Interactive Temporal Digital Forensic Event Analysis

Nikolai Adderley, Gilbert Peterson

Chapter 3

INTERACTIVE TEMPORAL DIGITAL FORENSIC EVENT ANALYSIS

Nikolai Adderley and Gilbert Peterson

Abstract Current digital forensic tools and applications lack the capability to visually present high-level system events and their associated low-level traces in a user interpretable form. This chapter describes the Temporal Analysis Integration Management Application (TAIMA), an interactive graphical user interface that renders graph-based information visualizations for digital forensic event reconstruction. By leveraging correlation and abstraction as core functions, TAIMA reduces the manual, labor-intensive efforts needed to conduct timeline analyses during digital forensic examinations. A pilot usability study conducted to evaluate TAIMA supports the claim that correlation and abstraction of low-level events into high-level system events can enhance digital forensic examinations.

Keywords: Automated event reconstruction, information visualization

1. Introduction

The discipline of digital forensics has been under constant pressure as advancements in digital device technology outpace the technical capabilities of digital forensic tools and applications [20]. Exacerbating the issue is the increased use of computers in the commission of crimes [10]. Continuous increases in the amount of heterogeneous data involved in investigations have made digital forensic analyses complex and time-consuming.

Current digital forensic analysis applications are primarily trace-based and force practitioners to rely on manual, labor-intensive practices for performing correlations and reconstructing events [18, 20]. As a result, it is difficult to establish a holistic understanding of an entire system

image and to identify patterns and anomalies in a reasonable amount of time.

Information visualization (InfoVis) and abstraction leverage human perceptual and intellectual capabilities to reduce forensic practitioner workload and analysis time [23]. Specifically, information visualization takes advantage of human visual and analytical capabilities to explore data; data exploration is conducted using visual displays that offer flexible data encodings in perceptually effective environments [8]. Abstraction reduces the amount of data displayed to users and minimizes the adverse effects of text-based information overload [19, 22]. Studies that combine information visualization and abstraction reveal that digital forensic practitioners appreciate displays that minimize the number of items for review, but still present relevant information [9].

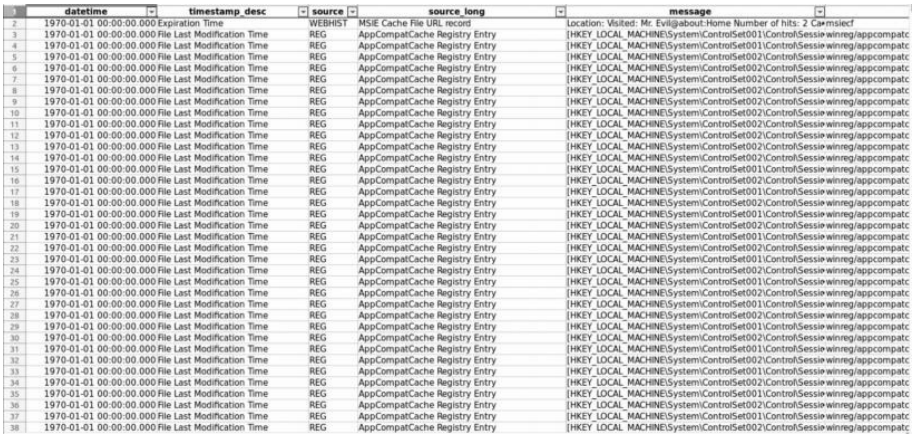
This chapter describes the Temporal Analysis Integration Management Application (TAIMA), a proof-of-concept information visualization application that enhances digital forensic investigations with an emphasis on the analysis phase of the digital forensic process. TAIMA leverages temporal system event reconstruction and information visualization to enrich a graphical timeline with discrete high-level system events. The information visualization component enables a practitioner to adjust the focus from a case-wide overview to a detailed view of low-level traces. The detailed view enables the practitioner to confirm the accuracy of reconstruction.

A usability study of TAIMA was conducted in which digital forensic practitioners completed a simulated digital forensic analysis task. During the study, all the participants were able to locate all the evidence items. The post-task survey results reveal that all the participants found TAIMA to be intuitive and easy to learn. Additionally, the participants felt that the visualization was effective at helping them complete their tasks. The results demonstrate the power of correlation and abstraction in supporting timeline analyses.

2. Related Work

Timeline analysis of digital system events helps identify when events occurred and the order in which they occurred. Timestamp data from multiple data sources (e.g., registry files, event logs, link files and prefetch files) help clarify the temporal proximity of system traces, test investigative hypotheses and identify additional system traces of interest that would have been overlooked without timeline examination [11].

Despite the critical role that timelines play in forensic investigations, most industry-standard digital forensic applications merely focus on data



	datetime	timestamp_desc	source	source_long	message
2	1970-01-01 00:00:00.000	Expiration Time	WEBHIST	MSIE Cache File URL record	Location: Visited: Mr. Evil@about:Home Number of hits: 2 Ca msiecf
3	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
4	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
5	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
6	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
7	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
8	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
9	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
10	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
11	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
12	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
13	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
14	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
15	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
16	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
17	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
18	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
19	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
20	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
21	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
22	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
23	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
24	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
25	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
26	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
27	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
28	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
29	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
30	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
31	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
32	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
33	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
34	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
35	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
36	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc
37	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet001\Control\Sessio winreg/appcompatc
38	1970-01-01 00:00:00.000	File Last Modification Time	REG	AppCompatCache Registry Entry	[HKEY_LOCAL_MACHINE\System\ControlSet002\Control\Sessio winreg/appcompatc

Figure 1. `log2timeline` CSV output.

collection. After importing artifacts into their applications, practitioners are left with vast amounts of low-level traces. The majority of industry applications either have limited capabilities for generating timelines or entirely lack the capabilities [3]. The applications that do generate timelines often provide static timelines or histograms, or simply export digital artifacts as comma-separated value (CSV) reports (Figure 1).

For example, Encase (version 8.07) [7] generates a calendar timeline with individual artifacts represented as dots on the calendar (Figure 2). Forensic Toolkit (FTK) [1] generates a histogram timeline (Figure 3).

Previous work in digital forensic visualization has highlighted the difficulty in handling the vast amounts of extracted data involved in timeline analyses [19]. Carbone and Bean [3] describe the lack of intuitive GUIs and integrated timelines when dealing with large heterogeneous datasets. Without careful planning, visualizations quickly become overwhelming; the overcrowded displays cause information overloads. Gudjonsson [6] notes that practitioners often struggle to complete forensic analyses using timelines that are overcrowded and stresses the importance of reducing the amount of reviewed data to facilitate timeline analyses.

Olsson and Boldt [18] demonstrate the advantages of using the CyberForensics TimeLab (CFTL) graphical timeline tool over Forensic Toolkit (FTK); the post questionnaire from their study reveals that participants solved a hypothetical case "significantly faster using CFTL than when using FTK." Teelink and Erbacher [23] demonstrate that visualization techniques assist practitioners in the forensic data analysis process. The two studies combined interactive capabilities with visualization tools. The combination resulted in practitioners experiencing improvements

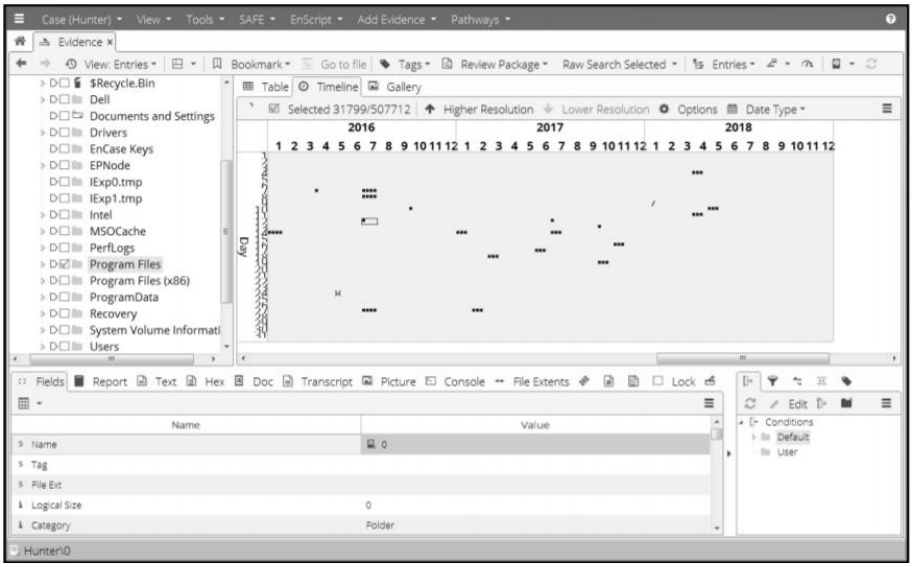


Figure 2. Encase timeline view.

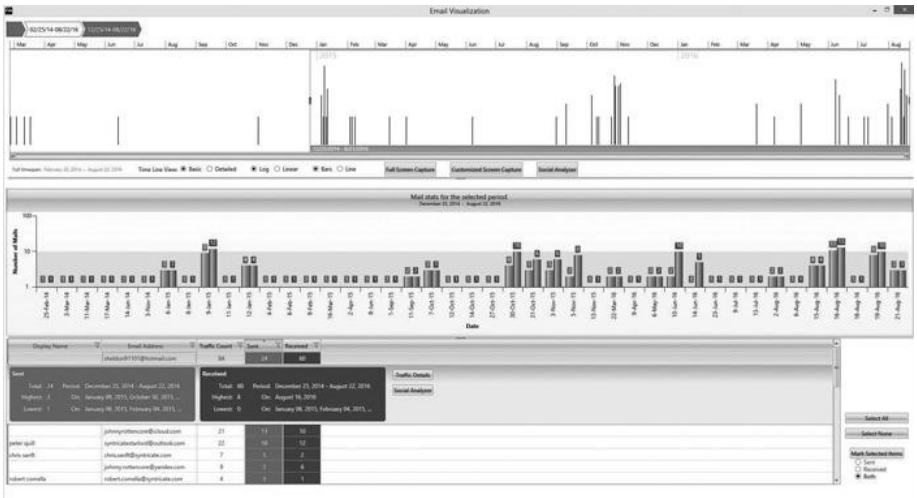


Figure 3. Forensic Toolkit timeline view.

in the digital forensic process and reductions in the time required to identify suspicious files.

As a visualization tool, TAIMA enhances digital forensic examinations by providing practitioners with an interactive environment integrated with visual representations of digital evidence. The interactive

capabilities and visualizations enabled practitioners to answer all the investigative questions posed in a user evaluation study.

3. TAIMA

TAIMA displays digital evidence on a graph-based timeline to enhance forensic analyses and facilitate event reconstruction. The primary purpose of TAIMA is to provide an overview of the types of system events that exist on a media image. TAIMA's intuitive GUI precludes users from having to learn a programming language or use a command line interface.

At the core of TAIMA is the rendering of a graph-based timeline that overcomes the effects of information overload. Using only a date/time range as a search parameter, a practitioner is able to identify suspicious files and events without labor-intensive manual exploration, or any plug-ins or code. TAIMA displays high-level system events on a single screen along with the locations of the traces that are related to the high-level events.

User accessibility was a core function goal during the TAIMA design phase. The application was intended to be used by technical as well as non-technical practitioners. Industry tools that provide extensive features and capabilities can be overwhelming. For example, Encase is not user friendly enough for non-technical practitioners. TAIMA, which is backed by the Neo4j graph-based database, provides non-technical users with the ability to issue database queries using only date/time ranges; no programming skills or additional plugins are required.

3.1 Design Principles

The development strategy for TAIMA followed Shneiderman's interactive GUI design principles [22]. Shneiderman proposed a user-centered GUI design guide model that supports the use of information visualization in digital forensic applications. He stresses the importance of providing a modern information visualization GUI that supports the processing of large volumes of heterogeneous data.

The TAIMA development process also followed the Visual Information Seeking Mantra (overview first, zoom and filter, details on demand) [19]. The mantra specifies information visualization design techniques and interactive controls for presenting data in an organized and intuitive manner that enables easy traversal. To fulfill the mantra requirements, TAIMA first presents an overview of all the high-level system events in a time-span as discrete color-coded tiles on a timeline. The graph-based timeline display enables a practitioner to view the temporal proximity

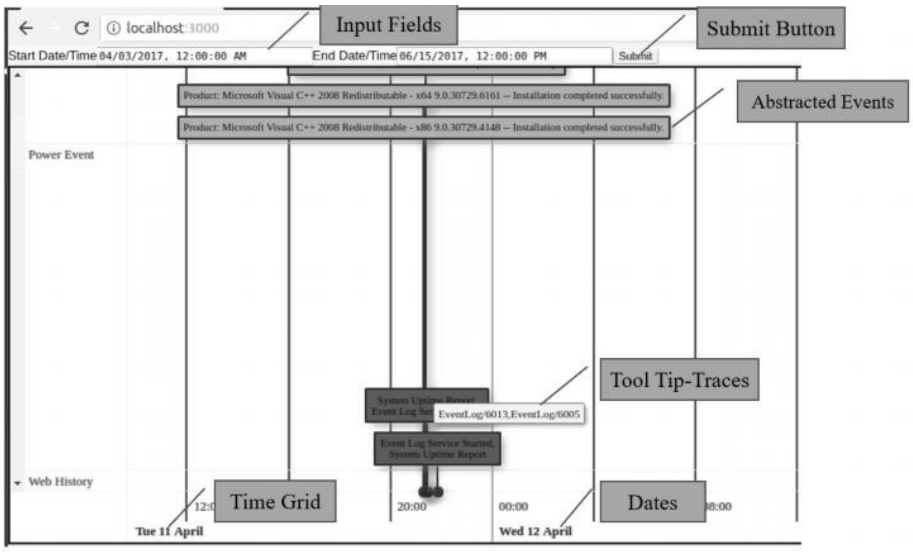


Figure 4. TAIMA GUI.

of system events quickly and also provides an easy way to identify (at a glance on a single screen) clusters of system events. Moreover, the practitioner can adjust the point-of-view by zooming in on a timeline to adjust the scale for detailed views of specific time intervals.

TAIMA was designed to minimize the effects of large, complex (heterogeneous) data volumes on the digital forensic investigation process, especially during the analysis phase. Accordingly, TAIMA was built using the GRANDstack (GraphQL, React, Apollo, Neo4j Database) architecture. GRANDstack is an ecosystem of software applications that are used to create full-stack web and mobile GUIs [5]. The integration of the applications allows for a scalable JavaScript web application backed by a Neo4j database [16]. An important advantage of the GRANDstack ecosystem is its provision of a modern web browser, which reduces the TAIMA learning time.

3.2 GUI Timeline

Graph-based digital forensic timelines mitigate many of the challenges encountered when attempting to analyze vast volumes of data [4, 8, 18].

Figure 4 shows the principal TAIMA GUI interactions. The user is presented with the React GUI front-end to enter a time interval of interest (start time and end time). Clicking the submit button sends a GraphQL query via the GraphQL service with the timestamps as search

parameters to the Neo4j database. GraphQL is a query language that enables developers to specify schema definitions to express the data to be requested from a datastore. The Apollo client integrates with GraphQL to process and route data requests to the Neo4j database [2]. The GraphQL server implements the logic for querying the modeled Neo4j database (via the addition of abstraction nodes) to search for high-level events based on their temporal attributes. After fetching the data, the Apollo client sends the results to the GraphQL server. The React GUI integration with the Apollo client is configured to store the results of the query within a React component to render the visualization. The database query results are automatically converted to graphical form and presented to the user on a graph-based timeline.

The TAIMA information visualization timeline component shown in Figure 4 displays discrete high-level system events chronologically. The graph-based timeline enables a practitioner to identify patterns and anomalies during data analysis while providing an overview that facilitates the overall understanding of system events.

Figure 4 also shows the various parts of a timeline: input fields, submit button, high-level events, traces via a tooltip, time grid and dates. The tooltip dialog box satisfies the "details on demand" interactive GUI requirement specified by Shneiderman [22]. This requirement enables a practitioner to view additional information about events by placing the mouse pointer over the event tile. A text-box displays the trace artifacts responsible for the event.

The visualization in Figure 4 shows four abstracted events. These include two program installation events (blue titles), denoted by "Installation Completed Successfully." Additionally, there are two power events (purple titles), denoted by "System Uptime Report/Event Log Service Started." The temporal proximity of the four high-level system events suggests that the system of interest was restarted due to the installation of Microsoft Visual C++ 2008 on 11 April 2017 at approximately 8PM local system time.

Figure 5 shows an overview of the activities on the timeline for the time interval 03 April 2017 12:00 AM to 15 June 2017 12:00 PM. Note that various programs (EXEs) were executed on the system of interest on 3 April 2017 and 11 April 2017.

3.3 Abstraction Technique

The abstraction technique implemented by TAIMA leverages the graph database generated via property graph event reconstruction (PGER) [21]. This technique extracts temporal traces from a media image and imports

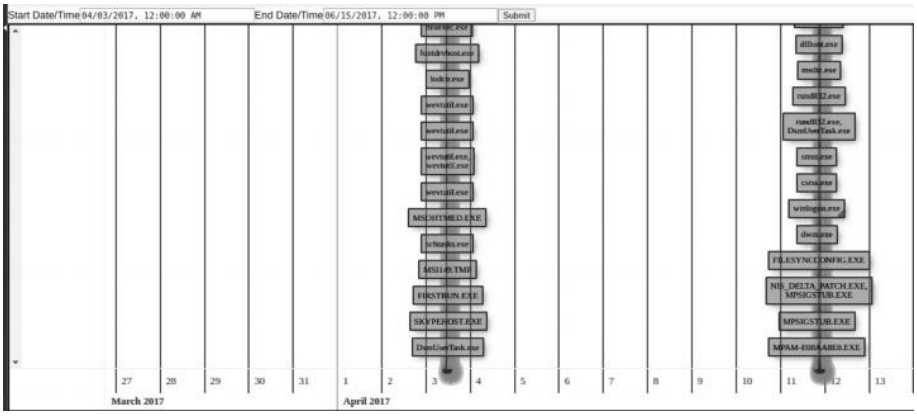


Figure 5. TAIMA global view.

them into a native labeled property graph that stores system events as nodes linked via cause-effect relationships. The links enable the temporal traces to be rapidly correlated and abstracted into a single narrative.

As a proof-of-concept tool, TAIMA supports five high-level abstraction system events: (i) program installation; (ii) power (startup/shutdown); (iii) program execution; (iv) file download; and (v) web history. The overall logic for creating abstraction events and relationships involves searching a time window for the declared low-level events. The time window is subsequently searched for related events. If the declared low-level events are found in the time window, then a high-level system event abstraction node is created; this creates a link. Multiple trace matching enforces the credibility of the high-level system event abstraction event. In addition to creating relationships, the low-level traces are used to enrich the abstracted event.

Program Installation. The Cypher query shown in Figure 6 creates a program installation abstraction node that is a relational correlation of four low-level traces associated with the particular program installation event.

Lines 1 through 5 of the Cypher query find all the action and object nodes associated with the event log entries that occurred within a ten-second window. Line 6 filters the object nodes down to three event log entries: (i) MsiInstaller/11707; (ii) MsiInstaller/1042; and (iii) MsiInstaller/1033. In Lines 7 through 8, the COLLECT command aggregates the action and object nodes based on time, and the DISTINCT commands filter the list to include only unique timestamps. This eliminates having multiple entries for a given timestamp; only unique timestamps

```
1   MATCH (:parser {parserName: "eventLog"}) <-[:PARSER] -
    (act:action)-[:EFFECTS]->(event:object)
2   MATCH (act)-[:AT_TIME]->(sec:Second)
3   MATCH p = (sec)-[:NEXT *10]->()
4   WITH p, event
5   UNWIND nodes(p) AS secNodes
6   MATCH (secNodes)<-[:AT_TIME]-(act:action)--(obj2:object)
7   WHERE obj2.filename IN ["MsiInstaller/1107",
    "MsiInstaller/1042", "MsiInstaller/1033"]
8   WITH act.timestamp as timestamp, COLLECT(DISTINCT
    act.message) as messages, COLLECT(DISTINCT obj2.filename)
    as filenames, COLLECT(DISTINCT act) as acts
9   CREATE (a:Abstraction{Event: 'Program Installation',
    Trigger:filenames, Description:messages, timestamp:timestamp})
10  FOREACH (act in acts | MERGE (act)-
    [:LVL1_ABSTRACTION_LINK]->(a))
    FOREACH (set in obj2s | MERGE (set)-
    [:LVL1_ABSTRACTION_LINK]->(a))
```

Figure 6. Program installation query.

are tracked. Line 9 creates the abstraction nodes. Finally, Line 10, creates a relationship (LVL1_ABSTRACTION_LINK) to the action (red) and object (blue) nodes associated with the program installation.

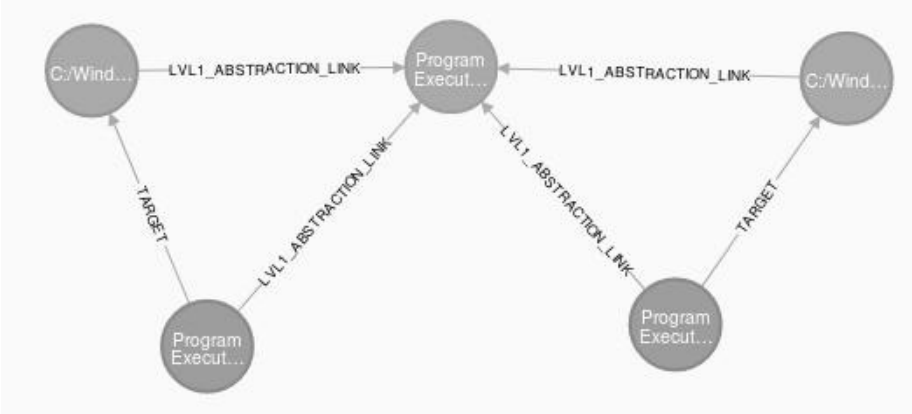


Figure 7. Program installation abstraction node linked to four traces.

Figure 7 shows a graph representation of the query result. The center node is the abstraction node, which is connected to four low-level nodes. High-level system events (i.e., center nodes) are only created

and added to the graph model if the criteria imposed by expert rules are satisfied. The event logs, MsiInstaller/11707, MsiInstaller/1042 and MsiInstaller/1033, constitute evidence that a program installation event occurred. New relationships (e.g., LVL1_ABSTRACTION_LINK) are added to the abstraction node to correlate the low-level traces connected to the program installation event. Aggregating low-level events and linking them to higher-level events reduce the amount of data presented to the practitioner and also increases the efficiency of the application (i.e., lower computational cost).

Power. Power events (shutdown, startup and sleep) are established by tracking the start and stop of the Windows Customer Experience Improvement (CEI) and Windows Event logs. The combination of these two traces is a strong indicator of power events. EventLog/6013 logs the total system running time after system boot-up. EventLog/6005 generates a log entry message that the Windows Event Log service was started.

Program Execution. The program execution abstraction is based on identifying prefetch file artifacts. The prefetch file, which is created every time an application is executed on a Windows system, contains information about the files associated with the application. Windows uses the prefetch information to reduce and optimize the loading time of the application during subsequent executions. Analyzing prefetch files provides insights into the files used by applications and the files that the Windows operating system loads at startup.

File Download. The download abstraction event nodes include Firefox and Chrome event log entries. The format string from the parser provides the complete path of the location of the downloaded file on the host.

Web History. The web history abstraction tracks the following traces: (i) source URL; (ii) web history; (iii) complete path of the downloaded file; and (iv) what happened to the file after it was downloaded.

4. Evaluating User Experience

TAIMA applies novel information visualization concepts and methods to digital forensic investigations. A usability evaluation study was conducted in order to examine their effectiveness. The study followed the user experience evaluation guidelines specified by Lam et al. [13]. Such an evaluation includes assessments that analyze individual responses and

attitudes towards a visualization [12]. The user evaluation performed in this research combined usability testing (UX) and the broadly-used post-study system usability questionnaire (PSSUQ), thereby drawing on their collective strengths [14].

The original PSSUQ comprises 19 items. However, this study employed a modified version that included 15 items. Since error handling was not implemented in TAIMA, the four related questions were removed from the PSSUQ.

Data from the user study included participant performance and open-ended feedback provided by a post-task questionnaire. The participant performance metrics included the task performance expressed in terms of the task completion time and completion rate (i.e., percentage of the six hacking software traces that were correctly found by a participant). Additionally, a subjective user satisfaction rating, captured via the post-task questionnaire, provided insights into the participants' overall satisfaction with TAIMA. The questionnaire also included an open-ended section for the participants to provide feedback about TAIMA.

4.1 Study Participants

According to Nielsen [17], approximately 90% of usability problems are discovered in usability evaluation studies with no more than five participants. This study had five participants, all of whom were either computer crime investigators or digital forensic analysts with one to two years of experience using industry-standard tools to analyze digital evidence.

Statistical analysis of the outcome was not appropriate due to the small sample size. The simulated hacking scenario reflected tasks that are typically performed in a digital forensic investigation. The participants also provided feedback on enhancing TAIMA.

4.2 Procedure

The usability testing portion of the evaluation employed a scenario involving an abandoned notebook computer suspected of being used for hacking purposes [15]. The participants were tasked with conducting a digital forensic analysis using TAIMA to identify hacking software applications on the abandoned computer. The task was deemed successful if all six hacking software applications were discovered.

Before starting their tasks, the participants were provided access to TAIMA to explore the visualization features and capabilities. The testing officially started after the participants verbally expressed that they

Table 1. Post-study system usability questionnaire (PSSUQ) usability metrics.

Item		Rating
1	Overall, I was satisfied with how easy it was to use this system	6.2
2	It was simple to use this system	6.4
3	I was able to complete the tasks and scenarios quickly using this system	5.8
4	I felt comfortable using this system	6.4
5	It was easy to learn to use this system	7.0
6	I believe I could become productive quickly using this system	6.4
7	Whenever I made a mistake using the system, I could recover easily and quickly	6.0
8	It was easy to find the information I needed	6.4
9	The visualization provided by the system was easy to understand	6.6
10	The visualization was effective at helping me complete the tasks and scenarios	6.6
11	The organization of information on the interface was clear	6.2
12	The interface of this system was pleasant	6.0
13	I liked using the interface of this system	6.0
14	This system has all the functions and capabilities I expect it to have	5.0
15	Overall, I was satisfied with this system	6.2

felt comfortable using TAIMA. After completing the task, the participants completed the PSSUQ.

The user study assessed two goals:

- Effectiveness of TAIMA at assisting forensic analyses by presenting digital evidence using best practice information visualization techniques.

- Effectiveness of the TAIMA infrastructure and processes at reducing the challenges associated with the examination and presentation of vast volumes of digital evidence.

The results of the assessment provided insights into the participants' attitudes towards the information visualization. Additionally, the assessment determined if TAIMA improved the analysis and presentation of large volumes of digital evidence.

4.3 Results

Table 1 and Figure 8 present the PSSUQ results. Note that higher scores denote better usability.

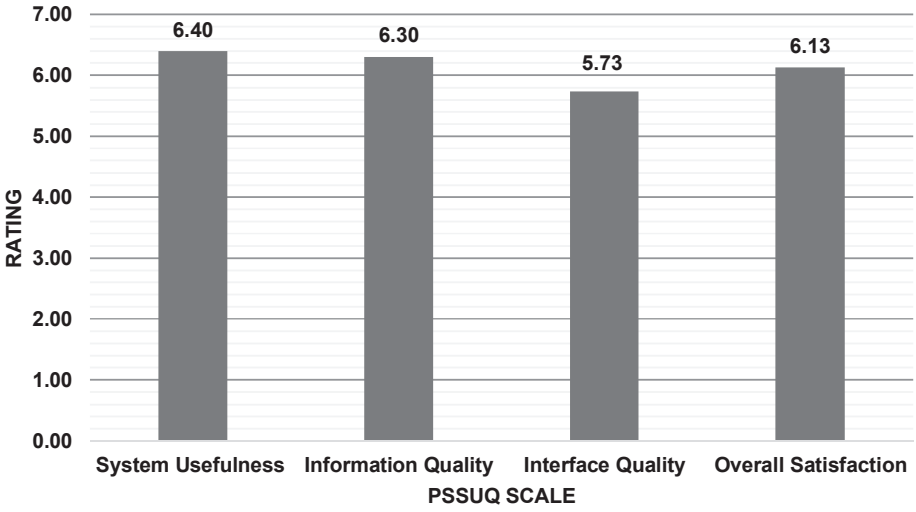


Figure 8. Post-study system usability questionnaire (PSSUQ) subscores.

The survey results reveal that the participants were highly satisfied with the usability of TAIMA – as indicated by a 100% (7 out of 7) rating for Item 5 (It was easy to learn to use this system). Item 9 (The visualization provided by the system was easy to understand) and Item 10 (The visualization was effective at helping me complete the tasks and scenarios) received the next highest rating of 94% (6.6 out of 7).

All the participants noted that they enjoyed not having to conduct an extensive search to find relevant artifacts. They also appreciated that important and relevant information was presented to them on one screen. Additionally, the participants found TAIMA to be easy to use due to its straightforward controls and intuitive display. This highlights the visualization effectiveness achieved by the intuitive display that reduces the data presented while still providing the critical information needed to complete the task.

One participant commented that the visualization was "extremely beneficial" and that it made observations of system activity "easy and fast." Another participant noted that it was easy to understand the visualization. This suggests that the integration of exploratory information visualization and abstraction techniques provides an accurate means to reconstruct timelines despite the challenges imposed by data complexity and data volume in digital forensic investigations.

Some items received low scores. The lowest score was received by Item 14 (This system has all the functions and capabilities I expect it to have). But this is a reasonable score because TAIMA is a prototype

system and the emphasis during development was on data reduction and accuracy.

The participants also provided suggestions on improving TAIMA. The suggestions included adding keyword search and file content viewing functionalities. This is because, during the testing, the participants wanted to search for particular files of interest instead of only using a data/time range. After these files were identified on the timeline, the participants did not want to have to search the timeline again for the files. In addition to viewing the locations of files of interest via a tooltip, participants were also interested in viewing the contents of the files. Also, TAIMA does not provide exporting or printing capabilities. The only option is to use print-screen to generate reports. Reporting is not only a desirable function, but also an essential part of the digital forensic process.

Finally, the participants felt that TAIMA provides limited functionality. They wanted more ways to customize TAIMA and wanted more control over the interface. For example, they wanted more filtering options. Also, after the results are returned and populated on the timeline, the participants wanted the ability to eliminate system events that were not of interest.

5. Conclusions

Establishing timelines is vital in digital forensic investigations. However, most digital forensic tools and applications merely present timelines as histograms or as raw trace entries in files. Additionally, traditional timeline analysis uses static text-based timelines that force practitioners to employ labor-intensive manual practices that often miss significant pieces of evidence.

The TAIMA prototype described in this chapter was specifically developed to mitigate these challenges. It leverages information visualization concepts and techniques to automate the creation of graph-based timelines of high-level system events. TAIMA enriches timelines with discrete high-level system events by presenting temporal data attributes in a practitioner-focused GUI.

The high usability ratings obtained by TAIMA in the user evaluation study reveal that it is intuitive, easy to learn, effective and accurate. These results satisfy the primary goal of the research – to address the significant challenges introduced by data complexity and data volume in digital forensic investigations.

Much of the future research and development activities related to TAIMA will be driven by the feedback received from the usability study

participants. Priorities include providing printing and data export capabilities. Even more important is supporting report generation, which is an essential component of the digital forensic process.

One of strengths of TAIMA is its ability to filter and reduce the numbers of events on timelines. Enabling users to customize filtering according to their needs should make the application even more effective. Future activities will also focus on the robust testing of TAIMA using a large image with real-world activities and complex system events.

References

[1] AccessData, Forensic Toolkit (FTK), Orem, Utah (`accessdata.com/products-services/forensic-toolkit-ftk`), 2020.

[2] Apollo Docs, Configuring the Cache, Apollo, San Francisco, California (`www.apollographql.com/docs/react/advanced/cach ing`), 2020.

[3] R. Carbone and C. Bean, Generating Computer Forensic Super-Timelines under Linux: A Comprehensive Guide for Windows-Based Disk Images, Technical Memorandum TM2011-216, Defence R&D Canada, Valcartier, Canada, 2011.

[4] Y. Chabot, A. Bertaux, C. Nicolle and T. Kechadi, Automatic timeline construction and analysis for computer forensic purposes, *Proceedings of the IEEE Joint Intelligence and Security Informatics Conference*, pp. 276–279, 2014.

[5] GRANDstack, Build Full Stack Graph Applications with Ease (`grandstack.io`), 2020.

[6] K. Gudjonsson, Mastering the Super Timeline with `log2timeline`, Information Security Reading Room, SANS Institute, Bethesda, Maryland, 2010.

[7] Guidance Software, EnCase Forensic User Guide, Version 8.07, Pasadena, California, 2018.

[8] G. Hales, Visualization of device datasets to assist digital forensic investigations, *Proceedings of the International Conference on Cyber Situational Awareness, Data Analytics and Assessment*, 2017.

[9] H. Hibshi, T. Vidas and L. Cranor, Usability of forensic tools: A user study, *Proceedings of the Sixth International Conference on IT Security Incident Management and IT Forensics*, pp. 81–91, 2011.

[10] P. Hitlin, Internet, social media use and device ownership in U.S. have plateaued after years of growth, *Fact Tank – News in Numbers*, Pew Research Center, Washington, DC, September 28, 2018.

[11] B. Inglot, L. Liu and N. Antonopoulos, A framework for enhanced timeline analysis in digital forensics, *Proceedings of the IEEE International Conference on Green Computing and Communications*, pp. 253–256, 2012.

[12] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair and T. Moller, A systematic review of the practice of evaluating visualization, *IEEE Transactions on Visualization Computer Graphics*, vol. 19(12), pp. 2818–2827, 2013.

[13] H. Lam, E. Bertini, P. Isenberg, C. Plaisant and S. Carpendale, Empirical studies in information visualization: Seven scenarios, *IEEE Transactions on Visualization and Computer Graphics*, vol. 18(9), pp. 1520–1536, 2012.

[14] J. Lewis, Psychometric evaluation of the post-study system usability questionnaire: The PSSUQ, *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 36(16), pp. 1259–1260, 1992.

[15] National Institute of Standards and Technology, Hacking Case, Gaithersburg, Maryland (`www.cfreds.nist.gov/Hacking_Case.html`), April 16, 2018.

[16] Neo4j, Introducing Neo4j, San Mateo, California (`neo4j.com`), 2020.

[17] J. Nielsen, Why you only need to test with 5 users, Nielsen Norman Group, Fremont, California (`www.nngroup.com/articles/why-you-only-need-to-test-with-5-users`), March 18, 2000.

[18] J. Olsson and M. Boldt, Computer forensic timeline visualization tool, *Digital Investigation*, vol. 6(S), pp. S78–S87, 2009.

[19] G. Osborne and J. Slay, Digital forensic infovis: An implementation of a process for visualization of digital evidence, *Proceedings of the Sixth International Conference on Availability, Reliability and Security*, pp. 196–201, 2011.

[20] G. Osborne, B. Turnbull and J. Slay, The "Explore, Investigate and Correlate" (EIC) conceptual framework for digital forensic information visualization, *Proceedings of the International Conference on Availability, Reliability and Security*, pp. 629–634, 2010.

[21] D. Schelkoph, G. Peterson and J. Okolica, Digital forensic event graph reconstruction, *Proceedings of the International Conference on Digital Forensics and Cyber Crime*, pp. 185–203, 2018.

[22] B. Shneiderman, The eyes have it: A task by data type taxonomy for information visualizations, *Proceedings of the IEEE Symposium on Visual Languages*, pp. 336–343, 1996.

[23] S. Teerlink and R. Erbacher, Improving the computer forensic analysis process through visualization, *Communications of the ACM*, vol. 49(2), pp. 71–75, 2006.