Topic Chains for Determining Risk of Unauthorized Information Transfer

by

Jeremy Wright

A Thesis Presented in Partial Fulfillment
of the Requirement for the Degree
Master of Science

Approved November 2014 by the
Graduate Supervisory Committee:

Violet Syrotiuk, Chair
Hasan Davulcu
Stephen Yau

ARIZONA STATE UNIVERSITY

December 2014

ABSTRACT

Corporations invest considerable resources to create, preserve and analyze their data; yet while organizations are interested in protecting against unauthorized data transfer, there lacks a comprehensive metric to discriminate what data are at risk of leaking.

This thesis motivates the need for a quantitative leakage risk metric, and provides a risk assessment system, called Whispers, for computing it. Using unsupervised machine learning techniques, Whispers uncovers themes in an organization's document corpus, including previously unknown or unclassified data. Then, by correlating the document with its authors, Whispers can identify which data are easier to contain, and conversely which are at risk.

Using the Enron email database, Whispers constructs a social network segmented by topic themes. This graph uncovers communication channels within the organization. Using this social network, Whispers determines the risk of each topic by measuring the rate at which simulated leaks are *not* detected. For the Enron set, Whispers identified 18 separate topic themes between January 1999 and December 2000. The highest risk data emanated from the legal department with a leakage risk as high as 60%.

To my mother who taught me to ask, "Why?"

To my sister who teaches me to see another perspective.

To my wife who's unfailing love,
boundless support, and selfless dedication makes everything I do possible.

To my daughter who makes it all worthwhile.

ACKNOWLEDGEMENTS

Foremost, I express my sincerest gratitude to my advisor Dr. Violet Syrotiuk for her continuous encouragement and support over relentless revision. I could not have had a better advisor and mentor for my Master's research.

Thank you Dr. Yann-Hang Lee. The opportunity to teach and freedom to design CSE-394 has afforded me more opportunity than I can measure.

Thank you Arun Balaji Buduru, and David Lucero for the initial ideas proposed in Cloud Assure. Thank you Dr. Stephen Yau for supervising our budding research on Cloud Assure. His feedback gave direction and focus to this related project. Whispers is an extension of a component founded in Cloud Assure.

Thank you Casey Kelso both his initial encouragement to start my Master's program, and his ongoing encouragement. His insight on practical matters concerning implementation were enlightening, and fruitful.

Thank you Chris Ryan for your invaluable friendship. Hacking Day got me started in software. I am eternally indebted to you for the hours we spent together as the Ex Cædens. We are finally over the wall.

Thank you Chrissy Kaiserman. Her keen eye for design, and forthright review made an impressive user experience for Whispers.

Thank you to Chris Robison for editing support, and an available kitchen table where I spent hours studying.

Finally, I thank my technical cadre: John Sutherland, Craig Fisher, Craig Brandenburg, and Chris Anderson. Our discussions continue to color my ideas about software in more ways than I can enumerate.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

A 2008 study commissioned by Cisco Systems of more than 2000 IT professionals in 10 countries revealed a startling statistic: 39% of these professionals were more concerned about threats from their peers than external attackers [1]. The reason for this concern? Almost half of the professionals questioned reported that in addition to employees dealing with more information than ever before, they are not receiving the proper training in data security. All this adds up to a shocking statistic: 11% of the professionals reported that either they or an employee they knew, accessed unauthorized information and sold it for profit. Personal devices make it ever easier to store data, making the problem of tracking access increasingly more difficult. According to a May 2009 US federal government report, American business losses between 2008 and 2009 due to cyber-attacks grew to more than $1 trillion of intellectual property [2]. Additionally, the impact of unauthorized data transfer goes beyond monetary damages. For example, hackers against the security company HBGary were sentenced to prison, some with sentences as high as 124 years [3].

From large sums of money, to jail time, the fallout of unauthorized data transfer are severe. Software such as firewalls, policy checkers, and virus scanners attempt to address unauthorized transfer, but protecting all data can be insurmountable in both expense or lost business agility due to strident security. To reduce the costs of protecting one's data, one first needs to identify the data truly at risk.

Corporations invest considerable resources to create, preserve and analyze their data; yet while organizations are interested in protecting against unauthorized data transfer, there lacks a comprehensive metric to discriminate what data are at risk of leaking. Google demonstrates a critical business advantage currently hinges on its ability to leverage their market data [4]. It stands then that data is valuable, and companies need to actively measure that value. Within the government space the risk is typically mitigated by clearances and compartmentalization of clearances. However, data changes quickly. In the face of ever present change, how does one assess the risk of new unclassified data? Furthermore, how is that risk changing? Are the people managing that data properly compartmentalized? We are missing a succinct metric to describe this risk. In a 1921 book on Economics, Frank H. Knight states that the difference between uncertainty and risk is that risk can be measured [5]. Without a comprehensive tool to estimate the risk within an organization, leaders are left with uncertainty about the state of their intellectual property.

This thesis implements a risk assessment system called Whispers to quantify the present and emerging risk within an organization's documents. Whispers uses unsupervised machine learning techniques to summarize documents and correlate them with related documents. These groups of documents form a topic group. It is essential, however insufficient to simply classify the data. Classification alone is not a clear measure of leakage risk. Data does not leak itself. As Cisco established, employees are leaking it [1], ergo the classification must be put in context of the people who create, maintain and manage that data. Classification combined with a social network forms a lens to clarify the picture of risk. Ergo, Whispers then traces the documents within a topic group to the authors, and recipients. The resulting cluster of people, form a social group of the topic. Whispers then applies a novel method to estimate a risk metric for each social group. The result is an impartial, automatic measure that security officers may use to quantify their security procedures and overall risk analysis.

Whispers discovers potential data leakages by measuring the connectedness of separate social groups. *Whispers* uses three primary tools to perform its work. First, topic modeling discovers the underlying thematic structure of the emails. From the topics, Whispers assembles the social groups surrounding those topics. The final predictive block identifies messages sent between social groups.

This thesis is divided into 2 primary parts. Part one, composed of Chapters 2, and 3 addresses the classification techniques evaluated in the design of Whispers, and summarizes other works on the problem of data leakage. Chapter 3 describes the design of Whispers as a risk assessment system. Part two, composed of Chapters 4 and 5, describes the results of Whispers' application on the Enron email data set [6] and the future work exposed by Whispers.

Chapter 2

RELATED WORK

Whispers is risk assessment system that uses information retrieval techniques. It leverages machine learning techniques to filter a sea of text to uncover leakage events within an organization. Hence, the existing research in information retrieval and data mining is essential to understanding Whispers' structure and design decisions. To color the related works in this chapter Whispers defines data leak as Definition 2.1; our focus is on data leakage via email. This chapter is divided into three primary sections Section 2.1 presents a preliminary work, Cloud Assure, which exposed the need for a risk metric. Section 2.2 presents Topic Models as an unsupervised technique for extracting underlying themes in a corpus. Finally, Section 2.3 discusses work with similar goals which attempts to draw similar conclusions.

Definition 2.1 *Data Leak := A leak is sending an email, intentionally, or unintentionally, containing a particular topic to someone outside that topic's social group (Definition 2.2).*

For example, Alice, Bob, Dave and Eve are employees of a company. If numerous people write emails to Alice and Bob about a topic on shoes, Alice and Bob form a shoes social group. Eve is external to the shoes social group. If Dave then emails Eve about shoes, Dave leaked data.

Definition 2.1 describes just one perspective on the data leakage definition. A more simple definition is, "data found in an unauthorized place" [7]. Cisco defines a special case of insider threat data leakage when a former employee fails to return company electronics [1]. This latter definition is too narrow for our application since Whispers intends to track risk of leakage for both current and former employees. Thus, the simpler definition is more useful, and Whispers' working definition 2.1 is simply a formal extension, expressing data leakage as a function of social groups. This offers the option to unambiguously, and qualitatively measure leakage while a more simple definition is open to interpretation.

2.1 Cloud Assure

Cloud Assure is a preliminary project published as part of the ASU Information Assurance Conference [8]. Cloud Assure's focus is to identify leaking emails by evaluating the trust of the senders and receivers. This project exposed the fact that many data leakage solutions require data to be leaked to demonstrate their overall value. This motivates identifying graduated levels of risk for an organization's intellectual property.

Figure 2.1, describes the high level structure of Cloud Assure. Essentially, Cloud Assure accepts email as input, and offers a boolean recommendation if the email should be sent. If Cloud Assure believes this email was a leak, Cloud Assure recommends not to send. Cloud Assure uses a Markov Decision Process (MDP) to estimate the potential risk based on individual's trust values. The finding however was that the trust metric could be easily gamed, and users could leak emails undetected, and still maintain a high trust value. Despite this issue, Cloud Assure made a strong contribution. It showed that emails leak in the context of their social network. Whispers is an extension of the Classifier block from Cloud Assure.

Originally, Cloud Assure was designed to be restrictive, such that instead of Cloud Assure's *recommendation* to send the email or not, Cloud Assure would drop the email if it determined the email would leak. However, this design decision demonstrated that a restrictive system requires emails to leak in order to build a history of decisions for the MDP; this is insufficient. Cloud Assure punted this issue by describing a training phase where the installing organization feeds past email with known leaks to provide a history for the MDP. How big this history needs to be is left to future work. Whispers started

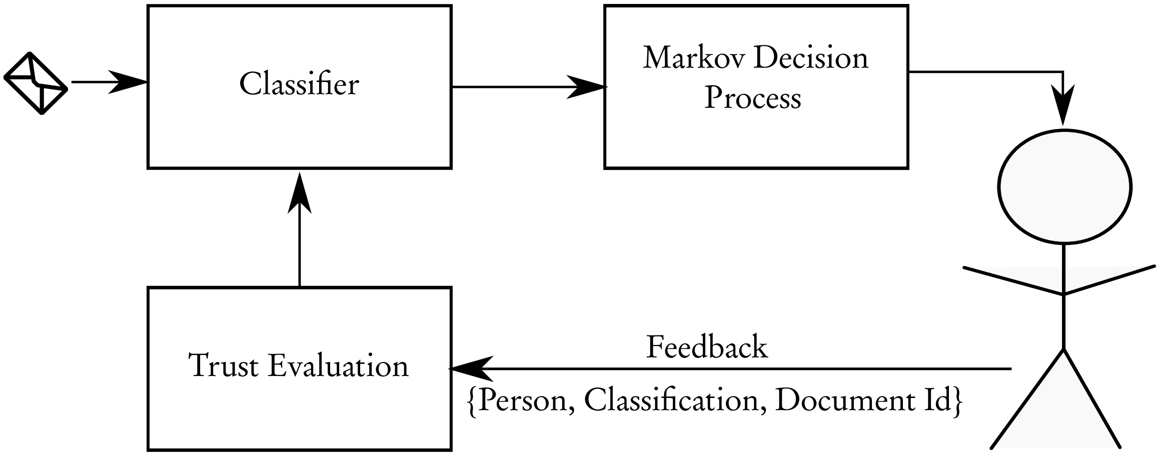

Figure 2.1: Simplified Block Diagram of Cloud Assure

as the classifier block of Cloud Assure. Cloud Assure needed a mechanism to classify documents into a sensitivity level. This level was used by the MDP as a leakage tolerance. The MDP process worked on a graph of the organization. For each email, it constructed the graph where nodes are users, and edges are past emails between two people. Near the recipients, the process adds a simulated user with an artificially high reward value. The MDP then iterates over the graph to see if the email reaches the simulated user. If the MDP reaches the simulated user it recommends the email not be sent since it believes the email is a leak.

This is a powerful idea. Cloud Assure demonstrated that the social network provides an extra dimension to data leakage, and that simulated leaks are an effective tool for detecting possible leakage paths. This allows us to define a social group (Definition 2.2).

Definition 2.2 *Social Group := A social group is the group of recipients corresponding to the emails clustered by the classification algorithm.*

Afterwards however, it became clear that the classifier was too constricting. Cloud Assure's MDP needs a numerical metric that reflects the *content* of the email. Cloud Assure uses Bayesian classification to cluster emails into sensitivity levels. This creates sufficient input for the MDP, but it lacked the ability to group emails of similar content. For example, within an organization, emails about engineering designs, and emails about financial reports could both be highly sensitive. The Bayesian classifier groups these two together, however these two subjects are quite different. Intuitively it follows that this content has different leakage potential, hence differing risks. Furthermore, Bayesian classification requires the groups to be known ahead of time, and installers must provide sufficient training data for each group. Email is simply too dynamic for this to be maintainable. With the motivation to build a risk metric established, Whispers needed something that could *discover* the classification groups in an unsupervised way. For this purpose, Whispers identified topic models Latent Dirichlet Allocation (LDA) and its big brother Correlated Topic Models (CTM) [9], [10]. Topic Models is a machine learning technique to uncover the latent topics within a set of documents [9]. Topic Models provides the unsupervised dynamic discovery of topics needed for a maintainable leakage system.

2.2 Topic Models

Whispers extracts underlying "topics" or groups of co-occurring words and tries to correlate communication of those topics with groups of authors and recipients. As mentioned earlier, simply classifying the documents is insufficient. Cloud Assure established that documents must be viewed through the lens of the social network. Whispers achieves this by applying the estimated topics against the set of authors and recipients to estimate leakage. This set of authors and recipients forms a *social group* defined in Definition 2.2. Lastly, since Whispers intends to be an off-line mechanism, unsupervised methods are preferred to supervised ones. At a high level, if a communique is outside of Whispers' expectation of the intended group, Whispers flags such document as a leak (Definition 2.1).

For the results to be meaningful, the topics must be dissimilar hence a reliable distance metric is essential to sufficiently separate the document clusters. The initial application of Topic Models for this work resulted in less than stellar results. The essential distance metric, the vocabulary was insufficiently culled of vapid words. To improve the identification of topics we leverage term frequency-inverse document frequency (tf-idf) in Section 2.2.4 to generate a vocabulary free of nonessential words in an unsupervised way.

2.2.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a machine learning technique which mathematically describes the human process of writing a document [11]. For example, a document is composed of a collection of words intended to persuade or inform the reader. LDA supposes that the collection of words are related by some list of common themes or topics [9]. Specifically, LDA supposes documents are written in the following fashion:

1. The author has a finite set of lists. Each list contains a vocabulary for a given topic.
2. The author chooses to start writing a document.
3. The author rolls a die to choose a topic list.
4. The author rolls the die again to choose a word from the selected topic list.
5. The author writes down the word.
6. The author continues rolling the die until the document is of sufficient length.
7. Once the document is complete, the author destroys the topic lists.

From the algorithm's perspective this process results in documents that look like Figure 2.2. Each color represents a topic list from which the author randomly selected a word. The histogram in Figure 2.2 shows the estimated topic proportions within the document.

Most of the literature on Topic Models uses a graphical algebra to represent the random variables. This method is called Plate Notation [12]. In plate notation:

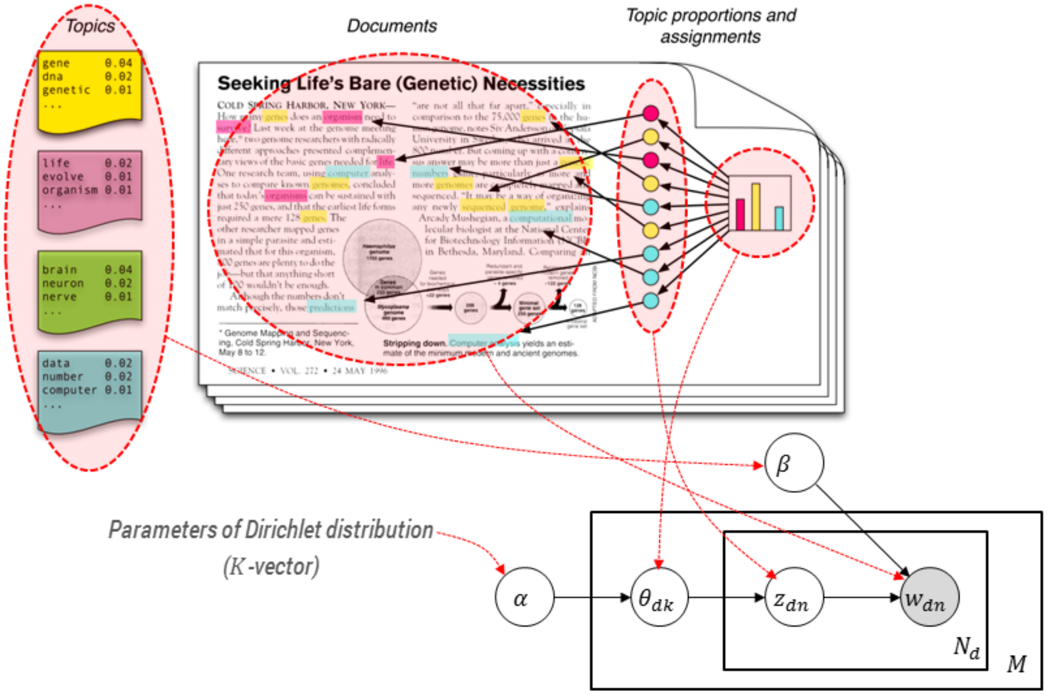

Figure 2.2: LDA's Generative View of a Document.

1. Rectangles, or "Plates" denote replication.
2. Unshaded circles denote unobserved random variables.
3. Shaded circles denote observed random variables.
4. Arrows denote conditional dependencies.

Figure 2.2 shows an example document and the parameters relating to the LDA Plates.

1. α is the parameter of the Dirichlet prior probability distribution on the per-document topic distribution.
2. β is the parameter of the Dirichlet prior probability distribution on the per-topic word distribution.
3. θ_{dk} is the topic distribution for document d for topic k.

4. z_{dn} is the topic for the nth word in document d.

5. w_{dn} is the nth word in document d.

6. N_d is the set of words in the document.

7. M is the set of documents in the corpus.

LDA has 3 inputs, the α and β values to set the Dirichlet distribution, and a vocabulary of words. The Dirichlet is used to estimate the distribution of word choices the author makes when “generating”, i.e., writing a document. As the author chooses words his choices are not uniform. Each topic in fact has a different distribution over its set of words. The Dirichlet provides a smooth distribution of topics, i.e., a distribution of distributions over words. As we'll see in Section 2.2.3 however, LDA is insufficient for Whispers' goals and restrictions on the distribution of words and how those words are chosen in the generative process will be relaxed.

LDA runs an estimation phase to fit the topic distribution, θ, for each document. The resulting distribution then uses an inference phase to infer the content of a yet unseen document according to the overall estimated topic proportions. LDA provides the basis for unsupervised classification. LDA is superior to Cloud Assure's naïve Bayesian classifier in that it classifies documents according to their content and allows new topics to emerge. Now financial data will likely be a separate topic from engineering data. This allows the leakage system to estimate leakage based on content rather than on the overly broad sensitivity level. LDA has become a fundamental tool for topic analysis [13]. However progress has generated two further lines of research that can improve Whispers' performance namely Topic Chains which introduced a time dimension to LDA, and Correlated Topic Models (CTM) which provides an extra dependence between topic distributions which allows Whispers to better estimate documents with varying proportions of differing topics, i.e., not only topics that are similar, but also link together documents that share a common *proportion* of common topics.

2.2.2 Topic Chains

Topic Chains is not a classification technique itself, but an extension of LDA to provide a time dimension [9], [14]. This technique adds additional context to documents allowing LDA to be more effective with shorter documents.

Kim targeted news articles to show how perspectives, and opinions change over time [14]. To achieve this, Kim segments a year's worth of news articles into two week increments, and fits an LDA distribution against each time slice. Using a similarity metric, Kim compares each time slice with its immediately previous time slice to assess which topics continue into the current time slice, which died off, and which topics emerged from the current time slice. With each time slice linked to the previous related time slice a chain forms for each topic. The chains show topic trends over time and Kim defines four special transition cases in Definitions 2.3, 2.4, and 2.5, 2.6.

Definition 2.3 *Topic := a topic is a major subject discussed in the corpus.*
Examples of topics are "winter Olympics", "health care reform", and "the stock market"[14].

Definition 2.4 *Long-Term Topic := if a topic lasts for a long time, we say it is a long-term topic.*
Examples of long term topics are "the stock market", "Afghanistan war", and "education"[14].

Definition 2.5 *Temporary issue := if a topic lasts for a short time, we say it is a temporary issue.*

Examples of temporary issues are “the winter Olympics”, “earthquake in Haiti”, and the “death of Michael Jackson”[14].

Definition 2.6 *Focus Shift := a topic chain exhibits different focuses for each individual topic in the chain.*
An example of a focus shift is “Greece, moratorium” to “Europe recession” in the “economy” long-term topic [14].

Kim further showed how topic chains can adjust the granularity of the extracted topics [14]. By tuning the similarity metric, more broad or more narrow categories emerge from the data. On a liberal setting, topic chains identify large categories like the sections of a newspaper, whereas tighter settings identify individual people, and places. Whispers leverages the topic chain method to fit time slices of emails as it provides a flexible tuning value for fitting topic similarity. Whispers is primarily focused on identifying risk of *topics*, and *long-term topics* as opposed to identifying when focus shift occurs within an organization. As a result Whispers uses a tighter similarity setting such that focus shift tends to show up as a new topic rather than as a shift in an existing one. As we in Section 4.6 a new topic emerges from a larger one. The new topic demonstrates a clear security violation speaking directly to the impact Whispers' can make.

2.2.3 Correlated Topic Models

LDA's generative processes has a problem however. The θ_{dk} parameter which tracks the topic assigns for words assumes that each word chosen from the topic list is independent. This is not expressive enough for complex documents with a mixture of topics. When an author writes more complex documents each word selection is not independent, but rather part of some influential concept the author wishes to convey. By relaxing this independence the generative process is more expressive in the documents it can convey, allowing words to be part of multiple topics. With respect to data leakage, the word *security* might relate to information assurance as much as it relates to a financial security. LDA's independence property prevents a word from contributing to multiple topics. Correlated Topic Models (CTM) however provides an extra dependence between topic distributions which allows Whispers to cluster a documents with similar content, and within each content cluster further cluster document with a similar *proportion* of similar topics. CTM calls this subgrouping of similar topic proportions, topic correlation. To enhance LDA's inability to estimate topic correlations, CTM uses a different underlying distribution. Rather than the Dirichlet, with its independence property, CTM uses the logistic normal distribution [10].

CTM follows LDA's generative document process in that a document is simply a bag of words. Also like LDA, CTM is sensitive to the provided vocabulary. Initially trivial techniques of removing very common and infrequent words resulted in poor topic separation. These attempts made broad strokes at the vocabulary cutting the 5%, and 10% frequent, and most infrequent words. This overly broad approach did not improve topic separation, which led to a preprocessing step of tf-idf.

2.2.4 Topic Frequency - Inverse Document Frequency

Term frequency-inverse document frequency (tf-idf) is an unsupervised technique for weighting document terms with the intent of filtering out terms which provide no information [15]. While quite an old technique, tf-idf is used extensively in information retrieval systems. In fact, Etsy.com the online handmade marketplace uses tf-idf extensively to filter keyword searches of potential customers [16]. Etsy.com is an interesting application of information retrieval as it demonstrates the importance of words relative to their domain. It's common practice to remove stop words, i.e., words used only for syntax and provide no information. Stop word lists tend to be language specific and for English include words such as a, the, they, etc. However some words independently provide value, yet in the context of their domain provide nothing useful. Etsy.com is a marketplace for people to sell their hand made items. As a result nearly every item on the site is unique. While a consumer may highly value hand made goods, the words “hand made” or “unique” as search terms on the site do not provide any differentiation between items. In a sense, “unique”, within Etsy.com’s domain is a stop word. Etsy.com uses tf-idf to find these vapid words and remove them from the search queries so the customer can find what they want quickly.

Tf-idf works on bag-of-words documents. It counts each word’s frequency relative to its inverse frequency within the corpus (Equation 2.3). The resulting vectors are unbounded positive real values. Smaller values denote lesser importance. Equation 2.1 describes the term frequency (tf) parameter. tf increases proportionally to the frequency of a word in a document. Conversely, Equation 2.2, the inverse document frequency (idf), decreases by the frequency of the word in the corpus [17]. Combining them provides a high per word score for words with high frequency in a few documents, but low frequency overall. Whispers uses this technique after stop-word removal to further clean the vocabulary prior to CTM estimation. The equations capture tf-idf, where

1. t is the vector of boolean terms. The term is 1 if the term occurs in document d, 0 otherwise.

2. d is the vector of terms within a document.

3. N is the total number of documents in the corpus.

4. $|\{d \in D : t \in d\}|$ is the number of documents where the term t appears.

5. $max\{f(w,d) : w \in d\}$ is the maximum frequency of any term in the document. This helps prevent long documents from dominating the term frequency.

$$tf(t,d) = \frac{1}{2} + \frac{\frac{1}{2} \cdot f(t,d)}{max\{f(w,d) : w \in d\}} \tag{2.1}$$

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|} \tag{2.2}$$

$$tf\_idf(t,d,D) = tf(t,d) \cdot idf(t,D) \tag{2.3}$$

While tf-idf has proved itself to be quite useful, there is little linguistic basis for it. Recently there is an effort to backfill grammatical reasoning to this quite useful technique [18].

2.3 Similar Works

In "Analyzing Group E-mail Exchange to detect Data Leakage", Zilberman et al. develops a plug-in for Microsoft Outlook^{TM} to provide users real-time recommendations on the intended recipients of an email [19]. Users of this system would author an email and add users to the "to" field. The plug-in then analyzes the content of the email and correlates that content with the recipients' social group. If a person in the "to" field is not the social group, the plug-in recommends that the email not be send to the outside user. Zilberman et al.'s primary focus was identifying "...the wrong Bob" where an author intends to send an email to their friend "Bob Saget", but instead accidentally sends it to their boss, "Bob Parsons".

Similar to Whispers, Zilberman et al. uses people clustered into groups according to common content. However Zilberman et al. uses *k*-means to group users into social groups instead of Whispers' topic model method. Whispers attacks a separate problem from "...the wrong Bob". While identifying leaks at the individual level is important for overall leakage protection, Whispers holds are more macroscopic view of the entire organization to identify chains and the potential of leaking data from those chains. To evaluate Zilberman et al.'s method, they simulate leaks by injecting users known to be outside the social group to emails and measure their false positive rate. Whispers extends this method from an on demand per email process to a holistic continuous online process. Whispers assesses risk by injecting leaks similar to Zilberman et al., but instead of making a recommendation on the single email, it injects hundreds of simulated leaks and aggregates its accuracy into a risk value for that topic.

2.4 Summary

Whispers is a risk assessment system that determines what data within in organization is truly at risk. Cloud Assure demonstrated that naïve Bayesian is insufficient as a classification technique for data leakage, but provided keen insight that the social network enhances the data leakage picture. Using CTM, Whispers is able to cluster documents of similar content, and further refine those clusters with documents of similar proportions of similar topics. tf-idf provides pithy input to the topic analysis by culling vapid words. Thus forms a fully unsupervised processing chain that classifies against content rather than sensitivity level. The following chapter provides a deeper understanding of the architecture of Whispers.

Chapter 3

DESIGN OF WHISPERS

Whispers equates leakage risk with its ability to find a seeded leak. More specifically, risk is synonymous to the false negative rate, i.e., the rate at which Whispers is unable to determine a leak occurred (Definition 3.1). Whispers continuously summarizes incoming emails within an organization. Since data trends over time, it follows that the potential leakage of that data also changes. With a continuous measurement system in place one may better assess which intellectual assets are at risk, and potentially how well the company is protecting them.

Definition 3.1 *False Negative* $:=$ *Declaring an email did not leak, when it did.*

Whispers provides a per topic risk metric to provide the estimated likelihood the organization can detect topic leakage. Such a metric allows businesses to focus their effort on the highly sensitive topics with a corresponding high likelihood of undetectable leaks. Whispers simulates leaks of various topics, then measures the rate at which it detects the simulated leak. The false negative rate of this process is the risk metric. If Whispers injects a leak, and then cannot find it, that speaks to the topic's volatility or ubiquity within the organization. If such a topic is also intellectually valuable to the organization, the organization can focus additional security measures on the topic to improve its security prior to an incident.

3.1 Architecture of Whispers

Leakage risk is inversely proportional to the likelihood of identifying simulated leaks. An accurate identification corresponds to low risk, i.e., one identifies that the signal ex-

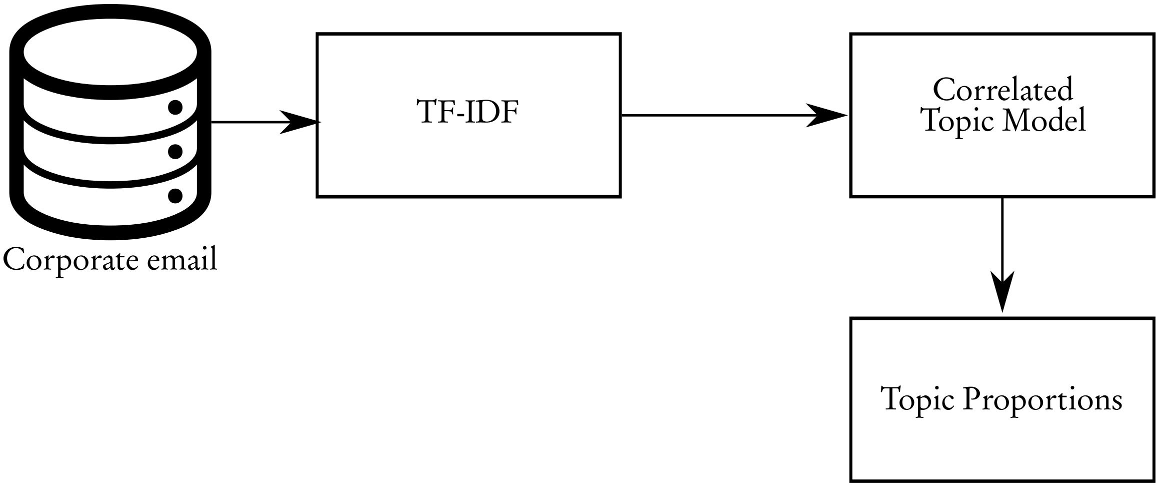

Figure 3.1: Block Diagram of Whispers Stage-1 As It Fits the Topic Proportions to the Corpus

ists, and that it leaks despite the inundation of irrelevant chaff. Conversely, an inaccurate identification corresponds to high data risk. There are at least two forms of inaccurate identification. First, if one inaccurately identifies a signal that does not exist, the organization impedes valid communication by mistakenly thinking data is leaking. Secondly, if one accurately identifies a signal, but misclassified the data, the organization is wasting resources protecting irrelevant information.

Whispers assesses risk in two stages. The first stage (Figure 3.1) runs tf-idf, and fits a CTM to the corpus. CTM outputs topic proportions which are used by the Classifier block in stage-2 (Figure 3.2).

Figure 3.2 shows the five components of Whispers' computation chain. The solid lines represent the processing path for email. The dotted lines represent the simulation side channel which feeds leaks to the Validator. This side channel allows the Validator to "cheat", by knowing the actual leak to compare against the computed answer from the Evaluator.

1. Email database.

2. Classification engine.

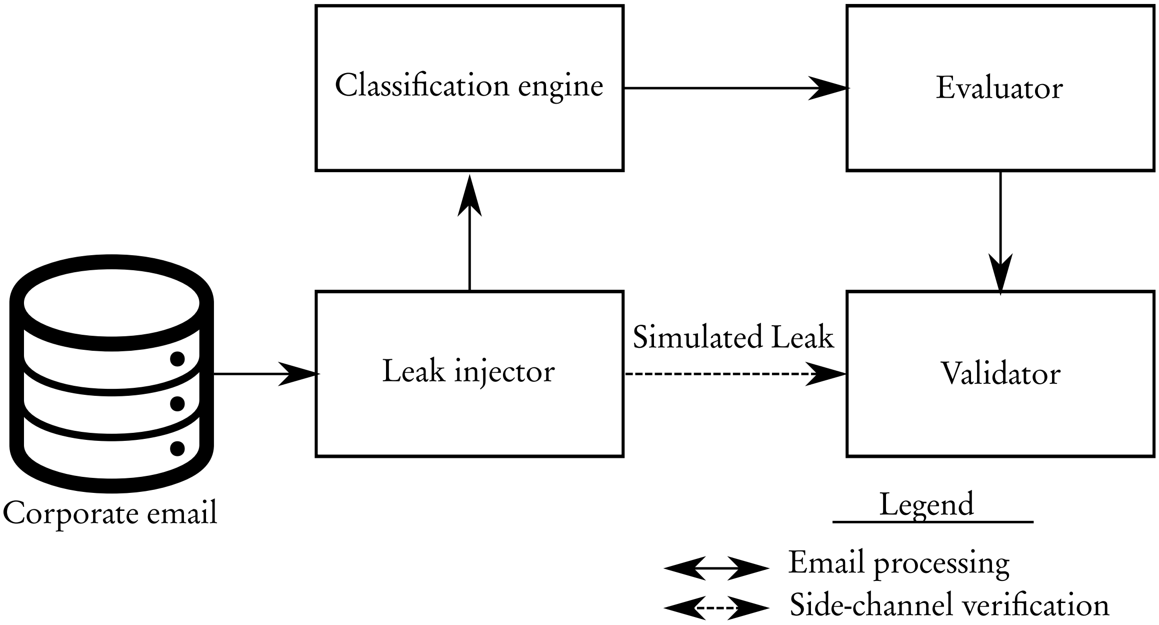

Figure 3.2: Block Diagram of Whispers Application for Continuous Measurement of Data Leakage Risk

3. Leak injector.

4. Evaluation engine.

5. Validator.

Next, we describe the components.

3.1.1 Email Database

The email database is a collection of all the email at the company. It contains everything from trivial e-mail exchanges to valuable intellectual property of the organization. While the information is valuable, it is too costly and error prone to manually classify all emails in the organization looking for potentially leaking information. Instead Whispers attaches to the primary email server and runs an unsupervised classification technique to extract the underlying topics within the organization's email.

3.1.2 Classification Engine

The classification engine is divided into two stages, an estimation component and an inference component. The estimation component summarizes all email up to the current date and correlates co-occurring words as topics (see Section 2.2.3). This is a computationally heavy process. Once estimation is complete the inference engine can use the estimated data to infer topics within a yet unseen email. This allows Whispers to infer the content of an email based on a corpus of past data. This is also an unsupervised process. Once the email is inferred, its topics are sent to the Evaluator to compare its topic content with the historical social groups of people.

3.1.3 Leak Injector

The leak injector simulates data leakage by adding new people to the recipient field of a withheld email. By injecting leaks, Whispers has the benefit of knowing who is leaking without needing to wait for an actual leakage incident. Being unsupervised, servers can be set aside to process reams of email to estimate the risk of topics within the organization. Also a benefit of being unsupervised, Whispers can evaluate emerging topics, i.e., topics of conversation yet unknown to the security officers of an organization. Whispers evaluates risk of these new topics as they emerge within the corpus.

The leak injector randomly selects users from the organization's roster, and adds them to the *To* field of the email. This simulates two possible forms of leakage. The first is that a malicious user sends information to a remote cohort. The second is that an innocent user unintentionally sends information to an inappropriate person. While the second form is leakage is more likely, both forms have the same potential result, loss of intellectual property.

Once the leak injector adds the name to the email, it forwards the email the organization's leak prevention system, in this case a classification engine. Additionally, via a side channel, it sends the leaked name to the Validator for closed loop validation of the final leakage result.

3.1.4 Evaluator

The Evaluator consumes the email with its accompanying classification and correlates the recipients with a social group. Using the topic distributions from the estimation phase, the Evaluator assembles a graph describing the social group for each topic. Recall that a leak is defined as sending an email containing a topic outside of the topic's target social group. The Evaluator then compares the *To* field of the email with the social groups of its content. If any user is outside the topic group, the Evaluator claims a leak.

If this were a real leak, and not a simulated process to evaluate risk, security officers have a decision point to alert the user, or drop the email silently. If the email is dropped this incurs a problem that the social groups of a topic cannot grow, i.e., a topic group remains static. Imagine a new employee starting at an organization. With such a strict security profile this employee could never send an email to anyone since such a new user is not part of any social group. Rules have to be designed to account for such cases. This issue is left for future work. Once this processing is complete, the Evaluator sends its decisions to the Validator for final scoring.

3.1.5 Validator

The Validator acts as a validation step and compares the output of the evaluation engine with the known injected user from the previous module. The Validator outputs the risk metric as a percentage from Equation 4.1.

3.2 Summary

Whispers is a general risk assessment system to analyze email and provide a per topic risk value. The following chapter evaluates Whispers' performance on a real data set.

Chapter 4

RESULTS

Assessing data leakage as a postmortem or audit type activity requires data to be leaked. Whispers' method is different. Whispers uses the organization's real email to feed an internal quality tracking system. It continuously simulates a leaked email, and sends it through the organization's leakage containment system. Whispers tallies the emails which are either flagged or passed along, all while knowing which emails are sensitive, and of those sensitive emails which emails contain simulated leaks. The false-negative rate of data leakage, i.e., the rate at which the system incorrectly flags an email as leaking correlates to the risk of that particular topic. Simulated leaks correlate to real risk, since the system proves itself unreliable with simulated leaks on real data, it follows it will be at least as unreliable against a malicious adversary.

4.1 Enron Email Corpus

Whispers is a risk assessment system intended to process email hence it needs an email corpus. Whispers assesses risk by simulating data leakage and measuring the accuracy of the simulated leak. A corpus of sufficient size is required to simulate the noise of real world email communication. Whispers uses the Enron email data set [6] for this purpose. This dataset contains 255,636 emails concerning both personal and business related topics. Excluding automated IT messages and infrequent senders, the majority of email is from 150 separate individuals during 1999 and 2001 [20]. Information assurance aside, the dataset is used extensively in several research areas ranging from business ethics [21], [22], to computer science topics such as spam detection [23], social networks [24] and artificial intelligence [25], as well as psychology papers attempting understanding how humans communicate [26], [27].

Whispers focus is to identify the topics users are communicating, and determine its ability to identify incorrect recipients on the email. Enron's wide topic content is perfect for this application. Whispers looks at the first seven quarters of the data as shown in Figure 4.1. Due to computation limits with the implementation, Whispers requires an exorbitant amount of time to process the successive quarters. The corpus provides an additional two years of documents, but Whispers' serial implementation is overwhelmed by the size of subsequent quarters. While a more parallel implementation will extend Whispers' reach, the current implementation is restricted to this subset. Section 5.1.2 provides additional details on this issue.

4.2 Database Versus Flat Files

Databases allow Whispers to better organize of the relationships between topic chains, the source documents, and the source authors than flat files. The database depicted in Figure 3.2 represents an organization's email database such as Microsoft Exchange, Postfix or some similar mail server. A database over flat files gives Whispers' a great deal of control both as a functional product, but also as a research project. The database features make it easy to mirror, duplicate, and log data manipulations while still maintaining data consistency. Whispers' started as a SQLite [28] backed application until the write performance on SQLite became a severe bottleneck for the LDA algorithm. SQLite is a single file database which makes backup and management of the database easy. The "server" runs as a library inside the user's application. While the simplicity is quite enticing, SQLite has a critical flaw regarding data processing: SQLite supports only one concurrent writer. Whispers is naively parallel. Each time slice is processed independently hence each quarter

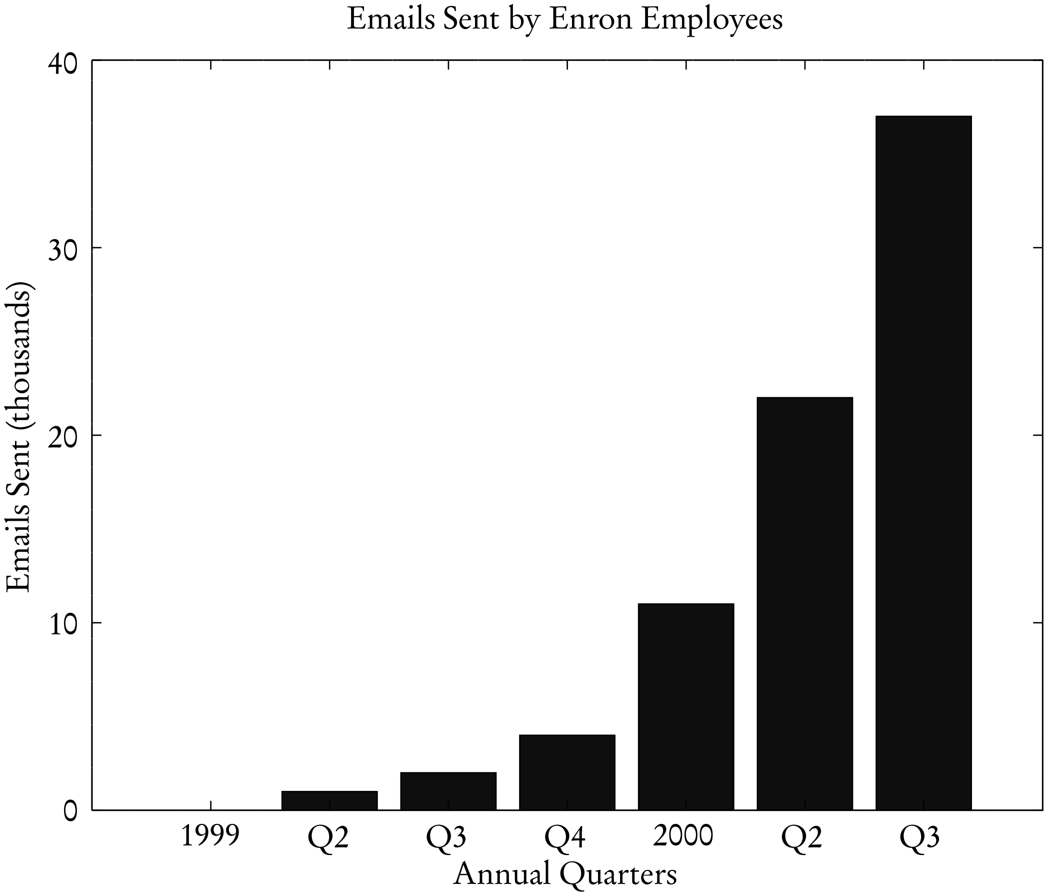

Figure 4.1: Distribution of Email by Annual Quarters.

is pounding the database. The processing time grew beyond a week for a single year of email.

Whispers had the option to switch from a database, or as the stock LDA implementation prefers, to flat files. The value of the database is simply too great to drop, and Whispers upgraded to PostgreSQL [29]. PostgreSQL has an advanced Full Text Search implementation built directly into the database query engine. Extending this is MADlib [30], is a set of python stored procedures backed by fast C++ libraries to perform topic modeling and other data analysis techniques apropos to Whispers. With MADlib in place, Whispers started processing email, yet the topics it extracted were quite poor (Table 4.1).

Table 4.1: Poor Topics Due to Insufficient Vocabulary Pruning.

wednesday	its	john	corp	company
take	help	find	power	good
had	fax	good	john	business
mark	were	friday	very	number
monday	group	work	below	energy

4.3 Improving Topics with Term Frequency-Inverse Document Frequency

A critical step in extracting quality topics from CTM is a quality vocabulary. Since Whispers' is intended to function unsupervised, it needed an unsupervised method to prune the vocabulary.

Processing starts with cleaning the text with tf-idf. Initial experiments did not perform this cleaning step, and the returned topics were less than useless. The topic model algorithms work by leveraging co-occurrence of words. The topic model algorithm must be provided a vocabulary, a list of words which are relevant and useful for the corpus at hand. If words which convey no information are left in the vocabulary, the topic model is left to assume the co-occurrence of useless words are somehow useful. Some of the topics generated by this situation are shown in Table 4.1. On the other hand, Table 4.2 shows the quality topics which Whispers can extract from the corpus. MADlib is truly a fantastic library, however it lacked an integrated tf-idf implementation, and the initially extracted topics needed some serious cleaning. A few supervised attempts were made, but this was strongly against the design goals of Whispers. This process entailed manually tuning the Dirichlet α and β to find a reasonable set of topics. This process resulted in topics similar to Table 4.1, and the cluttered vocabulary prevented the LDA from good separation and the model was unable to identify any more that five topics. Whispers needed an integrated tf-idf to clean the vocabulary. Consequently, MADlib was dropped in favor of an open

Table 4.2: Higher Quality Topics Due to Tf-Idf Vocabulary Pruning

legal	energy	stock	attachment	contract
power	service	market	copy	transact
product	power	trade	file	indian
trade	market	price	trade	counsel
meet	group	buy	email	deal

source project by Colorado Reed [31]. Reed provides a SciPy [32], Django [33] backed tool. This tool provided an excellent reference for integrating a tf-idf engine into the topic modeling processing chain. The result is sufficiently separated topics.

4.4 Chaining Topics

Once each quarter is processed, Whispers links each topic to the previous quarter using cosine similarity [34]. Each topic is a list of words sorted in order of words most characteristic of that topic. This list of words forms a vector, and each topic is compared to the previous quarter's topic vectors to find the most similar topics. This chains each topic together allowing Whispers to trace the topic over time. Even as the risk of the topic changes, the words which build up that topic contribute to the similarity between quarters. If the cosine similarity is not above a particular threshold, Whispers assumes the topic is new.

Whispers uses a similarity value of 12%. We tuned this value experimentally by attempting to maximize topic consistency, while minimizing the number of resultant topic chains. Values smaller that 8% created too many separate chains. Conversely, 15% included most topics and resulted in many topics matching quarter to quarter. 12% allows Whispers to not only trace topics over time, but also allows Whispers to drop topic chains when newer evidence demonstrates itself. Topics that are free to emerge as conversations

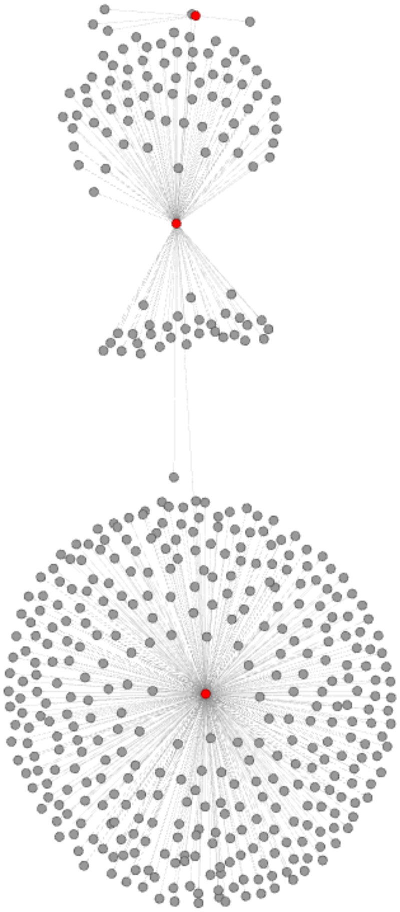

Figure 4.2: Topic Chain Showing Quarters Jan 1999 - July 1999. Center Nodes Represent The Quarter Anchors. Descendent Nodes Represent Documents Part Of That Topic In That Quarter Time Slice.

increase the co-occurrence probabilities estimated by CTM. Likewise topics with diminishing probability will fall away from the topic analysis. Each topic, clusters the recipients into social groups. Whispers has now established the lineage of each topic through the corpus. Figure 4.2 shows the linking of three quarters of data using this process.

4.5 Risk Estimation

Whispers' processing chain is nearly complete. The raw emails have been cleaned by tf-idf and, the resultant vocabulary is processed by CTM to form topics. The topics are

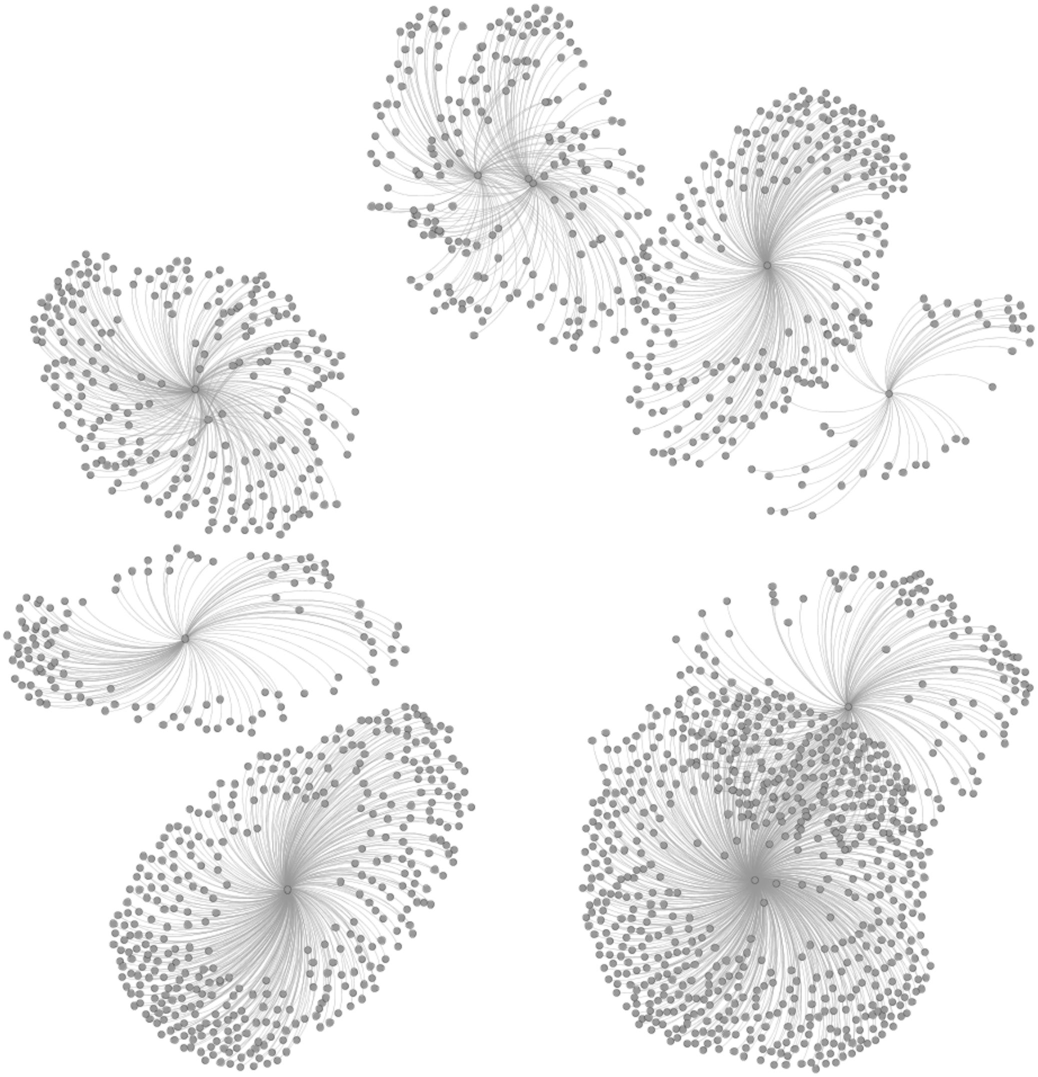

Figure 4.3: Document Graph for July 1999. Center Node Represent Topic Anchors. Descendent Nodes Represent Documents. Notice the Two Clusters in the Bottom-Right Overlap. This Overlap Demonstrates The Two Topics Share a Number of Documents Which is a Direct Effect of the Topic Proportion Feature of Correlated Topic Models (CTM).

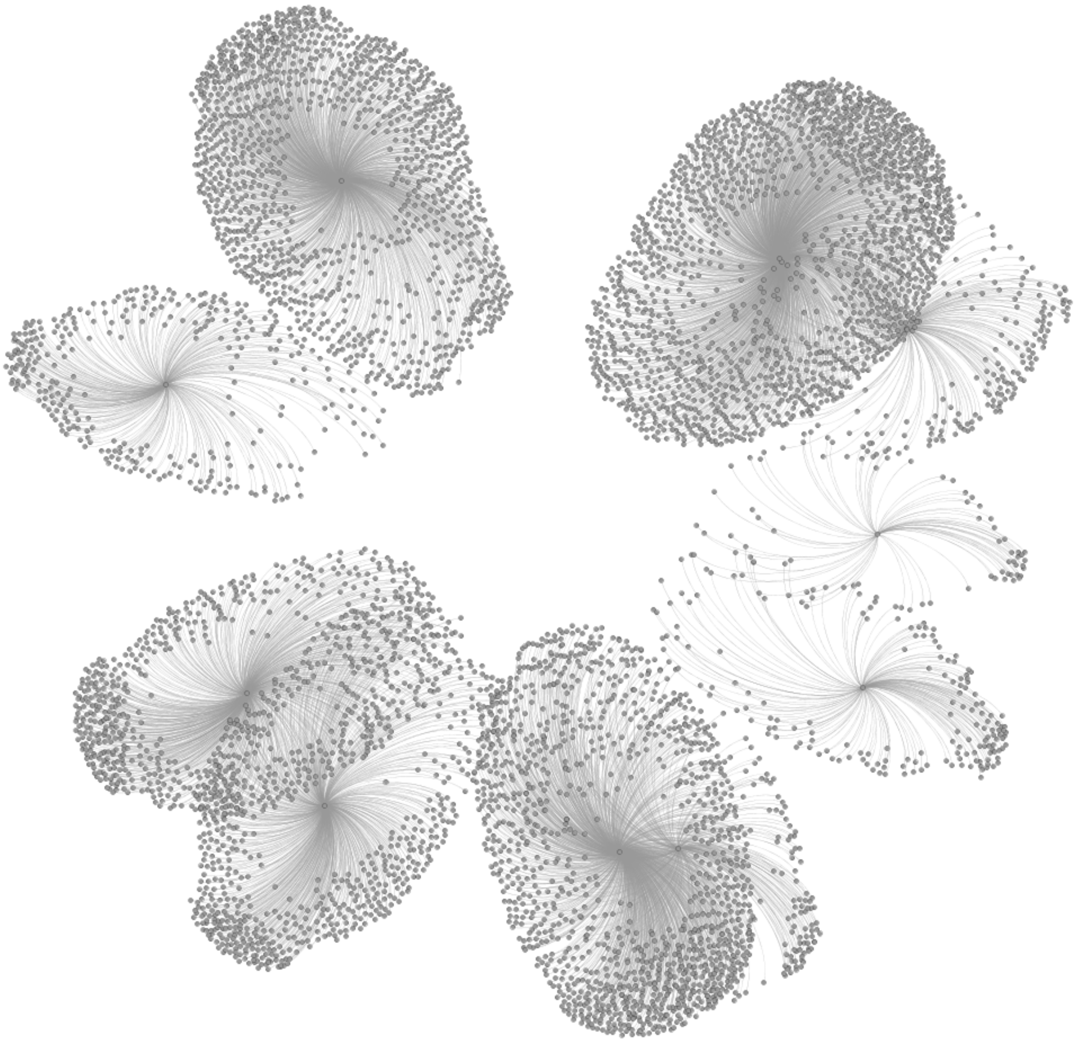

Figure 4.4: Topic Graph of Users for July 1999. Center Nodes Represent Topic Anchors. Descendent Nodes Represent Users. Notice The Shared Users Between the Top Right Two Clusters; This Represents Higher Risk Than The Fairly Isolated Cluster on the Middle Left.

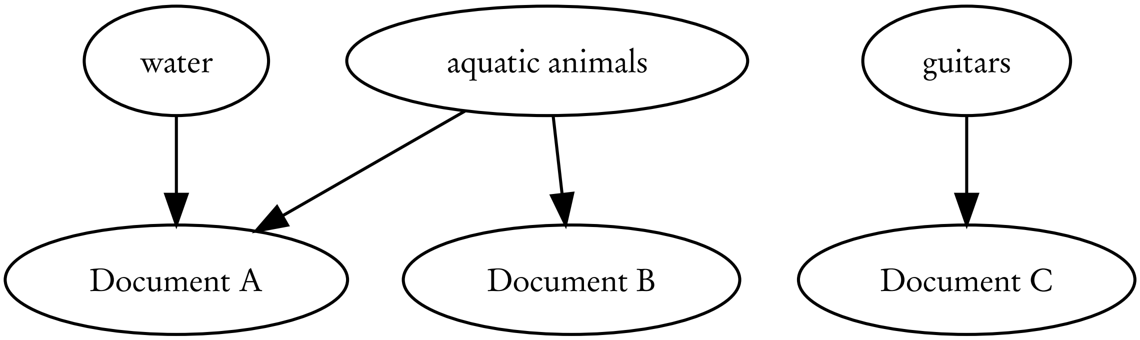

Figure 4.5: Example Document Graph.

chained together using cosine similarity. Whispers is now left with the task of injecting leaks and measuring its ability to discern the leaked recipient.

Whispers first assembles two directed graphs. The first graph is a document graph built from the CTM data by following the likelihood estimates generated by CTM. For example, take three documents. Document A contains words about fish, and oceans. Document B contains words about fish and turtles. Document C contains words about digital-signal-processor guitars. One builds a vocabulary of these documents using the tf-idf. The documents, and the vocabulary are given to the CTM algorithm. We then ask CTM to generate 3 topics. CTM returns 3 topics: 1 about aquatic animals, 1 topics about water, and 1 topic about guitars. CTM also returns an associated topic proportion for each document. Document A is 50% about aquatic animals, and 50% water. Document B is 100% about aquatic animals. Document C is 100% about guitars. The document graph then looks like Figure 4.5.

Figure 4.3 givens a document graph from July 1999. The same CTM step that outputs topic lists, also outputs topic distributions for each email. These distributions allow Whispers to construct a graph connecting the topic as a root node, with directed edges to each related document. Note that since documents are a mixture of topics, the in-degree of a document node is not always one. With the document graph, Whispers builds a user graph by following the recipients of the emails. The user graph forms the social group

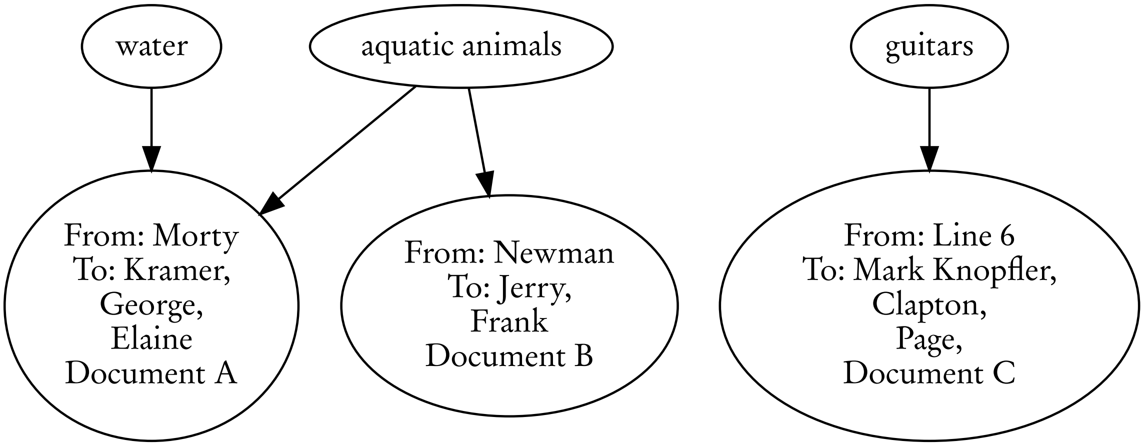

Figure 4.6: Example Document Graph With Users Added.

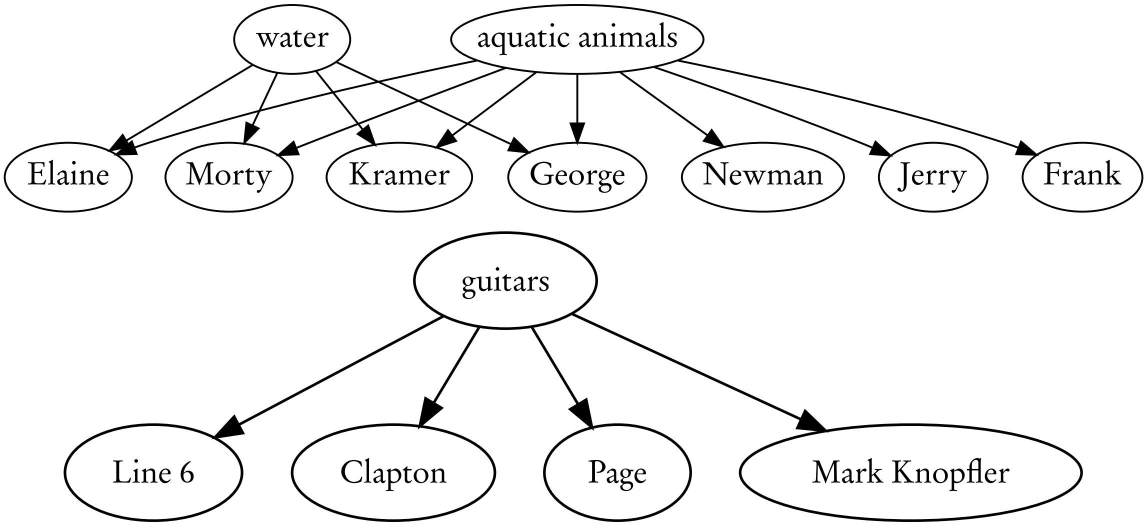

Figure 4.7: Example Document Graph With Users Added.

for the topics. Whispers follows the edges of the document graph and connects all the recipients within a topic group. The result is a graph rooted by topics with users as the descendants related to that topic. Recall our 3 document example about fish, and guitars from Figure 4.5. Imagine the document were sent to individual people as in Figure 4.6.

To build the topic group for each topic, Whispers collects all the users in document. Figure 4.7 gives the user graph for our 3 document example.

Now a given email will have a given topic distribution. Within the user graph, Whispers can start at the root node for the given topic. All descendants from that node form that email's social group. Whispers considers an email a leak if a recipient is outside this social group. Figure 4.4 illustrates a user graph for July 1999.

Algorithm 4.1 High Level Algorithm For Simulating Data Leakage

```
$M \leftarrow get\_message()$
$T \leftarrow M.topic()$
$G \leftarrow T.descendants()$
$R \leftarrow random\_user()$
$L \leftarrow M.recipients.append(R)$
**return** $((L \notin G) == R)$
```

Whispers uses Algorithm 4.1 to simulate a leak. This algorithm returns True if Whispers identifies the person outside of the social group, and False otherwise. For each email in the document graph, Whispers simulates 500 leaks. The final risk metric for the given topic is the average rate Whispers incorrectly identifies the injected user given by Equation 4.1. This results in Figure 4.8.

$$Risk = \frac{\text{False Negatives}}{\text{Total Emails}} \tag{4.1}$$

4.6 Interpreting Whispers Risk

Figure 4.8 identifies a number of interesting attributes about risk. CTM looked for 10 topics within the corpus in each quarter however, the final tally found 18 (Tables 4.3 - 4.5). The cosine similarity could not match all topics to each quarter and new topics emerge out of the data. Of these emerging topics the most interesting from a security perspective are Topics 16 and 17 shown in Figure 4.9 and Table 4.6.

Topic 16 and 17 emerge on the tail end of Whispers data range. Figure 4.9 shows the isolated risk of these two topic chains which discuss emails, passwords, and access.

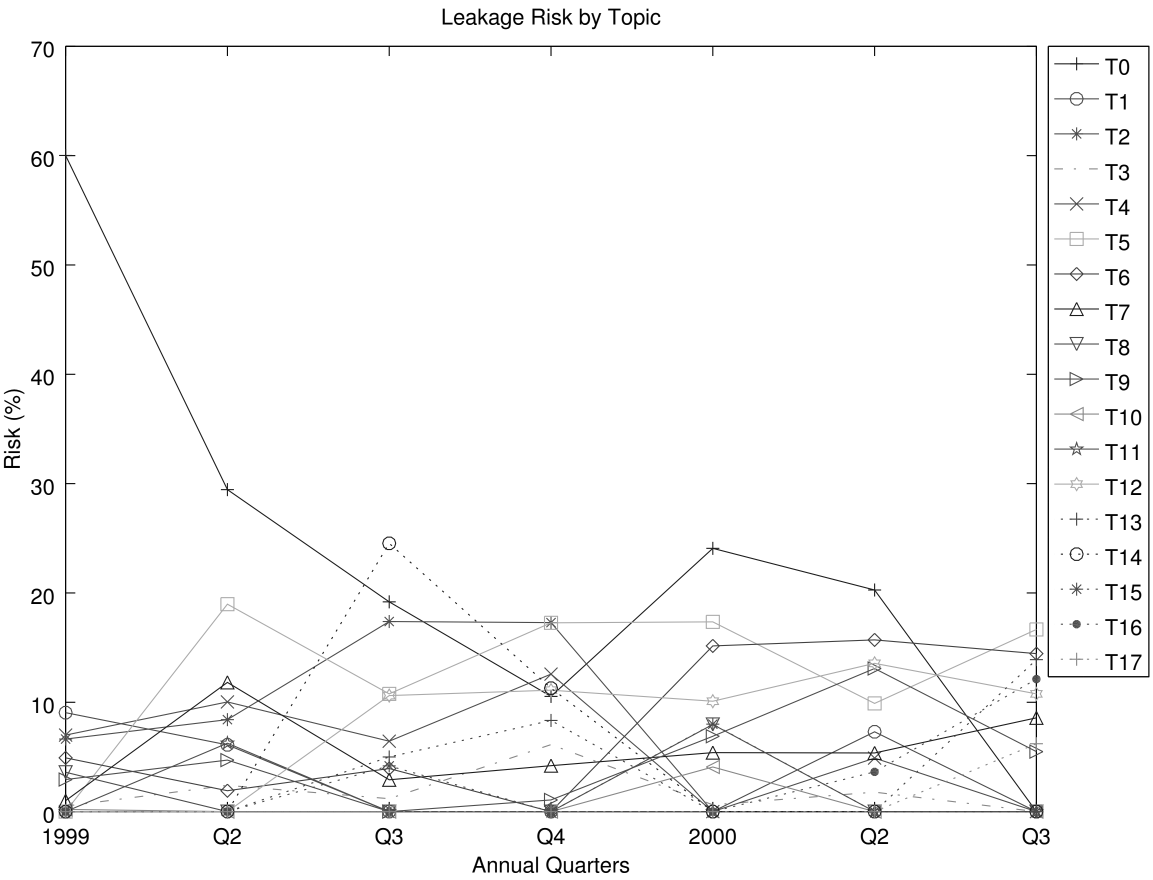

Figure 4.8: Topic Risk Over Time.

These keywords appear to be red flags with respect to information assurance. In fact this topic group includes 3640 emails in just two quarters. Certainly this topic group requires deeper analysis. Whispers tracks which documents contributed to each topic, and allows one to trace the most influential documents. By tracing the document graph we can read some of the influential emails. Figures 4.10 and 4.11 show clear security policy violations. Figure 4.10 shows an IT representative sending a password to a shared resource in cleartext. Figure 4.10 shows the same person losing his password to an internal database application, SAP. This sample within the topic chain could be a possible social-engineering attack attempt by a malicious agent posing as Eric Bass. Whatever the actual motivation here, Whispers identified a possible threat, and a clear policy violation.

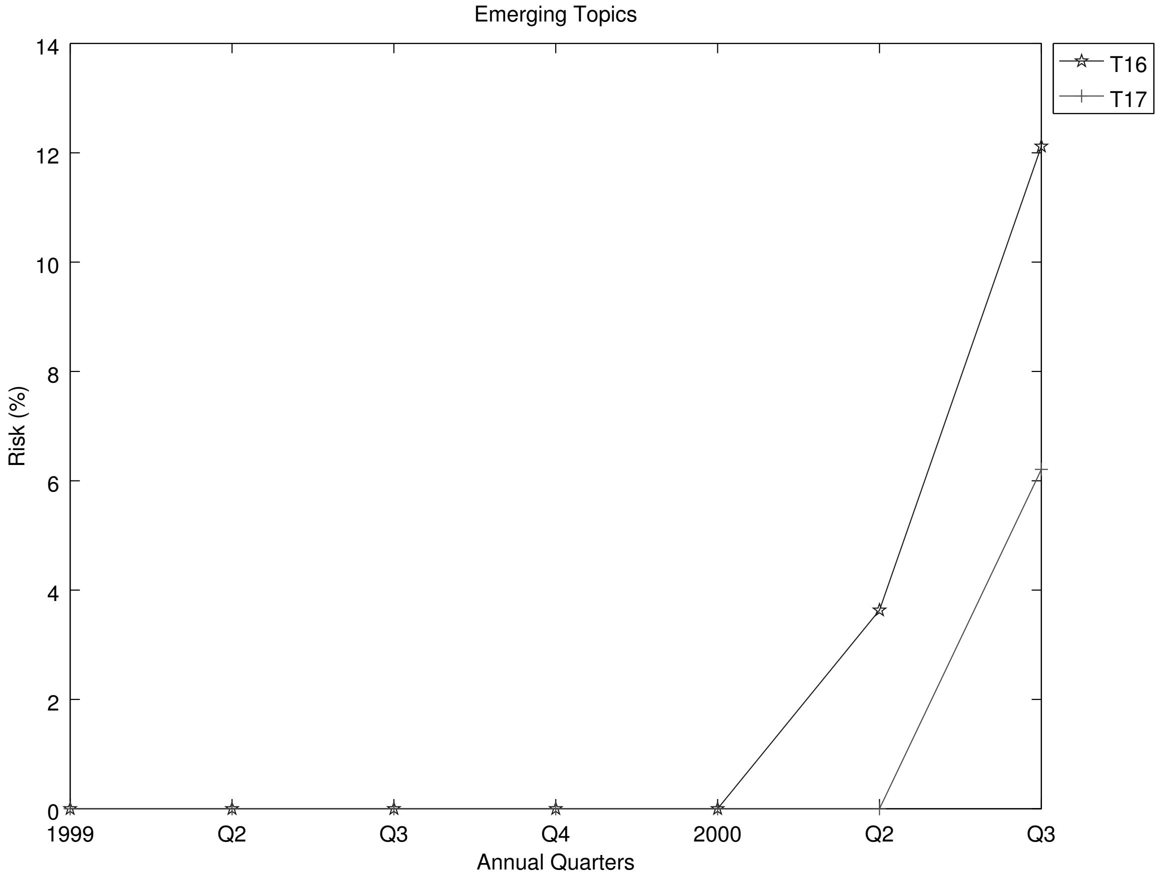

Figure 4.9: Topics 16 and 17 Emerged From the Tail of the Data Range Examined by Whispers. They Represent Whispers Ability to Identify New Topics in a Corpus.

Table 4.3: Topics Tracked by Whispers (0 - 5)

Topic-0	Topic-1	Topic-2	Topic-3	Topic-4	Topic-5
legal	contract	attachment	stock	energy	subject
power	transact	draf	market	service	forward
product	indian	file	trade	power	reply
trade	counsel	trade	price	market	time
meet	deal	email	buy	group	send

```
Forwarded by Eric Bass/HOU/ECT

Enron North America Corp.

From: Tracey Irvin

To: Eric Bass/HOU/ECT@ECT

Subject: Re: Statistical Database

Eric,

Your password is Packers and it is case-sensitive. Let me know if you

have any problems.

Enron North America Corp.

From: Eric Bass

To: Tracey Irvin/HOU/ECT@ECT

Subject: Statistical Database

I forgot my password to the statistical database. Who should I contact

to find this info.?

Thanks,

Eric

3-0977
```

Figure 4.10: Sample Documents in Topic 16 (1 of 2)

Table 4.4: Topics Tracked by Whispers (6 - 10)

Topic-6	Topic-7	Topic-8	Topic-9	Topic-10
brazilian	gas	risk	london	price
rate	deal	derivativ	equiti	model
transact	compani	manag	deal	option
customs	energi	insur	stock	trade
exchange	contract	transact	buy	call

```
From: Scott Becken

To: Eric Bass

Subject: Re: SAP Logon ID and Password

Your SAP user ID is P00501250.

Your initial password is your birthdate in YYYYMMDD format.

808333

- Forwarded by Scott Becken/Corp/Enron -

From: Eric Bass @ ECT

To: SAP COE/Corp/Enron@Enron

Subject: SAP Logon ID and Password

How do I go about getting my logon id and password? I seem to have

deleted the e-mail that was sent out. Thanks,

Eric Bass

x30977
```

Figure 4.11: Sample Documents in Topic 16 (2 of 2)

Table 4.5: Topics Tracked by Whispers (11 - 15)

Topic-11	Topic-12	Topic-13	Topic-14	Topic-15
buy	time	sara	meet	chairman
feb	work	brent	subject	deal
call	day	mari	will	contract
earn	go	mark	enron	unit
week	year	robert	sara	meter

Table 4.6: Topics Tracked by Whispers (16 - 17)

Topic-16	Topic-17
request	price
list	cost
access	california
password	generat
user	suppli
confidenti	demand

Topic 17 is less malicious and exemplifies Whispers ability for dynamic discovery of new threats. As Figure 4.12 shows, Whispers identified California's Summer 2000 energy crisis. Now we know this resulted in a huge financial for Enron with increased travel into the state for trades, and deal making [21], [22]. From a security perspective this means increased remote VPN access, and remote help desk support. Whispers acts as a predictive tool to suggest IT resource allocation.

```
Forwarded by Carla Hoffman/PDX/ECT

-- Enron Capital & Trade Resources Corp. --

From: "Pergher, Gunther" <Gunther.Pergher@dowjones.com>

To: "Golden, Mark" <Mark.Golden@dowjones.com>, ...

Subject: DJ BIG PICTURE: Wider Econ Risks In California’s Power Woes


DJ BIG PICTURE: Wider Econ Risks In Californiaś Power Woes

By John McAuley

Of DOW JONES NEWSWIRES

NEW YORK (Dow Jones)-Hot weather and a still-robust economy have

intensified electricity demand in the face of drum-tight power supplies

in California, the nation’s most populous - and, in economic terms, most

important - state.

The resultant rolling "brown outs" and the potential for blackouts

in the future could have a noticeable empirical and real impact on

industrial production in California and even in the national statistics.

...
```

Figure 4.12: Emerging Market Data From Topic 17

Chapter 5

CONCLUSIONS & FUTURE WORK

Data leaks. It is infeasible to lock down all communication. In the information business new information emerges continuously. A single leak can cost the company its life, yet no leakage system can block all leaks. One needs to identify the data at risk. Whispers presents a novel application of topic models for assessing data leakage risk within an organization. Section 4.6 demonstrated the experimental results of Whispers, and demonstrates its potential value as a risk assessment tool. Whispers' simulation loop assumed it is the sole component in assessing leakage, however the system can be extended with an organization's existing data leakage detection system, or incorporated within a larger solution such as Cloud Assure. As a component in a larger solution it provides an intuitive and impartial method to evaluate the effectiveness of various security policies.

Recall, Frank H. Knight states that the difference between uncertainty and risk is that risk can be measured [5]. Whispers transforms uncertainty into measurable risk.

5.1 Future Work

5.1.1 People's "Connectedness"

Whispers estimates risk by inserting a random user into a recipient list, then checking if all the recipients exist in the same topic group. Some people however are highly connected, and are part of multiple social groups in their normal business. These people could slip malicious information between social groups undetected. Whispers identifies the topics as high risk, but does not make an effort to identify the people who contribute to the topic's elevated risk. In a sense these highly connected people reduce Whispers' ability to identify simulated leaks since those people are connected to so many social groups. Instead of simulating leaks by adding random recipients, one could look at the degree of users in the social network to further identify risk. A personal risk metric paired with Whispers risk metric could provide a powerful and useful value for identifying and evaluating security policies.

5.1.2 Interpreted and Native Code

Whispers' implementation is a mixture of C++ and Python. Specifically the topic models and the database are implemented in native code (C and C++). Much Whispers however is implemented in Python augmented with NumPy. There is a clear opportunity for reducing the time Whispers take to evaluate the user graph. Whispers' reliance on interpreted code currently renders it unable to provide a real-time online answer to data leakage questions.

For the Enron email data set used during testing the topic estimation step took 28 hours 43 minutes to estimate topics for 266,000 emails on an AMD 1090T X6 processor. Further analysis showed a majority of the time is spent in Python's NLTK while processing tf-idf. Improving this implementation or distributing the computation could have a large impact on the practicality of Whispers. This however could be improved by converting the tf-idf algorithms from Python's NTLK library to a C++ implementation.

The leakage simulation, which is highly parallelizeable took 3 days on similar hardware. This time is not realistic for a real-time risk analysis tool. Similarly, the graph analysis is done in Python. The final year of data required 7,200 minutes to estimate.

5.1.3 Alternative Vocabulary Pruning

tf-idf is a standard method of determining valuable words in a corpus. However there is some promising research within the space of "information gain". "Information gain" during some experiments at Etsy.com actually identified and removed the stop words [16]. Traditionally as a preprocessor step, stop word removal utilizes standard lists specific to each language which enumerate words of syntax and structure which themselves provide no information. "Information gain" however showed it can identify stop words as meaningless. This could be especially effective in a noisy corpus, or in corpus with domain specific stop words, i.e., words which within the domain are so common they provide no value.

5.1.4 Whispers' Reliance on Now

Whispers only calculates topic models for one quarter at a time. After the topic model is formed for that quarter it attempts to match topics with the previous quarter. If a topic from previous quarter is masked by the present topics, Whispers can not evaluate its current risk and assumes no one is talking about that particular topic. Since no one is talking about that topic, Whispers is left to assume the topic has zero risk. Intuitively this does not make sense and Whispers could be extended to evaluate risk for topics hidden by the present assessment. Blei's Dynamic Topic Models (DTM) could be an excellent start in this effort [35], [36].

5.1.5 Privacy

Privacy is an issue with Whispers. Whispers looks at all email in the company. This presents serious privacy implications, especially in countries which restrict this type of data mining. The Enron email set is unique in that it is a curated data set and personal and private information have been largely removed. While this curating helps improve Whispers' accuracy, it exemplifies the need to anonymize and protect personal information for future implementations.

REFERENCES

[1] Cisco, *Data leakage worldwide white paper: the high cost of insider threats*, 2008. [Online]. Available: http://cisco.com/c/en/us/solutions/collateral/enterprise-networks/data-loss-prevention/white_paper_c11-506224.html (visited on 09/08/2014).

[2] Symantec, *What's new in symantec data loss prevention*, Apr. 2014. [Online]. Available: http://www.symantec.com/content/en/us/enterprise/fact_sheets/b-whats-new-in-dlp12-21299912-en.us.pdf (visited on 09/08/2014).

[3] P. Bright, *With arrests, HBGary hack saga finally ends*, Mar. 2012. [Online]. Available: http://arstechnica.com/tech-policy/news/2012/03/the-hbgary-saga-nears-its-end.ars (visited on 09/08/2014).

[4] *2011 data center efficiency summit: joe kava | google*, Jun. 2011. [Online]. Available: http://www.youtube.com/watch?v=APynRrGuZJA&feature=youtube_gdata_player (visited on 09/05/2014).

[5] F. H. Knight, *Risk, uncertainty and profit,* [Boston and New York, 1921. [Online]. Available: http://hdl.handle.net/2027/loc.ark:/13960/t2j687w8t.

[6] William W. Cohen, *Enron email dataset*, Aug. 2009. [Online]. Available: https://www.cs.cmu.edu/~enron/ (visited on 02/10/2014).

[7] *Data loss prevention software*, en, Page Version ID: 628333637, Oct. 2014. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Data_loss_prevention_software&oldid=628333637 (visited on 10/06/2014).

[8] A. B. Buduru, D. Lucero, and J. Wright, "CloudAssure: a data transfer decision framework for cloud-based systems using dynamic trust metrics", Tempe, AZ, Apr. 2013.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation", in *Journal of Machine Learning*, ser. Research 3, J. Lafferty, Ed., IEEE, Jan. 2003. (visited on 09/07/2014).

[10] D. M. Blei and J. D. Lafferty, "A correlated topic model of science", *The Annals of Applied Statistics*, vol. 1, no. 1, pp. 17–35, Jun. 2007, arXiv: 0708.3601, ISSN: 1932-6157. DOI: 10.1214/07-AOAS114. [Online]. Available: http://arxiv.org/abs/0708.3601 (visited on 09/13/2014).

[11] C. Bishop, *Introduction to bayesian inference*, Nov. 2009. [Online]. Available: http://videolectures.net/mlss09uk_bishop_ibi/ (visited on 09/08/2014).

[12] *Plate notation*, en, Page Version ID: 601839889, Aug. 2014. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Plate_notation&oldid=601839889 (visited on 09/14/2014).

[13] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-graber, and D. M. Blei, "Reading tea leaves: how humans interpret topic models", in *Advances in neural information processing systems*, 2009, pp. 288–296. [Online]. Available: http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2009_0125.pdf (visited on 09/14/2014).

[14] D. Kim and A. Oh, "Topic chains for understanding a news corpus", in *Computational Linguistics and Intelligent Text Processing*, Springer, 2011, pp. 163–176. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-19437-5_13 (visited on 02/05/2014).

[15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval", *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988, ISSN: 0306-4573. DOI: 10.1016/0306-4573(88)90021-0. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0306457388900210 (visited on 09/14/2014).

[16] *Beyond TF-IDF: why, what and how*, May 2013. [Online]. Available: http://www.youtube.com/watch?v=C25txE_dq90&feature=youtube_gdata_player (visited on 09/06/2014).

[17] *Tf-idf*, en, Page Version ID: 623618416, Oct. 2014. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Tf%C3%A2%C2%80%C2%93idf&oldid=623618416 (visited on 10/13/2014).

[18] A. Aizawa, "An information-theoretic perspective of tf-idf measures", *Information Processing & Management*, vol. 39, no. 1, pp. 45–65, 2003. [Online]. Available: http://www.sciencedirect.com.ezproxy1.lib.asu.edu/science/article/pii/S0306457302000213 (visited on 09/14/2014).

[19] P. Zilberman, S. Dolev, G. Katz, Y. Elovici, and A. Shabtai, "Analyzing group communication for preventing data leakage via email", in *2011 IEEE International Conference on Intelligence and Security Informatics (ISI)*, Jul. 2011, pp. 37–41. DOI: 10.1109/ISI.2011.5984047.

[20] B. Klimt and Y. Yang, "Introducing the enron corpus.", in *CEAS*, 2004. [Online]. Available: http://bklimt.com/papers/2004_klimt_ceas.pdf (visited on 06/19/2014).

[21] W. W. Bratton, "Enron and the dark side of shareholder value", *Tul. L. Rev.*, vol. 76, p. 1275, 2001. [Online]. Available: http://heinonlinebackup.com/hol-cgi-bin/get_pdf.cgi?handle=hein.journals/tulr76&section=51 (visited on 06/19/2014).

[22] R. R. Sims and J. Brinkmann, "Enron ethics (or: culture matters more than codes)", *Journal of Business ethics*, vol. 45, no. 3, pp. 243–256, 2003. [Online]. Available: http://link.springer.com.ezproxy1.lib.asu.edu/article/10.1023/A:1024194519384 (visited on 06/19/2014).

[23] R. Bekkerman, "Automatic categorization of email into folders: benchmark experiments on enron and SRI corpora", *Computer Science Department Faculty Publication Series*, Jan. 2004. [Online]. Available: http://scholarworks.umass.edu/cs_faculty_pubs/218.

[24] A. McCallum, X. Wang, and A. Corrada-Emmanuel, "Topic and role discovery in social networks with experiments on enron and academic email.", *J. Artif. Intell. Res.(JAIR)*, vol. 30, pp. 249–272, 2007. [Online]. Available: http://www.aaai.org/Papers/JAIR/Vol30/JAIR-3007.pdf (visited on 06/22/2014).

[25] J. Diesner, T. L. Frantz, and K. M. Carley, "Communication networks from the enron email corpus "it's always about the people. enron is no different"", en, *Computational & Mathematical Organization Theory*, vol. 11, no. 3, pp. 201–228, Oct. 2005, ISSN: 1381-298X, 1572-9346. DOI: 10.1007/s10588-005-5377-0. [Online]. Available: http://link.springer.com.ezproxy1.lib.asu.edu/article/10.1007/s10588-005-5377-0 (visited on 06/19/2014).

[26] R. Rowe, G. Creamer, S. Hershkop, and S. J. Stolfo, "Automated social hierarchy detection through email network analysis", in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, ser. WebKDD/SNA-KDD '07, New York, NY, USA: ACM, 2007, pp. 109–117, ISBN: 978-1-59593-848-0. DOI: 10.1145/1348549.1348562. [Online]. Available: http://doi.acm.org/10.1145/1348549.1348562 (visited on 06/19/2014).

[27] J. C. Coffee Jr, "What caused enron-a capsule social and economic history of the 1990s", *Cornell L. Rev.*, vol. 89, p. 269, 2003. [Online]. Available: http://heinonlinebackup.com/hol-cgi-bin/get_pdf.cgi?handle=hein.journals/clqv89&section=16 (visited on 06/19/2014).

[28] D. R. Hipp, *SQLite*, May 2014. [Online]. Available: http://www.sqlite.org/ (visited on 06/01/2013).

[29] M. Stonebraker, *PostgreSQL*, Jul. 2014. [Online]. Available: http://www.postgresql.org/docs/9.2/static/release-9-2-9.html (visited on 09/29/2014).

[30] F. Schoppmann, *MADlib: big data machine learning in SQL for data scientists*, Jul. 2014. [Online]. Available: http://madlib.net/product/.

[31] C. Reed, *Topic model analyzer*, Oct. 2013. [Online]. Available: https://github.com/cjrd/TMA.

[32] SciPy, *Scientific computing tools for python*, May 2014. [Online]. Available: http://www.scipy.org/about.html.

[33] A. Holovaty, *Django web framework for perfectionists with deadlines*, Sep. 2014. [Online]. Available: https://www.djangoproject.com/.

[34] *Cosine similarity*, en, Page Version ID: 604135823, Oct. 2014. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Cosine_similarity&oldid=604135823 (visited on 10/14/2014).

[35] C. Wang, D. Blei, and D. Heckerman, "Continuous time dynamic topic models", *arXiv preprint arXiv:1206.3298*, 2012. [Online]. Available: http://arxiv.org/abs/1206.3298 (visited on 10/12/2014).

[36] D. M. Blei and J. D. Lafferty, "Dynamic topic models", in *Proceedings of the 23rd international conference on Machine learning*, ACM, 2006, pp. 113–120. [Online]. Available: http://dl.acm.org/citation.cfm?id=1143859 (visited on 10/12/2014).