3-2017

Whitelisting System State in Windows Forensic Memory Visualizations

Joshua A. Lapso
Air Force Institute of Technology

Gilbert L. Peterson
Air Force Institute of Technology

James S. Okolica
Air Force Institute of Technology

Recommended Citation

Lapso, J. A., Peterson, G. L., & Okolica, J. S. (2017). Whitelisting system state in windows forensic memory visualizations. Digital Investigation, 20(March), 2–15. https://doi.org/10.1016/j.diin.2016.12.002

Whitelisting System State In Windows Forensic Memory Visualizations

Joshua A. Lapso*, Gilbert L. Peterson, James S. Okolica

Department of Electrical and Computer Engineering, Air Force Institute of Technology, USA

Abstract

Examiners in the field of digital forensics regularly encounter enormous amounts of data and must identify the few artifacts of evidentiary value. One challenge these examiners face is manual reconstruction of complex datasets with both hierarchical and associative relationships. The complexity of this data requires significant knowledge, training, and experience to correctly and efficiently examine. Current methods provide text-based representations or low-level visualizations, but levee the task of maintaining global context of system state on the examiner. This research presents a visualization tool that improves analysis methods through simultaneous representation of the hierarchical and associative relationships and local detailed data within a single page application. A novel whitelisting feature further improves analysis by eliminating items of less interest from view. Results from a pilot study demonstrate that the visualization tool can assist examiners to more accurately and quickly identify artifacts of interest.

Keywords: Memory forensics, Incident response, Information visualization, Forensic visualization tools, Single page web application, D3.js

1. Introduction

Modern criminal investigations frequently include evidence obtained from electronic devices such as computers, smart phones, tablets and even refrigerators. Hinshaw[1] estimates data storage is doubling every nine months, twice the rate of Moore's Law. As datasets grow with technology, the time required to analyze the data increases. Adding additional manpower is not a likely solution for reducing the temporal factor associated with data analysis[2], this is especially true in fiscally constrained environments.

*Corresponding author

Email addresses: joshua.lapso.1@us.af.mil (Joshua A. Lapso), gilbert.peterson@afit.edu (Gilbert L. Peterson), james.okolica@afit.edu (James S. Okolica)

Beebe and Clark [3] encourage further research in data mining such as information visualization (InfoVis), for immediate application in digital forensics discipline. This has proven successful in storage media[4], device interaction[5], memory[6] and triage[2] analysis methods. These visualization tools lead to more accurate identification of forensic artifacts by calling on the examiner's intuition and knowledge of the process. However, it still remains to be seen if forensic visualization tools provide faster results.

The objectives of this research are to unite three characteristics of analysis into a single memory visualization tool. The characteristics necessary to produce a successful memory visualization tool are:

1. Examiners must maintain local and global context throughout analysis.
2. Examiners must be able to quickly connect data.
3. A forensic visualization tool cannot be divorced from low level details that must be documented in an examiner's report.

To meet the objectives a fully functional memory analysis visualization tool is developed. To maintain global and local views and quickly connect data the tool simultaneously displays hierarchical and associative relationships. An additional aid to quickly connected data is a behavioral whitelisting function that filters processes of less interest from view.

The visualization tool uses an industry standard, single-page application model with database support. A searchable table module makes low-level details available to the examiner when needed. Additionally, sorting functions abstract the operating system and allow for future development.

The behavioral whitelisting feature, written as a server-side module, integrates with the database solution for speed and scalability. The whitelisting feature removes items of less interest from view, shrinking the examiner's search space. This feature allows an examiner to identify anomalies more rapidly.

A pilot study using human subjects supports that the visualization tool produces more accurate analysis and faster than traditional methods. The study evaluated participants completing forensic exercises with and without the memory visualization tool. Scored exercises establish a baseline for analysis, while post-study surveys provide qualitative data for content analysis. The themes selected for content analysis show the memory visualization tool produces more accurate artifact identification and with the whitelisting feature, reduces time spent on a task.

2. Background

Volatile memory, most commonly referred to as Random Access Memory (RAM), contains critical pieces of information about a system's state. This information is only available while the system is 'powered on' and is lost forever after a system is turned off[7]. There have been instances of Malicious Software (Malware) that only reside in RAM and thus would never be found on the other forms of computer storage media[8]. Also, other items of interest such as

active processes, open directory and file handles, current network connections, and command history can only be found in RAM[9][10]. Together these artifacts describe the full state of a machine at the time of discovery and allow examiners to infer user and system activity. Several methods exist for memory acquisition, both hardware[7] and software based[11][12][13] that first responders employ regularly.

There are numerous publications on the extraction of forensic evidence through hard drive imaging[4][14], registry keys[9], and memory captures[11][12][13]. However, there are fewer publications in InfoVis, for forensic analysis.

Forensic visualization tools establish state of the art analysis techniques for digital forensic examiners. Recent successes applying visual analysis techniques to storage media[4], device interaction[5], memory[6] and triage[2] encourage further research in this discipline.

Teerlink and Erbacher[4] prototyped a file search tool that visualizes file size, date, and type using unique shapes and colors. The result of their experiment showed that users found more files of interest with a visualization tool when compared to using traditional search strings. More compellingly, a single file, not found with search strings had a 100 percent find rate using their visualization tool. Had their experiment been a real investigation, key evidence would have been missed using traditional search strings.

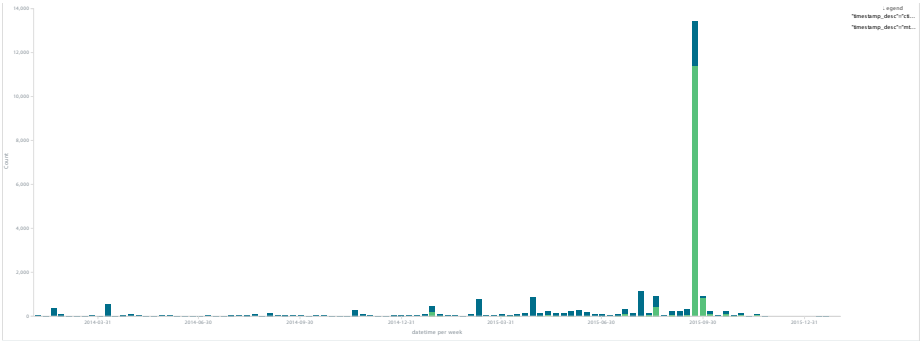
Osborne[5] introduced the Explore, Investigate and Correlate (EPIC) model which visualized events and interaction on and between devices. The EPIC tool provided two visualizations, one focused on inter-entity relationships and the other intra-device events such as "email, Short Message Service (SMS), Multimedia Messaging Service (MMS), phone call and website visit".

Henderson[2] explored a triage method for non-volatile storage media through a timeline visualization of modify, access and create disk activities similar to those in Figure 1. This tool would automatically generate a second timeline visualization focused around the period of greatest activity similar to Figure 1b. The resulting visualization limited the search space to a finite timeframe for investigators.

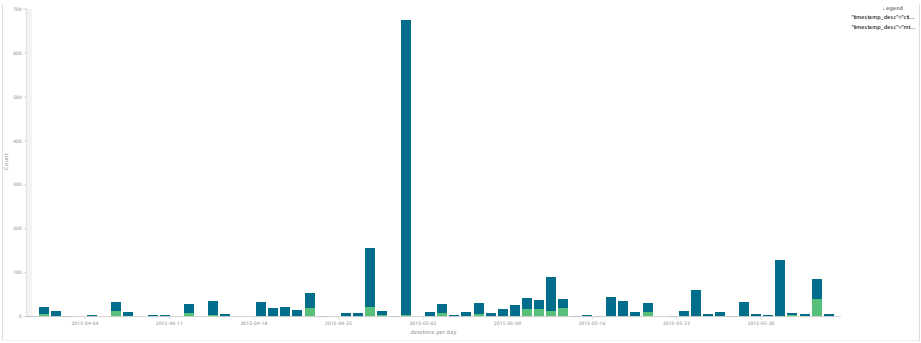
These visualization tools highlight the benefits of intuitive representations of large, interrelated datasets. However, there is still a need for data filtering to focus the scope of forensic analysts. The most obvious method for filtering data during static analysis is whitelisting. Whitelisting effectively skips over "known good" files and applications in order to shrink the search space. Traditional whitelisting methods, such as hashing and signature comparison, work well for file storage media such as hard drives, flash memory, and magnetic tape, but do not directly translate to system state information stored in volatile memory[15].

3. Memory Forensics Visualization

Modern incident response tools leave forensic examiners with an enormous collection of data and the daunting task of locating useful artifacts. This research seeks to improve speed and accuracy of artifact identification during the



(a) Full Timeline For Directory MAC Times.



(b) Period Of Highest Activity Timeline For Directory MAC Times.

Figure 1: Timeline Visualizations.

analysis process. Using a combination of InfoVis methods, the memory visualization tool provides global and local views of the memory capture data to the examiner in a single visualization structure. The examiner is then free to interact with and apply intuition to the analysis processes. In addition, through a novel white-listing process, the memory visualization tool filters items of less interest from view, effectively shrinking the examination space.

3.1. System Overview

Computer system forensic examiners and incident response teams work under various prescribed timelines derived from federal, state and private regulations[16]. Evolving technologies, most recently the arrival of vertical negative-AND (NAND) structures[17], drastically increase available storage and continue to put examiners behind the curve as existing timelines do not consider rapid leaps in future technology. The existing tools and methods provided to these examiners often generate ever-growing sets of data visualized as text or simple visualizations such as the trees seen in the graphical process tree reconstruction from the Digital Forensics Framework (DFF)[18] or histograms shown in Figure 1. Figure 1 presents a histogram of MAC times created using Plaso (log2timeline and

psort), Elasticsearch, Logstash, and Kibana (Plaso-ELK)[19]. These products provide a local view of data, but lack a global context beneficial to the examiner. This levies the task upon the examiner to manually or internally connect the data in order to make it useful.

Data sources in forensic examinations can have both hierarchical and associative relationships. Hierarchical relationships are seen in system structures such as the process tree, logical file system directory structure and registry (or equivalent configuration files) structure. Associative relationships are found between processes and their open network connections, file/registry handles, system modules, and/or services. In general, tree visualizations represent hierarchical data, while network visualizations depict connectedness. The problem lies in that neither visualization method simultaneously represents both types of data.

The memory visualization tool presents both types of data simultaneously. This hybrid visualization method uses three types of visualizations, as shown in Figure 2, on a single canvas. In the center of the visualization, nested circles represent the system's process tree, a hierarchical structure. Around the perimeter, a donut chart (i.e., a modified pie chart) represents system resources (e.g., network connections, file/registry handles, system modules, and services). System resources are hierarchical in nature, but at an arbitrary level are resources nodes with an associative relationship to nodes in the process tree. Associative relationships between process nodes and systems resource nodes are shown using lines (i.e., edges) as in standard network diagrams.

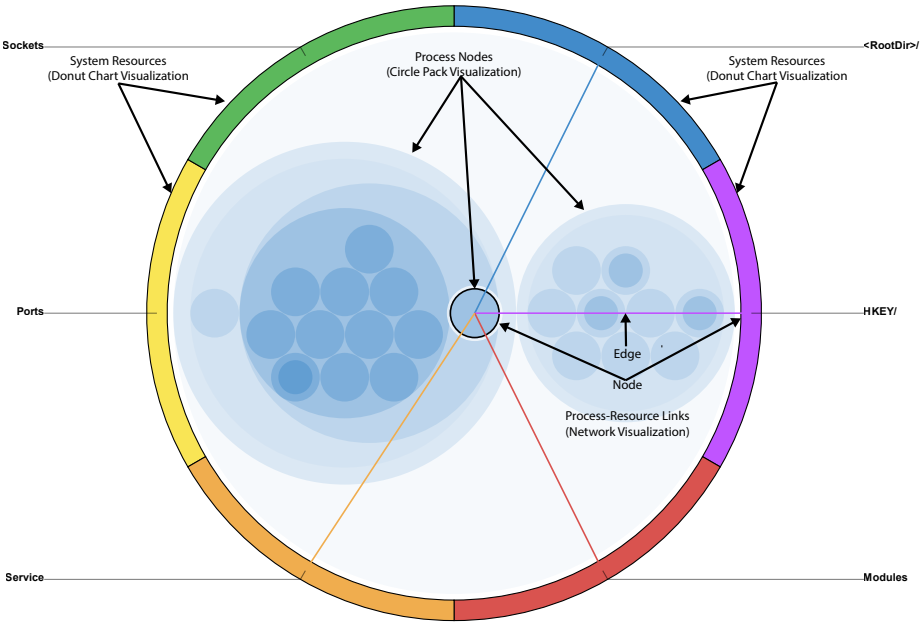


Figure 2: Overview Of Hybrid Visualization Components.

The goal of the memory visualization tool is to provide a tool that helps examiners perform a more accurate analysis in less time than current methods and tools. By providing both global and local views of the data, the examiner is freed from the task of determining connectedness of data, which may not be of interest, and can focus on applying knowledge and intuition to the analysis process. Furthermore, the global view provides the examiner with the proverbial "Big Picture" as they step further into the local views to get the descriptive raw data. Additionally, the visual representation of data allows novice examiners to better understand and explore the system under analysis, freeing up expert analysts for tasks more suited to their expertise.

In addition to these visualization techniques, the memory visualization tool applies a novel whitelisting process with the goal of shrinking the search space by eliminating items of little interest from view. The whitelisting process works off the premise that much of the data found on a live computer system is consistent and repetitive between instances of the same operating system. The tool filters consistent data from view based on a confidence percentage asserted by the examiner. The interactive nature of the tool allows the examiner to identify items of interest within a few clicks of the mouse rather than riffling through pages of text files.

The memory visualization tool, written in the MEAN Stack framework, is comprised of server-side Javascript using the NodeJS Engine and client-side Javascript using AngularJS running in a web browser as depicted in Figure 3. ExpressJS provides the routing interface between AngularJS and NodeJS. Mongoose connects AngularJS and NodeJS to the MongoDB database for find, update, remove, and insert operations. The whitelist module is a C++ node module that receives commands from the NodeJS Server and has direct access to both the memory image and the whitelist databases. Memory image feature files are uploaded to the NodeJS server from the client workstation and then imported to the memory image database.

3.2. User Interface Client

Figure 4 highlights the three main components of the memory visualization tool's user interface. The 'Image Select' and 'Functional Buttons' components control initialization of the 'Interactive Visualization' and toggle features within the 'Interactive Visualization' and 'Raw Text View' components. The 'Raw Text View' component displays raw text data from the source database in a sortable as well as searchable table. The majority of this section focuses on the 'Interactive Visualization' component, which is the primary InfoVis development in this tool.

Figure 4 shows the nine basic function buttons. These buttons are used to initialize the visualization tool, add or remove datasets from the database, and toggle features on or off during analysis. Additional user input controls are attached to components of the visualization. These components are discussed later in this section.

Table 1 describes the function of each button. Some buttons open an additional control window via modals. Modal windows allow the visualization to

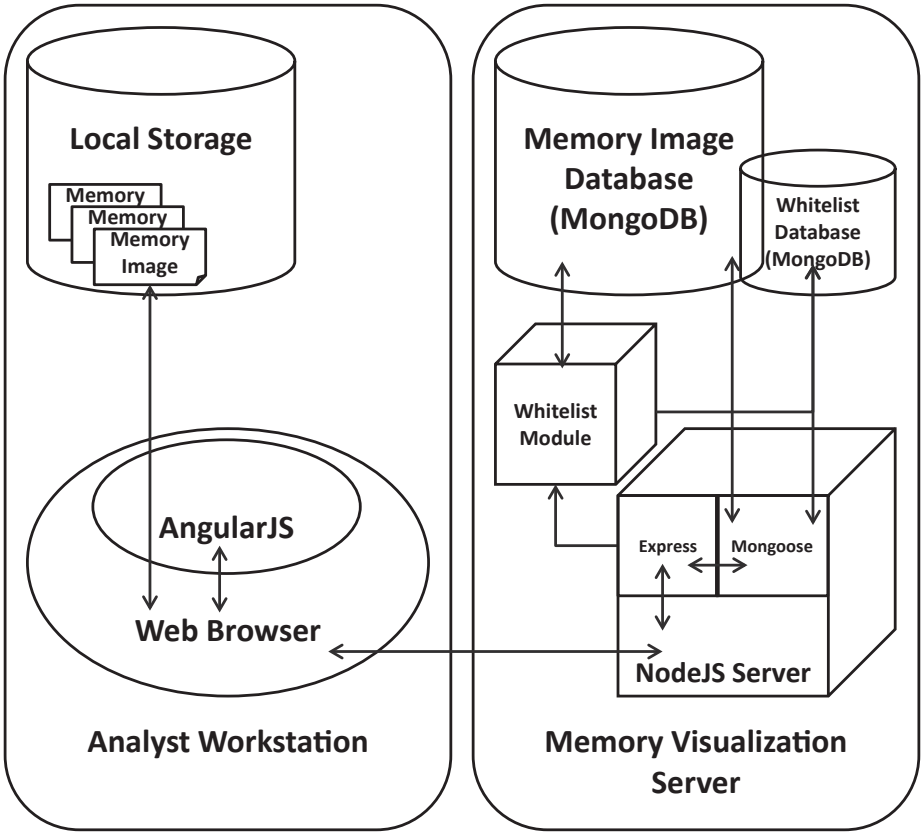


Figure 3: Memory Visualization Tool Diagram.

retain state while adding additional hypertext mark-up language (HTML) views and function scope.

Table 1: Function Button Descriptions.

Button	**Action**
Image Select Menu	A list of memory images currently loaded in the image database.
Visualize Dataset	Initializes the visualization tool with selected image.
System View	Resets the visualization tool to the initial global view.
Toggle Links	Enables or disables process to resource link display.
Toggle Text View	Enables or disables raw data display using datatables plug-in.
Toggle Whitelisting	Opens Whitelisting precision select modal.
Whitelist Memory Image	Opens Whitelisting image select modal.
Add Memory Image	Opens Memory image upload and import select modal.
Remove Memory Image	Opens Memory image remove select modal.

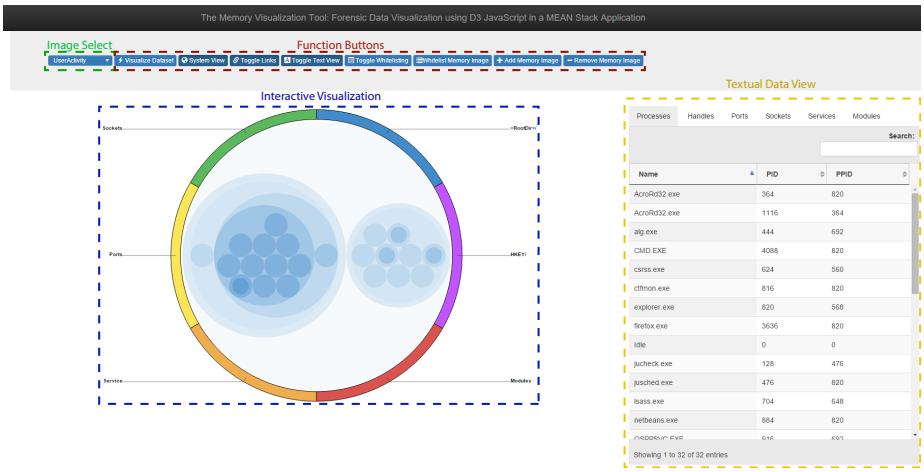


Figure 4: User Interface Orientation.

3.3. Process Nodes

Circle packing provides an intuitive representation of the process tree in a given memory image. This differs from a standard tree diagram in that relationships are implied by spacial position rather than lines, displaying greater amounts of data in a smaller space. Processes are nested inside of their parent process with the root node being Microsoft Window's `System Idle Process`. Figure 5 shows the process state of a "clean" Microsoft Windows XP SP3 system. The labels identify system processes (those initiated by the operating system) and user processes (those initiated by a user).

Process nodes offer two additional user controls. Holding a mouse curser over a node circle initiates tooltip window that shows the process name and process ID (PID) as shown in Figure 6. Process nodes also accept a mouse click event which highlights the node and sets that node as scope for the process to resource links, discussed later in this section.

3.4. Resource Arcs

A donut chart, much like a pie chart shows parts of a whole. When observing the initial system view shown in Figure 2, six system resources (File Handles, Registry Keys, Modules, Services, Ports, and Sockets), which are defined in Table 2 make up the outer donut.

Three additional user controls are attached to the resource arcs. A mouse-over displays the name of the resource over which the mouse is currently hovering. A mouse click steps through a hierarchical resource tree and display all branches and leaf nodes at each new level. Leaf nodes are opaque and clicking on them highlights all nodes associated with that specific resource. Rolling the mouse wheel up while hovering over an arc steps back one level in the resource tree until the initial system view is reached.

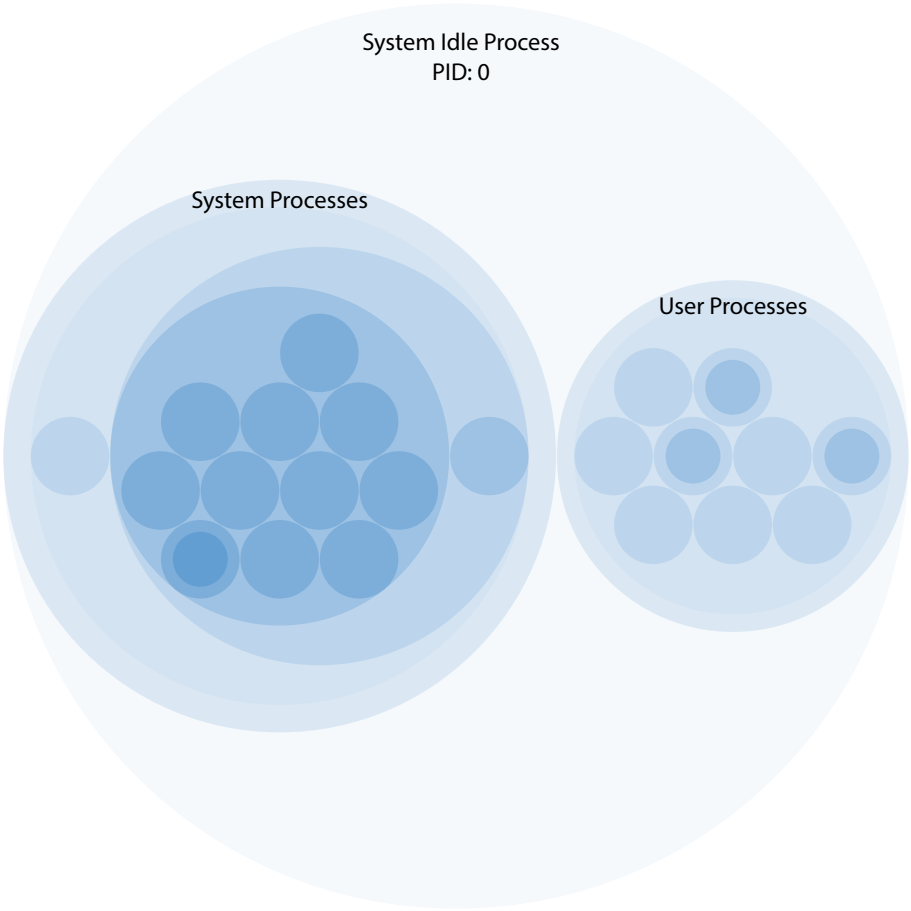


Figure 5: Process Node Hierarchy.

Table 2: System Resource Definitions.

Resource	**Definition**
File Handle	A unique identifier* linking an open file to owning PID.
Registry Key Handle	A unique identifier* linking a registry key to owning PID.
Module	Core executable programs and shared system libraries.
Service	A background program providing a specific function.
Port	The operating system end-point of an network connection.
Socket	The process end-point of a network connection.

** Except when a file handle held by a process is duplicated, or process inherits the file handles of the parent*[20].

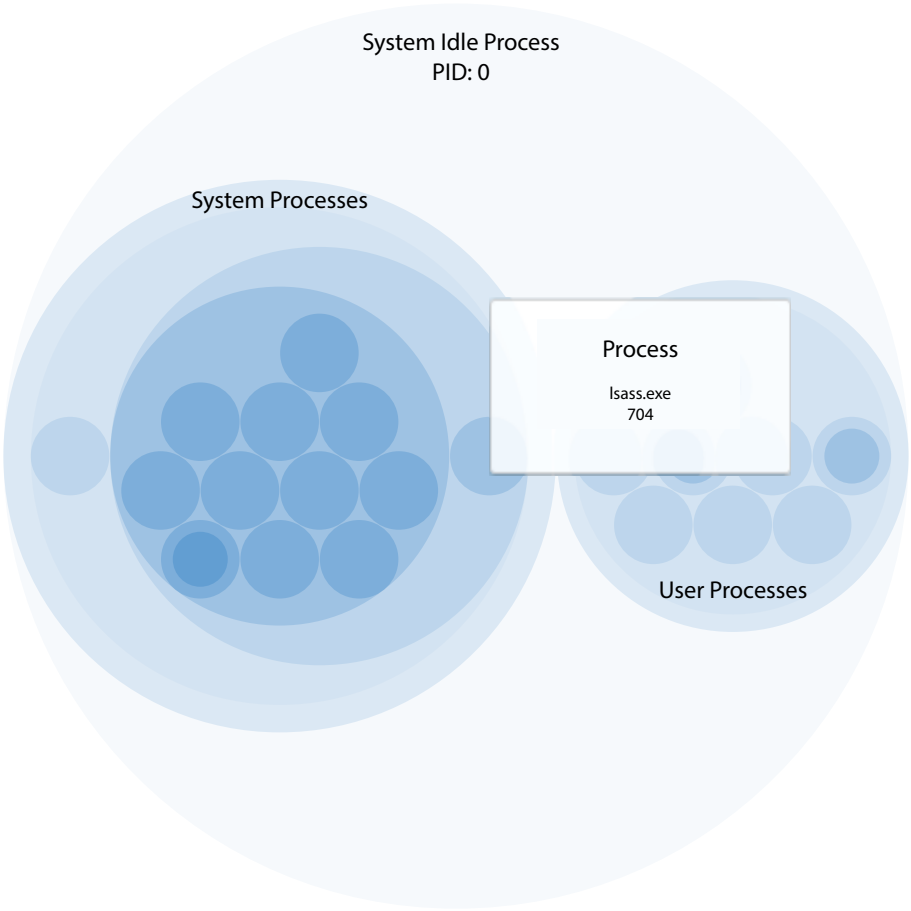


Figure 6: Mouse Over Control.

3.5. Process and Resource Links

Links show a relationship between connected data. Much like edges in a network diagram, links show a one-to-one, one-to-many, many-to-one, or many-to-many relationship between process nodes and resource arcs. Figure 7 shows the global system view with links enabled for the selected process `lsass.exe`.

Selecting a resource arc steps into that particular resource and draws links from the selected process node to the resources in the current display. Figure 8 depicts the service links for selected node `lsass.exe`.

As noted in the resource arc discussion above, clicking on an opaque resource highlights all associated process nodes, and with links enabled, draws links from the select nodes to all associated resources. Figure 9 illustrates this functionality.

Using the mouse-wheel up control or selecting the 'System View' control button steps backwards in the resource tree while leaving the multiple nodes

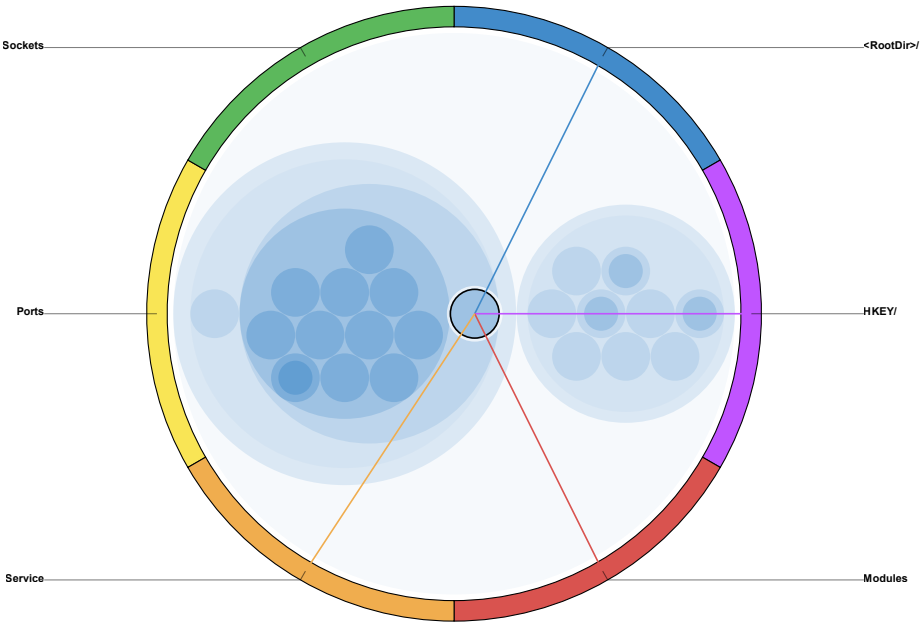


Figure 7: System View Links For lsass.exe.

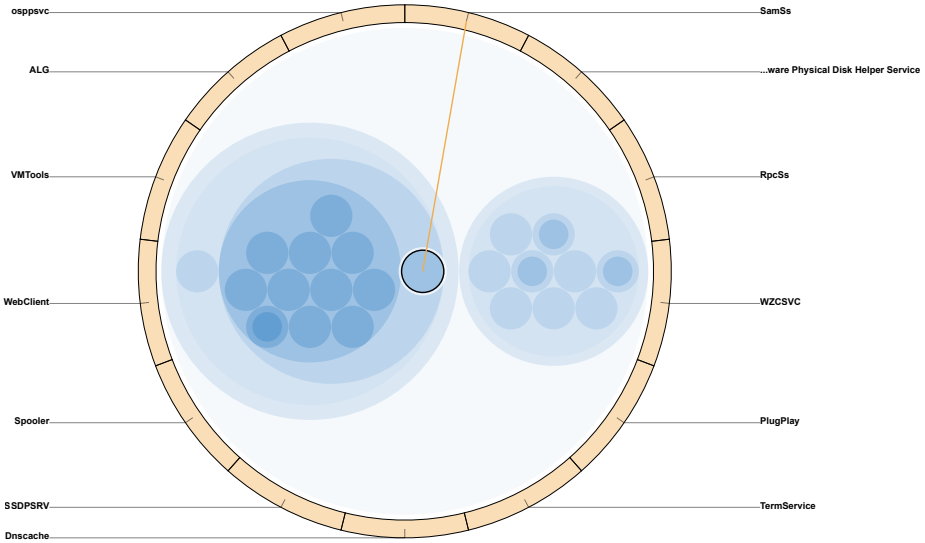


Figure 8: Service Links For lsass.exe.

selected as seen in Figure 10.

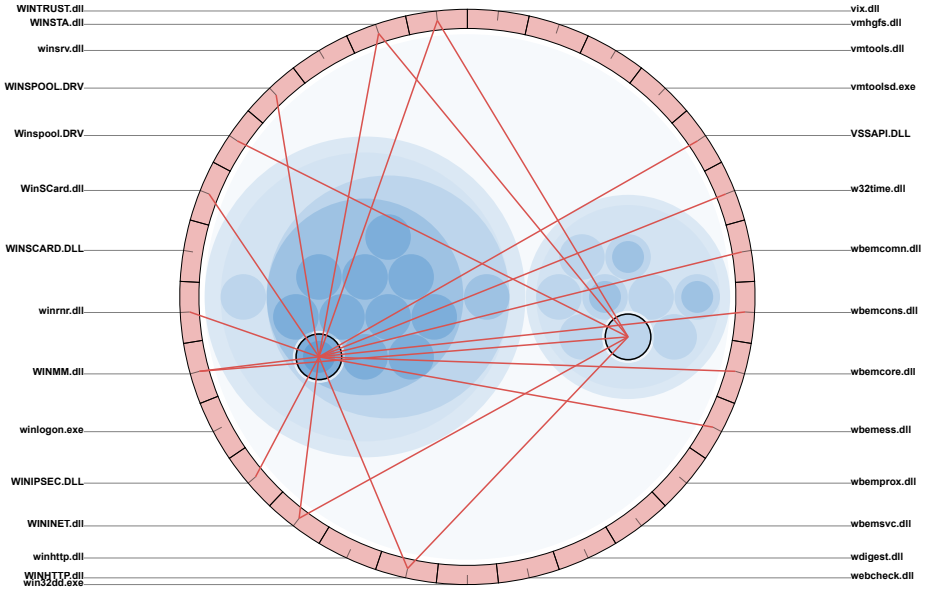


Figure 9: Module Links for Multiple Nodes.

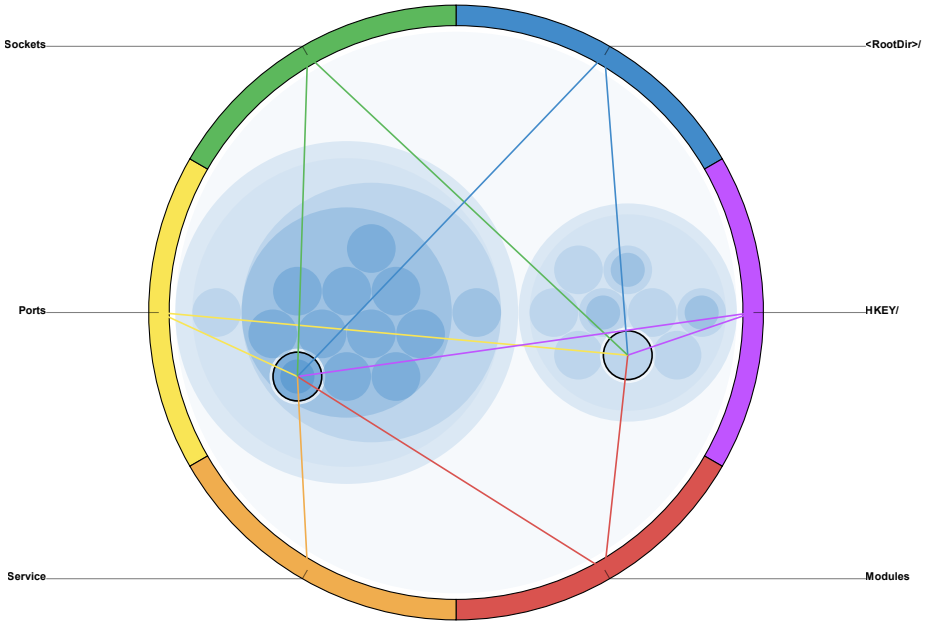


Figure 10: System View Links for Multiple Nodes.

3.6. Textual Data View

The textual data view using the DataTables Javascript plug-in provides raw text data to the analyst with a few additional controls and features. Using HTML tabs, the examiner can switch the table view between the system resources. Figure 11 shows the process list for a clean Microsoft Windows XP Image. DataTables provides a text search function that limits the table with each letter typed. Furthermore, each header is alphabetically or numerically sortable.

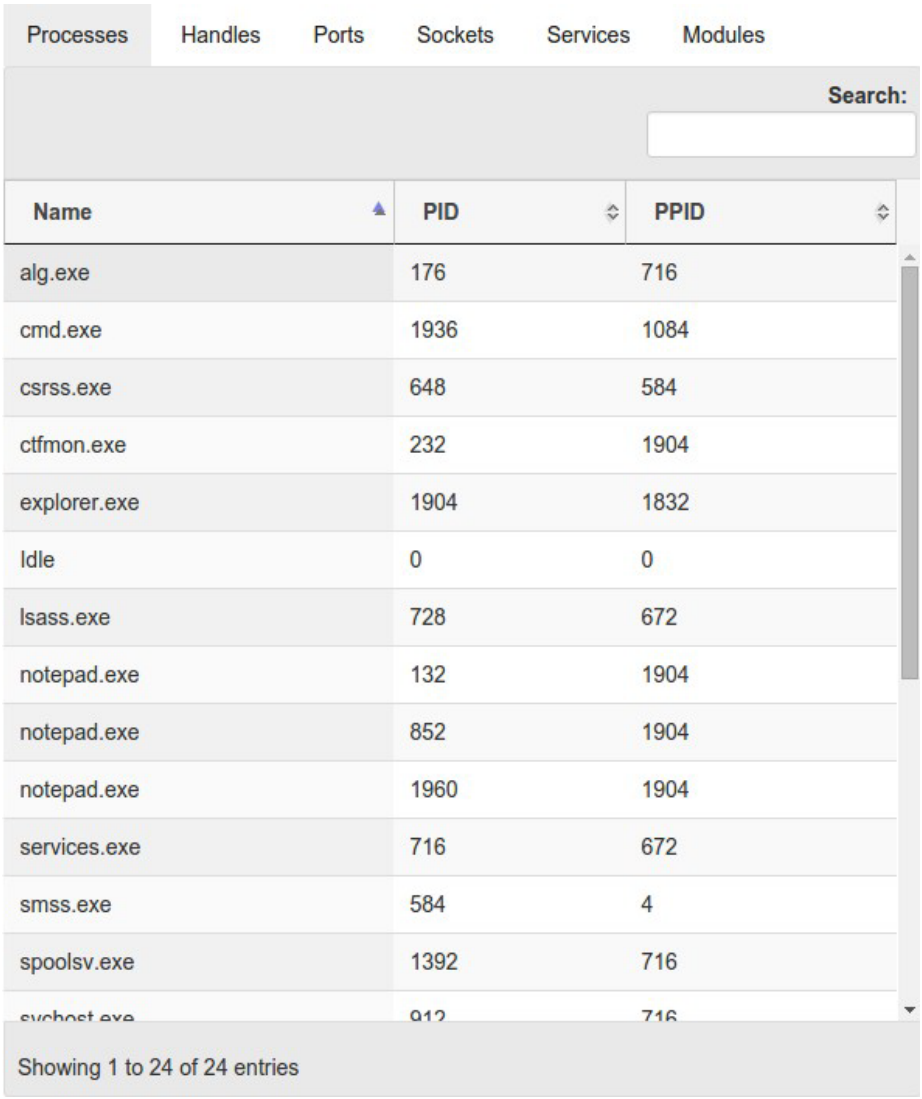


Figure 11: Process List In DataTables.

As the examiner types in the search field, the table is limited to entries with matching strings. Typing "lsa" in the search field limited the table to a single entry for `lsass.exe` as shown in Figure 12. Mouse click event listeners are appended to each table row and highlight an associated process node when clicked.

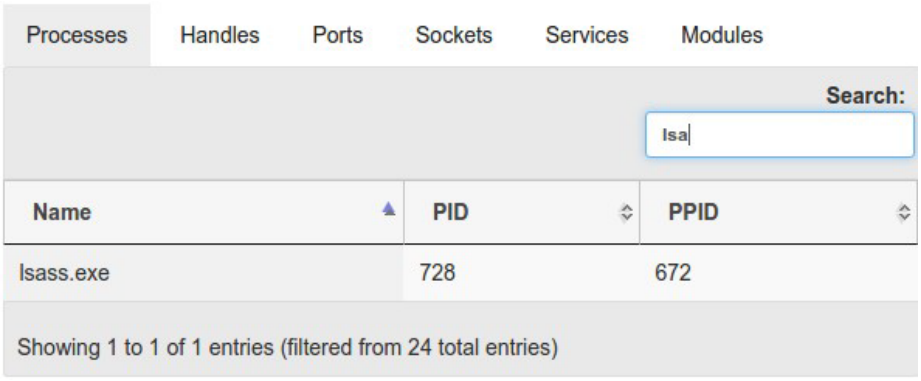


Figure 12: lsass.exe Search In DataTables.

3.7. Whitelisting

The memory visualization tool uses a behavioral whitelisting algorithm. This approach to whitelisting looks at a process by application name and its associated resources to determine if an application behaves similarly to other applications of the same name. When a given application behaves the same as other applications bearing the same name, it is likely that application is genuine. For instance, a compromised version of `svchost.exe` behaves differently than genuine versions from a Microsoft release.

The whitelisting process has two main functions: load new images and update resulting percentages. Product versioning is accounted for in the whitelisting process. During both the load and results functions, each memory image is only compared to those of the same major and minor version (i.e., Processes in Windows 5.1 (aka Windows XP) are not compared against other versions of Windows such as 7, 8, 8.1 or 10). The whitelisting process is depicted by Figure 13. This process works best with a very large database of memory images.

The load function iterates through each process in the image being loaded and compares it to processes already in the whitelist database. If a given process behaves the same way as a similarly named processes in the database, the whitelisting module increments the count for the number of times that application appears in the whitelist database. However, if the application behaves differently (i.e., different loaded modules, parent process, location, open files, or registry keys) or it is the first time an application of that name is added to the

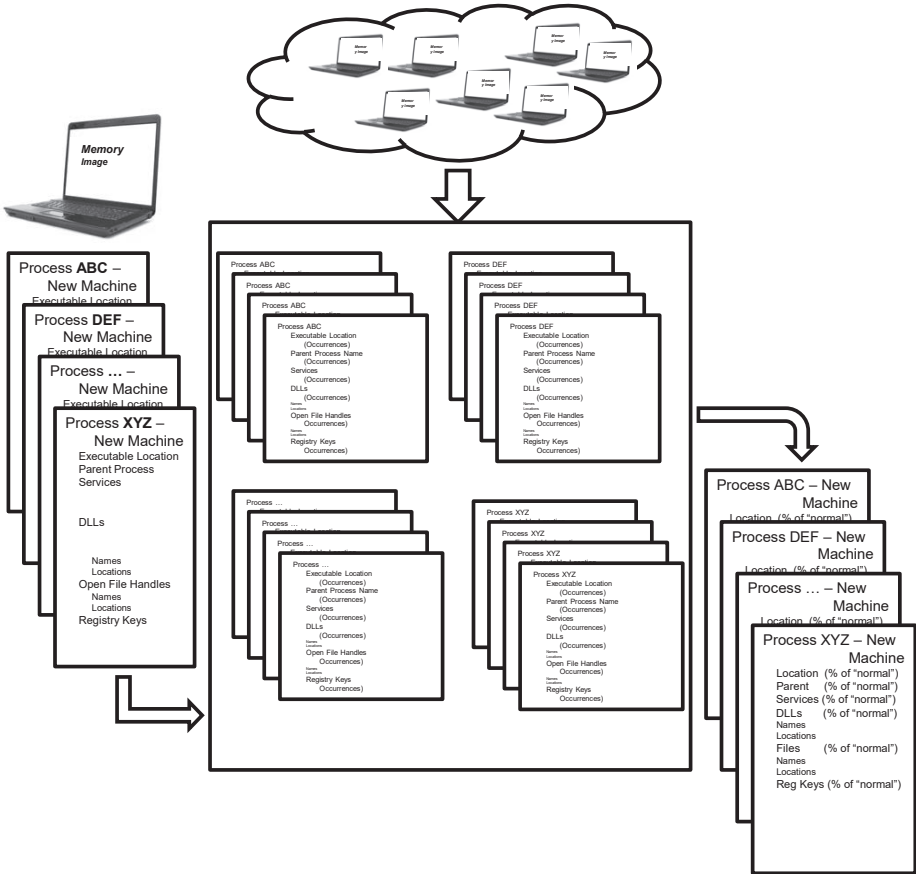


Figure 13: Whitelist Method Diagram.

whitelist, a new application entry is created and the application is assigned a unique application identifier (AppID).

The results function looks at each image in the working database. The function attempts to match each process in a selected image to an AppID and assigns the associated percentage denoting how often that AppID appears in the whitelist database. If no matching AppID exists, the process is assigned zero percent. The percentages are updated in the working database for use in the visualization as described earlier in this chapter.

4. Methodology

Digital forensic tools share a common analysis medium, humans. The binary nature of forensic artifact detection presents two distinct outcomes, *Found* and *Not Found*. The merits of a tool cannot simply be measured by how well

it detects and presents anomalies in a collection to the human examiner, because each tool also relies on the human examiner, of varying expertise, to correctly determine whether or not an anomaly is an artifact of interest. As such, quantitative methods do not adequately reflect the efficacy of a given interactive tool. This section describes a pilot study in digital forensics involving human subjects, where memory capture data are analyzed using the qualitative methodology known as content analysis[21].

4.1. Pilot Study Introduction

To test the efficacy of the memory visualization tool, a pilot study, involving human participants, examines the core principles of the tool. The study evaluates if the visualization improves the accuracy of artifact identification and reduces analysis time.

The study compares the memory visualization tool with a traditional textual data approach. Participants employ any search, sort and filter functions available in commercial or open-source text tools (e.g., Microsoft Office, Libre Office, grep, sort) as long as the display is textual. Both methods used the same source memory image data.

To evaluate the research goals, researchers collect data from the written submission of each exercise along with participant feedback. The written submission evaluates a participant's successful completion of the forensic exercise and time taken to complete the exercise. Post-study surveys use word and content analysis to support the hypothesis.

Participants respond on either the text-based method survey or the visualization method survey, based on participant's assigned method of analysis. Each survey question focuses on the participant's observations of the method, tool or perception of their own performance. Word counts identify the initial presence of themes, while direct quotations from participants provide content supporting each theme. Additionally, a single user experience question focuses on the participant's perceived understanding of the data. The final question in the survey, sought to draw out future work and/or recommended modifications to the user interface.

The pilot study also examines a single user experience objective. As the primary objective, researchers evaluate whether or not the memory visualization tool increases the participant's understanding of the data through simultaneous visualization of global and local views. As a secondary objective, researchers examine whether or not intuition plays a role in a participant's understanding of the data.

4.2. Experimental Procedures

During the exercise portion of the pilot studies, each participant is presented with data from three fictitious scenarios described in Table 3. Two of the scenarios involve scripted malicious activity, while the third scenario represents normal user activity. Each participant is required to answer questions for each scenario. Each scenario becomes more complex.

Table 3: Fictional Scenario Descriptions.

Name	Description
Scenario One	One machine running Microsoft Windows XP SP3. The user is browsing virus writing tutorial websites with the Firefox browser. The user has the Netbeans Java IDE open and is currently editing a file called "NewVirus.java". Adobe Acrobat is open reading a document named "How_To_Write_A_Virus.pdf". Lastly, Microsoft Word is open and currently editing a file named "The_Secret_Plan.docx".
Scenario Two	Three machines running Microsoft Windows XP SP3. One of the machines is running an instance of Poison Ivy[22]. The second machine only contains normal user activity. The third machine is running an instance of BadProcess.exe hidden by FUTo[23] rootkit.
Scenario Three	Five machines running Microsoft Windows XP SP3 and three servers running Windows 2008 Server. All machines are accessible to the incident response team. Each machine contains normal user activity along with some questionable activity. One machine is exploited using a Meterpreter[24] Reverse TCP Shell.

4.3. The Pilot Study Participants

The pilot study utilizes eleven cyber operations master's degree students. These participants self-identified as knowledgeable in the subjects of computer operating systems, computer networking, and malware. Participants are split into two groups using a psuedo-random list generator. Participants in the first group must use text-based tools. Participants in the second group must use the memory visualization tool with the whitelisting function. All participants are provided data from scenarios one, two and three and asked various open-ended questions pertaining to the data. Each question is timed.

Questions one through four of the exercise present participants with data from scenario one and ask them to provide specific details about the state of a single system. Question five asks participants to examine data from scenario two and answer questions about the presence of malware. Lastly, questions six uses data from scenario three and asks participants to report anything suspicious. Each exercise is scored for completeness and accuracy. Time and accuracy results are compared between groups.

4.4. Data Collection Methods

The independent variable in this study was which tool a participant used. Half of the participants used the text-based tools and half of the participants used the visualization tool (between groups). The controlled variable was the

scored exercise. The dependent variables were survey responses, accuracy of artifact identification and measured time to complete the exercise.

Time was self reported by participants. Each participant was provided a stopwatch and asked to record time taken to complete each question; no limit was established. The accuracy of exercise completion was scored based on researcher knowledge of the scenario. Incompleteness was scored as incorrect.

The post-exercise surveys introduce the qualitative aspect of the pilot studies. The survey responses are the primary interest of the researchers. Each response to the open ended questions are analyzed for keywords and themes. Accuracy and time information are used to bolster survey content by assigning more weight to responses from higher scoring participants.

4.5. Assumptions

1. Digital forensic methods exist for obtaining an accurate physical memory image.
2. The analysis is limited to process lists, network connections, system services, open file handles, system registry keys, and loaded modules.
3. The operating systems under consideration are limited to Microsoft Windows Operating Systems.
4. The researchers have knowledge of the laboratory exercises with which to evaluate accuracy of artifact identification.
5. Participants have a scholastic background (i.e., knowledge of operating systems, malware, networking and incident response) and a refresher lesson in analysis techniques in order to reduce learning effects during the experiment.

4.6. Hypothesis

The primary hypothesis of the pilot study is that the memory visualization tool would produce more accurate artifact identification than traditional text-based methods. It was hypothesized that any improved accuracy over text-based methods was due to the visualization tool's ability to simultaneously display hierarchical and associative relationships (i.e., simultaneous global and local view). It was also hypothesized that simultaneous global and local views led to better understanding of the data.

5. Results

During the pilot study, we collected two forms of data from participants. Each participant submitted a written solution to each forensic scenario. Secondly, each participant completed an open response survey. The written solutions received a percentage score for completeness and correctness. The surveys were analyzed for content and themes. We drew conclusions about the efficacy of the memory visualization tool from both sources of data.

5.1. Data Analysis

Our pilot study contained twelve pieces of quantitative data shown in Table 4: six scored questions and six associated time recordings. After initial review of the quantitative data, we removed the second visualization participant from the study. It was clear to us that this participant did not possess the required knowledge to successfully complete the forensic exercise (Assumption 3). We formed this conclusion based on their low score, slow time, and comments from their post-study survey. No other outliers were seen in the quantitative data.

Figures 14a and 14b show individual scores and times respectively. The quantitative data suggests that participants using the memory visualization tool scored higher and had faster completion times than their counterparts using text-based methods.

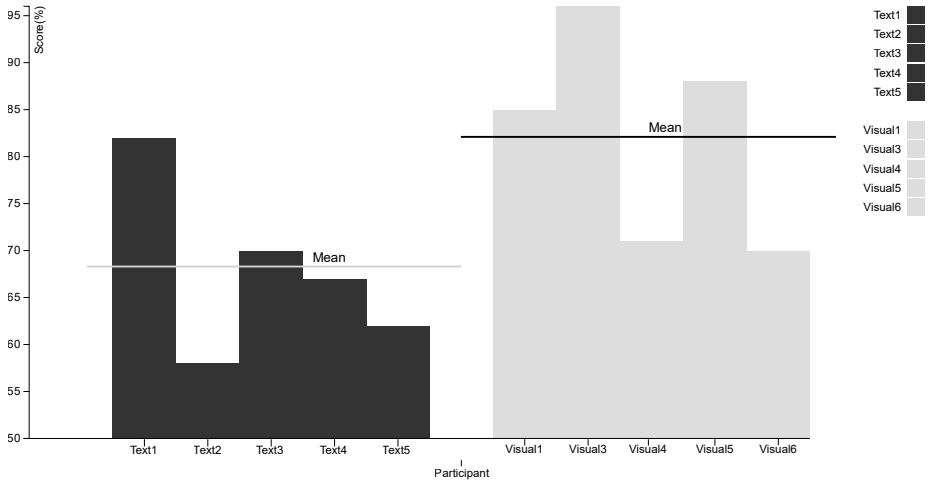
Table 4: Pilot Study Scores and Time By Participant.

	Question 1		**Question 2**		**Question 3**		**Question 4**		**Question 5**		**Question 6**		Average Total	
Participant	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time	Score	Time
Text1	1.00	0:15:15	1.00	0:12:52	1.00	0:06:04	1.00	0:06:12	0.44	0:42:29	0.50	0:49:43	0.82	2:12:35
Text2	0.85	0:07:12	0.50	0:12:50	0.50	0:01:08	0.88	0:05:30	0.78	0:27:46	0.00	0:28:01	0.58	1:22:27
Text3	0.69	0:03:14	0.50	0:20:26	1.00	0:01:55	0.94	0:04:24	0.56	0:20:13	0.50	0:39:18	0.70	1:29:30
Text4	0.92	0:07:20	0.50	0:07:20	1.00	0:07:20	0.63	0:07:20	0.22	0:44:30	0.75	0:45:00	0.67	1:43:01
Text5	0.92	0:11:15	0.50	0:11:15	1.00	0:11:15	0.94	0:11:15	0.33	0:29:10	0.00	1:00:00	0.62	2:14:10
Visual1	0.92	0:01:30	1.00	0:06:45	1.00	0:04:13	1.00	0:09:35	0.67	0:38:28	0.50	0:38:41	0.85	1:39:12
Visual3	1.00	0:08:11	1.00	0:09:27	1.00	0:02:03	1.00	0:05:43	1.00	0:31:27	0.75	0:34:19	0.96	1:31:10
Visual4	0.77	0:02:33	0.25	0:16:30	1.00	0:01:28	0.81	0:04:59	0.44	0:15:28	1.00	0:26:56	0.71	1:07:54
Visual5	1.00	0:09:30	1.00	0:10:30	1.00	0:03:00	1.00	0:10:00	0.56	0:21:30	0.75	0:27:00	0.88	1:21:30
Visual6	0.69	0:00:30	0.50	0:10:00	1.00	0:01:15	0.94	0:11:50	0.33	0:03:00	0.75	0:19:30	0.70	0:46:05
Text Average	0.862	0:06:06	0.6	0:11:43	0.9	0:05:10	0.875	0:07:37	0.511	0:32:01	0.35	0:42:12	0.683	1:41:40
Visual Average	0.877	0:04:27	0.75	0:10:38	1	0:02:24	0.95	0:08:25	0.6	0:21:59	0.75	0:29:17	0.821	1:17:10

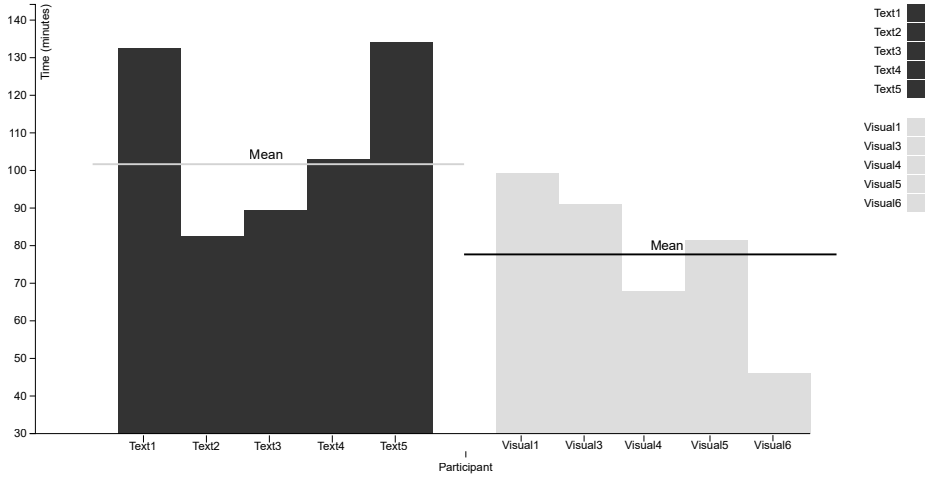
We begin the qualitative analysis by examining the survey responses at the most basic level. Figures 15a and 15b show the most commonly used words while Tables 5 and 6 provide the frequency of usage. It is interesting to note, the most dominant word in the text-based methods surveys is *data*. Using this information, we confirmed the presence of our themes in the survey data and continued the analysis.

The themes in the pilot study align with the hypothesis, improved accuracy with the visualization tool is attributed to its ability to represent hierarchical and linked data simultaneously. Additionally, this representation of data helps the user to better understand the data and apply intuition. Lastly, we attribute reduced time between artifact identification to the efficacy of the whitelisting feature. We continue to analyze the content of the open response surveys.

When discussing which components of the visualization tool made the tasks easier, one participant thought the process node visualization (i.e., circle packing) was the most helpful, stating, "it allowed me to very easily see what a particular process was using" while two others stated the "automatic links to resources" or "link view" was the most helpful. Another participant wrote the "leveled local views", using the resource circle, simplified the tasks. We interpret "leveled local views" to mean global view, because a single local view does not have levels. The most affirming response stated, " [T]he whitelisting tool was very useful, but it made me nervous to turn it up too far since I was worried that I might miss something."



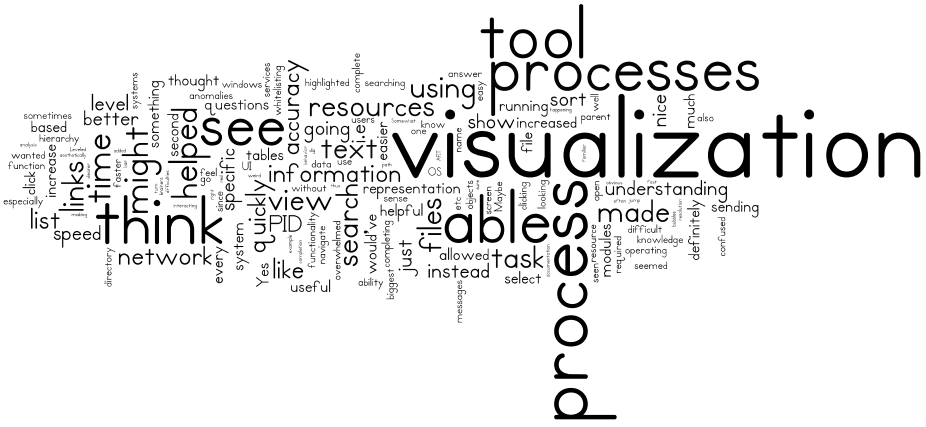
(a) Scores By Individual



(b) Times By Individual

Figure 14: Pilot Study Score and Time Charts By Individual.

We asked our participants how they perceived their accuracy of artifact identification, keeping in mind that they have not yet seen their scores. The responses were overwhelmingly positive. All five participants stated that the visualization tool increased their accuracy. One of the participants stated, "Without the visualization, I probably would not have been able to complete any of the tasks...', while another wrote "I can't imagine using spreadsheets" to complete the tasks. There were also concerns that the visualization tool made artifact discovery too easy and as one participant wrote, "[T]he tool made me more confident than I should have been...I might have rushed through the data too fast and missed something." Two participants noted the visualization tool im-



(a) Visualization Method Survey Word Cloud

(b) Textual Methods Survey Word Cloud

Figure 15: Pilot Study Two Word Clouds.

proved their accuracy by making "anomalies more obvious" and helping spot "odd behavior" by a process.

We then shifted our focus to how they perceived their time on task. One participant thought the visualization tool "definitely" increased completion speed noting, "I was able to, in a few seconds, locate suspicious processes". Three other participants attributed their speed to the simultaneous global and local views. One participant wrote, "Being able to see the links throughout the UI and navigate through the hierarchy allowed for a better understanding." A second participant stated, "The [resource] links and process bubbles made some of the questions no brainers", while a third felt "the visualization helped me literally see connections and dig into specific resources with more clarity than I think I would have otherwise."

Table 5: Frequently Used Words Visualization Survey.

NO. Occurences	Word	NO. Occurences	Word
6	able	1	notice
4	understanding	1	located
3	links	1	leveled
3	helped	1	level
2	view	1	interacting
2	quickly	1	interacted
2	hierarchy	1	integrated
2	helpful	1	information
1	whitelisting	1	increase
1	whitelist	1	help
1	visually	1	filtering
1	understand	1	filter
1	tree	1	efficient
1	showing	1	easier
1	show	1	connected
1	shading	1	bubbles
1	perceive	1	aided

Table 6: Frequently Used Words Text-Based Methods Surveys.

NO. Occurences	Word	NO. Occurences	Word
19	data	2	tables
6	time	2	sorting
6	task	2	scanning
6	excel	2	representation
5	filtering	2	puzzle
4	within	2	pen
4	together	2	paper
4	think	2	organization
4	relationships	2	manually
4	information	2	links
4	found	2	indicators
4	find	1	scattered
4	between	1	overload
3	spreadsheets	1	filters
2	text	1	filter
2	tedious	1	distracting

When asked if the visualization tool increased their understanding of the process and data being analyzed, the participants gave a resounding “yes”. The participants agreed the visualization tool helped them “see” or “understand” the data better. The visualizations allowed them to understand “how objects were

connected" as opposed to making the connections manually. One participant stated, "the visualization tool did seem to direct me" and "it increased my knowledge, by allowing me to see at a glance" the internal workings of an active system.

The quantitative results of the pilot study were mostly positive. Participants using the memory visualization tool showed higher scores and lower completion times than their counterparts using text-based methods. More importantly, key themes in the survey data supported our primary hypotheses. While these responses did not clearly support that the faster analysis times were attributable to the whitelisting feature, they do support the hypothesis that the simultaneous global and local views improve understand of the data and allow the users to apply intuition. Furthermore, these results provide the level of confidence required to move forward from pilot studies into real world testing.

6. Conclusions and Future Work

This research developed and evaluated how a fully functional memory analysis tool helps examiners maintain global and local views and quickly connect data by simultaneously displaying hierarchical and associative relationships. A pilot study confirmed the efficacy of the visualization tool through qualitative analysis of key themes contained in post-study survey data. Most importantly, this research confirmed the hypothesis that a visualization tool that provides context throughout analysis and shrinks an examiner's search space will make an examiner more accurate and faster.

Currently, the whitelisting algorithm has a couple shortcomings to address. The whitelisting does not differentiate between similarly named applications with varying numbers of open network connections and more research needs to be done on uniqueness when there are many handles. Lastly, the next version of the memory visualization tool should implement a code extraction feature. This would allow forensic examiners to extract unpacked executable code segments from a memory image using the memory visualization tool's user interface. This would most likely require an additional database schema.

7. Acknowledgements

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. We would like to thank all of our pilot study participants and Brendan Baum for his research and development on the initial proof of concept for memory visualization.

References

[1] F. Hinshaw, Data warehouse appliances Driving the Business Intelligence Revolution, DM Review 14 (9) (2004) 30.

[2] G. Henderson, Triage visualization for digital media exploitation, Master's thesis, Naval Postgraduate School (2013).

[3] N. Beebe, J. Clark, Dealing with Terabyte Data Sets in Digital Investigations, in: S. Pollitt, Mark and Shenoi (Ed.), Advances in Digital Forensics, Springer US, 2005, pp. 3–16. doi:10.1007/0-387-31163-7_1.

[4] S. Teerlink, R. F. Erbacher, Improving the computer forensic analysis process through visualization, Communications of the ACM 49 (2) (2006) 71. doi:10.1145/1113034.1113073.

[5] G. Osborne, H. Thinyane, J. Slay, Visualizing information in digital forensics, in: G. Peterson, S. Shenoi (Eds.), Advances in Digital Forensics VIII, Vol. 383 of IFIP Advances in Information and Communication Technology, Springer Berlin Heidelberg, 2012, pp. 35–47. doi:10.1007/978-3-642-33962-2_3.

[6] J. B. Baum, Windows Memory Forensic Data Visualization, Master's thesis, Air Force Institute of Technology (2014).

[7] B. D. Carrier, J. Grand, A hardware-based memory acquisition procedure for digital investigations, Digital Investigation 1 (1) (2004) 50–60. doi:10.1016/j.diin.2003.12.001.

[8] N. Davis, Live Memory Acquisition for Windows Operating Systems, Citeseer.

[9] H. Carvey, Windows Forensic Analysis, Elsevier, 2007. doi:10.1016/B978-159749156-3/50008-3.

[10] M. Burdach, Physical memory forensics, Black Hat, 2006.

[11] M. H. Ligh, A. Case, J. Levy, A. Walters, The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory, 1st Edition, Wiley, Indianapolis, 2014.

[12] J. Okolica, G. L. Peterson, Windows operating systems agnostic memory analysis, Digital Investigation 7 (SUPPL.) (2010) S48–S56. doi:10.1016/j.diin.2010.05.007.

[13] M. Cohen, Rekall memory forensic framework (2013).

[14] S. L. Garfinkel, Digital forensics research: The next 10 years, Digital Investigation 7 (2010) S64–S73. doi:10.1016/j.diin.2010.05.009.

[15] S. Chawathe, Effective whitelisting for filesystem forensics, in: Intelligence and Security Informatics, 2009. ISI '09. IEEE International Conference on, 2009, pp. 131–136.

[16] W. Gortney, CHAIRMAN OF THE JOINT CHIEFS OF STAFF MANUAL 6510.01B CYBER INCIDENT HANDLING PROGRAM, Department Of Defense, 1400 Defense Pentagon Washington, DC 20301-1400 (dec 2014).

[17] A. Goda, K. Parat, Scaling directions for 2d and 3d nand cells, in: Proceedings of the 2012 IEEE International Electron Devices Meeting (IEDM), 2012, pp. 2.1.1–2.1.4. doi:10.1109/IEDM.2012.6478961.

[18] Arxsys, Features - arxsys (2014).

[19] C. Vandeplas, Finding the needle in the haystack with ELK, in: Digital Forensics and Incident Response (DFIR), SANS Institute, Prague, 2014.

[20] Microsoft Developer Network, File handles (Windows) (2015).

[21] M. Savin-Baden, C. Major, Qualitative Research The essential guide to theory and practice, 1st Edition, Routledge, Abingdon and New York.

[22] FireEye Inc., Poison Ivy: Assessing Damage and Extracting Intelligence, Tech. rep., Milpitas, CA (2014).

[23] P. C. Silberman, FUTo, Uninformed 3 (2006) 14.

[24] T. M. Project, Metasploit's meterpreter (2004).