Visualization for network forensic analyses: extending the Forensic Log Investigator (FLI)

Paul Michael Miller
Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/rtd
Part of the Computer Sciences Commons

Recommended Citation
DEDICATION

To my family – Dad, Mom, Amy, and Molly:

You have encouraged me every time I’ve left to chase down a dream, and always been there to support me when I’ve come home. You’ve also helped me stay balanced, reminding me of what is most important – and that even I need a break sometimes. Thank you so much for your support – I couldn’t have finished without it. I love you all very much.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CHAPTER 1. INTRODUCTION</td>
<td>VI</td>
</tr>
<tr>
<td>1.1</td>
<td>MOTIVATION</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>REVIEW OF LITERATURE</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1</td>
<td>Visualization</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1.1</td>
<td>Scientific Visualization</td>
<td>3</td>
</tr>
<tr>
<td>1.2.1.2</td>
<td>Information Visualization</td>
<td>3</td>
</tr>
<tr>
<td>1.2.1.3</td>
<td>Security Visualization</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>DIGITAL FORENSICS AND DIGITAL EVIDENCE</td>
<td>4</td>
</tr>
<tr>
<td>1.3.1</td>
<td>Investigative Context, Evidence of Offense, and Associative Information</td>
<td>6</td>
</tr>
<tr>
<td>1.3.2</td>
<td>Handling procedures</td>
<td>7</td>
</tr>
<tr>
<td>1.4</td>
<td>FORENSIC PROCESS</td>
<td>8</td>
</tr>
<tr>
<td>1.5</td>
<td>NETWORK FORENSICS</td>
<td>9</td>
</tr>
<tr>
<td>1.5.1</td>
<td>Network Forensic Process</td>
<td>10</td>
</tr>
<tr>
<td>1.5.2</td>
<td>Network Forensics Visualization Development</td>
<td>12</td>
</tr>
<tr>
<td>1.6</td>
<td>COMPUTER FORENSICS</td>
<td>12</td>
</tr>
<tr>
<td>1.6.1</td>
<td>Role of computer</td>
<td>13</td>
</tr>
<tr>
<td>1.7</td>
<td>INTRUSION DETECTION</td>
<td>14</td>
</tr>
<tr>
<td>1.7.1</td>
<td>Stepping Stone Analysis</td>
<td>14</td>
</tr>
<tr>
<td>1.7.2</td>
<td>Multistage Attack Analysis</td>
<td>15</td>
</tr>
<tr>
<td>1.7.3</td>
<td>Connection Analysis</td>
<td>15</td>
</tr>
<tr>
<td>1.8</td>
<td>ATTACKER MODELING</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>CHAPTER 2. DESIGN &amp; ARCHITECTURE</td>
<td>17</td>
</tr>
<tr>
<td>2.1</td>
<td>OVERVIEW</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>ANALYSIS FRAMEWORK</td>
<td>18</td>
</tr>
<tr>
<td>2.2.1</td>
<td>FLI Analyzer</td>
<td>18</td>
</tr>
<tr>
<td>2.2.1.1</td>
<td>Connection Analysis</td>
<td>19</td>
</tr>
<tr>
<td>2.2.1.2</td>
<td>Stepping Stone Analysis</td>
<td>19</td>
</tr>
<tr>
<td>2.2.1.3</td>
<td>Multi-Stage Attack Analysis</td>
<td>20</td>
</tr>
<tr>
<td>2.2.1.4</td>
<td>Full Analysis</td>
<td>21</td>
</tr>
<tr>
<td>2.2.2</td>
<td>FLI Analysis Result</td>
<td>22</td>
</tr>
<tr>
<td>2.3</td>
<td>VISUALIZATION FRAMEWORK</td>
<td>22</td>
</tr>
<tr>
<td>2.3.1</td>
<td>Displaying Results</td>
<td>23</td>
</tr>
<tr>
<td>2.3.1.1</td>
<td>Processing a Result</td>
<td>23</td>
</tr>
<tr>
<td>2.3.1.2</td>
<td>Map-Based Display</td>
<td>23</td>
</tr>
<tr>
<td>2.3.1.2.1</td>
<td>Geocoding</td>
<td>24</td>
</tr>
<tr>
<td>2.3.1.2.2</td>
<td>Internal Address Mapping</td>
<td>24</td>
</tr>
<tr>
<td>2.3.1.3</td>
<td>Additional Information</td>
<td>24</td>
</tr>
<tr>
<td>2.3.2</td>
<td>Visual Encodings &amp; Visual Metaphors</td>
<td>24</td>
</tr>
<tr>
<td>2.3.2.1</td>
<td>Connection Analysis</td>
<td>24</td>
</tr>
<tr>
<td>2.3.2.2</td>
<td>Stepping Stone Analysis</td>
<td>26</td>
</tr>
<tr>
<td>2.3.2.3</td>
<td>Multi-Stage Attack Analysis</td>
<td>28</td>
</tr>
<tr>
<td>2.3.2.4</td>
<td>Full Analysis</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>CHAPTER 3. INFRASTRUCTURE</td>
<td>32</td>
</tr>
<tr>
<td>3.1</td>
<td>COMPUTER LANGUAGES</td>
<td>32</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Java</td>
<td>32</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Java Server Pages</td>
<td>32</td>
</tr>
<tr>
<td>3.1.3</td>
<td>JavaScript</td>
<td>32</td>
</tr>
<tr>
<td>3.1.4</td>
<td>XML</td>
<td>32</td>
</tr>
<tr>
<td>3.1.5</td>
<td>XPath</td>
<td>32</td>
</tr>
<tr>
<td>3.1.6</td>
<td>SQL</td>
<td>33</td>
</tr>
</tbody>
</table>
3.1.7. SQL/XML

3.2. SOFTWARE FRAMEWORKS

3.2.1. Google Maps API

3.2.2. Maxmind GeoIP API

3.2.3. Oracle DBMS

3.2.4. Adobe Flex 2.0

3.2.5. Java Enterprise Edition Application Server

3.2.6. Hibernate API

CHAPTER 4. SUMMARY AND DISCUSSION

4.1. DESIGN GOAL ACHIEVEMENT

4.2. USE IN INVESTIGATIVE PROCESS

4.3. COMPONENT REVIEW

4.3.1. Analysis Framework

4.3.2. Visualization Framework

4.3.3. Categorization

4.4. EVALUATION

4.5. FUTURE WORK
# LIST OF FIGURES

| Figure 1.3.1. | Digital Forensics Research Suitability Guidelines  | 4 |
| Figure 1.4.1. | Digital Forensics Investigative Process           | 8 |
| Figure 1.5.1.1 | Network Forensic Process                         | 10 |
| Figure 1.5.2.1 | Network Forensics Visualization Development      | 11 |
| Figure 2.1.1. | FLI Overall Architecture                         | 16 |
| Figure 2.1.2. | FLI Map Overall Process for Analysis & Visualization | 17 |
| Figure 2.2.1.1.1 | Connection Analysis Flowchart                   | 18 |
| Figure 2.2.1.2.1 | Stepping Stone Analysis Flowchart                | 18 |
| Figure 2.2.1.3.1 | Multi-Stage Attack Analysis Flowchart            | 19 |
| Figure 2.2.1.4.1 | Full Analysis Flowchart                          | 20 |
| Figure 2.3.2.1.1 | Connection Analysis Example Screenshot           | 25 |
| Figure 2.3.2.1.2 | Comparison of Connection Analysis Results        | 26 |
| Figure 2.3.2.2.1 | Stepping Stone Analysis Example Screenshot       | 27 |
| Figure 2.3.2.3.1 | Multi-Stage Analysis Example Screenshot          | 28 |
| Figure 2.3.2.4.1 | Full Analysis Example Screenshot                 | 29 |
ABSTRACT

In a network attack investigation, the mountain of information collected from varying sources can be daunting. Investigators face significant challenges in being able to correlate findings from these sources, given difficulties with time synchronization. In addition, it is difficult to obtain summary or overview information for one set of data, much less the entire case. This, in turn, makes it nearly impossible to accurately identify missing information.

Identifying these information gaps is one problem, yet another is filling them in. Investigators must rely on legal processes and requests to obtain the information they need. However, it is extremely important they are aware of cases or events that cross jurisdictional boundaries. Where tools exist to assist in evidence overview, they do not contain the necessary geographic information for investigators to quickly ascertain the location of those involved.

In addition to these difficulties, investigators need to perform several types of analysis on the evidence that has been collected. Several of these analyses cannot typically be performed on data from multiple log files, since they are based on timing data. Furthermore, it is difficult to understand results from these analyses without visual representation, and there are no tools to bring them together in a single frame.

This thesis details the design and implementation of an analysis and visualization extension for the Forensic Log Investigator, or FLI. FLI is a web-based analysis and visualization architecture built on advanced technologies and enterprise infrastructure. This extension assists investigators by providing the ability to correlate evidence and analysis across traditional log file and analysis method boundaries, identify information gaps, and perform analysis in accordance with published evidence handling guidelines.
CHAPTER 1. INTRODUCTION

1.1. Motivation

There have been a number of changing factors that have led to the development and extension of the Forensic Log Investigator as a tool for practitioners of network forensics. The evolution of the Internet has connected people and systems across nearly every categorical classification. From the individual to the business, the public sector to the private sector, the virtuous to the vicious, use of computer networks has become part of the daily routine. This influx of Internet citizenry has brought with it an increase in cyber-societal issues similar to those in the physical world.

One of the major draws influencing this change is the perceived value of networked targets. In particular, as networks have transitioned from informational to operational, the value of data and systems accessible online has drastically increased.

Progress can indeed be a double-edged sword; as the sophistication of networked applications has increased, so have the tools available to an attacker. These attackers are primarily concerned with maintaining anonymity, and put to use a number of concealment tactics to thwart attribution attempts. While there is a philosophical debate on the issue of anonymity, our focus here is on accountability for law enforcement and internal review. There are many tactics for concealment, such as using stepping stones, anonymizing networks, and distributed attacks. Each utilizes different methods and principles to complicate attack attribution.

In short, we hope that our work on this topic may someday help investigators to quickly understand a sequence of events, explore possible leads, identify information gaps, and present their findings. We believe that the current state of available tools is inadequate to sift through the mountains
of evidence involved in technically complex cases, and the most effective tools are often unusable due to poor evidence handling routines. The major design goals for this analysis and visualization extension are:

- Provide ability for investigators to visually correlate evidence and analysis across multiple logs
- Provide ability for investigators to visually correlate results from multiple types of analysis
- Assist in identifying information gaps
- Properly handle evidence when performing analysis

1.2. Review of Literature

1.2.1. Visualization

Webster’s defines visualization as “the act or process of interpreting in visual terms or of putting into visible form.” A visualization, then, is an application that uses images or symbols to convey information. In the field of computer science, visualization is based in computer graphics and human computer interaction. In this paper, we use the terminology defined by [10]. A visual encoding is a mapping of an information vector to a display element, such as analysis type to line color. FLI Map visual encodings are defined in Section 2.3.2. Visual metaphors are a combination of visual encodings in a single display. FLI Map visual metaphors are defined by analysis type – see Sections 2.3.2.1 – 2.3.2.4. Visualization, as a field of research, is generally broken down into two areas – information visualization and scientific visualization, based on the utilization of spatial data. Scientific visualizations typically rely on spatial information included or implicit in the data itself. Information visualizations, on the other hand, tend to use spatial layout in order to represent something about the data. Finally, there is a subfield (typically considered part of information visualization) called security visualization. This area focuses on the application
of visualization principles to information security systems. We found that our
needs did not correspond directly to the contexts generally discussed in such
research, which resulted in a need to draw on principles from all three areas.

1.2.1.1. **Scientific Visualization**

Scientific visualization was really the genesis of the field, as many of the
first applications of computer graphics fell into this discipline. Such visualizations
are essentially an extension of human perception, often providing us with ways to
“see” things that are too small, too fast, or too dangerous, for us to observe on
our own. In other cases, they are simply data points that our body is incapable of
sensing like infrared radiation or [10]. The data observed or produced inherently
contains the spatial layout, such as a visualization application for ocean currents
[8].

1.2.1.2. **Information Visualization**

Information visualization does not contain any inherent spatial data, but
rather uses spatial layout as a component of the visual metaphor. This spatial
positioning has been described as the most effective way to visually encode all
types of data, whether it is nominal, ordinal, or quantitative. Research in the area
has indicated that certain visual encodings are more effective than others, but
that their efficacy is dependent upon the data they represent [8]. Examples of
such visual encodings are color hue, shape, size, color brightness, and density.

Business-type charts and graphs are a simple example of this type of
visualization. Profit and time do not contain any implicit spatial information, but
we can apply an ordering to profit and time such that profit increases in a vertical
direction, while time increases in the horizontal. The spatial layout of our data
points then provides an instant visual cue as to the trends within the data.
1.2.1.3. **Security Visualization**

For analyzing log data, several approaches have been used in prior research. Self-organizing maps were researched early on, and provided effective trend analysis and anomaly detection capabilities. Force-based mapping techniques [9] have shown to be effective for anomaly and intrusion detection. However, these systems do not relate the location information for IP addresses often critical to investigators. Furthermore, they are designed for real-time or playback visualizations, as opposed to investigative applications. Playback applications most closely mirror an investigative application, but generally lack broad overview capabilities. This deficiency is in a critical area for the investigator, who needs to quickly understand the evidence that has been collected.

1.3. **Digital Forensics and Digital Evidence**

Digital Forensic Science came into its own as a field in 2001, at the First Digital Forensic Research Workshop (DFRWS). At this workshop, researchers and practitioners met to discuss the major challenges, objectives, and ideas in the field. They developed and published a set of objectives based on the area of operation.

![Figure 1.3.1: Digital Research Suitability Guidelines [2]](image)

As illustrated in Figure 1.3.1, the objectives of forensic application differ from industry to industry. There are a number of challenges facing both researchers and practitioners as they work to complete their objectives.
Extending this work, Sarah Mocas [4] defined five primary properties that are desirable for forensic tools and processes.

- **Integrity**
  Actions taken to secure and collect electronic evidence should not change that evidence.

- **Authentication**
  Must be able to show in court this evidence is what its presenter claims.

- **Reproducibility**
  This is a basic principle of the scientific process, which strengthens the belief that an experimental result is correct.

- **Non-interference**
  Tools and processes must maintain integrity, or only result in identifiable interference. Thus, the interference can be accounted for and disregarded.

- **Minimalization**
  This principle is a requirement based on Federal law, and assures that the minimal amount of data needed was used.

The practitioners of digital forensics have seen many changes throughout the last few decades. As identified by [2], keeping up with technology is a major difficulty in the field. In fact, a 1995 survey found that 70% of law enforcement agencies did not have a written procedures manual for digital evidence [5]. As time has progressed, these agencies have developed definitions and principles to govern their work with digital evidence. Some important definitions from [5] include:

- **Digital Evidence**: Information of probative value stored or transmitted in digital form.
- **Data Objects**: Objects or information of potential probative value that are associated with physical items. Data objects may occur in different formats without altering the original information.

- **Original Digital Evidence**: Physical items and the data objects associated with such items at the time of acquisition or seizure.

- **Duplicate Digital Evidence**: An accurate digital reproduction of all data objects contained on an original physical item.

1.3.1. **Investigative Context, Evidence of Offense, and Associative Information**

The context in which an investigation takes place is critical to determining what evidence can be used and how it can be used. This investigative context contains three sets [4]:

- A set of reasons for initiating the investigation.
  
  This set may include varied items, both technical and non-technical. Examples such as complaint reports, data loss, or intrusion alarms are all possible reasons for initiating an investigation.

- A set of constraints on the scope of the investigation.
  
  These constraints may include both legal constraints and resource constraints. These constraints are likely to affect the tools useful to an investigator.

- A set of potential outcomes.
  
  The interesting subset of this is the set of outcomes desirable to the investigator. This can be anything from prosecution to system restoration.

The contents of the sets will likely vary as the industry of the practitioner changes, in accordance with the suitability chart in Figure 1.3.1. In addition to the above components, the investigative context for a digital investigation also includes the technical environment – which can be either static, such as log files.
collected from an imaged hard drive, or dynamic, such as logs still being actively modified.

Finally, Mocas [4] established a distinction between evidence of offenses and associative information. Evidence collected can directly support or establish the occurrence of a misdeed, which can be direct or circumstantial evidence of an offense. Examples of digital evidence of offense would be a log showing unauthorized access or data exfiltration, the unauthorized transmission of data to an external entity. It may also serve as associative information to connect or disconnect an actor or event from that evidence. Examples of such would be subscriber IP address information during the time of offense.

1.3.2. Handling procedures

“In order to ensure that digital evidence is collected, preserved, examined, or transferred in a manner safeguarding the accuracy and reliability of the evidence, law enforcement and forensic organizations must establish and maintain and effective quality system. Standard Operating Procedures (SOPs) are documented quality-control guidelines that must be supported by proper case records and use broadly accepted procedures, equipment, and materials.” [5]

This statement serves as a guiding principle for evidence handling procedures. The standards laid out in accordance with this principle provide a set of constraints for handling digital evidence. The standards below are those relevant to technical procedures such as analysis and visualization:

- **Standards and Criteria 1.3** – Procedures used must be generally accepted in the field or supported by data gathered and recorded in a scientific manner.
- **Standards and Criteria 1.5** – The agency must use hardware and software that is appropriate and effective for the seizure or examination procedure.

- **Standards and Criteria 1.6** – All activity relating to the seizure, storage, examination, or transfer of digital evidence must be recorded in writing and available for review and testimony.

- **Standards and Criteria 1.7** – Any action that has the potential to alter, damage, or destroy any aspect of original evidence must be performed by qualified persons in a forensically sound manner. [5]

### 1.4. Forensic Process

The First Digital Forensics Research Workshop outlined a linear investigative process for practitioners. This process outlines the full set of tasks to be carried out by an investigator, without eliminating those that may not be considered “forensic.”

![Digital Forensics Investigative Process Diagram]

*Figure 1.4.1. Digital Forensics Investigative Process*
1.5. **Network Forensics**

Network forensics is defined as “The use of scientifically proven techniques to collect, fuse, identify, examine, correlate, analyze, and document digital evidence from multiple, actively processing and transmitting digital sources for the purpose of uncovering facts related to the planned intent, or measured success of unauthorized activities meant to disrupt, corrupt, and or compromise system components as well as providing information to assist in response to or recovery from these activities.” [2]

In addition to providing a definition of the field, researchers and practitioners at the conference identified the following major issues:

- **Time**
  
  This is one of the most important issues in the application of network forensics. Synchronization and integrity of dates and times associated with events, and is even more important when work spans multiple jurisdictional or time zones.

- **Complexity**
  
  The level of detail is high for singular system analysis. Most tools developed for this application do not translate well into networked analyses. Analysts and examiners also need tools to correlate and understand data relationships in the volumes of log data, particularly from intrusion detection systems.

- **Collection**
  
  Collection is difficult given the traditional requirements on collectors. Freshness of data and what to collect are also concerns.
- **Paradigm Distinctions**
  Law enforcement needs more data integrity, chain of custody, and proven & accepted methods in its tools. Civilian and military applications are focused on intelligence and operations.

- **Collaboration**
  Researchers identified a need for collaborative tools that are capable of simultaneous use or can spread data to associated parties quickly.

- **Legal Hurdles**
  There are no established rules of engagement for effective and legal operation in many networked arenas. Researchers were particularly concerned with the *Digital Millennium Copyright Act* and *Security Systems Standards and Certification Act*.

- **Emerging Technologies**
  Focus groups identified the explosion in wireless technologies and integration as a major issue for network forensics, both for complexity and legal hurdles.

**1.5.1. Network Forensic Process**

![Network Forensic Process Diagram](image_url)

*Figure 1.5.1.1. Network Forensic Process*
Figure 1.5.1.1 depicts the process for performing network forensics analysis, as described in [3]. This process details the application of the overall digital forensics process described in Figure 1.4.1 to the network forensics field. Text-based analysis is a fundamental capability for performing network forensics, and is the traditional means of investigation. Erbacher et al. [3] identified four major additional capabilities: pattern matching, stream identification, data browsing/examination, and domain knowledge. It is understood that these capabilities must be available to, and integrated into, the visualization functionality.

Visualization based analyses work with the investigator to improve the analysis process and results [3]. The primary goal is to improve comprehension of the events and analysis results. These visualizations must be integrated with traditional forensic analyses in order to be effective.

Finally, analysis results must be recorded so that they can be compared to prior or future analyses of the data. This allows for reproducibility (see Section 1.3) and aids in proving legal admissibility [3]. It can also be used to allow investigators to review previous findings without being forced to perform analysis again.
1.5.2. **Network Forensics Visualization Development**

Figure 1.5.2.1 outlines the development process for network forensics analysis tools, focused on forensic analysis visualizations. This process is continual for the development of visualization and interactive techniques. Iteration in the upper section of the diagram allows for the incorporation and improvement of new text-based analysis methods.

1.6. **Computer Forensics**

The field of computer forensics is far more developed than that of network forensics. FLI and the visualization extension are truly network forensic applications, and an overview of the computer forensics field is not necessary;
however, there is one specific consideration from the field that bears weight in our analysis and may be useful for understanding some of the design goals and decisions.

1.6.1. **Role of computer**

When computers are involved in an incident, criminal or otherwise, they have three basic roles. Below, each is discussed along with its relevance to the FLI Map visualization extension.

- **Target**
  
  Many times, a computer itself is the target of an attack. For example, attackers may be attempting to steal data directly off a machine. In FLI, this is often referred to as the “anchor” computer (denoted by IP Address), or the “victim.” This serves as a starting point for the forensic analyses.

- **Tool**
  
  As mentioned in the motivation, advancing computer technology has not only provided rich targets for attackers but also new means for them to reach those targets. Attackers wishing to target computer systems will often employ other computers as tools to carry out an attack, for everything from a simple scripted attack to complex, distributed, multi-stage attacks. Endpoints of stepping stone and multi-stage attack analyses fall into this category.

- **Latent Evidence Repository**
  
  Finally, a computer may contain latent evidence of probative value to an investigation. This may include, but is not limited to, log files, user files, operating system cache files, deleted files, or memory contents. This application deals specifically with the log files
containing evidence from hops along an attack path. Most of the computers displayed fall into this category.

1.7. **Intrusion Detection**

The field of intrusion detection is mentioned here, as many of the analysis methods utilized by FLI Map are rooted in intrusion detection research. They were originally developed to function in the real-time, on-the-fly world of the intrusion detection system, or IDS. Most recognize the potential for application after the fact, but focus design goals on real time analysis. We will discuss related work in the field, along with associated visualizations if any exist.

1.7.1. **Stepping Stone Analysis**

Stepping stone analysis has become a very important part of the computer security field at large, as many attackers log in to long chains of computers to conceal their true location. There are essentially three types of analysis methods:

- **Content Based**
  
  These algorithms either modify or compare the content of the transmissions themselves. These methods are completely ineffective against encrypted traffic, such as secure shell (SSH) connections.

- **Active**
  
  Active methods, such as sleepy watermark tracing, modify the timing of individual packets to produce an overall “fingerprint” for the stream. These systems would have to be in place prior to an event occurring. It may also be argued that they violate the non-interference principle of digital forensic tools, as they modify a timing value and then use it to determine involvement.

- **Passive**
  
  Passive methods are based solely on timing information, and do not require additional capabilities to be deployed prior to the event occurring.

There are few papers detailing approaches to visualizing stepping stone analysis results or evidence. We were unable to locate any research with such visualization as its primary focus.

1.7.2. Multistage Attack Analysis

Multi-stage attack analysis is a complicated area. First, the ability to detect such attacks is based on observables, or specific actions that indicate attacker intention [15]. Some of this inference is based on attacker profiling. Marcus Rogers conducted an in-depth study of computer criminal behavior, which is informative for some types of attackers [17]. Building evidence graphs for this type of evidence has also been addressed [16].

1.7.3. Connection Analysis

Much of the research in this area is focused on misuse detection. The work related to the FLI visualization intersects with information visualization, and aims to make anomalies easily identifiable. These anomalies can typically be investigated further to determine if it is misuse or not. Examples include [3, 7, 9].

1.8. Attacker Modeling

Another important, but often misunderstood, component of investigation is attacker modeling, sometimes called profiling. The practice is most widely used in the development of multi-stage attack analysis systems. However, attacker modeling is also informative for developing integrated analysis visualizations, where output from one analysis method feeds into another. It is important to
have a model of attack behavior, so that it can be analyzed as behaviors change and be adjusted accordingly.

Another important component of modeling is that it often provides the only support investigators have to identify an attacker’s mission. Missions can sometimes be inferred, but not supported [15]. Attacks generally move in phases, which vary based on the model. These phases, or attack states, can be observed as states themselves or in the transitions between states. An example would be a port-scanning alert – it may indicate that an attacker is conducting system discovery, or that they have *begun* or *returned* to the system discovery phase of the attack.
CHAPTER 2. DESIGN & ARCHITECTURE

2.1. Overview

This application is an extension of the FLI Map component of the Forensic Log Investigator, developed by Thieu Pham [1]. It is designed to provide network forensics examiners to perform complex analyses and quickly interpret the results. This extension consists of two major components, an analysis framework and a display framework. The application is integrated with the larger FLI software package, which utilizes client-server architecture.

The architecture of the FLI Map component also draws extensively on the Google Maps API. In addition, it utilizes Java, Java Server Pages (JSP), JavaScript, an Oracle database for evidence, and a flat-file database for geocoding IP addresses.

The overall process for FLI Map visualization involves three basic steps; receive an analysis request from the user, perform the analysis, and display the results. The flowchart in Figure 2.1.2 below provides an overview of the process. The upper half of the diagram represents the analysis framework, which begins with the analysis request and produces an analysis result. The bottom half of the diagram shows the display framework process, receiving an analysis result and producing a map with the result drawn on it.
2.2. **Analysis Framework**

The analysis framework provides the capability for the visualization extension to process analysis requests. It is designed as a separate component so that it can be used by other FLI components in the future. This component functions as the text-based analysis of the network forensics visualization (see Section 1.5).

2.2.1. **FLI Analyzer**

The FLI Analyzer consists of four individual processes, each of which handles one type of analysis. The final component is designed to include additional analyses as they are implemented.
2.2.1.1. **Connection Analysis**

This analysis answers a very basic question, but one that is often relevant in forensic investigations – “who connected to whom?” The algorithm accepts both a starting, or anchor, IP address and the number of analysis levels desired. It recursively determines connecting IP addresses down a the desired number of levels.

2.2.1.2. **Stepping Stone Analysis**

![Stepping Stone Analysis Flowchart](image-url)

Figure 2.2.1.2.1. Stepping Stone Analysis Flowchart
The stepping stone analysis is an implementation of the data tick analysis algorithm [11]. This analysis method was chosen for its robustness against both delay and chaff. Other methods, such as ON/OFF, Deviation, IPD, State-Space, Multiscale, Detect-Attacks, and Detect-Attacks-Chaff (as named in [11]) were susceptible to different attack packet timing perturbations.

2.2.1.3. **Multi-Stage Attack Analysis**

![Multi-Stage Attack Analysis Flowchart](image)

**Figure 2.2.1.3.1. Multi-Stage Attack Analysis Flowchart**

Multi-stage attack analysis is based primarily on attacker modeling and intrusion detection system alerts. Since observables do not yet exist for attack phases such as logistics or detailed preparations, it
focuses on a breakdown of system discovery and attack events, with a focus on components of these states relevant to an investigation. It classifies results into categories based on the likely attack phase involved - reconnaissance, infiltration, attack, data exfiltration, or concealment. For example, data exfiltration is really a part of the attack execution phase, but has particular interest to investigators because it has a significant impact on response. For example, the response exfiltration of email records would likely be different than the response for exfiltration of a credit card database.

### 2.2.1.4. Full Analysis

![Full Analysis Flowchart](image)

This analysis is designed to combine the other methods in an elegant manner. Streams connecting to the anchor are analyzed for stepping stone paths and multi-stage attack involvement. Endpoints
from the multi-stage analysis are further analyzed to determine if such an attack was carried out via a stepping stone chain. Finally, end nodes are analyzed for connections. This final step is crucial to identifying information gaps in the current evidence repository. See Figure 2.3.2.4.1 below for an example.

2.2.2. FLI Analysis Result

In order for the results to be fully utilized, a common data format for results needed to be defined. Additionally, the common format allows for maximum flexibility and will not require modification to include additional types of network forensic analysis, however, it is easily extensible. The display framework is flexible in order to accommodate such extensions in the event that they become necessary.

This “analysis result” data type is implemented to contain results in the form of FLIEventMapInfo and sub results. A sub result is another AnalysisResult item, containing the results from a sub-component of the larger analysis request.

An example of this is shown in Figure 2.3.2.2.1 below, specifically referencing a stepping stone analysis result. The first hops away from the anchor are results, while the remaining legs out from each of these are contained in their own sub result.

2.3. Visualization Framework

The visualization framework for FLI Map is responsible for displaying the results of each analysis method. The process (see Figure 2.1.2) displays the provided analysis result on a map, utilizing the Google Maps API. The related processes and visual encodings are described in these sections. This component functions as the visualization and interactive techniques portions
of the network forensics visualization (see Section 1.5). It can also be used to produce legal presentation materials.

2.3.1. **Displaying Results**

The process of displaying a result must iterate through the set of results provided, and display them properly on a map. There are essentially two primary pieces of information displayed.

- **Actors** – individual IP addresses representing computers.
- **Associations** – a link between actors.

An analysis result contains actors as objects, with associations stored based on the structure of the result itself.

2.3.1.1. **Processing a Result**

When processing results, FLI Map is handling a single analysis result, which may contain additional results, also called sub-results. Individual results denote actors with an association to the analysis result’s “anchor.” A sub-result, then, has an anchor that is also an individual result for its parent result. Each result also contains type and subtype information, which further describes the associations between actors in the result. Section 2.3.2 describes the visual encodings for these information vectors in detail.

2.3.1.2. **Map-Based Display**

FLI Map, as the name implies, is a cartographic visualization. This provides visually identifiable location information crucial to investigators. When displaying actors determined by results, identified by IP addresses, each must be placed to a physical location, using geocoding. Additionally, cases where IP addresses are internal – and therefore not geocodable – must be handled.
2.3.1.2.1. Geocoding

The geocoding of IP addresses is performed primarily by the analysis methods as they collect results. This provides that each result with a valid public IP address has latitude and longitude information. However, some results do not have valid addresses, and they are mapped to locations across the top of the map (in the Arctic Sea).

2.3.1.2.2. Internal Address Mapping

Some results have internal IP addresses. In this case, a valid public IP address is obtained using the IP address of the logging device that captured it (collected along with the log itself).

2.3.1.3. Additional Information

The results provided by each algorithm contain additional information that is available by drill-down. This information includes the IP address of the log that collected the information, the number of separate events that qualify, and detailed location information such as city, county, and area code. This is done to assist investigators seeking to verify the jurisdictions involved in any investigation.

Another important piece of information for investigators is the party who is registered for the IP address. This information is obtainable by drill-down to the major WHOIS databases – ARIN, APNIC, and InterNIC.

2.3.2. Visual Encodings & Visual Metaphors

The FLI Map visualization provides the following information, using the encodings denoted below each. Following each encoding is a description of its use to display specific information vectors.
- Where are the actors physically located? (Physical Location of Actors)
  - Marker placement
    Encoding the physical location of each actor is based on geocoding the IP Address being mapped, using the Maxmind GeoIP API. Markers for internal addresses are mapped to the IP address of the log they were collected from. In the event that an IP address is not geocodable, or both addresses are internal, the marker will be displayed at the top of the map. This is done to prevent results from not displaying, and the investigator from having incomplete information.

- Who is associated with whom? (Association of Actors)
  - Line endpoints
    The associations between the actors are encoded as lines from one to another. Since the different analyses methods produce associations between multiple sets of actors, understanding the structure of the results relies primarily on this encoding.

- How are they associated? (Analysis Method Producing Association)
  - Line color
    While the association of actors is important, it is only a portion of a more complete view of their association. To an investigator, it is imperative to know whether the association is based on simple connection, if the association is part of a stepping stone chain, or if an attack has been identified. Each necessitates different actions on the investigator’s part. This encoding is based on the type and subtype of the analysis result, and provides investigators the ability to visually identify how actors are associated.
2.3.2.1. Connection Analysis

- Who is the focus of this analysis? (Anchor Location)
  - Marker icon

  A final piece of important information is visual identification of the starting point for the analysis itself. In FLI, this is referred to as the Anchor, or Victim, IP Address. This special actor is displayed with a different marker image than other actors. In addition, the map is centered upon this marker when the result is first displayed.

Connection analysis is a simple form of analysis designed to answer the question “Who connected to who?” In FLI, the analysis will review subsequent hops, out to the specified level. The example above shows only direct connections to the anchor IP address. The visualization currently limits the
number of levels to analyze to three in order to keep overall processing time at an acceptable level. The figure below illustrates the difference in results for each analysis type. Each hop out increases the number of nodes displayed, and each new level is displayed with lines representing each connection.

Figure 2.3.2.1.2. Comparison of Connection Analysis Results.
A) Direct Connections (1 Level) B) Secondary Connections (2 Levels) C) Tertiary Connections (3 Levels)
2.3.2.2. Stepping Stone Analysis

Figure 2.3.2.2.1. Stepping Stone Analysis Example Screenshot

Figure 2.3.2.2.1 shows the results of a stepping stone analysis performed on the same anchor above. Connections not correlated are not displayed, and those that are displayed in red. Also, stepping stone analysis is not limited to levels, and simply continues processing until no more streams correlate. While the stepping stone algorithm used is capable of solving the indirect stepping stone analysis problem, the method for retrieving events limits it to the direct stepping stone analysis problem. This decision to limit events considered was a performance trade off, as analyzing every possible connection stream dramatically increased processing time. This analysis is only capable of identifying streams that include the anchor as a hop in the chain.
2.3.2.3. **Multi-Stage Attack Analysis**

![Multi-stage Analysis Example Screenshot](image)

**Figure 2.3.2.3.1. Multi-stage Analysis Example Screenshot**

Multi-stage attack analysis has a slightly more in-depth encoding than the other singular methods. This is due to sub-typing of analysis results, based on event category. The line colorings are kept to similar colors so that visual identification as a multi-stage analysis result is maintained, but differentiation between subtypes, or attack model phases, is possible. The result above shows (from left to right) an attack, reconnaissance, and data exfiltration.
2.3.2.4. Full Analysis

Figure 2.3.2.4.1. Full Analysis Example Screenshot

The figure above illustrates the advantages of a combined visualization tool. First, investigators can easily identify the stepping stone chain involving the anchor, as they would have been able to see in the standard stepping stone analysis only visualization. However, we can also identify a second chain, unconnected to the first, which connects several of the nodes identified by the multi-stage analysis. This gives the investigator a much richer understanding of what took place; for example, they may infer that these events were related, as opposed to separate alarms – even though the attacks were distributed geographically and temporally. Furthermore, an investigator can quickly see that one endpoint of the stepping stone chain involving the anchor is outside the logs already collected, since we have no further connection data for that IP address.
The identification of this information gap can lead the investigator to quickly determine where he or she needs to collect additional information. Also, the visual encoding includes the information necessary to identify the jurisdiction. In the example above, the address in question is located in Oakland, CA.
CHAPTER 3. INFRASTRUCTURE

3.1. Computer Languages

This section describes the various languages utilized by the FLI Map visualization.

3.1.1. Java

Java is a high-level object oriented programming language, developed by Sun Microsystems. It provides a flexible backend to FLI, as it is compiled into machine-independent byte code.

3.1.2. Java Server Pages

Java Server Pages is a Java-based technology that assists in the creation of dynamic web content through server-side production of HTML pages.

3.1.3. JavaScript

JavaScript is a Java-like scripting language for client-side execution. It is responsible for pre-submission request handling, and managing the Google Maps API interaction within the client browser.

3.1.4. XML

XML, or eXtensible Markup Language, is a highly structured yet general purpose markup language. It is recommended by World Wide Web Consortium (W3C) as cross-platform, human readable, and portable. XML is widely used, and there are a number of applications and tools capable of working with and processing XML. Its extensible nature supports multiple dialects, transforms, and styling specifications.

3.1.5. XPath

XPath is essentially an XML query language, capable of addressing elements and values within an XML document. It operates on the logical structure of the document, using path notation. XPath also has functionality for comparing and manipulating those values.
3.1.6. **SQL**

SQL, or Structured Query Language, is designed for use with relational databases. SQL provides data retrieval and management capabilities.

3.1.7. **SQL/XML**

SQL/XML is the combination of the two technologies mentioned above for the specific purpose of dealing with XML data stored in relational databases. It allows the integration of traditional relational database primitives with XML-type data. It is retrieved and managed with a combination of SQL queries and XPath statements.

3.2. **Software Frameworks**

3.2.1. **Google Maps API**

The Google Maps API is heavily utilized by FLI Map for visualization of analysis results. The API provides functionality to utilize pre-developed maps and controls, as well as extensions for defining your own components.

3.2.2. **Maxmind GeoIP API**

The Maxmind GeoIP API provides IP Address geocoding services. In addition to providing latitude and longitude, it also provides further information to investigators such as country, city, and area code.

3.2.3. **Oracle DBMS**

Oracle Database is a DataBase Management System that supports SQL/XML. It provides the evidence storage for FLI. Oracle DMBS is both multi-threaded and multi-user capable.

3.2.4. **Adobe Flex 2.0**

Adobe Flex is a set of technologies that provide advanced interactive capabilities for websites, based on the Flash platform. Flex is a
presentation-layer technology, but other components, such as Flex Data Services, provide additional functionality.

3.2.5. **Java Enterprise Edition Application Server**

Java’s Enterprise Edition Application Server provides management of web application communication between the server and clients. JRun is the JEE application server utilized by FLI.

3.2.6. **Hibernate API**

The Hibernate API is an object-relational mapping solution for Java. It is primarily utilized by the larger FLI package.
CHAPTER 4. SUMMARY AND DISCUSSION

This chapter provides an overview of the work completed. First, the FLI Map analysis and visualization extension is compared to its design goals, and reviewed for how well each goal was met. Second, its place in the forensics process is discussed. Third, its major components are summarized. Finally, future work is outlined.

4.1. Design Goal Achievement

• Provide ability for investigators to visually correlate evidence and analysis across multiple logs

  This goal is accomplished with the combination of database-driven text-based analysis and map-based visualization. Each of the four visual metaphors correlates evidence across multiple log files, relying on the schema and import structure of FLI. The flexibility of FLI’s SQL/XML backend handles widely varying queries.

  This ability is present, but does not assist as greatly when performing stepping stone analysis. This is due to the time synchronization problem for logs collected from different locations. However, if the logs are synchronized, it will be capable of correlating results.

• Provide ability for investigators to visually correlate results from multiple types of analysis

  This design goal is handled entirely by the Full Analysis visual metaphor, as it is the only one to currently include multiple types of analyses. However, this capability is particularly powerful when combined with the above ability to correlate results across multiple log files.

  This combination allows investigators to gain a level of comprehension and certainty of result that would be extremely difficult, if not impossible, to
obtain by text-based analysis. The use of a visual metaphor increases the amount of information that can be viewed at one time.

- Assist in identifying information gaps

The visual encodings for association between actors provide investigators the opportunity to visually identify where associations are not present. Domain knowledge, as discussed by [3], is a critical investigator skill that has not been replaced by analysis. This knowledge leads investigators to know where associations should be present, but is not – an indication of missing information.

- Properly handle evidence when performing analysis

FLI Map and Analyzer provide reproducible results while maintaining the integrity of information in the database. Furthermore, the analysis never writes any information back to the database, so it cannot interfere with the evidence contained within. Finally, it logs analysis requests and results to a file for later review discovery.

The FLI framework does not yet include full case or evidence management features. As a result, maintaining minimalization during analysis is not fully accomplished. The inclusion of current case information to the package will allow for this goal to be realized fully.

4.2. **Use in Investigative Process**

The FLI Map visualization and analysis framework apply to the examination, analysis, and presentation phases of the model described in [2]. The may also assist in identification, if you consider the detection of information gaps part of the identification process. In the [3] model for network forensics, the work contained in this thesis applies to the text-based analysis, visualization-based analysis, analysis activities, analysis visualization, analysis validation, and legal presentation phases. The body of work is
primarily in the visualization-based analysis, analysis activities, and analysis visualization; the other components are either rudimentary or were developed elsewhere.

4.3. Component Review

The extension for FLI described in this thesis is broken down into two components, one that provides the text-based analysis backend, and another that handles the visualization and interaction with results.

4.3.1. Analysis Framework

The majority of the analysis framework consists of components that leverage existing packages, algorithm implementations, and APIs. Most of the work on this framework was integration, however, the full analysis method combining the varied types of text-based analysis was developed alongside the implementation of existing analysis algorithms.

The Analyzer and AnalysisResult classes in the fli.util package were developed as part of this thesis. Analyzer is a static class containing the procedures for analysis, while AnalysisResult was developed as a data structure for analysis output. The new development that went into this framework consisted of integrating analysis implementations with the FLI framework and evidence repository database.

4.3.2. Visualization Framework

The Adobe Flex component, FLI Map, consists of a Java Server Pages frame. This architecture allows most of the processing to be handled at the server. Keeping within the requirements for digital evidence, particularly integrity and non-interference, evidentiary data does not traverse the network prior to analysis, and is not subject to modification in transit.

The development of this component comprises the work on analysis and interaction with data, and is primarily novel. The logic for each visual
metaphor is present in this component. This framework also handles user input, and provides default levels for analysis parameters.

4.3.3. Categorization

While this extended FLI Map is a visualization application, it is not easily categorized using prior research as a guide. It utilizes geographic spatial information to map public IP addresses, but the spatial information is not truly inherent in the data – particularly if you consider the possibility of spoofed IP addresses. Its use as a forensic application separates it in some respects from security visualization applications, although this seems to be the most appropriate category.

4.4. Evaluation

Evaluating these components must be based on their function. The analysis component, which provides text-based analysis, should be evaluated for effectiveness and performance. However, the methods utilized were already established, and their evaluation should be left to such research. However, the visualization component was primarily novel work, and evaluation is necessary.

There are several approaches to evaluating visualization applications, and they are summarized in [10]. Recommended in this work is evaluation based on “how and why” a given visualization is effective. This evaluation is based on the visual encodings used as well as the overall visual metaphor. Findings from prior visualization research provide a basis for this evaluation. The “How” part of the evaluation is primarily handled in the visual encodings and metaphors section above (2.3.2). The “Why” is addressed here.

The choice of markers to encode an address is effective because it associates the information with a visual symbol of the device represented – a computer. The use of a map-based display is effective as it provides the user
an orientation they are already familiar with. The use of a line was effective to represent association because it provides a visual connection to mirror the virtual. Our choice of color to represent analysis type is effective because color has been found to be effective for encoding nominal data. Additionally, the use of slightly varying colors to represent sub-types was effective because it allows distinction without disrupting the overall visual metaphor. Essentially, this allows quick visual correlation of related streams, with a cognitive distinction of sub-type following the pre-attentive grouping.

4.5. **Future Work**

The next phase of developing this visualization includes the addition of an information visualization for internal IP addresses, built into the larger map-based visualization. This visualization has the ability to use marker placement as an encoding, and some further research must to be done to determine the best choice of an information vector for this encoding. Other effective encodings, such as line width, marker size, and line glyphs are still available for encoding additional information vectors, should additional analysis methods require them.

Future work could include visualization for other types of network forensics text-based analysis, such as data mining. Also, the multi-stage analysis method could be expanded to handle multiple attacker models, based on named actors or class [15]. Learning machines could be applied in an attempt to supplant investigator domain knowledge.

Further work on FLI, particularly in case and evidence management, will enable more specific queries and analyses, which should improve the performance of the analysis framework. In addition, they are required to prepare the analysis component for the law enforcement use, due to the minimalization requirement [4].
BIBLIOGRAPHY


ACKNOWLEDGEMENT

I would like to thank Dr. Yong Guan for his guidance and support on this thesis. I would also like to thank Theiu Pham for his extraordinary willingness to support my extension of his FLI software package. I would also like to thank my committee members, Dr. Johnny Wong and Dr. Doug Jacobsen for their time and efforts to support this project.

To my fellow Cyber Corps students, Chris Hoff, Jason Maughan, Steve Eilers, and Lucas Witt: it has been a privilege to work with you over the last two years. Thank you for your support, encouragement, and for challenging me to do better. I wish you all the best of luck. I would also like to thank Dr. Jan Wiersema and Dr. Barb Licklider for their guidance and development of the Cyber Corps leadership development program. I extend my gratitude to the National Science Foundation and the Department of Homeland Security, who sponsor the Scholarship for Service Program that supported me during my studies at Iowa State University.

Finally, I would like to thank my family. This work is dedicated to you in thanks for your love and support throughout my life. I could never express how much you mean to me. I love you all.