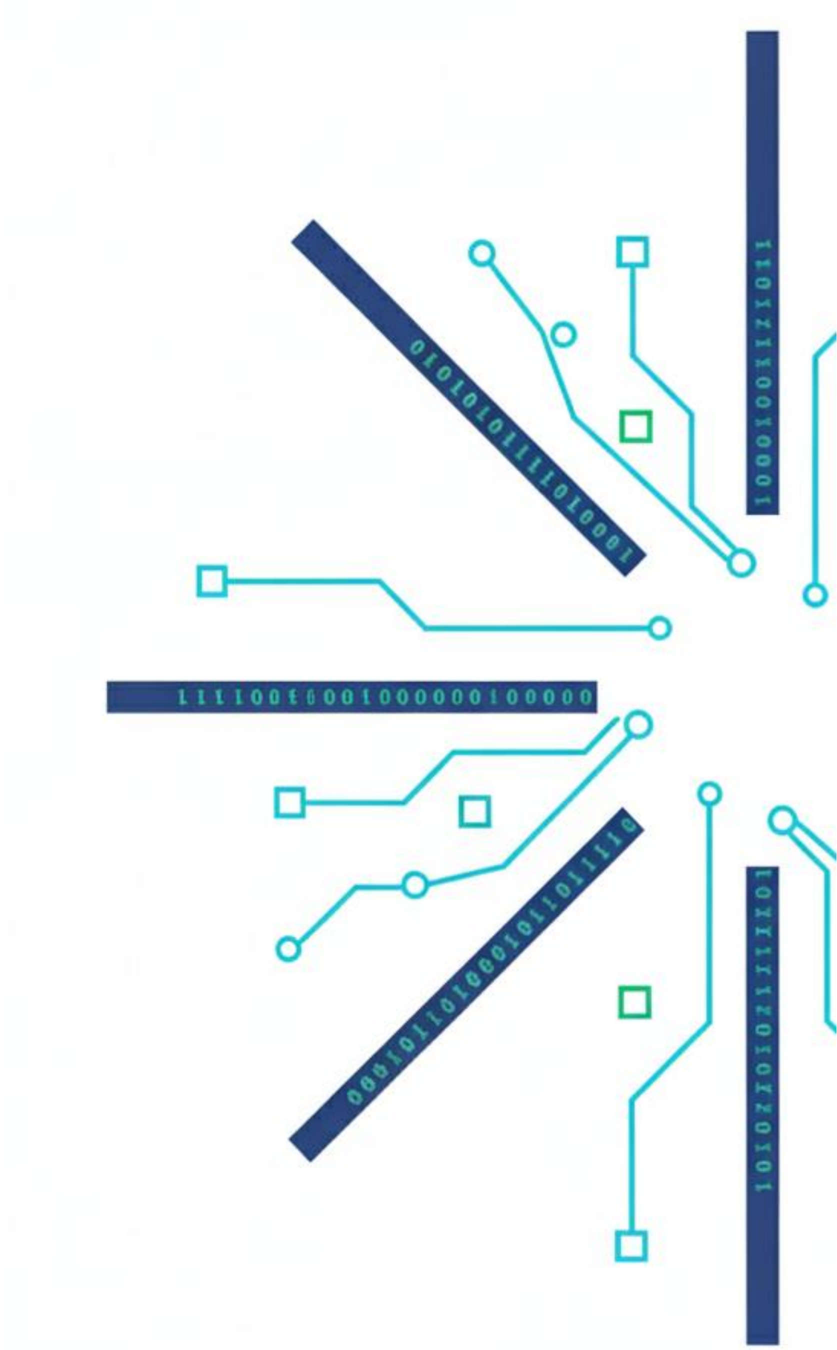
D14 - Syntetic dataset



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.

Outline

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.

2 Introduction

An important aspect of digital forensics data research is the ability to create a dataset that meets certain expectations and requirements. In general, there is no dataset that can be used for all research purposes in the field of digital forensics [1, 2]. There are several challenges researchers face when using, creating, and sharing datasets. It is important to note that, in general, such datasets are missing in this area, or there is a lack of documentation and formal description of its construction [3, 4]. For the purposes of our research and project, we need to work with a dataset that describes real scenarios that can occur in real security incidents. The goal is to create a suitable dataset for comparing methods in the analysis of digital evidence, which could be used to investigate various problems. In general, there are various datasets that researchers work with, but either they do not share them or not all of them are usable in this type of research. There are also frameworks that can be used to generate datasets, but there is no guarantee that the dataset is correctly created and meets all the conditions necessary for this type of research.

We are aware that such a dataset **may not cover all situations** that can occur in real-world environments. According to the project objectives, the presented deliverable - the dataset - is **created primarily from datasets used in CTF competitions and is appropriately supplemented with realistic data**. For these reasons, the dataset consists of several complementary parts:

- **A dataset created through the simulation of various attacker techniques**, which is described in more detail in Chapter 4.
- **A dataset created from disk images originating from CTF competitions**, which is described in more detail in Chapter 5. This part of the dataset includes:
  - evaluated and processed outputs from forensic tools (modified EZ tools dataset), and
  - the modified **embedded dataset**.

The structure of this output follows the **article schema of the journal *Data in Brief***, which is focused on the presentation and description of research datasets. A brief overview of the dataset is provided in **Tab. 1**.

Subject	Computer Science
Specific subject area	Cybersecurity, digital forensics.
Type of data	Tabular data (CVS files) containing data, SQL database, and embedding data from several forensics artefacts of the Windows operating system.
Data collection	The data were obtained from two sources, categorized by their type. Data covering various attacker techniques were collected from a simulated environment provided by IstroSec and subsequently gathered using the Athena tool. The second type of data consists of processed images of devices (computers and laptops) downloaded from various CTF competition websites and repositories. The purpose

	of these CTFs is to provide practical experience in digital forensics, incident response, and threat hunting. Specifically, we used data from the case titled *The Stolen Szechuan Sauce* from the DFIR Madness Portal, data from Magnet CTFs (2019, 2022), and data from the NIST data leakage case. These datasets are freely available on the respective portals and are widely used for training purposes, supporting both beginners and professionals in the fields of incident response and digital forensics.
Data source location	Data covering various attacker techniques were stored in IstroSec laboratory. The second type of data is from disk image files of the domain controller and also disk image files from the desktop from the case titled The Stolen Szechuan Sauce. Specifically, we have used files DC01 Disk Image (E01) and Desktop Disk Image (E01). Next image files are Magnet CTF 2019 Windows Desktop, Magnet CTF 2022 Windows Laptop, and NIST Data Leakage Case.
Data accessibility	Repository name: Mendeley Data – ADFIR forensic dataset Data identification number: 65g9gm8zrd Direct URL to data: 10.17632/65g9gm8zrd.1
Related research articles	Hennelová, Z., Marková, E., & Sokol, P. (2025, August). The Impact of Anti-forensic Techniques on Data-Driven Digital Forensics: Anomaly Detection Case Study. In *International Conference on Availability, Reliability and Security* (pp. 131-148). Cham: Springer Nature Switzerland. Antoni, Ľ., Sokol, P., Krišáková, S. P., Kotlárová, D., Krídlo, O., & Krajči, S. (2025). Formal concept analysis and attribute dependencies of NIST data leakage case.

Tab. 1 - Specifications Table

3 Value of data

From the perspective of digital forensic analysis, it is essential to pay attention to the value and potential of available data, as these form the foundation for effective security incident analysis. The value of the presented dataset can be identified in the following aspects:

This dataset enables the effective **application of data analysis and machine learning methods** in the field of digital forensic analysis within the Windows operating system environment using the NTFS file system. The dataset is composed of a combination of data from simulated attacks and data originating from Capture The Flag (CTF) competitions, which together provide a realistic view of system behavior and attacker activities. Since NTFS is the most widely used file system in the Windows operating system, these data represent a key example of the application of analytical and machine learning methods in the field of cybersecurity, specifically in digital forensic analysis. Within our research, methods such as Formal Concept Analysis [5, 6] and outlier detection [7] were applied to these data.

The dataset is intended for researchers focused on the application of data analysis and machine learning in cybersecurity, as well as for practitioners interested in **automating security incident response processes** and digital forensic analysis in the Windows operating system and NTFS file system environment.

The presented dataset is suitable for the **development of automated security tools** aimed at identifying relevant digital evidence in Windows systems, analyzing relationships between evidence, their attributes, and other contextual dependencies arising from forensic artifacts of the Windows operating system and the NTFS file system.

The **identification of digital traces and behavioral patterns** of system and attacker activities, based on data from simulated attacker techniques as well as CTF scenarios in the Windows operating system environment, can significantly contribute to the development of more effective tools for the detection and prevention of cyberattacks. This enables organizations to more effectively protect their information systems and data against security threats.

In addition to retrospective forensic analysis, the dataset is also applicable to **predictive analysis and the prevention of security threats** in the Windows operating system and NTFS file system environment. The application of data analysis and machine learning methods to combined data from simulated attacks and CTF competitions makes it possible to identify behavioral patterns and anomalies that may indicate potential security risks or compromises within IT infrastructure.

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

4 Simulated attackers' techniques dataset

The simulated attackers´ techniques dataset created within this project was designed for training and validating machine learning (ML) models aimed at automating digital forensic analysis. Its primary objective is to realistically capture the behaviour of modern attackers in the **post-exploitation phase**, with a particular focus on advanced techniques used by APT groups. Unlike purely synthetic or randomly generated datasets, this dataset combines automated attack simulation with manually executed advanced offensive activities, ensuring a high level of realism and forensic relevance.

4.1 Data description

The baseline layer of network and system activity was generated using the **Cymulate platform**, which provided a realistic background of legitimate operations and simulated standard intrusion vectors. Within this environment, an advanced scenario emulating the behaviour of the **APT-19** group was executed, covering a wide range of techniques from malicious code execution to persistence and defence evasion.

The dataset includes a broad spectrum of **Tactics, Techniques, and Procedures (TTPs)**, particularly:

- abuse of PowerShell (payload download via Invoke-WebRequest, execution of Base64-encoded commands, hidden PowerShell windows),
- execution of commands as Windows services,
- process injection using **Reflective DLL Injection**,
- service installation and execution via PowerShell,
- registry modifications (persistence mechanisms, hiding file extensions, manipulation of security-related settings),
- abuse of legitimate system binaries (**Living off the Land Binaries – LOLBINs**), including the use of Alternate Data Streams (ADS),
- manipulation of the PATH environment variable to hijack execution flow.

On top of this automated baseline, manually injected attack sequences targeted key stages of the post-exploitation attack chain. **Persistence** was achieved using the **SharPersist** tool, which created scheduled tasks, services, and registry entries to ensure long-term access through a backdoor. This was followed by **environment enumeration and privilege escalation**, performed using **SharpUp** and **winPEAS**, which generated a large volume of system queries and a distinct “noisy” pattern in logs, characteristic of aggressive post-exploitation reconnaissance.

For **credential access**, the dataset captures the use of **SafetyKatz** and **SharpKatz** to extract NTLM and Kerberos authentication material from the LSASS process, including the creation and removal of memory dumps. The **defence evasion** phase is represented by the use of **SharpKiller**, an AMSI bypass technique that disables in-memory scanning, simulating sophisticated adversaries who attempt to blind security monitoring before proceeding with further malicious actions.

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

From a **threat landscape** perspective, the dataset reflects the current shift from traditional file-based malware towards **Living off the Land (LotL)** techniques and **.NET tradecraft**. Many tools are loaded directly into memory with minimal disk footprint, forcing detection mechanisms—and ML models in particular—to focus on behavioural anomalies, process context, and API usage rather than static file signatures. The intentional placement of tools in trusted directories such as C:\Windows\Tasks, combined with legitimate-looking filenames, further supports research into context-aware detection of masquerading processes.

Thanks to the inclusion of techniques actively used by both state-sponsored actors and modern **Ransomware-as-a-Service (RaaS)** groups, the dataset has strong **practical applicability**. It enables the development and evaluation of ML-based detection approaches capable of identifying fileless attacks, advanced persistent threats, and early-stage activities aimed at weakening or bypassing security controls in real-world environments.

More detailed information on the dataset creation process is provided in the deliverable: **ADFIR – D13 – Method for Dataset Creation.**

4.2 Database schema

This module defines the hierarchy of the project, individual collections, and the identification of devices.

Column Name	Data Type	Description
Id	int	Unique Project Identifier (PK).
Guid	uniqueidentifier	Globally unique project identifier.
DtCreatedUtc	datetime2(2)	The date and time the project was created.
DtUpdatedUtc	datetime2(2)	The date and time the project was last modified.
Name	nvarchar(128)	The name of the project.
Code	nvarchar(250)	Internal code name of the project.
ProjFolder	nvarchar(256)	The path to the folder where the project data is stored.

Tab. 2 - Project

Column Name	Data Type	Description
Id	int	Unique parameter identifier (PK).
ProjectId	int	A foreign key that points to the Project table.

Name	nvarchar(64)	The name of the parameter.
Value	nvarchar(1024)	The value of the parameter.

Tab. 2 - ProjectParameter

Column Name	Data Type	Description
Id	int	Unique session identifier (PK).
Guid	uniqueidentifier	A globally unique session identifier.
AgentVersion	bigint	The version of the agent who performed the data collection.
ProjectId	int	A foreign key that points to the Project table.
HostId	int	A foreign key pointing to the Host table.
DtStartUtc	datetime2(2)	The start time of data collection.
DtEndUtc	datetime2(2)	Time of end of data collection.
Result	int	The return code of the collection result (Success/Error).

Tab. 3 - Session

Column Name	Data Type	Description
Id	int	Unique Guest Identifier (PK).
Guid	uniqueidentifier	Device GUID.
DtRegisteredUtc	datetime2(2)	Date of registration of the device in the system.
HostName	nvarchar(256)	The network name of the device.
HardwareId	uniqueidentifier	Unique Hardware ID (HWID).
OsId	uniqueidentifier	The identifier of the operating system.

Tab. 4 - Guest

Column Name	Data Type	Description
Id	int	Unique artifact identifier (PK).
SessionId	int	Link to the Session session.
ArtConfigId	int	The ID of the configuration used for this artifact.

Descriptor	nvarchar(512)	The original identifier of the source (e.g., the path to the file).
Checksum	bigint	Artifact checksum for integrity verification.
Status	tinyint	The status of the artifact being transmitted or processed.

Tab. 5 - ArtObject (Base Entity)

4.2.1 NTFS File System (SchemaMft)

This module maps the Master File Table (MFT) structure.

Column	Data type	Description
Id	bigint	Primary key.
ArtObjectId	int	Link to ArtObject.
Signature	int	The signature of the record (usually "FILE").
UpdateSeqOffset	smallint	Offset to update sequence array.
Lsn	bigint	Log Sequence Number ($LogFile).
SequenceNumber	smallint	Sequence number for reuse of the record.
ReferenceCount	smallint	The number of hardlinks per file.
Flags	smallint	Flags (InUse, Directory).
BaseRecordRef	bigint	Reference to the base MFT record (if it is an extension).

Tab. 6 - MftEntry (Main Entry)

Column	Data type	Description
MftAttrHeaderId	Bigint	FK on the attribute header.
DtCreatedUtc	datetime2(0)	Creation Time.
DtLastModifiedUtc	datetime2(0)	Date of modification of the content (Modification Time).
DtMftEntryLastMod...	datetime2(0)	MFT Change Time (MFT Change Time).
DtLastAccessUtc	datetime2(0)	Last Access Date (Access Time).
FileAttributeFlags	Int	File flags (Read-only, Hidden, System...).

Tab. 7 - MftStandardInformation (Timestamps and Attributes)

Column	Data type	Description
MftAttrHeaderId	bigint	FK on the attribute header.

ParentDirectoryRef	bigint	Reference to the MFT record of the parent folder.
DtCreatedUtc	datetime2(0)	Time of creation (from the point of view of the file name).
AllocatedFileSize	bigint	The allocated size of the file on the disk.
RealFileSize	bigint	The actual size of the file data.
FileName	nvarchar(512)	The name of the file/folder.
Namespace	tinyint	Namespace type (POSIX, Win32, DOS).

Tab. 8 - MftFileName (Name and Parent)

4.2.2 Execution Artifacts: Prefetch (SchemaPrefetch)

Processed .pf files used to optimize application startup.

Column	Data type	Description
Id	bigint	Primary key.
FileSize	int	The size of the prefetch file.
ExecutableFilename	nvarchar(60)	The name of the executable file (e.g., CMD.EXE).
PrefetchHash	int	The hash of the file path (distinguishes the same names from other paths).
RunCount	int	The total number of times the app was launched.
VolumeCount	int	The number of volumes from which the application was launched.

Tab. 9 - PrefetchFileHeader

Column	Data type	Description
PrefetchFileHeaderId	bigint	FK to the header of the prefetch file.
LastRunTime	datetime2(0)	The timestamp of one of the most recent runs.

Tab. 10 - PrefetchLastRunTime

Column	Data type	Description
PrefetchStartTime	int	The time since the start of the run when the file was loaded.
PrefetchDuration	int	How long did it take to load the file.

FileName	nvarchar(512)	The name of the loaded file (dependencies, DLLs).

Tab. 11 - PrefetchFileMetrics

4.2.3 LNK files and shortcuts (SchemaShellLink)

Detailed decomposition of the binary structure of Shell Link (LNK) files.

Column	Data type	Description
Id	bigint	Primary key.
LinkFlags	int	Flags determining the presence of other data structures.
FileAttributesFlags	int	Attributes of the target file (Hidden, System...).
CreationTime	datetime2(3)	The time the target file was created.
WriteTime	datetime2(3)	Write time of the target file.
FileSize	int	The size of the target file.

Tab. 12 - ShellLinkHeader

Column Name	Data Type	Allow Nulls	Description
Id	bigint	No	Unique identifier (PK).
ShellLinkHeaderId	bigint	Yes	Foreign key to ShellLinkHeader.
NameString	nvarchar(256)	Yes	Optional link description.
RelativePath	nvarchar(256)	Yes	The relative path to the target file.
WorkingDir	nvarchar(256)	Yes	Working directory for running the application.
CommandLineArguments	nvarchar(256)	Yes	Command-line arguments (Key for forensic analysis).
IconLocation	nvarchar(256)	Yes	The path to the file with the icon.

Tab. 13 - ShellLinkStringData (Text Data)

Column	Data type	Description

MachineId	nvarchar(256)	NetBIOS the name of the machine where the link was created.
Droid1	uniqueidentifier	Volume ID.
Droid2	uniqueidentifier	File ID.

Tab. 14 - ShellLinkTrackerData (Distributed Link Tracking)

4.2.4 Event Logs (SchemaEventLog)

This module is used to store processed Windows Event Logs (.evtx).

Column Name	Data Type	Description
Id	bigint	Unique Row Identifier (PK).
ArtObjectId	int	Foreign key to ArtObject (source file).
EventId	int	Event ID (e.g., 4688, 4624).
Version	tinyint	A version of the event structure.
Qualifiers	smallint	Event qualifiers (often 0).
[Level]	tinyint	Severity level (Info, Warning, Error).
Task	smallint	The category of the task within the provider.
Opcode	tinyint	Operational code (e.g., Info, Start, Stop).
Keywords	bigint	Keyword Bitmask (Audit Success/Failure).
RecordId	bigint	The original sequence number of the log entry.
ProviderName	nvarchar(255)	The name of the source (e.g., Microsoft-Windows-Security-Auditing).
ProviderId	uniqueidentifier	GUID of the provider.
ChannelName	nvarchar(255)	Channel name (Security, System, Application).
ProcessId	int	PID of the process that logged the event.
ThreadId	int	Thread ID.
ComputerName	varchar(255)	The name of the computer where the event originated.
UserId	varbinary(128)	User SID (binary).

DtCreatedUtc	datetime2(2)	The exact time of the occurrence of the event.
ActivityId	uniqueidentifier	Activity ID for Correlation (ETW).
RelatedActivityId	uniqueidentifier	Related activity ID.
Description	nvarchar(MAX)	Full text description of the event (if mined).

Tab. 15 - EvtRecord

Column Name	Data Type	Description
Id	bigint	Unique Property Identifier (PK).
EvtRecordId	bigint	Foreign key on EvtRecord.
Name	nvarchar(128)	The name of the parameter (e.g., TargetImage, IpAddress).
Value	nvarchar(MAX)	The value of the parameter (e.g., C:\Windows\System32\cmd.exe).

Tab. 16 - EvtRecordProperty (Dynamic Data)

4.2.5 Program Compatibility Assistant (SchemaPca)

Column	Data type	Description
ExecutablePath	nvarchar(512)	Full path to the running application.
LastExecutedUtc	datetime2(3)	Last Run Time.

Tab. 17 - PcaLaunchDictionary

Column	Data type	Description
RuntimeUtc	datetime2(3)	Run/record time.
ExecutablePath	nvarchar(512)	The path to the file.
ExitCode	nvarchar(64)	The return code of the end of the process.
FileVersion	varchar(32)	File version.

Tab. 18 - PcaGeneralDatabase

4.2.6 Recycle Bin (SchemaRecycleBin)

Column	Data type	Description
DeletedFileSize	bigint	The size of the file in bytes.
DtDeletedUtc	datetime2(3)	The time the file was deleted (moved to the trash).
DeletedFileName	nvarchar(512)	The original name and path of the deleted file.

Tab. 19 - RecycleBinFileEntry

4.2.7 Windows Error Reporting (SchemaWer)

Windows error message analysis module. The main table is WerReport.

Column Name	Data Type	Description
Id	int	The primary key of the report.
ArtObjectId	int	Binding to the ArtObject table (source file).
Version	int	Version of the WER report.
EventType	varchar(32)	Event type (e.g., "APPCRASH", "BEX").
EventTimeUtc	datetime2(3)	The time of occurrence of the event in UTC.
ReportType	int	Numerical report type.
UploadTimeUtc	datetime2(3)	The time when the report was sent to the Microsoft server.
ReportIdentifier	uniqueidentifier	A unique GUID identifying this report.
OriginalFilename	nvarchar(512)	The original name of the file that caused the error.
AppPath	nvarchar(512)	The path to the app that crashed.
AppName	nvarchar(512)	The name of the app.

Tab. 20 - WerReport

Column Name	Data Type	Description
Id	int	The identifier of the record.

Report Id	int	Link to the main report (WerReport).
[Key]	nvarchar(512)	The name of the metadata key (e.g., "ProcessId").
Value	nvarchar(512)	A value (e.g., process number).

Tab. 21 - WerReportProcessMetadata

Column Name	Data Type	Description
Id	int	The identifier of the record.
Report Id	int	Link to the main report.
Path	nvarchar(512)	The full path to the loaded module/library.

Tab. 22 - WerReportLoadedModule

Column Name	Data Type	Description
Id	int	The identifier of the record.
Report Id	int	Link to the main report.
Name	nvarchar(512)	The name of the parameter (e.g., "AppVersion").
Value	nvarchar(512)	Parameter value (e.g., "1.0.0.0")

Tab. 23 - WerReportSig

Column Name	Data Type	Description
Id	int	Record ID.
Report Id	int	Link to WerReport.
Name	nvarchar(512)	Name.
Value	nvarchar(512)	Value.

Tab. 24 - WerReportDynamicSiq

Column Name	Data Type	Description
Id	int	Record ID.
Report Id	int	Link to WerReport.
[Key]	nvarchar(512)	Status key.
Value	nvarchar(512)	The value of the state.

Tab. 25 - WerReportState

Column Name	Data Type	Description
Id	int	Record ID.
Report Id	int	Link to WerReport.
[Key]	nvarchar(512)	Key (e.g., OS version).
Value	nvarchar(512)	Value.

Tab. 26 - WerReportOsInfo

Column Name	Data Type	Description
Id	int	Record ID.
Report Id	int	Link to WerReport.
Value	nvarchar(512)	The text displayed to the user.

Tab. 27 - WerReportUI

4.2.8 USN Journal (SchemaUsn)

Column Name	Data Type	Description
Id	bigint	Record ID.
ArtObjectId	int	Binding to ArtObject.
FileReferenceNumber	bigint	File ID (MFT Index).
ParentFileReferenceNumber	bigint	Parent folder ID.
Usn	bigint	Update Sequence Number.
TimeStamp	datetime2(0)	A time of change.
Reason	int	The reason for the change (flags).
FileName	nvarchar(512)	The name of the file.
FileAttributes	int	File attributes.

Tab. 28 - UsnRecordV2

Column Name	Data Type	Description
Id	bigint	Record ID.

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

FileReferenceNumber	binary(16)	128-bit file ID.
ParentFileReferenceNumber	binary(16)	128-bit parent ID.
Usn	bigint	Serial number.
TimeStamp	datetime2(0)	A time of change.
Reason	int	The reason for the change.
FileName	nvarchar(512)	The name of the file.

Tab. 30 - UsnRecordV3

Column Name	Data Type	Description
Id	bigint	Record ID.
FileReferenceNumber	binary(16)	128-bit file ID.
Reason	int	The reason for the change.
RemainingExtents	int	Remaining extents.
ExtentSize	smallint	The size of the extent.

Tab. 31 - UsnRecordV4

4.2.9 Windows Task Scheduler (SchemaWinScheduler)

Column Name	Data Type	Description
Id	int	Job ID.
ArtObjectId	int	Binding to an XML file.
URI	varchar(512)	The path of the task in the system.
SecurityDescriptor	varchar(128)	Security settings.
Author	nvarchar(128)	Author of the task.
Date	datetime2(0)	Creation date.
Description	nvarchar(1024)	Job description.
Enabled	bit	Whether the task is active.
Hidden	bit	Whether the task is hidden.

Tab. 32 - WinSchedulerTask

Column Name	Data Type	Description
Id	int	Principal ID.
TaskId	int	Attachment to the task.
UserId	varchar(256)	User ID / SID.
LogonType	varchar(32)	Login type.
RunLevel	varchar(16)	Permission level.

Tab. 33 - WinSchedulerTaskPrincipal

Column Name	Data Type	Description
Id	int	Share ID.
TaskId	int	Attachment to the task.
Command	nvarchar(512)	Command / Path to EXE.
Arguments	nvarchar(MAX)	Command parameters.
WorkingDirectory	nvarchar(512)	Work directory.

Tab. 34 - WinSchedulerTaskExecAction

Column Name	Data Type	Description
Id	int	Action ID.
TaskId	int	Binding to the task.
Server	nvarchar(256)	SMTP server.
[To]	nvarchar(512)	Beneficiary.
Subject	nvarchar(512)	Subject.
Body	nvarchar(MAX)	Message body.

Tab. 35 - WinSchedulerSendEmailAction

Column Name	Data Type	Description
Id	int	Action ID.
TaskId	int	Binding to the task.
ClassId	uniqueidentifier	CLSID of the COM object.

Data	xml	Data for the handler.

Tab. 36 - WinSchedulerComHandlerAction

Column Name	Data Type	Description
Id	int	Action ID.
TaskId	int	Attachment to the task.
Title	nvarchar(256)	The title of the window.
Body	nvarchar(4000)	Message text.

Tab. 37 - WinSchedulerShowMessageAction

Column Name	Data Type	Description
Id	int	Trigger ID.
TaskId	int	Binding to the task.
Type	smallint	Trigger type.
StartBoundary	datetime2(0)	Effective since.
EndBoundary	datetime2(0)	Expiry.

Tab. 38 - WinSchedulerTaskTrigger

Column Name	Data Type	Description
Id	int	Record ID.
TriggerId	int	Binding to Trigger.
Delay	bigint	Delay (seconds).

Tab. 39 - WinSchedulerBootTrigger

Column Name	Data Type	Description
Id	int	Record ID.
TriggerId	int	Binding to Trigger.
Subscription	nvarchar(256)	Subscribe to events (Query).

Tab. 40 - WinSchedulerEventTrigger

Column Name	Data Type	Description
Id	int	Record ID.
TriggerId	int	Binding to Trigger.
UserId	nvarchar(256)	The user to whom it relates.

Tab. 41 - WinSchedulerLogonTrigger

Column Name	Data Type	Description
Id	int	Record ID.
TriggerId	int	Binding to Trigger.
RandomDelay	bigint	Random delay.

Tab. 42 - WinSchedulerTimeTrigger

4.2.10 Windows 10 Timeline Activity (SchemaWin10Timeline)

Column Name	Data Type	Description
Id	int	Activity ID.
AppId	varchar(1024)	The JSON ID of the application.
ActivityType	int	Activity type.
StartTime	datetime2(0)	Start time.
EndTime	datetime2(0)	Time of the end.
Payload	varbinary(1024)	Activity content.
PlatformDeviceId	varchar(128)	Device ID.
ExpirationTime	datetime2(0)	Record expiration.

Tab. 43 - Win10TimelineActivity

Column Name	Data Type	Description
Id	int	Operation ID.
ArtObjectId	int	A link to an artifact.
OperationOrder	int	The order of the operation.
AppId	varchar(1024)	App ID.

Payload	varbinary(1024)	Operation data.

Tab. 44 - Win10TimelineActivityOperation

Column Name	Data Type	Description
Id	int	Record ID.
ActivityId	uniqueidentifier	Activity ID.
PackageName	varchar(256)	The name of the package.
Platform	varchar(256)	Platform.

Tab. 45 - Win10TimelineActivityPackageId

Column Name	Data Type	Description
Id	int	Record ID.
AssetPayload	varbinary(1024)	Binary content.
LastRefreshTime	datetime2(0)	Last updated time.

Tab. 46 - Win10TimelineAsset (parser)

Column Name	Data Type	Description
Id	int	Record ID.
AppId	varchar(256)	App ID.
AppTitle	varchar(256)	App caption.

Tab. 47 - Win10TimelineAppSettings

Column Name	Data Type	Description
Id	int	Record ID.
[Key]	varchar(128)	Metadata key.
Value	varchar(128)	Value.

Tab. 48 - Win10TimelineMetadata

4.2.11 Windows Registry (SchemaWinReg)

Column Name	Data Type	Description

Id	bigint	Key ID (offset).
KeyName	nvarchar(255)	The name of the key.
LastWrittenTimestamp	datetime2(3)	Last Write Time.
Parent	int	Parent ID.
NumberOfSubkeys	int	The number of subkeys.
NumberOfKeyValues	int	Number of values.

Tab. 49 - WinRegKeyNode

Column Name	Data Type	Description
Id	bigint	Value ID.
CellId	bigint	The ID of the key cell.
ValueName	nvarchar(MAX)	The name of the value.
DataType	int	Data type (REG_SZ, etc.).
DataSize	int	Data size.
DataOffset	int	Offset of data in a file.

Tab. 50 - WinRegKeyValue

Column Name	Data Type	Description
Id	bigint	Block ID.
Signature	int	Signature (regf).
LastWrittenTimestamp	datetime2(3)	The time the file was last modified.
RootCellOffset	int	Offset of the root key.

Tab. 51 - WinRegBaseBlock

Column Name	Data Type	Description
Id	bigint	Record ID.
SecurityDescriptor	varbinary(2048)	ACL permissions (binary).
ReferenceCount	int	Number of links.

Tab. 52 - WinRegKeySecurity

Column Name	Data Type	Description
Id	bigint	Record ID.
CellId	bigint	Cell ID.
ListSegmentOffset	int	The offset of the list segment.

Tab. 53 - WinRegBigData

Column Name	Data Type	Description
Id	bigint	Record ID.
Offset	int	Position in the file.
Size	int	The size of the bin.

Tab. 54 - WinRegHiveBinHeader

Column Name	Data Type	Description
Id	bigint	Record ID.
Size	int	Cell size.
Offset	int	Cell position.

Tab. 55 - WinRegHiveBinCell

5 CTF dataset

In addition to the Simulated Attackers' Techniques dataset described in the previous section, we used **Capture The Flag (CTF) competitions** as a data source for the creation of datasets for digital forensics research, primarily because access to real-world data from security incidents is often problematic and highly limited. Data originating from real-life cyber incidents is typically not publicly available, and obtaining such data is constrained by organizational permissions, legal and regulatory requirements, and strict privacy and data protection obligations. These limitations significantly hinder reproducibility and large-scale experimentation in digital forensics research.

CTF datasets offer a practical and ethically sound alternative, as they are specifically designed to emulate realistic attack scenarios while remaining legally accessible and well-documented. The activities performed during CTF challenges are intentionally malicious, but they are executed in controlled environments, allowing forensic artifacts to be safely collected, shared, and analyzed. As a result, CTF-based datasets provide valuable ground truth, making them particularly suitable for training, testing, and benchmarking forensic analysis methods and machine learning models.

The creation of the dataset from CTF scenarios followed the procedure described in deliverable D13 – Method for Dataset Creation. The dataset consists of two main parts.

The first part is the **Modified Embedding Dataset**, which was created using the *plaso* tool to generate supertimelines, followed by the construction of an embedding-based representation of the extracted forensic data. This approach enables advanced analytical and machine learning techniques to be applied to temporal forensic artifacts.

The second part is the **Modified EZ Tools Dataset**, which was generated using forensic tools developed by Eric Zimmerman for parsing disk images and extracting Windows forensic artifacts. The extracted data was subsequently enriched with labels to support supervised analysis and evaluation.

Detailed information on the preparation of datasets from CTF disk images is provided in deliverable D13 – Method for Dataset Creation.

5.1 CTF use cases

Within our dataset, we have pre-processed several representative forensic cases originating from simulated attacks and CTF (Capture The Flag) scenarios carried out in a controlled environment. All cases are based on the Windows operating system with the NTFS file system, which ensures a uniform structure of forensic artifacts and consistent interpretation of evidence. Data preprocessing was designed to unify heterogeneous forensic sources, reduce noise, and preserve the temporal and contextual relationships between individual events. The

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

following cases represent various attack and post-incident scenarios used in the creation and evaluation of the dataset.

In our dataset we have preprocessed the following cases:

- CTF case The Stolen Szechuan Sauce,
- Magnet CTF case 2019,
- Magnet CTF case 2022, and
- NIST Data Leakage Case.

5.1.1 The Stolen Szechuan Sauce

The **CTF case The Stolen Szechuan Sauce** focuses [8] on a forensic investigation of a corporate data breach scenario. The main objective is to determine how a secret Szechuan sauce recipe belonging to CITADEL company ended up on a dark website. The company requested a forensic analysis of its Domain Controller and network host to identify malicious applications installed on the system and determine place and time of software installation. The case also provides us with information – whether any information has been created, modified or deleted and whether there has been a data breach. We are working with artifacts from the company's Domain controller server (**DC server**) and from the **Desktop** (network host).

5.1.2 Magnet CTF 2019, and 2022

The **Magnet CTF case 2019** [9], and **Magnet CTF case 2022** [10] are forensic challenge scenarios designed to test specific digital forensic analysis skills. All of the three use cases were part of the capture the flag competition. It is not a classic process of digital forensic analysis, but rather answers to specific questions, such as "when was the disk image acquired", "when was the software installed", and others.

5.1.3 NIST / Data Leakage Case

The **NIST Data Leakage Case** [11] represents a training-oriented forensic scenario focused on investigating various forms of data leakage. The primary objective of this scenario is to become familiar with different types of data exfiltration and to practice appropriate forensic investigation techniques. The case describes an internal data breach in which an employee deliberately abused his authorized access to transfer confidential information to a competing company. The leakage was carried out through a combination of email communication, personal cloud storage services, and an attempted physical transfer of data using external storage media. Despite the presence of established security policies and deployed DLP/DRM mechanisms, the suspect sought to bypass these controls, requiring forensic analysis to identify evidence of the data leakage and reconstruct activities performed on the seized electronic devices.

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

5.2 Modified embedding dataset

To ensure **experimental reproducibility**, consistent data processing, and the reusability of trained components, all auxiliary artifacts generated during dataset preparation were stored in standardized and widely supported formats.

The structure of this part of the dataset is as follows:

- **global_scaler.pkl** – a pickle file containing the fitted scaler, used during inference to adjust the size of delta values.
- **tokenizer** – a directory containing files required for reconstructing the tokenizer.
    - **merges** – a text file specifying how to merge individual vocabulary outputs.
    - **vocab** – the learned vocabulary used to represent text in the CTF files.
- **processed_windows** – a directory containing preprocessed windows (20 rows each) created from CTF files, formatted as <embedding, delta, label>.
    - **val** – a subdirectory containing windows used for validation during training.
    - **train** – a subdirectory containing windows used for training.
    - ***_windows.pt** – windows saved in a PyTorch-optimized format (in the dataset, these are provided as archives: *_windows.pt.zip).
    - ***_windows.txt** – windows in a human-readable text format.

The **vocabulary and tokenizer** used for processing the textual representation of forensic artifacts were serialized and stored in **JSON format**. This format provides an explicit and transparent mapping between tokens and their numerical identifiers, is human-readable, and can be easily transferred across different experimental environments, easily loaded into industry-standard frameworks (such as tokenizers from Hugging Face). Storing the tokenizer as JSON guarantees that identical tokenization rules and vocabulary are applied during both training and evaluation phases, thereby preventing inconsistencies in text preprocessing that could negatively affect model performance.

For the normalization of numerical features, most notably the temporal deltas between consecutive events, a scaling mechanism (e.g., MinMaxScaler) was applied. After fitting, the scaler was persisted as a **pickle file**, enabling it to be reloaded and consistently applied to validation or newly acquired data without recalculating scaling parameters. Persisting the scaler is essential for maintaining uniform numerical transformations across the entire experimental pipeline and for ensuring comparability between models trained on different dataset partitions.

Instead of storing the embeddings themselves (which would be large), encoded windows are stored in a PyTorch-optimized format.

In Tab. 56 statistics of this part of dataset is provided.

File	Size of dataset [kB]	type

vocab	tokenizer	786	JSON
merges	tokenizer	446	TXT
Magnet_CTF_2019_windows	processed windows - train	7 278 650	PyTorch model file
Magnet_CTF_2019_windows	processed windows - train	1 071 261	TXT
NIST_Data_Leakage_windows	processed windows - train	6 883 445	PyTorch model file
NIST_Data_Leakage_windows	processed windows - train	936 726	TXT
SSS_DC_windows	processed windows - train	2 422 781	PyTorch model file
SSS_DC_windows	processed windows - train	362 009	TXT
SSS_Desktop_windows	processed windows - train	1 698 838	PyTorch model file
SSS_Desktop_windows	processed windows - train	236 472	TXT
Magnet_CTF_2022_windows	processed windows - val	8 350 791	PyTorch model file
Magnet_CTF_2022_windows	processed windows - val	1 173 086	TXT
global_scaler.pkl		0,609	Python pickle file

Tab. 56 – Size of files

5.3 Modified EZ Tools Dataset

The **Modified EZ Tools Dataset** was created using a set of forensic tools developed by Eric Zimmerman, a recognized expert in DFIR and the author of widely used tools such as RECmd, EvtxECmd, SrumECmd, PECmd, SBECmd, and others. These tools enable structured parsing of a wide range of forensic artifacts from the Windows operating system, either directly from disk images or extracted digital traces. Further details on the tools and their use for dataset creation are provided in Deliverable D13 – Method for Dataset Creation.

The resulting dataset contains normalized and machine-processable representations of low-level system artifacts, primarily exported in CSV format, which were subsequently enriched with labels to support analytical and machine learning tasks.

The dataset is organized into logical categories representing different aspects of system and user behavior. Each category corresponds to a specific group of Windows forensic artifacts, providing complementary perspectives on system activity, user behavior, and potential attacker actions.

The structure of this part of the dataset is as follows:

- CTF use case (Magnet CTF 2019 / Magnet CTF 2022 / NIST Data Leakage Case / Stolen Szechuan Sauce DC / Stolen Szechuan Sauce Desktop) directory:
    - EventLogs directory,
    - FileFolderAccess directory,
    - FileSystem directory,
    - ProgramExecution directory,
    - Registry directory,
    - SRUMDatabase directory.

A detailed overview of each category is provided in the following subchapters. A comprehensive summary of all artifacts and related information is available in Deliverable D12 – Method for Automated Collection of Digital Evidence.

5.3.1 EventLogs

This section of the dataset contains parsed Windows system logs (.evtx files) processed using EvtxECmd. EventLogs represent one of the most important sources of temporal and contextual information in digital forensic investigations.

The dataset includes system, security, and application logs, as well as extended logging sources, such as Sysmon, when available. These artifacts record events related to user authentication (logins and logouts), privilege usage, service and driver activity, process creation and termination, network connections, registry changes, and error states.

Key extracted information includes precise timestamps, event identifiers (Event ID), event sources, descriptive messages, and structured data fields. This category is essential for reconstructing timelines, detecting suspicious behavior, and correlating system events with other forensic artifacts.

5.3.2 FileFolderAccess

The FileFolderAccess category focuses on forensic artifacts related to user interactions with files and directories, providing insight into how users navigated the system and which files or folders they accessed.

Primary data sources include Shellbags (parsed using SBECmd or registry plugins), Jump Lists (JLECmd), RecentDocs entries, WordWheelQuery (Windows Explorer search history), and Open/Save MRU lists. Together, these artifacts reveal information about opened files and folders, network shares, USB device usage, and access to cloud synchronization services such as OneDrive or Dropbox.

This category is crucial for identifying user intent, reconstructing workflows, and detecting attempts to search for, access, or exfiltrate sensitive data.

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

5.3.3 FileSystem

The FileSystem category captures low-level file system activity, primarily derived from the NTFS $MFT (Master File Table) parsed using MFTECmd. The $MFT provides a comprehensive view of file and directory metadata across the entire disk volume.

The dataset includes file timestamps, such as creation, modification, access, and metadata change (MACB times), as well as filenames, paths, sizes, and flags indicating deleted or resident files. This information allows for detailed reconstruction of the file lifecycle, including creation, modification, deletion, and potential manipulation.

Supplementary artifacts, such as LNK shortcuts (parsed using LECmd), enable correlation of file system activity with user interactions and program execution.

5.3.4 ProgramExecution

This category aggregates artifacts that provide evidence of program execution within the system and is critical for identifying which applications or binaries were run, how often, and in what context.

Data sources include Prefetch files (parsed with PECmd), Amcache, AppCompatCache (Shimcache), UserAssist registry entries, Background Activity Monitor (BAM/DAM), and Jump Lists. Collectively, these artifacts reveal executed executables, execution counts, last run times, command-line arguments, and, in some cases, persistence mechanisms.

The ProgramExecution category is particularly relevant for detecting malicious code execution, lateral movement tools, living-off-the-land binaries (LOLBins), and unauthorized software usage.

5.3.5 Registry

The Registry category consists of artifacts parsed from Windows Registry hives using RECmd or Registry Explorer with specialized plugins. The Windows Registry serves as a central repository of system configurations, user settings, and application states, making it a rich source of forensic evidence.

The dataset includes artifacts such as UserAssist, TypedURLs, MRU lists, autorun keys (Run and RunOnce), Shellbags, USB device history (USBSTOR), service configurations, installed programs, persistence mechanisms, and browser-related data. Multiple plugins were used to extract specific artifacts, such as Bam, RecentApps, or TypedPaths.

Registry artifacts are essential for understanding system configuration changes, identifying persistence, and tracking historical user activity and peripheral device usage.

5.3.6 SRUMDatabase

The SRUMDatabase category is derived from the System Resource Usage Monitor database (SRUDB.dat), parsed using SrumECmd. This database stores historical statistics of system resource usage and is extremely valuable for long-term activity reconstruction.

This section of the dataset contains detailed information on application usage (CPU time, bytes read and written), network activity (data sent and received per application and network interface), energy consumption, and events such as push notifications. SRUM data is typically retained for approximately 30–60 days, enabling activity analysis even when other forensic artifacts have been deleted or overwritten.

This category provides a unique quantitative view of application behavior and network activity and is highly effective in identifying anomalous or suspicious patterns over extended time periods.

5.4 Characteristics of modified EZ tools dataset

This section focuses on the values and characteristics of data extracted from datasets created using EZ tools. It provides an overview of key statistical properties of artifacts obtained forensic images from CTF use cases.

5.4.1 Szechuan Sauce – DC

The following table Tab. 57 shows statistics from artifacts obtained from the CTF Szechuan Sauce DC.

Artefact		Size of dataset [kB]	Number of records
EventLogs		44 200	87 280
FileDeletion		0,299	1
FileFolderAccess	AutomaticDestinations	9	9
	CustomDestinations	6	6
	LECmd_Output	7	12
	Administrator_NTUSER	0,256	1
	Administrator_UsrClass	5	27
FileSystem	MFTECmd_$MFT_Output	49 900	111 850
	MFTECmd_$J_Output	17 300	82 085
ProgramExecution	Amcache_UnassociatedFileEntries	1	2
	Windows81_Windows2012R2_SYSTEM_AppCompatCache	52	280

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

Registry	RECmd_Batch_AllRegExecutablesFoundOrRun_Output	19	29
	RECmd_Batch_BasicSystemInfo_Output	81	206
	RECmd_Batch_InstalledSoftware_Output	15	34
	RECmd_Batch_RECmd_Batch_MC_Output	6 300	17 372
	RECmd_Batch_RegistryASEPs_Output	16 700	42 987
	RECmd_Batch_SoftwareASEPs_Output	529	1 320
	RECmd_Batch_SoftwareClassesASEPs_Output	10 600	28 194
	RECmd_Batch_SoftwareWoW6432ASEPs_Output	6 600	17 038
	RECmd_Batch_SystemASEPs_Output	6 200	17 459
	RECmd_Batch_UserActivity_Output	107	198

Tab. 57 – Size of files

5.4.2 Szechuan Sauce – Desktop

The following table Tab. 58 shows statistics from artifacts obtained from the CTF Szechuan Sauce Desktop.

Artefact		Size of dataset [kB]	Number of records
EventLogs		33 100	40 917
FileFolderAccess	AutomaticDestinations	51	47
	CustomDestinations	4	5
	LECmd_Output	18	28
	ricksanchez_UsrClass	1	6
	Administrator_UsrClass	7	34
	mortysmith_UsrClass	1	5
	Admin_UsrClass	0,822	3
FileSystem	MFTECmd_$MFT_Output	58 600	132 615
	MFTECmd_$J_Output	9 900	43 463
ProgramExecution	Amcache_DriverPackages	2	4
	Amcache_DeviceContainer	4	16
	Amcache_AssociatedFileEntries	37	83
	PECmd_Output	2 000	196
	Amcache_DriveBinaries	126	371
	Amcache_UnassociatedFileEntries	6	15
	Amcache_ShortCuts	6	35
	Amcache_ProgramEntries	56	85
	Windows10Creators_SYSTEM_AppCompatCache	52	266
	Amcache_DevicePnps	109	201

Registry	RECmd_Batch_AllRegExecutablesFoundOrRun_Output	109	194
	RECmd_Batch_BasicSystemInfo_Output	50	149
	RECmd_Batch_InstalledSoftware_Output	15	42
	RECmd_Batch_RECmd_Batch_MC_Output	6 400	16 401
	RECmd_Batch_SystemASEPs_Output	5 700	15 411
	RECmd_Batch_SoftwareASEPs_Output	955	2 758
	RECmd_Batch_RegistryASEPs_Output	26 600	77 615
	RECmd_Batch_UserClassesASEPs_Output	457	1 114
	RECmd_Batch_SoftwareClassesASEPs_Output	16 300	50 182
	RECmd_Batch_SoftwareWoW6432ASEPs_Output	6 900	20 921
	RECmd_Batch_UserActivity_Output	892	1 718
SRUMDatabase	SrumECmd_vfuprov_Output	19	105
	SrumECmd_NetworkUsages_Output	41	217
	SrumECmd_NetworkConnections_Output	2	10
	SrumECmd_AppTimelineProvider_Output	280	1488
	SrumECmd_PushNotifications_Output	2	11
	SrumECmd_AppResourceUseInfo_Output	169	766

Tab. 58 – Size of files

5.4.3 NIST data Leakage case

The following table Tab. 59 shows statistics from artifacts obtained from the CTF NIST Data Leakage case.

Artefact		Size of dataset [kB]	Number of records
EventLogs		4 100	5 199
FileFolderAccess	AutomaticDestinations	39	42
	CustomDestinations	44	53
	LECmd_Output	23	36
	informant_UsrClass	19	97
	informant_NTUSER	3	12
	temporary_UsrClass	2	10
	admin11_UsrClass	1	6
FileSystem	MFTECmd_$MFT_Output	43 700	98 904
	MFTECmd_$J_Output	69 400	317 137
ProgramExecution	PECmd_Output	1 100	95

	Windows7x64_Windows2008R2_SYSTEM_AppCompatCache	51	262
	Windows7x64_Windows2008R2_SYSTEM_AppCompatCache	57	305
Registry	RECmd_Batch_AllRegExecutablesFoundOrRun_Output	80	127
	RECmd_Batch_BasicSystemInfo_Output	184	472
	RECmd_Batch_InstalledSoftware_Output	155	376
	RECmd_Batch_RECmd_Batch_MC_Output	12 300	33 161
	RECmd_Batch_SystemASEPs_Output	11 800	33 102
	RECmd_Batch_SoftwareASEPs_Output	718	1 865
	RECmd_Batch_RegistryASEPs_Output	35 400	92 418
	RECmd_Batch_UserClassesASEPs_Output	4	10
	RECmd_Batch_SoftwareClassesASEPs_Output	24 700	66 624
	RECmd_Batch_SoftwareWoW6432ASEPs_Output	14 400	37 640
	RECmd_Batch_UserActivity_Output	247	446

Tab. 59 – Size of files

5.4.4 Magnet CTF 2019

The following table Tab. 60 shows statistics from artifacts obtained from the CTF Magnet 2019.

Artefact		Size of dataset [kB]	Number of records
EventLogs		93 300	109 433
FileDeletion		0,313	1
FileFolderAccess	AutomaticDestinations	32	30
	CustomDestinations	1	2
	LECmd_Output	14	24
	SelmaBouvier_UsrClass	3	16
	Administrator_UsrClass	6	28
FileSystem	MFTECmd_$MFT_Output	94 200	210 592
	MFTECmd_$J_Output	63 700	292 405
ProgramExecution	Amcache_DriverPackages	1	3
	Amcache_DeviceContainers	4	14
	Amcache_AssociatedFileEntries	9	21
	PECmd_Output	2 300	212

	Amcache_DriveBinaries	111	330
	Amcache_UnassociatedFileEntries	22	55
	Amcache_ShortCuts	11	65
	Amcache_ProgramEntries	59	104
	Windows10Creators_SYSTEM_AppCompatCache	152	742
	Amcache_DevicePnps	44	83
Registry	RECmd_Batch_AllRegExecutablesFoundOrRun_Output	71	130
	RECmd_Batch_BasicSystemInfo_Output	46	142
	RECmd_Batch_InstalledSoftware_Output	5	13
	RECmd_Batch_RECmd_Batch_MC_Output	5 600	14 732
	RECmd_Batch_SystemASEPs_Output	5 100	14 254
	RECmd_Batch_SoftwareASEPs_Output	835	2 434
	RECmd_Batch_RegistryASEPs_Output	24 500	72 291
	RECmd_Batch_UserClassesASEPs_Output	190	466
	RECmd_Batch_SoftwareClassesASEPs_Output	15 500	48 040
	RECmd_Batch_SoftwareWoW6432ASEPs_Output	6 900	20 900
	RECmd_Batch_UserActivity_Output	450	879
SRUMDatabase	SrumECmd_vfuprov_Output	108	569
	SrumECmd_NetworkUsages_Output	2 700	14 892
	SrumECmd_NetworkConnections_Output	204	1 374
	SrumECmd_AppTimelineProvider_Output	2 400	13 652
	SrumECmd_PushNotifications_Output	34	176
	SrumECmd_AppResourceUseInfo_Output	30 400	153 256
	SrumECmd_EnergyUsage_Output	2	12

Tab. 60 – Size of files

5.4.5 Magnet CTF 2022

The following table Tab. 61 shows statistics from artifacts obtained from the CTF Magnet 2022.

Artefact		Size of dataset [kB]	Number of records
EventLogs		163 900	157 383
FileFolderAccess	AutomaticDestinations	50	56
	CustomDestinations	18	24
	LECmd_Output	32	55
	Patrick_usrClass	7	36
FileSystem	MFTECmd_$MFT_Output	198 200	418 398

	MFTECmd_$J_Output	74 400	344 746
ProgramExecution	Amcache_DriverPackages	27	30
	Amcache_DeviceContainers	2	7
	Amcache_AssociatedFileEntries	68	175
	PECmd_Output	8 400	786
	Amcache_DriveBinaries	142	391
	Amcache_UnassociatedFileEntries	74	188
	Amcache_ShortCuts		
	Amcache_ProgramEntries	75	120
	00_Windows10Creators_SYSTEM_AppCompatCache	101	498
	03_Windows10Creators_SYSTEM_AppCompatCache	120	554
	Amcache_DevicePnps	103	166
Registry	RECmd_Batch_AllRegExecutablesFoundOrRun_Output	122	234
	RECmd_Batch_BasicSystemInfo_Output	110	337
	RECmd_Batch_InstalledSoftware_Output	33	95
	RECmd_Batch_RECmd_Batch_MC_Output	10 500	31 093
	RECmd_Batch_SystemASEPs_Output	10 000	30 725
	RECmd_Batch_SoftwareASEPs_Output	2 000	5 702
	RECmd_Batch_RegistryASEPs_Output	57 200	165 565
	RECmd_Batch_UserClassesASEPs_Output	3 000	7 770
	RECmd_Batch_SoftwareClassesASEPs_Output	36 700	112 130
	RECmd_Batch_SoftwareWoW6432ASEPs_Output	13 800	41 203
	RECmd_Batch_UserActivity_Output	329	647

Tab. 61 – Size of files

6 Bibliography

[1] C. Grajeda, F. Breitinger, I. Baggili, Availability of datasets for digital forensics–and what is missing, Digital Investigation 22 (2017) S94–S105.

[2] L. Luciano, et al., Digital forensics in the next five years, in: Proceedings of the 13th International Conference on Availability, Reliability and Security, 2018.

[3] H. Studiawan, F. Sohel, C. Payne, Sentiment analysis in a forensic timeline with deep learning, IEEE Access 8 (2020) 60664–60675.

[4] E. Casey, The chequered past and risky future of digital forensics, Aust. J. Forensic Sci. 51 (6) (2019) 649–664.

[5] P. Sokol, et al., The analysis of digital evidence by Formal concept analysis, The 16th International Conference on Concept Lattices and Their Applications (CLA 2022), 2022.
[6] P. Sokol, et al., Formal concept analysis approach to understand digital evidence relationships, Int. J. Approx. Reason. 159 (2023) 108940.
[7] E. Marková, P. Sokol, K. Kováčová, Detection of relevant digital evidence in the forensic timelines, 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), IEEE, 2022.
[8] Case 001 – the stolen szechuan sauce, 2020, [online] Available: https://dfirmadness.com/the-stolen-szechuan-sauce/.
[9] MagnetCTF 2019 Windows Desktop, 2019, [online] Available: https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/magnet/2019%20CTF%20-%20Windows-Desktop.zip.
[10] Magnet CTF 2022 Windows, 2022, [online] Available: https://digitalcorpora.s3.amazonaws.com/corpora/scenarios/magnet/2022%20CTF%20-%20Windows.zip.
[11] Data LeakageCase, [online] Available: https://cfreds-archive.nist.gov/data_leakage_case/data-leakage-case.html.