Exploratory Analysis of File System Metadata for Rapid Investigation of Security Incidents

Michal Beran* František Hrdina† Daniel Kouřil‡ Radek Ošlejšek§ Kristína Zákopčanová¶

Masaryk University

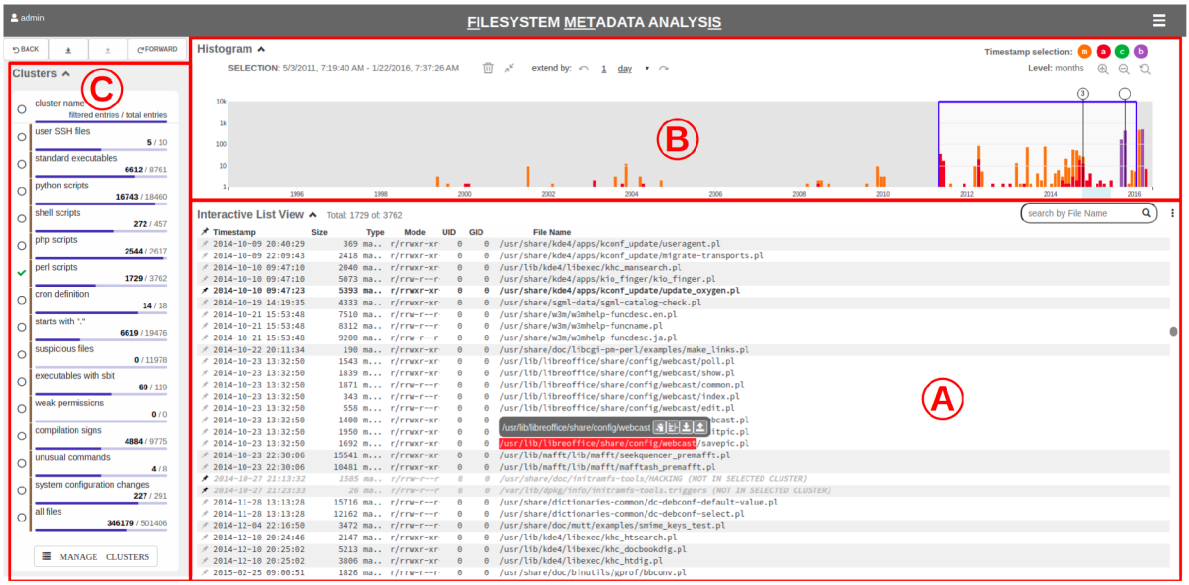

Figure 1: FIMETIS is a tool providing an interactive exploration of file system snapshots. Analysts can quickly investigate cybersecurity incidents via three complementary views: A – *list view* with file system records, B – *histogram* with a timeline, and C – data *clusters*.

Abstract

Investigating cybersecurity incidents requires in-depth knowledge from the analyst. Moreover, the whole process is demanding due to the vast data volumes that need to be analyzed. While various techniques exist nowadays to help with particular tasks of the analysis, the process as a whole still requires a lot of manual activities and expert skills. We propose an approach that allows the analysis of disk snapshots more efficiently and with lower demands on expert knowledge. Following a user-centered design methodology, we implemented an analytical tool to guide analysts during security incident investigations. The viability of the solution was validated by an evaluation conducted with members of different security teams.

Keywords: incident investigation, digital evidence, file system metadata, data analysis

Index Terms: Human-centered computing—Visual analytics; Security and privacy—Systems security—File system security; Applied computing—Computer forensics—Evidence collection, storage and analysis;

*e-mail: beran@ics.muni.cz
†e-mail: hrdina@ics.muni.cz
‡e-mail: kouril@ics.muni.cz
§e-mail: oslejsek@fi.muni.cz
¶e-mail: zakopcanova@mail.muni.cz

1 Introduction

Cybercrime has rapidly developed over the past years [10], and cybersecurity threats are expected to present significant risks for the future [1]. For computer systems to be able to face the constantly changing threat landscape, it is necessary to develop and maintain capabilities for responding to cybersecurity attacks. A vital part of the response process consists of the investigation of the evidence, which reveals the nature of the incident and performed activities.

The investigation depends heavily on a proper evaluation of all collected evidence. Methods of digital forensics [8,17] are employed for systematic scrutiny of the data. It is a continuous process where hypotheses are formulated based on observations followed by steps to either confirm or deny the theory.

A simplified scheme of an investigation workflow is depicted in Figure 2. First, the suspicion of an incident is reported in the form of a preliminary report. Then, data sources for digital evidence of the incident are collected. They capture either the broader state of involved computer networks and communication history (net flows, PCAPs) or the state of involved devices (system logs, the content of disks, memory snapshots, etc.).

The iterative investigation is often time-consuming and requires a high level of expert knowledge. The amount of data collected is often high, which only complicates the analysis. While the forensic investigation methods provide a great platform to derive particular results, a user-oriented approach is missing to simplify the overall process.

Permanent storage devices are a crucial part of contemporary

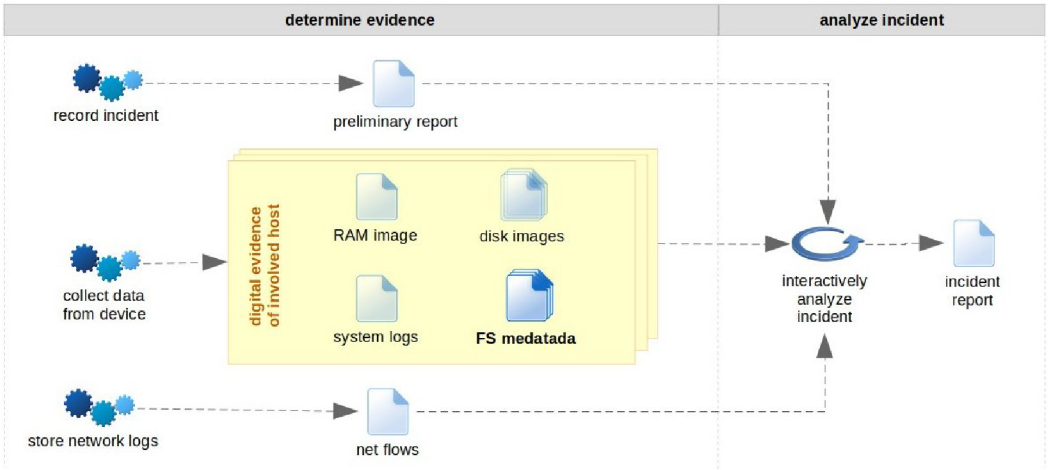

Figure 2: Incident investigation process. The *FIMETIS* tool deals with file system metadata only.

computer systems and data retrieved from these devices provide significant input for the investigation. The state of permanent storage can be captured in multiple ways. The most straightforward and complete approach is to analyze the complete disk content. However, as current media tend to be quite large—it is not uncommon for disks to provide several terabytes of capacity—the analysis becomes time- and resource-demanding. Moreover, analyzing disk content encounters privacy issues when the data contain sensitive information [9].

One way of coping with the volume and privacy problems is to work only with file metadata, extracted from permanent storage, which include the file owner, size, name and dates of last manipulations. However, even though such a dataset is much smaller in size compared to raw disk images, it is still necessary to process hundreds of thousands of records already in case of a standard storage. Moreover, it requires deep knowledge about the relationships among files, their purpose in the system, and importance for the attacker.

In this paper, we propose visual-analytic methods that make the investigation of file system metadata significantly more efficient and are also available to analysts with no deep domain knowledge. We describe an application called *FIMETIS* (FIlesystem METadata analysIS) that was developed to verify the visual-analytic concepts. Evaluation of this tool has shown that the user interface is easy to learn and well supports analytical tasks. Even less skilled participants were able to investigate and reconstruct a real incident in limited time at surprising precision and level of details.

2 Related Work

Many tools and approaches dealing with individual types of data sources for digital evidence can be found.

So far, big attention has been paid to the investigation of network communication. NetCapVis [27] provides a post-incident visual analysis of PCAP files that capture network traffic. TVi [3] is a tool that combines multiple visual representations of network traces to support different levels of visual-based querying and reasoning required for making sense of complex traffic data. Visualization techniques proposed by Gray et al. [11] provide conceptual network navigation for situational awareness in network communication.

Analysis of system logs was researched as part of ELVIS [14] and CORGI [15], for instance. These tools, both proposed by the same authors, provide security-oriented log visualizations that allow security experts to visually explore and link numerous types of log files through relevant representations and global filtering. A top-down approach to the log exploration is provided by the Visual Filter [26] tool, which represents the whole log in a single overview and then allows the investigators to navigate and make context-preserving sub-selections.

Disks and permanent storage provide another valuable source of information for the digital investigation. Disk and file systems analysis can be performed in several layers [7]. Approaches addressing specific features are, for example, Change-link 2.0 [18], which provides several visualizations to capture changes to files and directories over time, or the work of Heitzmann et al. [13], who proposed a visual representation of access control permissions in a standard hierarchical file system using treemaps.

This paper deals with the utilization of file system metadata as they have lesser demands on volumes and do not threaten data sensitivity. The utility of metadata for digital forensics has been articulated previously [4], and various techniques for metadata-based analyses have been proposed since then. The use of metadata to provide a fingerprint of actions performed with files has been suggested to streamline file system analysis [16].

Metadata attributes are also known to be useful to reconstruct a timeline of previous activities [12] and have been demonstrated to locate suspicious files [21]. These techniques address the particular sub-problems of the analysis. To facilitate the whole investigation process, it is necessary to support interactive work, which would support the above-mentioned analytical techniques and make them easily accessible to users.

Only a few papers can be found on approaches supporting interactive work with the data of digital evidence, which is essential for the whole forensic investigation process. Our literature survey revealed two works dealing with timelines constructed from file system activities, which are very relevant to our research.

The Zeitline [5] tool represents activities as generic events. The user interface enables analysts to group events and then make the timeline hierarchical, to filter obtained data trees, and locate specific events by queries.

In the CyberForensic TimeLab [19], the timeline is implemented as a histogram using bars to represent the number of pieces of evidence at a specific time. The investigator can highlight interesting parts of the timeline and zoom in to get greater detail of that particular time span.

Both the tools are designed as generic, enabling analysts to create timelines from multiple resources, e.g., from file system metadata as well as system logs, and their user interfaces reflect this universality. In contrast, our approach focuses solely on file snapshots build from metadata only. We aim to make the analysis of this specific data

maximally effective, focusing not only on the timeline but also on other data available for files. To reach this goal, we follow a user-centered design methodology, which is extended with a mechanism guiding the investigator during the process. Although our design shares some visual elements with the CyberForensic TimeLab, e.g., histograms, our solution provides an interface fine-tuned for a single specific use case – a forensic analysis of file system snapshots. On the other hand, the visual-analytics concepts proposed in this paper are sufficiently general that they could be extended to other types of timeline in the future.

3 Design Methodology

In this project, we applied the user-centered approach guided by the design study methodology framework [25], mainly reflecting its *core* stages: discover, design, implement, deploy.

In the *discover* stage, we gained a better understanding of the workflows of the digital investigation and elicited user requirements on the tool in order to simplify the analytical tasks.

The initial insight into the application domain was provided by a co-author of this paper, who is a member of the cybersecurity team of Masaryk University. Based on his initial input, we conducted semi-structured informal interviews with two other domain experts who have long-term experience with practical investigations of cybersecurity incidents. The first respondent works as a senior security specialist at CESNET – an academic institution in the Czech Republic providing IT services to Czech academia. The second expert is a member of the incident response team at Masaryk University. All three of them have long-term experience with practical investigation of cybersecurity incidents. Each interview lasted about two hours.

Based on these interviews, we distilled a generic workflow of the investigation process and formulated requirements for a file system analysis. The results are presented in Section 4.

In the *design* stage, we proposed the visual elements and the interactive dashboard reflecting the functional requirements. The design was proposed and refined iteratively. User interfaces were continuously prototyped under consultation with the domain expert (co-author of the paper). Proposed visual encoding is described in Section 5.

In the *implement* stage, we iteratively developed the analytical dashboard. We paid attention to the observation that cybersecurity experts investigate incidents rarely, and evidence collection is a long-term interactive process. Architecture and implementation of the tool are described in Section 6.

In the *deploy* stage, we evaluated the tool. As the investigation of real cybersecurity incidents is a sensitive process, we could not perform a usability study *in the wild*. Moreover, as the developed tool deals with only part of this process, we conducted a qualitative evaluation focused directly on the tool. However, we used data from a real incident. The evaluation is described in Section 7 and results are summarized in Section 8.

4 Requirement Analysis

The interviews conducted during the *discover* stage of the design methodology revealed that incident investigators would benefit from an interactive tool for file system exploration. Specific requirements were inferred from the characteristics of the data and the analytical workflow.

4.1 Data Characteristics and Abstraction

The investigation of cybersecurity incidents aims to provide answers to key questions related to the incident, like when the activities happened, what data was changed during the incident, where the activities originated from, etc. The process of investigation is driven by methodologies stipulated by digital forensics. The whole process comprises three main stages during which the evidence is acquired, analyzed, and the final report is produced. A simplified schema of the process is depicted in Figure 2.

During the acquisition phase, the investigator needs to identify and collect the data that is likely to provide evidence about the case. The number of possible data sources from which digital evidence can be collected is vast. In case of forensic examinations performed directly on the machine, it is common to gather data from permanent storage (hard disk or external device like USB storage). There are also other sources of digital evidence, such as network traffic or its metadata, state and content of volatile memory, or information about authentication attempts. The rest of the paper deals with analysis of files and their metadata. It keeps the investigation domain limited in size while making it possible to evaluate the main principles.

File metadata describes information about the file, maintained by the operating system together with the file data. The exact scope of metadata depends on the operating system used, however, nowadays, it is common for all widely used file systems to recognize the file name, file ownership (specifying the user and a group), content size, and access rights. Besides these, several timestamps are maintained, indicating the time when key activities with the file or the metadata were last performed:

- *a-time*: the time when the file content was last read (accessed),
- *m-time*: the time when the file content was last modified,
- *c-time*: the time when the metadata record was last changed (e.g., during the change of access rights),
- *b-time*: the time when the file was created. The *b-time* timestamp is supported only by advanced file systems.

All the timestamps, except for *b-time*, change during the file life-time based on the operations performed. When a timestamp is updated, the previous value is overwritten and lost, which means they always refer only to the last performed actions.

Timestamps are an essential source of information for the reconstruction of events relevant to the investigation. They can help understand when certain operations took place but also reveal the nature of the activities performed. For instance, when a file is copied from another computer, the copying process usually retains the original timestamp. Such a file has the *m-time* value set to a date before the *b-time* and *c-time* values, which both will refer to the time when the copying process finished. A brand-new file created on the system has all the timestamps set to the same value upon creation. The difference in the timestamps can reveal where the file originates from.

Even if they do not reveal the actual file content, all file metadata attributes play a big role in the incident analysis. One of the most important reconstructions is determination of the timeline of actions performed in the analyzed system. A timeline emphasizes crucial activities conducted during the incident. For instance, it specifies when the attacker accessed the system for the first time or when a specific system configuration got changed.

A timeline constructed from metadata is a list of records ordered by the timestamps. Since there are multiple timestamp types assigned to a file, a single file can occur multiple times in the list, whenever its timestamps differ. A typical timeline contains hundreds of thousands of records, which need to be further analyzed.

In addition to providing input to recover the timeline, metadata can be used for efficient filtering of files, based on unique *fingerprints* they form, such as similarities of file locations, common access rights, or suspicious ownership.

4.2 Requirements

Based on the interviews, data abstraction, and the analytical workflow, we identified five functional requirements:

R1: Exploration of the file system structure. During the investigation, the analysts have to pay attention to different parts of

the file system, e.g., files in a specific directory, files with specific extensions, or all log files. However, the interviewed domain experts emphasized that the interactive hierarchical exploration of the file system is not helpful. Instead, they need a global temporal view of the file system data with the possibility to navigate in the file system structure effectively. The analytical tool should support analysts in the efficient switching between different parts of the file system and narrowing the area of interest by offering filtering functions that would localize the data by various aspects and meaning encoded in the available file system metadata.

R2: Exploration of temporal relationships. Disk snapshots have strong temporal characteristics. Each record provides the timestamp of the last manipulation, e.g., the creation, modification, or access. However, every file or directory usually appears multiple times in the dataset as the manipulation timestamps differ, which increases the data volume to be inspected. Also, the recorded data period is often very long, containing timestamps from a time long before the system was installed (but from when the files were created). Therefore, providing a scalable temporal view on the data with efficient filtering, zooming, and preserving time coherence is very important for making the analysis effective.

R3: Detection of file system anomalies. Some combinations of file locations and attributes can be considered unusual or deserving analyst's attention. For example, publicly writable files or directories, hidden files outside of users' homes, executables with administrator's privileges, files masking their names (e.g., a binary file with a *.txt* extension or named with only white spaces). The analytical tool should provide multiple views on various combinations of location paths and attributes in order to localize potential anomalies easily, and then further explore the corresponding files using **R1** and **R2** principles.

R4: Traces of the execution of suspicious commands. Some commands are seldom used by administrators but often used by attackers. For example, the `shred` Unix command is often used to wipe data content. The tool should allow analysts to verify whether or not such commands were used. Command execution can be identified by the *a-time* attribute. Once the command execution is confirmed, the analyst can use interactions reflecting **R1** and **R2** to explore details, analyze the impact of the execution, and either confirm or reject the hypothesis that an attacker executed the command.

R5: Traces of batch processing. Besides the execution of specific commands (**R4**), attackers often use scripts to perform reconnaissance on the system or to compile programs or libraries before installing them into the system. These batch activities can be recognized by the execution of multiple commands or the creation of multiple files in a short time, while manual tasks take a longer time. However, batch processing can represent a legal activity, e.g., the legal compilation or the result of regular system updates. Therefore, the tool should support analysts in efficiently identifying batch processes in the huge amount of file system data and then allowing them to analyze suspicious activities further using **R1** and **R2**.

While the requirements **R1** and **R2** reflect the generic investigation workflow, requirements **R3–R5** are related to more specific analytical questions that are often asked during the file system investigation. Besides these functional requirements, we set two complementary qualitative requirements that affect the architecture and implementation. These requirements follow the practice emphasized by the interviewees where cybersecurity experts investigate incidents rarely, and every investigation takes a lot of time (hours or days).

R6: Easy to use. Even practicing incident investigators analyze disks rarely (see Section 7). Therefore, they should be able to use the tool even after a long period without the need for repeated learning.

R7: Persistence. The data and interactions have to be persistent so that an analyst can pause the investigation process and continue later on. Persistence is also important for recalling previous investigations and comparing hypotheses and results.

5 Visual Design

In this section, we summarize the design rationale, visual encoding, and interaction capabilities. The user interface consists of three coordinated views [20, 24], where a change in one view to the dataset affects other parts of the dashboard.

5.1 List View

The *List View* (Figure 1 – A) is a dominant part of the dashboard providing a view on the raw data. Records are sorted by the timestamp by default (**R2**), but they can be re-ordered according to the file system structure (**R1**) by clicking on the *File Name* or *Type* columns. Individual columns can be shown or hidden via the *List View* menu (the three dots in the up-right corner of the *list view* area).

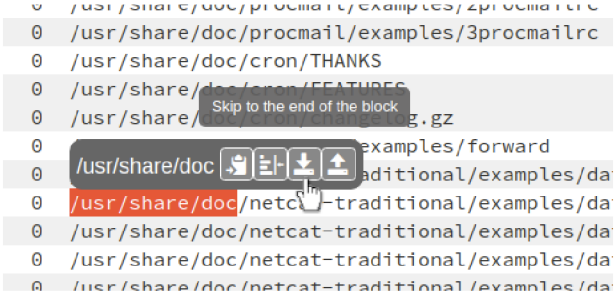

Figure 3: Detail of smart block skipping in the *List View*.

Analysts can browse records traditionally by scrolling the list up and down, or they can use *smart block skipping* (Figure 3) that significantly increases the efficiency of the list exploration. By clicking on a timestamp or a file path, the prefix is highlighted, and a context menu appears that enables analysts to skip records with the same prefix. Using this feature, analysts can quickly navigate to the next or previous date, hour, or sub-directory, and then accelerate the data exploration either from structural (**R1**) or temporal (**R2**) perspective.

The background of lines with the same timestamp is brushed to visually distinguish different time blocks (**R2**).

Search operation in the list works at two levels (the *name selection* label in Figure 4). Typing text into the input search field highlights the corresponding parts of the file paths. If the text is confirmed or the user clicks at the magnifier icon, then the list of records is filtered out, and only relevant lines remain displayed, enabling the analyst to pay attention to only desired files and directories (**R1**,**R4**). Data filtered out in this way remains in the *Histogram* (see subsection 5.2) to preserve a broader context, but they are grayed out.

Records of high importance can be bookmarked (the *bookmarks* label in Figure 4). Bookmarked records are emphasized in the list, displayed in the *Histogram* view, and used for fast navigation (**R2**). Bookmarks are persistent throughout the whole analysis and can be removed only on demand. Moreover, as they provide a broader context with significant events, the bookmarked lines are always visible in the *List View*, even if they do not fit all filters of the dashboard at the moment.

5.2 Histogram

The *Histogram* section (Figure 1 – B) provides an interactive view on data distribution.

The *y*-axis encodes the number of records. The axis has a logarithmic scale to deal with high peaks that often appear in the data but still preserve the visibility of low numbers that can be important for analysts.

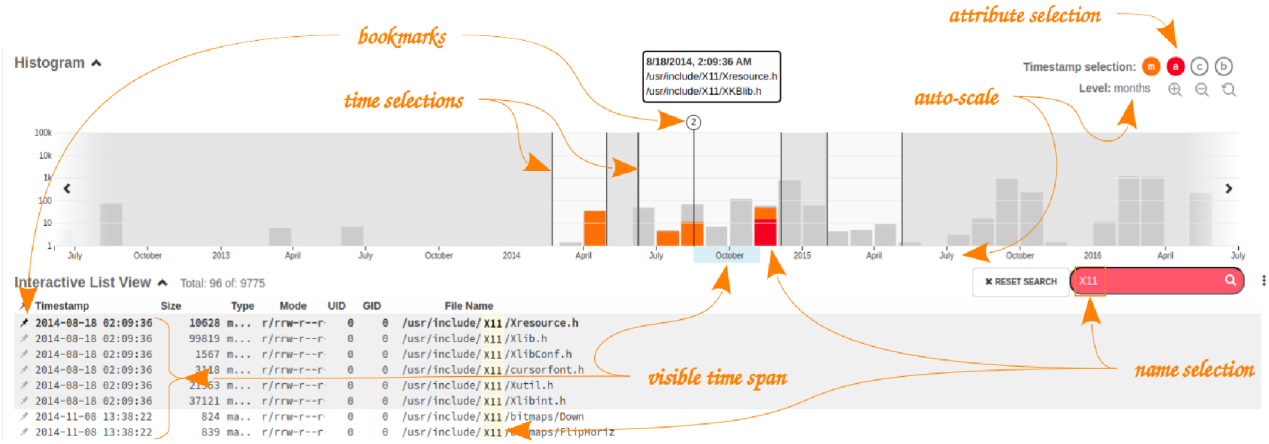

Figure 4: Navigation and filtering in the *List View* and *Histogram*.

The *x*-axis is scaled automatically (the *auto-scale* label in Figure 4). When zooming in, the *x*-axis automatically changes from years to months, days, and hours, and vice versa. The bars are recalculated and aggregated accordingly, representing the distribution in a specific year, month, day, etc. Zooming can be performed either by mouse, keyboard, or via icons in the upper-right corner.

Different colors in the histogram encode different file system operations (values of the *Type* column in the *List View*). Color encoding is shown in the *Timestamp selection* section. A detailed description of the metadata attributes is provided when the mouse is located over an icon. Similarly, hovering the mouse pointer above a bar in the histogram triggers a pop-up tool-tip with attribute type, time, and an exact number of records. Clicking on a bar scrolls the *List View* to the corresponding entries.

The *Timestamp selection* is also used for per-attribute filtering (the *attribute selection* label in Figure 4). Attributes can be switched on or off in the histogram by clicking on the icons. The *List View* is updated accordingly – only the records with selected attributes are shown in the list.

The histogram also serves as a time focusing tool (the *time selection* label in Figure 4). Using a mouse, the analyst can draw multiple span windows and thus restrict the lines shown in the *List View*. A context menu appears when a user selects a selection span window. This menu enables the user to perform common operations, like extending the span, zooming into the span, or erasing the span. Some of these operations are available via direct mouse interaction in the histogram as well.

Due to restricted space on the web page, the *List View* displays only part of all the records at any one time (the rest is available via scrolling). Visible records represent span, which is emphasized in the *x*-axis of the histogram as a cyan stripe (the *visible time span* label in Figure 4). This stripe supports the visual correlation between the *List View* and the histogram.

Entries bookmarked in the *List View* are shown in the histogram as push-pin icons. If they are too dense, they are aggregated into a single icon with a number of merged bookmarks. Details are provided as a tool-tip triggered on the mouse hover. Click on the icon scrolls the *List View* into the corresponding entry (to the first record in the case of aggregated push-pin). Push-pins that are out of selection spans are not clickable.

Span selectors, bookmarks, and automatically adaptable *x*-axis represent a powerful combination enabling analysts to scale and explore data from the time perspective (**R2**).

The structural exploration (**R1**) is less dominant in the histogram view. It is mainly restricted to the per-attribute filtering of records. On the other hand, the per-attribute filtering combined with the path filtering of the *List View* provides a generic approach to solve **R3** and **R5**. For example, a C/C++ compilation process accesses header files and the `gcc` compiler binary. A proper combination of the filters can reveal these traces. Moreover, the compilation unusually touches a huge amount of header files, leaving peaks in the histogram, especially when performed in calm nighttime.

5.3 Clusters

Clusters (Figure 1 – C) represent a generic mechanism enabling analysts to select files or directories with a specific ”fingerprint”. Clusters are defined by the combination of modification attributes (entries with *m-a-c-b* modification types) and regular expressions applied to the file names. Taking into account analytical requirements **R3 – R5** and needs of domain experts, we predefined several clusters covering the most common investigation tasks for UNIX file systems. Additional clusters can be easily appended.

- *All files* –The default cluster with no filtering.
- *User SSH files* – Configuration files and SSH keys stored in the users’ home directories.
- *Standard executables* – Files stored in the standard system directories for binaries, e.g., `/bin`, `/sbin`.
- *Python/shell/PHP/perl scripts* – Several clusters based on standard file extensions, e.g. `.py`, `.sh`.
- *Cron definitions* – Files stored in the default locations of `cron` jobs, i.e., regularly executed services.
- *Starts with ’.’* – Hidden files or directories.
- *Suspicious files* – Files or directories with names consisting of dots and white spaces.
- *Executables with sbit* – Executables that can run under a different user or group privileges than the original user or group.
- *Weak permissions* – Executable files writable for general users.
- *Compilation signs* – Access to C/C++ header files and the compiler executables.
- *Unusual commands* – Commands that are rarely used by common system administration, but often by attackers, e.g., `wget`, `curl`, and `shred`.
- *System configuration changes* – Important files related to the system configuration, e.g., `/etc/init.d` or `/etc/passwd`.

In the current implementation, only one cluster can be selected at one time. The number of all records fulfilling cluster criteria is shown as a “total entries” number. The “filtered entries” indicator shows the number of records satisfying other filtering criteria of the dashboard, and then they are listed in the *List view* and included in the *histogram*. A bar under each cluster box visually emphasizes the

ratio between the filtered and total records, enabling the analysts to identify the impact of currently used filtering criteria on clusters.

6 System Architecture and Implementation

FIMETIS is designed as a client-server application. The client part is implemented as a web application built on the Angular framework. Interactive visualizations use the D3.js library. The server part provides services for file system data management (import, export) and interactive data processing via the client. The Flask REST API handles the client-server communication. Flask is a lightweight web server gateway interface written in Python, which mediates access to the backend API – the center of the application logic and communication with databases. This architecture enables a concurrent investigation of multiple sources. It is possible to open two file systems simultaneously in two different explorer windows, for instance, and explore them side by side.

Persistence (**R7**) is guaranteed by two database systems. The file system snapshots are stored in the NoSQL Elasticsearch database. Configuration data, user accounts, interactions (e.g., bookmarks), and other operational data related to the analysis are stored in the rational Postgresql database.

7 Evaluation

To gather feedback on how well the tool fulfill the requirements **R1–R5**, and to identify possible refinements for the future design process iteration, we conducted a qualitative evaluation. The evaluation was held in June 2020.

7.1 Participants

We conducted the user study with five cybersecurity professionals who represent the target audience of the tool. All of them are members of the university cybersecurity research team or a security team in another organization. One participant works as an incident investigator in a private company. The average age of all participants was 30.2 years (*SD=3.5*); all of them were males. Two of them participated in initial interviews from which the requirements were derived. However, they did not participate on the design of the tool.

All the participants were cybersecurity professionals. However, they differ in the experience with practical investigation of incidents using file system analysis. Their skills are summarized in Table 1.

ID	Age	Occupation	INC
P1	34	researcher in cybersecurity	<3
P2	32	researcher in cybersecurity	0
P3	32	incident investigator – network analyst	<3
P4	26	lead security analyst	>10
P5	27	incident investigator	>10

Table 1: Demographic information of our participants. Occupation – position related to network administration and incident investigation, INC – number of incidents investigated by the analyst using disk analysis.

7.2 Data sets

During the evaluation, we used two datasets that were captured from computers affected by real incidents. The files were maintained using the *ext4* file system, which is commonly used on UNIX servers. We used different mechanisms to capture the primary data, yielding some records without the *b-time* timestamp (see 4.1). The first dataset contained 308 311 records and was used for the tool demonstration and familiarization of participants with the dashboard. The second dataset consisted of 505 742 records and was used for the evaluation.

We carefully analyzed the second dataset using FIMETIS to reconstruct the incident to establish a baseline for the evaluation. Navigating through the predefined clusters, we gradually collected a list of crucial findings relevant to the incident. We identified six clusters that are most relevant to providing evidence of the incident.

- *User SSH files* – Displays access to SSH key files used by the attacker to control remote access to user's account.
- *Suspicious files* – A bunch of files is visible in `/var/tmp/...`. The directory name is suspicious (`...` is often seen during attacks) and it contained files named using IP addresses, suggesting it was used as a cache for network scans.
- *Executables with sbit* – In addition to standard Unix commands, the output reveals file `/var/lib/.s`, which is definitely not legit (tries to hide itself and elevates the executable rights using the root s-bit parameter).
- *Unusual commands* – Two HTTP command-line clients can be seen in the output that are used recently: `wget` and `curl`.
- *System configuration changes* – Changes to the machine user accounts can be identified in the output.
- *Compilation signs* – Several compilations of C-language codes are present in the dataset.

However, these pieces of evidence are often hidden in a huge amount of other entries. Therefore, using the list view and histogram is necessary to focus attention on relevant parts of the dataset. Having put all the collected information together, we compiled a precise summary of the incident and its timeline:

S1: 2016-05-25, 00:40: The attacker illegally logged in the account of user *martin* using SSH for remote access. Further analysis showed that the attacker abused unsecured NFS access to `/home` directory, allowing to upload of files and execution of privileged binaries. This is the only part of the analysis that could not be done just with the file system metadata, but the provided file system evidence gave a precise lead about what to check in the system logs and configuration.

S2: 2016-05-25, 02:40: The attacker installed a trojan code. A purportedly malicious `libselinux` library was downloaded using the `wget` command, and the system configuration (in file `/etc/ld.so.preload`) was changed to likely inject the library into every newly created process. The SSH service was restarted to activate the trojan code (either a backdoor and/or credential-stealing). A suspicious s-bit file `/var/lib/.s` was installed simultaneously, probably to trigger the illicit activities.

S3: 2016-05-25, 19:20: There are suspicious activities in the account of user `roberto`. This account was probably also compromised a few hours later by the attacker as both the accounts show similar signs, e.g., an empty file named 1. The reason is uncertain. However, there is no evidence that this account was used for suspicious activities.

S4: 2016-05-25, 21:22: The attacker re-compiled and re-installed the trojan code. The attacker was probably not satisfied with the version they deployed at the beginning of the day, so they returned, re-compiled the `libselinux` library, and then produced another binary on the spot.

S5: 2016-05-25, 22:08: The attacker created a hidden directory '`/var/tmp/...`', where they compiled some suspicious tools, e.g., `pcap` or `nmap`, and installed them into the system. Following that, they started a network scan and used the directory to store results obtained for individual network targets. Since then, the data was kept being captured and logged into this directory. The directory is used for a massive scan spanning almost two days, which is visible from the relevant histogram, see Figure 5.

S6: 2016-05-26, 23:12: The system files with user account and passwords (`/etc/shadow` and `/etc/passwd`) were modified

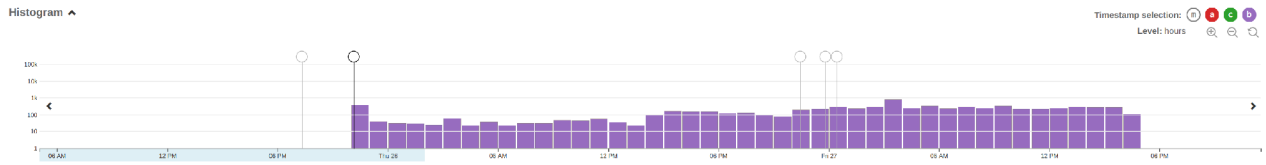

Figure 5: Indication of a continuous creation of files generated by the network scanner.

one day later. It is uncertain whether this activity is related to the incident or not.

7.3 Apparatus

The server part of the FIMETIS application was deployed on a common cloud machine, equipped with 8GB RAM, 80GB disk space and 4 CPUs. We conducted the evaluation online using Google Meet. The participants used Google Chrome on their computers or laptops with resolutions ranging from FullHD to UHD. Their interaction and comments were recorded for later analysis.

7.4 Procedure

The user study was divided into four parts. First, the participants were introduced to the general procedure, signed a consent form, and filled the demography questionnaire. Then, the experimenters presented the tool, explained all its features using the first dataset, and let the participant familiarize with the tool for 5–10 minutes.

Next, the participants were to find the following signs of the file system manipulation and usage:

- T1: Files or directories with suspicious names.
- T2: System files (configurations or executables) possibly modified by the attacker.
- T3: Executables or libraries that were not installed from its package (i.e., either directly downloaded or manually compiled on the system).
- T4: Privileged executables (with root s-bit) possibly used in the attack.
- T5: Suspicious or unusual commands possibly executed by the attacker.
- T6: Possibly compromised user accounts.

These tasks address requirements **R1–R5**. Together, they should provide an overview of what happened during the incident. While the tasks *T1,T2,T4, and T6* reflect different aspects of the detection of file system anomalies (**R3**), *T5* and *T3* are related to the execution of suspicious commands (**R4**) and traces of batch processing (**R5**) respectively. All the tasks require iterative exploration of the file system structure (**R1**) and temporal relationships (**R2**).

The participants had the tasks printed out so that they could easily make notes. The experimenter asked the participants to solve the tasks iteratively in any order. They were asked to think aloud. At the end of this evaluation phase, they had to summarize the incident upon their observations.

Although the real investigation of an incident lasts many hours or can even spread to several days, we restricted the participants to roughly one hour. The study's goal was not to get all the details about the attack, which is usually not possible without additional pieces of information such as system logs or network traffic, but to ascertain whether the analyst can get a quick insight into the incident using our tool.

When the incident investigation ended, the participant filled the usability questionnaire (Simple Ease Question, SEQ [23]), and System Usability Scale, SUS [22]. Finally, the experimenter interviewed participants on their final thoughts and feature requests.

7.5 Limitations

This user study has several limitations. The number of participants is relatively low. The reason lies in the time demands put on the evaluation process, which took roughly two hours per participant. To minimize the impact of this limitation, we involved security practitioners – possible users of the tool. On the other hand, we aimed to cover a wide range of expertise. Therefore, we engaged both highly skilled experts who have practical experience with collecting evidence from file systems and professionals who lack these specific skills as they focus on other cybersecurity domain, e.g., network analysis or cybersecurity research.

We are also aware that the evaluation was performed with only one test case, and then the results could be affected by the specific attack vector hidden in the dataset. We strove for authenticity, and then we preferred a real incident from artificial data. On the other hand, we aimed to choose an incident which is typical in a sense. The selected dataset contains the digital evidence of common attack steps like the abuse of user accounts, privilege escalation, installation of backdoor, and using the compromised host for further illegal activities.

7.6 Results

Usability & learnability: User experience with the tool was evaluated by the System Usability Scale (SUS). SUS is a de facto standard method for assessing systems' usability regardless of their purpose. The average SUS score of FIMETIS was 88.5. According to the adjective ratings [2], the score corresponds to *excellent* ratings and proves compliance with **R6**.

SUS questions #4 (I think that I would need the support of a technical person to be able to use this product) and #10 (I needed to learn many things before I could get going with this product) can also be used to interpret learnability [22]. The average answers 1.2 and 1.8, respectively, on the Likert scale from 1: "strongly agree" to 5: "strongly disagree" suggest that FIMETIS is also easy to learn.

Preferences in using visual-analytic elements: FIMETIS is designed as a generic tool where hypotheses can be verified in various ways using the combination of diverse visual-analytical elements. To explore if some elements are more popular then other, we analyzed videos captured during the evaluation. We measured the usage of key interactions and data filtering concepts: filtering data by attributes, using predefined clusters, filtering data by span windows, searching and filtering by path, and using push-pins.

The results are summarized in Figure 6. Push-pins represent the maximal number of bookmarks used by the analyst at the same time (20 push-pins in the participant P5). The other axes encode the relative time the analyst used the element. The time is expressed as the percentage of the investigation time. It is to be pointed out that the *name filtering* is used occasionally for temporal filtering and navigation during the interaction with the *List View*. Therefore, its usage can be underestimated in the radar charts.

The radar charts depicted show that different analysts preferred different combinations of elements. Usually, only 2–3 elements are used intensively, while others are ignored either completely or used significantly less. Another interesting observation, which is not captured in the radar charts, is that the analysts used only one span window. P1 did not use this element, and P3 used two span windows simultaneously, but only for a very short time.

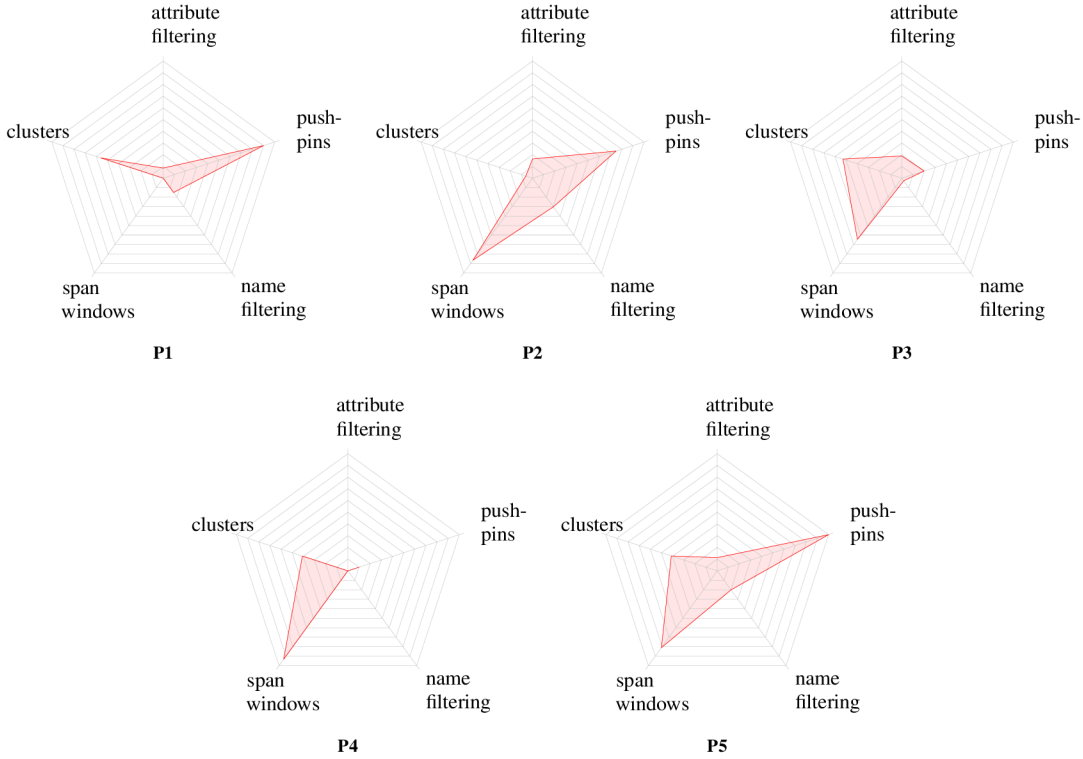

Figure 6: Approximate utilization of visual-analytic elements by individual P1–P5. The *push-pins* axis encodes maximal number of bookmarks used simultaneously. Other axes represent the relative time (as the percentage of investigation time) when the element was used.

Precision of the attack timeline: To evaluate the ability of the FIMETIS tool to provide a quick insight into the incident timeline, incident scenarios reported by participants were compared with the baseline scenario *S1–S6*. The precision was ranked by the authors of the paper. The results are summarized in table 2.

	S1	S2	S3	S4	S5	S6
P1	●	◐	●	○	●	●
P2	●	●	●	○	●	●
P3	◐	◐	◐	○	●	●
P4	●	●	●	◐	●	●
P5	◐	◐	◐	○	●	●

Table 2: Precision of the attack reconstruction: ○ overlooked/not identified, ◐ identified partially, ● identified correctly.

S1 (compromising the account 'martin') was identified by all participants. However, P3 and P5 identified the account together with 'roberto'. They did not decide who was the primary target of the attacker.

S2 (installation of a trojan code) was identified by all participants, but the level of observed details varied. All the participants discovered the `/var/lib/.s` as part of the attack vector, but P1, P3, and P5 did not provide more details about this attack phase. Moreover, the `selinux` library was completely overlooked by them. P2 did not mention the restart of the SSH server, but SSH was correctly identified as the service used for the escalation of privileges. P4 noticed and described all the details related to this attack phase, including the usage of `/etc/ld.so.preload`.

S3 (suspicious manipulation with the account 'roberto') was identified by all participants and considered part of the attack. Neither participant found the real abuse of this account. However, P3 and P5 did not decide whether the 'roberto' or 'martin' was the primary access point for the attacker.

S4 (re-compilation and new installation of the trojan code) was overlooked by all participants except P4. This analyst noticed the re-installation but overlooked the re-compilation of the trojan code at the compromised computer.

S5 (a hidden directory) was identified by all participants very quickly. The directory contained almost 12.000 records combining source code of multiple tools, traces of their compilation and usage, and data files gathered by the attacker. Nevertheless, the analysts were able to spot tools and data relevant to the attack vector and directly describe their purpose in the attack (P2, P3, P4, P5) or at least mention them as a tool worth further exploration (P1).

S6 (modification of the user account database) was identified by all participants. P1 noticed the changes but finally considered as not being linked to the incident. P2 did not provide more details. Other analysts considered the changes to be part of the attack when the attacker probably created a new user for later access.

Tasks difficulty: To evaluate the usability of the tool for solving individual tasks *T1–T6*, we analyzed the SEQ answers. We used this method because our tasks were too complex for metrics such as task duration time or completion rate, and the method performs as good as more complicated measures of task difficulty [23]. The participants responded to a single question associated with individual tasks ("Overall, how difficult or easy did you find this task?"), using a scale from 1 (very easy) to 5 (very difficult). The box plot is depicted in Figure 7.

Overall, the participants considered tasks rather easy with the FIMETIS tool. This result correlates with the analysts' success to correctly reconstruct the incident in limited time at an appropriate level of detail. The only exception was finding out executables or

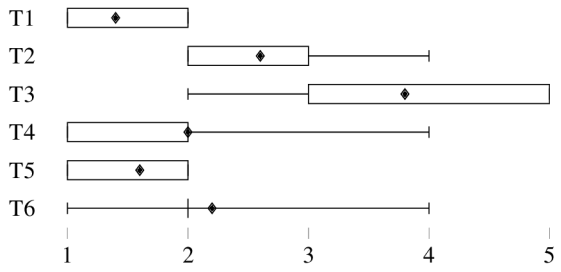

Figure 7: Distribution of answers to SEQ tasks (min/max values, lower/upper quartile, and average). Lower score is better (1 = Very easy, 5 = Very difficult).

libraries that were not installed from its package (*T3*). This task is considered rather difficult. However, this result also corresponds to the low success rate of revealing the re-compilation of a trojan code (step *S4* of the incident). The reason probably lies in the complexity of the task, which forces the analyst to iteratively combine multiple views and combine multiple features of the tool.

8 Discussion and Future Work

The work we presented in this paper focuses on the design and user evaluation of a visual-analytics tool that aims to support efficient disk snapshot exploration as part of the cybersecurity incident investigation workflow.

We collaborated with three skilled investigators on the clarification of forensic processes and the specification of requirements. The evaluation conducted with five cybersecurity experts revealed that the analytical tool built upon these requirements is intuitive and easy to use. All of the analysts were able to provide an incident report at surprising precision in very limited time. Moreover, it seems that the results obtained from less and more skilled analysts are subtle. We are aware that it could be affected by the attack vector of the incident selected for the evaluation, but this unexpected finding is promising for further development.

Another interesting observation was made regarding the usage of proposed visual-analytics concepts and their combinations. We noticed different workflows in using the tool by different analysts. This finding indicates that the tool is sufficiently generic. It supports various approaches to the verification of hypotheses and collecting the evidence. Moreover, the results captured in Figure 6 suggest that there could exist a favorite combination of analytical elements. For example, the analysts P2 and P5 used predominantly span windows with name filtering and a lot of push-pins, while P3 and P4 preferred span windows and clusters combined with only a few push-pins. Exploring such behavioral patterns would bring insight into analytical strategies. However, it requires a much deeper evaluation and analysis in future work.

Our work is still in progress. During the user study, we collected user feedback and requests for additional useful features.

File system attributes management: Multiple analysts forgot to cancel the per-attribute filtering during the investigation. This mistake led to false hypotheses and delay in the investigation. Emphasizing this filter or indicating that the *List View* contains only entries with selected modifications are required.

Dealing with file system records: The *List View* is the primary source of information for investigators, and efficient manipulation with records has shown to be the key factor for the investigation process. In spite of searching, filtering, and smart navigation techniques implemented in the *List View*, the analysts requested even more features for rapid navigation in the list. Especially, scrolling the list to a record by CTRL+F hotkey was missing. Currently, only highlighting and filtering out the data by the typed text is implemented in the tool. Also, the support of regular expressions and hiding records matching the typed text temporarily were required. Complementary hierarchical views to the strictly temporal ordering of records, e.g., using treemaps to convey space requirements of file system parts, reveal anomalies, and navigate to them quickly, will be considered in the future work.

The current implementation of FIMETIS serves as an analytical and decision-making tool for file system metadata analysis (Figure 2). Although the evaluation proved the usefulness of the tool, users ask for the support of other parts of the investigation process as well. Reaching this goal requires making significant extensions to current functionality and then to the design. In what follows, we outline key requirements and their possible impact on visualizations and GUIs.

Incident report creation: Incident reports are key outputs of the investigation process. As a lot of clues and pieces of the incident evidence appear during the interaction, it would be useful to use them for the report creation. Apart from online notes that have already been integrated into the new version of FIMETIS, investigators' feedback revealed possible changes in using bookmarks for this purpose. Currently, bookmarks are very simple. They are represented as push-pins referring to interesting records (points in time) and used for fast navigation (jumping to these records). Multiple analysts were asking for the possibility to distinguish between push-pins by color, tagging them, and making their own notes. Once the concept of bookmarks is moved from push-pins to advanced annotations, it would be possible to use them for the direct generation of incident reports or their parts.

Analysis of system logs: File system metadata represents only one source of information for investigators. Other data sources, like system logs or network traffic data, are often available to provide a broader context. Especially so-called super-timelines, i.e., file system metadata merged with system logs, are often used for forensic investigation. Extending FIMETIS with system logs should be possible. Both types of data sources are time series. The proposed approaches to file system exploration seem to be reusable also for system logs. However, further research and evaluation are needed. It is especially necessary to balance between unified exploration, when an analyst uses both data types together, and distinguishing both contexts as they represent different knowledge with possibly different uncertainty.

Other information sources: Ability to analyze other data sources like network traffic or memory snapshots are required by forensic investigators as well. However, they encode very different data with very different abstractions that require the application of specific visual-analysis techniques and concepts. Therefore, narrowly focused tools are designed that provide comprehensive visual-analytics interfaces [6]. Joining these information sources into a single "silver bullet" analytical tool can be counter-productive and going against the **R6** requirement.

We aim to address the aforementioned features and enhancements in future work. As the FIMETIS application is already used in practice for the investigation of real-world incidents (three incidents were successfully investigated by the security teams of Masaryk University and CESNET so far), we aim to utilize this experience to extend the functionality of the application further. Especially, we plan to introduce advanced user-defined clusters and the support of multiple timelines, e.g., records of system logs. These extensions will require changes in the current design and the development of new visual-analytic methods to cope with even bigger and more variable data.

Acknowledgments

This work was supported by ERDF "CyberSecurity, CyberCrime and Critical Information Infrastructures Center of Excellence" (No. CZ.02.1.01/0.0/0.0/16_019/0000822).

References

[1] R. Anderson, C. Barton, R. Boehme, R. Clayton, C. Ganan, M. Levi, T. Moore, and M. Vasek. Measuring the Cost of Cybercrime. In *Proceedings of the 18th Annual Workshop on the Economics of Information Security*, 2019.

[2] A. Bangor, P. Kortum, and J. Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3):114–123, May 2009.

[3] A. Boschetti, L. Salgarelli, C. Muelder, and K.-L. Ma. TVi: a visual querying system for network monitoring and anomaly detection. In *Proceedings of the 8th international symposium on visualization for cyber security*, pages 1–10, 2011.

[4] F. Buchholz and E. Spafford. On the role of file system metadata in digital forensics. *Digital Investigation*, 1(4):298 – 309, 2004.

[5] F. P. Buchholz and C. Falk. Design and Implementation of Zeitline: a Forensic Timeline Editor. In *Proceedings of the fifth annual DRFWS Conference*, 2005.

[6] B. Cappers. *Interactive visualization of event logs for cybersecurity*. PhD thesis, Department of Mathematics and Computer Science, Dec. 2018. Proefschrift.

[7] B. Carrier. *File System Forensic Analysis*. Addison-Wesley Professional, 2005.

[8] E. Casey. *Handbook of Digital Forensics and Investigation*. Academic Press, Inc., 2009.

[9] L. Caviglione, S. Wendzel, and W. Mazurczyk. The Future of Digital Forensics: Challenges and the Road Ahead. *IEEE Security & Privacy*, 15(6):12–17, 2017.

[10] Gartner, Inc. Gartner Forecasts Worldwide Information Security Spending to Exceed $124 Billion in 2019. `https://muni.cz/go/c7a9e9`, August 2018.

[11] C. C. Gray, P. D. Ritsos, and J. C. Roberts. Contextual network navigation to provide situational awareness for network administrators. In *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–8. IEEE, 2015.

[12] C. Hargreaves and J. Patterson. An automated timeline reconstruction approach for digital forensic investigations. *Digital Investigation*, 9:S69 – S79, 2012.

[13] A. Heitzmann, B. Palazzi, C. Papamanthou, and R. Tamassia. Effective visualization of file system access-control. In *International Workshop on Visualization for Computer Security*, pages 18–25. Springer, 2008.

[14] C. Humphries, N. Prigent, C. Bidan, and F. Majorczyk. Elvis: Extensible log visualization. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, pages 9–16, 2013.

[15] C. Humphries, N. Prigent, C. Bidan, and F. Majorczyk. Corgi: Combination, organization and reconstruction through graphical interactions. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, pages 57–64, 2014.

[16] S. Kälber, A. Dewald, and F. C. Freiling. Forensic Application-Fingerprinting Based on File System Metadata. In *Proceedings of the IEEE 2013 Seventh International Conference on IT Security Incident Management and IT Forensics*, pages 98–112, 2013.

[17] J. Kävrestad. *Fundamentals of Digital Forensics: Theory, Methods, and Real-Life Applications*. Springer International Publishing, 2018.

[18] T. R. Leschke and C. Nicholas. Change-link 2.0: a digital forensic tool for visualizing changes to shadow volume data. In *Proceedings of the Tenth Workshop on Visualization for Cyber Security*, pages 17–24, 2013.

[19] J. Olsson and M. Boldt. Computer forensic timeline visualization tool. *Digital Investigation*, 6:S78 – S87, 2009. The Proceedings of the Ninth Annual DFRWS Conference.

[20] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, pages 61–71. IEEE, 2007.

[21] N. Rowe and S. Garfinkel. Finding Anomalous and Suspicious Files from Directory Metadata on a Large Corpus. In *Proceedings of the Digital Forensics and Cyber Crime*, 2011.

[22] J. Sauro. *A Practical Guide to the System Usability Scale: Background, Benchmarks & Best Practices*. CreateSpace Independent Publishing Platform, 2011.

[23] J. Sauro and J. S. Dumas. Comparison of three one-question, post-task usability questionnaires. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1599–1608, New York, NY, USA, 2009. ACM.

[24] M. Scherr. Multiple and coordinated views in information visualization. *Trends in Information Visualization*, 38:1–33, 2008.

[25] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, Dec 2012.

[26] J.-E. Stange, M. Dörk, J. Landstorfer, and R. Wettach. Visual filter: graphical exploration of network security log files. In *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*, pages 41–48, 2014.

[27] A. Ulmer, D. Sessler, and J. Kohlhammer. Netcapvis: Web-based progressive visual analytics for network packet captures. In *2019 IEEE Symposium on Visualization for Cyber Security (VizSec)*, 2019.