Chapter 7

A FRAMEWORK FOR EMAIL INVESTIGATIONS

Automated Information Extraction and Linkage Discovery

Anthony Persaud and Yong Guan

Abstract Email communications are commonly used by criminal entities to perpetrate illegal activities such as fraud and phishing scams as well as for transmitting threats and viruses. Due to the complexity of the task, it is often difficult for investigators to manually analyze email-related evidence. Automated techniques for email analysis can significantly enhance computer crime investigations. This paper proposes a framework for email investigations that incorporates automated techniques for information extraction and linkage discovery. The application of text/data mining and link analysis techniques assists in inferring social networks and in accurately correlating events and activities from email-related evidence.

Keywords: Email investigations, linkage analysis, information extraction

1. Introduction

Numerous crimes are committed using email communications. Emails are commonly used to perpetrate phishing scams as well as to transmit threats and viruses. In many cases, email communications provide evidence of conspiracy, helping identify new suspects and linking them to specific criminal activities. For example, in the Enron scandal, investigations of email correspondence showed that several top executives conspired to commit fraud and money laundering [6]. More recently, email evidence suggested that Merck executives may have known about the deadly side-effects of Vioxx since March 2000, long before it was removed from store shelves [3, 11].

Due to the complexity of the task, it is often difficult for investigators to manually analyze email-related evidence. Automated techniques for email analysis can significantly enhance computer crime investigations. This paper proposes a framework for email investigations that incorporates automated techniques for information extraction and linkage discovery. The application of text/data mining and link analysis techniques assists in inferring social networks and in accurately correlating events and activities from email-related evidence.

2. Problem Definition

Email messages comprise header information and a message body. Information in email messages ranges from partially structured information to unstructured information.

Simple Mail Transfer Protocol (SMTP) headers are examples of fixed information fields in email messages that provide formatted information on routing, time, date and addresses. Email message bodies contain unstructured information because no regulations are specified for message content. For example, a message can be written in a language other than English, or it may contain undefined acronyms, or it may use different writing styles and not have correct punctuation.

Analyzing email header information and the frequency of messages can provide insights into the communication patterns of individuals. The message body is essential to understanding the context of these patterns. Manual analysis of email-related evidence is arduous and time consuming. Unfortunately, the mixture of structured and unstructured information in email messages makes it difficult to create a fully automated process for analysis. Therefore, the main goal of this work is to provide a framework for automating information extraction and analysis during investigations of email-related evidence.

3. Related Work

Link discovery encompasses a broad range of topics such as discovering social networks, analyzing fraudulent behavior, detecting preemptive threats and modeling group activities. The InFlow organizational network analysis tool constructs social maps from email messages [8]. InFlow uses the `To` and `From` header fields and the frequency of emails sent between individuals to create social maps of organizations.

Other research uses email analysis to understand user behavior. Boyd and Potter [2] created Social Network Fragments as a self-awareness application for digital identity management. The system uses address fields from user email files to construct a social behavior map. For

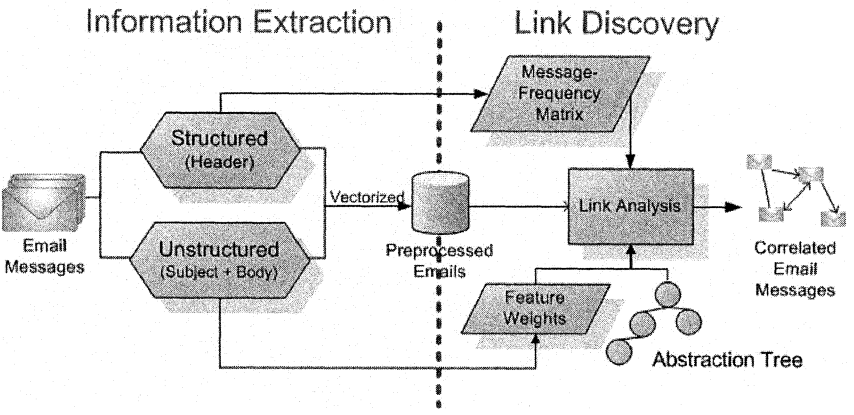

Figure 1. Email analysis framework.

example, the system helps identify a user's interests during a certain time period by analyzing his/her mailing list subscriptions.

4. Email Investigation Framework

Figure 1 presents an overview of the proposed email investigation framework. The framework involves two phases: information extraction and link discovery.

The information extraction phase involves structured information extraction and unstructured information extraction, which condense and summarize email messages using a feature vector format. Additionally, as described below, the processes produce a message frequency matrix and a set of feature weights.

The link discovery phase analyzes the vector-formatted email files, message frequency matrix and the corresponding feature weights, producing correlated pairs that manifest hidden relationships between communicating parties.

5. Information Extraction

This section describes the techniques used to extract structured and unstructured information.

5.1 Structured Information Extraction

The main goal of structured information extraction is to build messaging relationships. Therefore, the focus is on extracting information contained in the `To`, `From`, `Cc`, `Bcc`, `Reply-To` and `Delivered-To` address

Table 1. Message frequency matrix.

	$addr_1$	$addr_2$	...	$addr_{n-1}$	$addr_n$
$addr_1$	0	f_{12}	...	$f_{1(n-1)}$	f_{1n}
$addr_2$	f_{21}	0	...	$f_{2(n-1)}$	f_{2n}
...	...	...	...	...	...
$addr_{n-1}$	$f_{(n-1)1}$	$f_{(n-1)2}$	...	0	$f_{(n-1)n}$
$addr_n$	f_{n1}	f_{n2}	...	$f_{n(n-1)}$	0

fields. The address fields in SMTP headers are described in RFC 821 [9]. The format of email addresses is defined by RFC 2822 [10].

Address information is stored in a message frequency matrix that records the frequency of communication between pairs of email addresses. A sample message frequency matrix is presented in Table 1, where $addr_x$ is the primary email address of individual x and f_{xy} is the frequency of messages sent by $addr_x$ and received by $addr_y$. Note that `Received-By` is used instead of `Sent-To` because multiple individuals may have received an email message based on addresses in the `Cc` and `Bcc` fields.

Two rules are followed when collecting email addresses. First, all known email addresses used by a sender are recorded as a single (primary) address. For example, if an individual uses `me@email.com` and `myself@email.com`, one of the addresses, say `me@email.com`, is designated as the primary address. It does not matter which address is chosen as long as frequencies are recorded for a single address.

The second rule is that mailing list addresses, when known, are expanded to their corresponding membership addresses. Each membership address should be recorded in the message frequency matrix. For example, suppose mailing list `mlist@email.com` contains the addresses `them@email.com`, `him@email.com` and `her@email.com`. Suppose a message is sent from `me@email.com` to `mlist@email.com`. Then, the mailing list address `mlist@email.com` is expanded to {`them@email.com`, `him@email.com`, `her@email.com`}. In addition, the message frequency count is increased by one for each communication pair: {`me@email.com`, `them@email.com`}, {`me@email.com`, `him@email.com`}, {`me@email.com`, `her@email.com`} and {`me@email.com`, `mlist@email.com`}.

Figure 2 presents the general procedure for extracting structured information. First, the sender's address is extracted from the `From` and `Reply-To` fields (multiple email addresses for a user are mapped to the same user). Next, receiver addresses are extracted from all address fields. A receiver address that is a mailing list address is expanded to its corresponding membership addresses, and all the membership addresses are added to the receiver list (duplicate entries are discarded). Finally, for

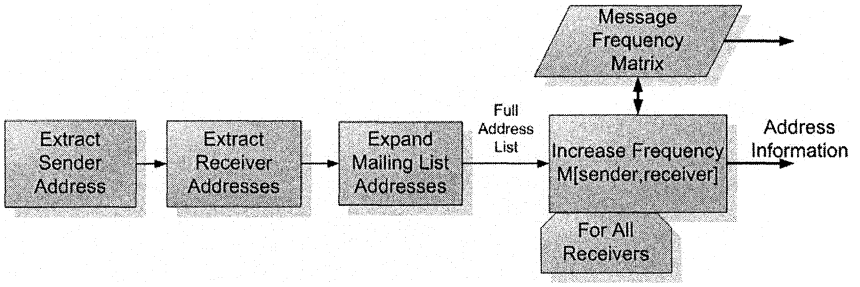

Figure 2. Structured information extraction.

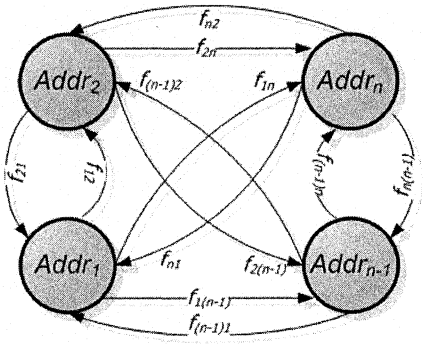

Figure 3. Social network graph from message frequency matrix.

each [`sender`, `receiver`] pair, the message frequency matrix is updated by incrementing the frequency value of the communication pair by one. After all the email messages are processed, the frequency values in the matrix are normalized based on the total number of communicating entities.

A directed network graph (social network graph) can be constructed from the set of communication pairs in the matrix (see Figure 3). Modeling address information as a directed social network graph helps establish asymmetric relationships between communicating parties (e.g., group leader and members). For example, a high message frequency between `me@company.com` and `you@company.com` may indicate that a strong relationship exists between the individuals. Since both email addresses belong to the same domain (`company.com`), one might infer that the two individuals are employees of the same company. It is normal to see high frequencies for emails between co-workers.

On the other hand, if `you@company.com` has a high frequency of communication with `him@competitor.com`, it could be inferred that `you@company.com` is passing secrets to `him@competitor.com` at a competing company.

Table 2. Example feature vector.

t	F[t]
$term_1$	$freq_1$
$term_2$	$freq_2$
$term_3$	$freq_3$

5.2 Unstructured Information Extraction

The unstructured sections in an email message include the `Subject` field, the message body and attachments. This paper focuses on plain-text content; the analysis of email attachments is a topic for future research.

The subject line in an email header can be considered to be unstructured information. The combination of the subject and message body is defined as the "feature string." The feature string contains terms that describe its content. These features (key terms) range from unique non-dictionary terms and numerical sequences to complex components such as web addresses (e.g., `www.google.com`). Relevant features must be extracted from the feature string to produce a summarized version of each message. Internet document indexing schemes (e.g., [1, 5]) can be applied to extract information from feature strings.

When processing a feature string, single terms should be extracted that best summarize the contents. This can range from using simple grammatical rules (e.g., ignoring articles and prepositions) to using a large dictionary list. When extracting these features, generalizations between words should be used. For example, the numerical sequence {`six`, `6`, `VI`} could be summarized as {`6`}, known nouns and acronyms {`U.S.`, `USA`} as {`USA`}, and verb variations {`run`, `running`, `ran`} as {`run`}.

Extracting semantic word pairs is essential to obtaining a full understanding of a feature string. For example, extracting the word pair {`Air Force`} is better than extracting {`Air`} and {`Force`} separately. This procedure can be implemented using natural language processing algorithms that recognize [`adjective`, `noun`] sequences, and provide table-lookups of known semantic word pairs.

After all the features are extracted from a feature string, a feature vector is constructed by extracting and recording the frequencies of each feature. Table 2 provides an example of a feature vector where $F[t]$ represents the frequency of feature t in the feature string.

The higher the frequency of a specific feature in a message, the greater the relevance between the topic of the email and that feature. A feature vector can be used as a comparison point when performing linkage anal-

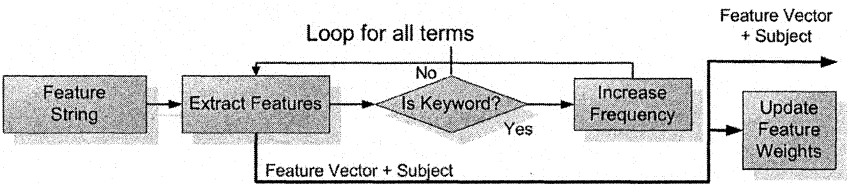

Figure 4. Unstructured information extraction.

ysis between email messages, i.e., clustering email message pairs that have strong relevance to each other.

Figure 4 shows the general procedure for extracting unstructured information. First, the subject and the message body are concatenated to produce a feature string, and the subject line is stored as the discussion thread in the feature vector. Next, each single term and semantic word pair in the feature string is checked to see if it can be considered to be a feature. Terms that are not considered keywords (or word pairs) are ignored. A term that is considered to be a keyword (or word pair) is extracted, and an entry is created for it in the feature vector for the email if none exists (and the frequency is set to one). If the feature already exists in the feature vector, the current frequency of that feature is incremented by one.

After all the features have been collected, the feature weights list is updated to include all the features extracted from the current email. The frequency count is increased by the frequency value in the feature vector for all the terms in the feature weights list.

5.3 Feature Weights

Suppose email evidence is collected from Company `XYZ`. Then, it is not appropriate to use the term `XYZ` as a unique feature. This is because `XYZ` has a high probability of being found in the evidence, e.g., because employees may mention the company's name in their correspondence or a disclaimer that mentions the company's name is appended to every email message. Therefore, the list of extracted features from an entire email set must be collected to produce feature weight statistics that help in evaluating feature uniqueness during link analysis.

Feature weights are calculated as follows. If k_i is a feature in feature set K, and f_i is the frequency of k_i. Then, the weight $\phi(k_i)$ of feature k_i is defined as $\phi(k_i) = 1 - \frac{f_i}{|K|}$.

6. Link Discovery

After email messages are processed for structured and unstructured data, various link analysis schemes can be used to discover hidden relationships between email users.

6.1 Link Analysis

Email evidence is analyzed by comparing each pair of email messages and calculating a suspicion value using the information contained in feature vectors and the message frequency matrix. This scheme potentially yields high true positive rates while maintaining acceptable false positive rates. These benefits come at the cost of performing $\binom{n}{2}$ comparisons.

6.2 Multiple Levels of Abstraction

Linkage analysis schemes often encounter obstacles when comparing email messages. Data specificity is a problem. For example, suppose one email has the feature `Merlot` and another has the feature `Chardonnay`. Comparing these features directly does not produce a correlation because the two features do not have direct lexical similarity. One solution is to improve the correlation between the feature vectors using multiple levels of abstraction (MLA).

Individuals write text in different contexts and perspectives; therefore, perfectly matching the features of two email messages will be very uncommon. Creating a decision tree based on taxonomic data can help produce higher correlation values between emails.

Several algorithms, e.g., the Attribute Value Taxonomy-Guided Decision Tree Algorithm [12], operate at different levels of specificity. WordNet[7], a lexical database for the English language, produces synonyms for various terms that represent a single lexical concept. These resources can be adapted to work with email messages at various levels of abstraction.

Consider the simple abstraction tree shown in Figure 5. It can be determined that `Merlot` and `Chardonnay` are both types of `Wines` using one level of abstraction ($\alpha = 1$). Therefore, if the features `Merlot` and `Chardonnay` are generalized to `Wines`, there is a direct lexical similarity, which produces a higher correlation match between the features. The increase in correlation can be fine-tuned using different levels of detail and precision in an abstraction tree. The main reason for using MLA is to reduce the inter-cluster distance between email messages by enhancing their relevance.

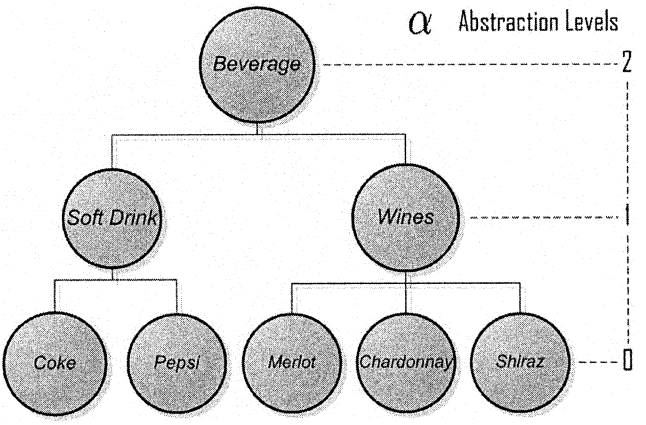

Figure 5. Example abstraction tree.

6.3 Suspicion Level

A suspicion level S, where $0 \leq S \leq 1$, is used to determine whether or not email pairs are correlated. The inter-cluster distance between message pairs is determined by comparing the set of features between feature vector pairs. The suspicion level is calculated by combining MLA with the corresponding weights of the intersection of related features.

Let F_i and F_j be feature vectors from emails i and j, where $F_x[y]$ is the frequency of feature y in F_x. Furthermore, define $A_{ij} = F_i \cup F_j$ and let $M_{ij} = F_i \cap F_j$ using MLA. Therefore, $M_{ij} \subseteq A_{ij} \subseteq K$, where K is the set of features extracted from the emails. Additionally, let $\phi(k_i)$ be the feature weight of feature k_i. Then, the suspicion level S is given by:

$$S = 0 \leq \frac{\sum_{p=1}^{|M_{ij}|} \phi(m_p) * \max(F_i[m_p], F_j[m_p])}{\sum_{q=1}^{|A_{ij}|} \phi(a_q) * \max(F_i[a_q], F_j[a_q])} \leq 1 \quad (1)$$

When performing link analysis, a threshold should be chosen to determine the suspicion level that is needed to consider an email message pair correlated.

Figure 6 outlines the procedure for performing link analysis for email messages. For each pair of email feature vectors (e_1, e_2), the subject and initial header information are used to determine whether or not the emails belong to the same discussion thread. If they are, the two emails are reported as correlated. If not, matching features are identified using the feature vectors for each message and MLA. Next, a suspicion level is computed using the message frequency matrix and the statistical feature weights of the matching features of the two email messages. If the

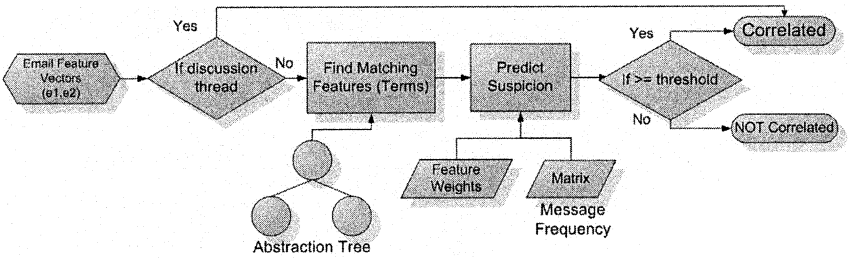

Figure 6. Link analysis.

Table 3. Results for different generalization heights.

α	False Positives (4704)	True Positives (246)	PPV
0	46 (0.98%)	74 (30.08%)	61.66%
1	106 (2.25%)	96 (39.02%)	46.60%
2	233 (4.95%)	143 (58.13%)	38.03%

suspicion value is higher than a user-defined threshold, the two emails are reported as correlated. Otherwise, the emails are reported as not correlated.

7. Results and Discussion

A training data set of 100 email messages was obtained from a personal inbox and modified manually for use in the experiment. A database of 150,843 words [4] was used to determine if a word was considered to be a feature in a feature string. A static two-level abstraction tree that specifically related to the content of the training email messages was implemented. A minimum threshold value of 0.7 (70% suspicion) was used as the cut-off point to decide whether or not two emails were correlated. Email pairs in different clusters reported as correlated were considered to be false positives. Email pairs in the same cluster reported as not correlated were considered to be false negatives. Pairs in the same cluster reported as correlated were designated as true positives.

The results obtained for various generalization heights (α) are presented in Table 3. Note that the positive predictive value (PPV) is defined as the probability that an email message pair is a true positive when restricted to the entire set of email message pairs reported as correlated. The main findings are discussed below.

A trade-off exists when using MLA. Using an abstraction tree in link analysis provides the ability to increase the total number of true positives at the cost of increasing the total number of false positives.

The higher the level of abstraction used in link analysis on the training set, the greater the number of email pairs that become relevant to each other. This increases the number of correct relationships found between email messages; however, more pairs of unrelated emails are identified as being relevant. For the training data set, an abstraction level (α) of one produced acceptable false positive and true positive rates.

The number of features extracted from email messages significantly affects correlation outcomes. It was found that the number of extracted features from an email increases the probability that an email message pair is correlated. The process of selecting features from a feature string is a user-defined process. If the selection process is not strict (i.e., most terms are considered features), then more features are generalized using MLA to increase correlations between message pairs.

8. Conclusions

Email messages contain valuable information that can be used to deduce social networks, and correlate events and activities in cyber crime investigations. The proposed email investigation framework, based on information extraction, linkage analysis, message frequencies and multiple abstraction levels, can automate the analysis of email evidence. While the initial results on email correlation are promising, statistical techniques must be employed to enhance linkage analysis based on abstraction trees and feature weights. Research efforts should also focus on integrating advanced text mining and data mining techniques in the email investigation framework.

Acknowledgements

This research was supported in part by NSF Grant DUE-0313837 and the GEM Fellowship Program. The authors also wish to thank Dr. Diane Cook and anonymous reviewers for valuable discussions and helpful comments on earlier versions of this paper.

References

[1] M. Berry (Ed.), *Survey of Text Mining: Clustering, Classification and Retrieval*, Springer-Verlag, New York, 2003.

[2] D. Boyd and J. Potter, Social network fragments (smg.media.mit.edu/projects/SocialNetworkFragments), 2004.

[3] Cable News Network, Merck's Vioxx e-mail scrutinized (money.cnn.com/2004/11/01/news/fortune500/merck), November 1, 2004.

[4] CTAN, English dictionary list (ctan.tug.org/tex-archive/systems/win32/winedt/dict/english.zip).

[5] B. Davison, Unifying text and link analysis, *Proceedings of the IJCAI Workshop on Text Mining and Link Analysis* (www.cs.cmu.edu/~dunja/TextLink2003/Papers/DavisonTextLink03.pdf), 2003.

[6] Federal Energy Regulatory Commission, Information released in the Enron investigation (www.ferc.gov/industries/electric/indus-act/wec/enron/info-release.asp), 2005.

[7] C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database*, MIT Press, Cambridge, Massachusetts, 1998.

[8] V. Krebs, Discovering social networks and communities in email flows (www.orgnet.com/email.html), 2003.

[9] J. Postel, Simple Mail Transfer Protocol, RFC 821, August 1982.

[10] P. Resnick, Internet message format, RFC 2822, April 2001.

[11] Wall Street Journal, Merck down 6.4% on report company knew of Vioxx risk early (online.wsj.com), November 2004.

[12] J. Zhang and V. Honavar, Learning from attribute value taxonomies and partially specified instances, *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 880-887, 2003.