Annual ADFSL Conference on Digital Forensics, Security and Law

2017
Proceedings

May 16th, 3:15 PM

Exploring Digital Evidence with Graph Theory

Imani Palmer
University of Illinois at Urbana-Champaign, ipalmer2@illinois.edu

Boris Gelfand
Los Alamos National Laboratory, bgelfand@lanl.gov

Roy Campbell
University of Illinois at Urbana-Champaign, rhc@illinois.edu

EXPLORING DIGITAL EVIDENCE WITH GRAPH THEORY

Imani Palmer[1], Roy Campbell[1], and Boris Gelfand[2]

[1]Department of Computer Science,University of Illinois at Urbana-Champaign
[2]Advanced Research in Cyber Systems, Los Alamos National Laboratory

ABSTRACT

The analysis phase of the digital forensic process is the most complex. This phase grows more complicated as the size and ubiquity of digital devices increase. There are many tools aimed at assisting the investigator in the analysis process; however, they do not address growing challenges. In this paper, we discuss the application of graph theory, a study of related mathematical structures, to aid in the investigation process of digital forensic examiners. Graph theory is used to study the pairwise relations between objects. We explore how graph theory can be used as a basis for further analysis. We demonstrate the potential of the application of graph theory through its implementation in a case study.

Keywords: No keywords defined

1. INTRODUCTION

Digital forensics is a forensic science which aims to incorporate scientific principles to the investigation of digital evidence. The goal is to understand the sequence of events from the provided set of evidence (Raghavan, 2013).

The digital forensics investigative process is facing a crisis. As the proliferation of technology in society advances the capabilities of digital forensic examination process have diminished. Currently, there are many challenges to the investigative process. There have been many tools developed in order to settle these challenges; however, most tools are designed to help examiners find specific pieces of evidence, not to assist in investigations (Garfinkel, 2010). This has impacted the credibility of digital forensics in the courtroom.

In 2004, a substitute teacher, Julie Amero, was in the middle of teaching a class when the school computer began displaying pop-ups from a pornographic website. She was convicted of four felony counts of risk of injury to a child (Eckelberry et al., 2007). After further investigation, spyware was found to be the source of the pop-ups and Amero was able to overturn her conviction (Alva & Endicott-Popovsky, 2012). The digital forensic process failed to properly assess the evidence accurately and failed to supply the legal system with proper analysis.

Investigators find it increasingly difficult to locate vital events in massive amounts of evidence. Having an overview of the evidence can be crucial to an investigation, as

well as, examining patterns from the data can help analysts locate information and guide them in their search (Fei, Eloff, Venter, & Olivier, 2005).

We believe graph theory can support the decision making of digital forensic investigators and assists them in conducting data analysis in a more efficient manner. The techniques used to analyze graphical representations of evidence offer investigators a relational view. We will demonstrate the potential of graph theory to serve as an initial starting point for further analysis of evidence.

The remainder of this paper is organized as follows. In Section 2 provides a related work and background on research in evidentiary analysis. The following sections discuss graph theory and its application in digital forensic investigations. The final section presents the conclusions and directions for future work.

2. BACKGROUND

The digital forensic process involves four main stages: collection, preservation, analysis, and visualization (Kessler, 2010). While there has been a research to improve upon each of these areas. The analysis stage is still marred by numerous factors. These factors include the lack of standardization, accreditation, as well as, human bias and error. This has severely impacted the credibility of forensic analysts in the courtroom (Nagy, Palmer, Sundaramurthy, Ou, & Campbell, n.d.).

The analysis of digital evidence is performed by evaluating the data to identify digital evidence that supports an existing theory, that which does not support an existing theory, and that which shows tampering. Analyzing every bit of data is a daunting task when confronted with the increasing size of storage systems. In digital forensics, the acquired data is typically at the lowest and most raw format, which is often too difficult for humans to understand. The skills required is great and is not efficient to require every forensic analyst to be able to do so. Currently, we have solved this problem by using tools to translate data through one or more layers of abstraction until it can be understood. For example, to view the contents of a directory from a file system image, the file system structures must be processed so that the appropriate data structures are displayed. The data that represents the directory contents exists the acquired file system image file, but in a format that is too low to identify. The directory is a layer of abstraction in the file system (Carrier, 2003).

There are many tools that focus on the abstraction of evidence. Examples of these tools include EnCase (Software, n.d.), SleuthKit (Carrier, 2010), Caine (Giustini, Andreolini, & Colajanni, 2010), and Volatility (Foundation, n.d.); however, as the growing size and proliferation of devices require not only analysis, but a correlation of evidence. This has lead to the development of many tools focused on timeline reconstruction (Hargreaves & Patterson, 2012).

Zeitline is an open-sourced graphical tool that allows forensic investigators to import various events and then order and classify them into one or more timelines. Events may be grouped into super-events, creating a hierarchy of events (Buchholz & Falk, 2005). FACE (Case, Cristina, Marziale, Richard, & Roussev, 2008) expands on this work by adding automated analysis and correlation of disk images, memory images, network captures, and configuration files, in order to provide a more coherent view of the state of the target system and allowing investigators to quickly understand it. The reliance on time has shown to be a problem. A study that measures and compares the accuracy and effectiveness of various event reconstruc-

tion techniques show they have very high false-positive rates, up to 96% (Jeyaraman & Atallah, 2006).

More recently, the literature has begun to explore other methods in order to analyze evidence. Self-Organizing Maps (SOM) is a type of artificial neural network which is used to visualize low-dimensional views of high-dimensional data. This visualization reveals interesting patterns from data. These patterns are able to aid in the investigator's decision making. The output of SOM provides excellent visualizations of the evidence; however, input data to SOM require data to be manually transformed, with a significant amount of human labor overhead (Fei et al., 2005). The use of Self-Organizing Maps also hasn't been fully explored in investigator contexts and would need to be further examined.

The following section provides a brief overview of graph theory. Graph theory is used to enable investigators to visualize the evidence as well as assist them in locating information of interest. We believe that relying on these relationships we can answer many questions about the system while providing a coherent view of the entire system. This knowledge is displayed through both visualization and data. Essentially, our approach applies graph theory algorithms, to improve the investigative process.

3. GRAPH THEORY

Graph theory is the study of graphs. Graphs are a mathematical representation of a network used to model pairwise relations between objects. A graph G consists of a set of nodes V that are representative of objects, with certain pairs of these nodes connected by edges E. The edges determine the relationship between the nodes. A graph may be either directed or undirected. An undirected graph means there is no distinction between two nodes associated with each edge. A directed graph means that its edges may be directed from one node to another, this relationship is better defined and can represent many ideas such as node A happened before node B, node A is parent of node B and etc. An example of a diredcted graph is shown in Figure 1.

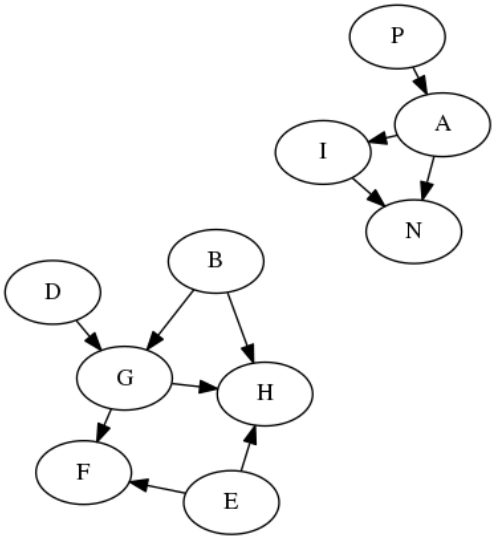

Figure 1. A Directed Graph

In graph theory, the degree of a vertex in a graph is the number of connections it has to other vertices. The degree distribution is the probability distribution of the known degrees over the entire graph. Centrality is an indicator of the most important vertices within a graph. The concept of centrality aims to quantify the influence of a vertex in a graph. We also rely on link analysis to aid in the examination process. Link analysis is a data analysis technique used to evaluate relationships between vertices. Relationships may be identified among various types of vertices, including organizations, people, and transaction. Link analysis has been used for investigation of criminal activity, computer security analysis, search engine optimization, market research, medical research, and art.

Previous digital forensic methods fail to find information that is anomalous or even slightly altered (Garfinkel, 2010). Graph theory is able to determine possible correlations among the evidence. This is achieved through analysis of the graphs. As we analyze the graph, we are able to interpret more from the evidence.

4. APPLYING GRAPH THEORY TO MEMORY FORENSICS

This section focuses on the application of graph theory to memory forensics. We demonstrate the potential of this analysis with a case study from The Honeynet Project - Banking Troubles. In this case study, a company has requested forensic analysis to be performed on an incident that occurred. One of its employees received an email from a fellow co-worker with an attached PDF file. Upon opening the email, the employee did not seem to notice anything; however, they did notice unusual activity in their bank account. Company X was able to obtain a memory image of the employee's virtual machine upon suspected infection (Spitzner et al., 2004).

We first define the vertices and edges of our graph. We are given a memory image from a Windows XP SP2 x86 as our sole source of evidence. Memory images contain a great deal of volatile evidence. We determine our vertices to be both processes and network connections. The edges are exemplified by processes that fork other processes or make network connections. This forms a directed graph as a parent process initializes a child process or network connection. The graphical representation of this memory image is shown in Figure 2.

4.1 Visualize Graphical Representations

The graph represented in Figure 2 is centered around user activity. This graph provides the examiner with an understanding of events surrounding the user. Next, we rely on link analysis algorithms in order to determine which vertices are the most important to all the vertices.

4.2 Determine High-Level Actions

We rely on Hyperlink-Induced Topic Search (HITS) in order to determine which vertices are important to other vertices. We believe that in determining these vertices will allow inferring the high-level actions taken by the user.

HITS was originally designed as a method of filtering results from web page search engines in order to identify results most relevant to a user query. The output of this algorithm is two scores for each vertex. The authority value, which estimates the value of the vertex, and its hub value, which estimates the value by the links to other vertices. We focus on the hub value in order to understand the high-level actions occurring in the system. A high-level action is an activity that either the system or user can partake. This includes the opening of a web browser, a system update or using a specific application. These high-level actions typically lead to other more specific actions such as sending an email, creating a file or removing unnecessary memory. The hub values for each vertex of Figure 3 is shown in Table 1.

services.exe and *svchost.exe* have the highest hub values. From this, we understand that there are many other processes from which these are spawned. This is aligned with what is already known about these properties. *services.exe* is the Services Control Manager, which is responsible for

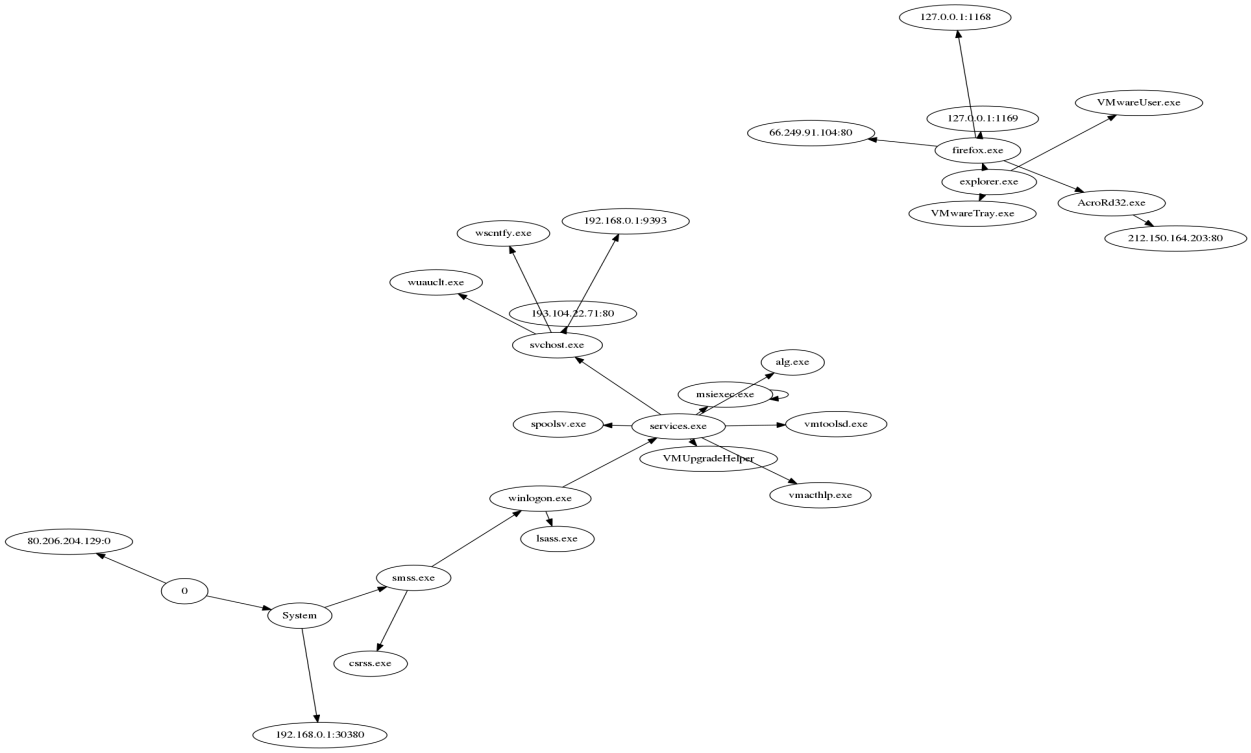

Figure 2. A Directed Graph Representation of Case Study

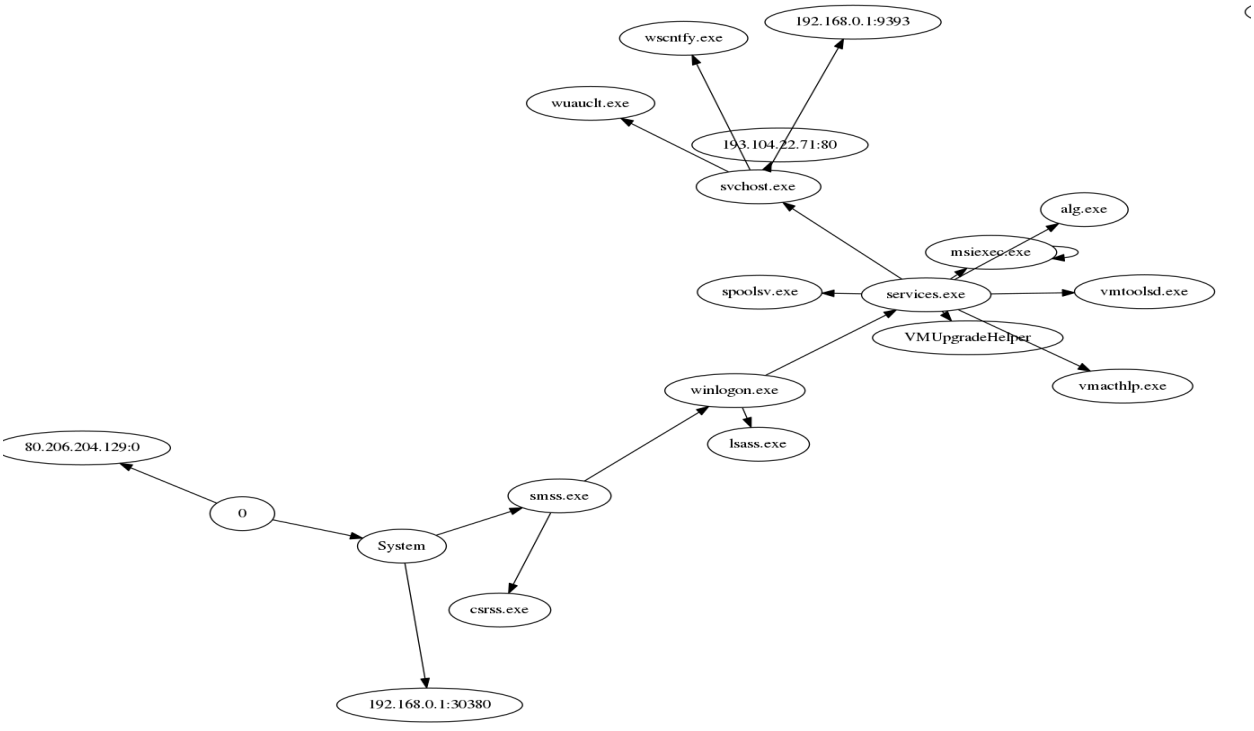

Figure 3. Graph of Computer Activity

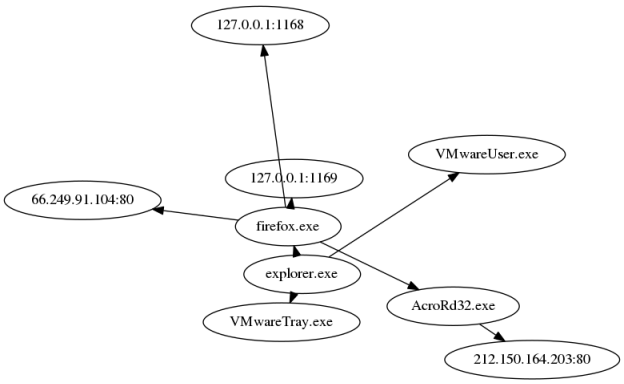

Figure 4. Subgraph of Figure 2

Vertex	Hub Value
firefox.exe	0.9999999760813284
explorer.exe	2.391867162501559e-08
AcroRd32.exe	1.8807909163296387e-37
VMwareUser.exe	0.0
127.0.0.1:1169	0.0
127.0.0.1:1168	0.0
VMwareTray.exe	0.0
66.249.91.104:80	0.0
212.150.164.203:80	0.0

Table 2. Hub Values of Vertices from Figure 2

Vertex	Hub Value
services.exe	0.8603796047843144
msiexec.exe	0.1396203891209935
svchost.exe	6.094692040769568e-09
System	1.4190310707244948e-18
0	1.4190310707244948e-18
smss.exe	1.4190310707244948e-18
winlogon.exe	1.4190310707244948e-18
80.206.204.129:0	0.0
vmtoolsd.exe	0.0
192.104.22.71:80	0.0
192.168.0.1:9393	0.0
192.168.0.1:30380	0.0
wuauclt.exe	0.0
csrss.exe	0.0
spoolsv.exe	0.0
vmacthlp.exe	0.0
VMUpgradeHelper	0.0
alg.exe	0.0
lsass.exe	0.0
wscntfy.exe	0.0

Table 1. Hub Values of Vertices from Figure 3

running, ending, and interacting with system services. It is used to start services, to stop services, or to change the service's default from automatic to manual startup. *svchost.exe* is the Service Host, it is a system process that hosts multiple Windows services. IT is essential in the implementation of shared service processes, where a number of services can share a process in order to reduce resource consumption. Next, we explore the hub values of Figure 2 shown in Table 2.

The hub values from Figure 2 give us three vertices: *firefox.exe*, *explorer.exe*, and *AcroRd32.exe*. The high-level action determined by *firefox.exe* is the opening of a web browser. Here we infer that the user used a web client in order to open their email. Another high-level action is stems from *AcroRd32.exe*. We understand from this that Adobe Reader was used to viewing the attached PDF. Lastly, *explorer.exe* is the Windows Explorer. It manages the Windows Graphical Shell. This means that the user relied upon the graphical user interface in order to interact with the system. The hub values have provided us with higher-level knowledge of the user actions, as well as, knowledge about the action of the system. We believe examining hub values of the

graphical representations of the evidence will also alert an examiner of anomalous behavior, however, this insight it out of the scope of this paper. In the next section, we will be exploring a ranking of vertices in order to better determine which events are important.

4.3 Construct Ranking

PageRank is a link analysis algorithm that computes the ranking of the vertices in the graph based on the structure of the incoming edges. PageRank was first developed as a method for computing a ranking for every web page based on the graph of the web. The rank value indicates the importances of a particular page. We employ this concept to our case study. We believe this ranking will identify key pieces of evidence from our memory image that we should further examine. First, we demonstrate its value in reference to Figure 3. This ranking is shown in Table 3.

The highest ranking value from Table 3 is the vertex representing the process *msiexec.exe*. *msiexec.exe* is a program required to run on startup, it adds, modifies and removes applications provided as a Windows Installer package. This fits with our understanding of PageRank this vertex is in contact with many different vertices making it important. This can also contribute to the detection of anomalous behavior. If a process is more or less active typically noted it would be shown in from its PageRank value. This will explore more in future work. We next examine the PageRank values of Figure 2.

In Table 4, we identify the vertex representing the network connection *212.150.164.203:80* with the highest PageRank value. This also aligns with our knowledge from the case study. It appears that this network connection made after being spawned by AcroRd32.exe. This

Vertex	**PageRank Value**
msiexec.exe	0.23557310777643373
services.exe	0.050167587835275466
lsass.exe	0.050167587835275466
csrss.exe	0.0492131817109793
winlogon.exe	0.0492131817109793
192.168.0.1:30380	0.046967564815315235
smss.exe	0.046967564815315235
80.206.204.129:0	0.04168384563715685
System	0.04168384563715685
193.104.22.71:80	0.036762326111687275
192.168.0.1:9393	0.036762326111687275
wuauclt.exe	0.036762326111687275
wscntfy.exe	0.036762326111687275
vmtoolsd.exe	0.03534358323175769
spoolsv.exe	0.03534358323175769
svchost.exe	0.03534358323175769
VMUpgradeHelper	0.03534358323175769
alg.exe	0.03534358323175769
vmacthlp.exe	0.03534358323175769
0	0.0292517283888176

Table 3. PageRank Values of Vertices from Figure 3

Vertex	PageRank
212.150.164.203:80	0.17315667694672304
firefox.exe	0.10674309394441647
VMwareUser.exe	0.10674309394441647
VMwareTray.exe	0.10674309394441647
127.0.0.1:1169	0.1058593956290246
127.0.0.1:1168	0.1058593956290246
AcroRd32.exe	0.1058593956290246
66.249.91.104:80	0.1058593956290246
explorer.exe	0.08317645870392903

Table 4. PageRank

allows us to infer that the PDF downloaded by the user was most likely malicious. We are also shown that *firefox.exe* is an important vertex. This is aligns with a previous knowledge shown in the calculation of the hub values. PageRank allows us to identify key pieces of evidence. After we have aided in overall analysis process of the examiner, we help formulate valid hypotheses based on the evidence provided with graph traversal.

4.4 Establish Valid Hypothesis

Complex arguments ought to be separated in small ones. The synthesis is the recomposition of the partial solutions of the decomposed problem. In the context of forensic investigations solving a problem should be interpreted as *collecting information to prove or disprove the occurrence of an event in the real world*. In other words, in order to be able to draw conclusive assessment about a case, detectives need to find significant tests to evaluate the simplest hypotheses. They have to analyze the scene of the crime in order to find elements that may enable them to estimate their rational belief in hypotheses. In other words, detectives perform tests aimed at collecting data that are relevant (i.e., provide information about discrimination between a hypothesis and its negation) in the assessment of a given hypotheses. We denote mapping between evidence set E_i and hypothesis H as $H \rightarrow E_1, E_2, E_3, ..., E_n$.

Graph traversal is the process of visiting each vertex in a graph. There are multiple algorithms to aid in graph traversal the shortest path problem. The shortest path problem deals with the problem of finding a path between two nodes in a graph such that the sum of the weights of its constituent edges is minimized.

The problem of finding the shortest path between two intersections on a road map (the graph's nodes correspond to intersections and the edges correspond to road segments, each weighted by the length of its road segments). The shortest problem can be defined for graphs whether undirected or directed. We will now evaluate how each of these elements of graph theory can contribute to a digital forensic investigation.

We use the use case example from the previous section in order to further elaborate this point. We rely on the shortest path algorithm to determine an occurrence of events. We have seen from our previous section that the network connection 212.159.164.203:80 had a high PageRank. We rely on shortest path problem to help us determine what events occurred before this network connection was made. We are able to find a path from explorer.exe to 212.150.164.203:80. The corresponding path is explorer.exe - firefox.exe - AcroRd32.exe - 212.150.164.203:80. From our prior knowledge of the case, we know that an employee received an email with an attached pdf from a friend. This is the believed course of action that caused the potential malicious activity. We are able to corroborate these events with the shortest path. The process explorer.exe represents the user interface. From this user interface the user opens up Firefox and we can tell that a pdf downloaded and accesses this network connection. We have successfully used elements of graph theory to

provide a more logical view of events from events.

5. CONCLUSIONS & FUTURE WORK

Graph theory can serve as the basis for further analysis of data generated from digital forensics tools. In particular, the graphical representation of evidence allows an investigator to not only visualize, but perform data analysis on evidence. This analysis enables forensic investigators to locate information of interest efficiently.

Initial work with graph theory has identified several areas for future research. The first area is an exploration of relationships among the evidence. This research has begun in previous works; however, it still needs to be continued alongside the exploration of time-dependent graphs. Digital evidence has multiple relationships that are both dependent on time and not. Exploring this area can lead to further insight and greater knowledge. The second area of exploration is the potential to develop algorithms based on graph theory. In digital forensics, outlier detection is not enough to detect everyday user actions. However, through the exploration of link analysis, this might be possible to determine potentially unique events. This area of exploration will require a lot of well-documented datasets open to the public. The third area of exploration is the automation of this tool. The automation of determining relationships among artifacts as well as the interpretation of them. This work is also already in progress by many previous works.

We explored the potential benefits of using graph representation in digital forensics. It is possible to get a high-level view of the system without requiring extensive knowledge of the operating system and its applications. In this paper, we successfully showed the potential of using graph representation in the analysis. We show this by exploring a case studies.In the future, we believe that this work can be greatly improved by exploring more relationships. We plan to further these efforts by building a prototype and implementing more forms of analysis.

ACKNOWLEDGEMENTS

The authors would like to thank the Systems Research Group at the University of Illinois at Urbana-Champaign. Thanks are also due to the anonymous reviewers for their valuable feedback.

The material in this paper is based upon work supported in part by the Air Force Research Laboratory and the Air Force Office of Scientific Research, under agreement number FA8750-11-2-0084.

REFERENCES

Alva, A., & Endicott-Popovsky, B. (2012). Digital evidence education in schools of law. *The Journal of Digital Forensics, Security and Law: JDFSL*, *7*(2), 75.

Buchholz, F. P., & Falk, C. (2005). Design and implementation of zeitline: a forensic timeline editor. In *Dfrws.*

Carrier, B. (2003). Defining digital forensic examination and analysis tools using abstraction layers. *International Journal of digital evidence*, *1*(4), 1–12.

Carrier, B. (2010). The sleuth kit. *TSK). http://www. sleuthkit. org/sleuthkit/.[Online]*.

Case, A., Cristina, A., Marziale, L., Richard, G. G., & Roussev, V. (2008). Face: Automated digital evidence discovery and correlation. *digital investigation*, *5*, S65–S75.

Eckelberry, A., Dardick, G., Folkerts, J., Shipp, A., Sites, E., Stewart, J., & Stuart, R. (2007). *Technical review of the trial testimony state of connecticut vs. julie amero.* Technical Report, http://www. sunbelt.

Fei, B., Eloff, J., Venter, H., & Olivier, M. (2005). Exploring forensic data with self-organizing maps. In *Ifip international conference on digital forensics* (pp. 113–123).

Foundation, V. (n.d.). *Volatility.* Retrieved from `www.volatilityfoundation.org`

Garfinkel, S. L. (2010). Digital forensics research: The next 10 years. *digital investigation*, *7*, S64–S73.

Giustini, G., Andreolini, M., & Colajanni, M. (2010). Open source live distributions for computer forensics. In *Open source software for digital forensics* (pp. 69–82). Springer.

Hargreaves, C., & Patterson, J. (2012). An automated timeline reconstruction approach for digital forensic investigations. *Digital Investigation*, *9*, S69–S79.

Jeyaraman, S., & Atallah, M. J. (2006). An empirical study of automatic event reconstruction systems. *digital investigation*, *3*, 108–115.

Kessler, G. C. (2010). *Judges' awareness, understanding, and application of digital evidence* (Unpublished doctoral dissertation). Nova Southeastern University.

Nagy, S., Palmer, I., Sundaramurthy, S. C., Ou, X., & Campbell, R. (n.d.). An empirical study on current models for reasoning about digital evidence.

Raghavan, S. (2013). Digital forensic research: current state of the art. *CSI Transactions on ICT*, *1*(1), 91–114.

Software, G. (n.d.). *Encase.* Retrieved from `www.guidancesoftware.com`

Spitzner, L., et al. (2004). *The honeynet project.*

