Behavior-Based Modeling and Its Application to Email Analysis

SALVATORE J. STOLFO, SHLOMO HERSHKOP, CHIA-WEI HU, WEI-JEN LI, OLIVIER NIMESKERN, and KE WANG
Columbia University

The Email Mining Toolkit (EMT) is a data mining system that computes *behavior profiles or models* of user email accounts. These models may be used for a multitude of tasks including forensic analyses and detection tasks of value to law enforcement and intelligence agencies, as well for as other typical tasks such as virus and spam detection. To demonstrate the power of the methods, we focus on the application of these models to detect the early onset of a viral propagation without "content-based" (or signature-based) analysis in common use in virus scanners. We present several experiments using real email from 15 users with injected simulated viral emails and describe how the combination of different behavior models improves overall detection rates. The performance results vary depending upon parameter settings, approaching 99% true positive (TP) (percentage of viral emails caught) in general cases and with 0.38% false positive (FP) (percentage of emails with attachments that are mislabeled as viral). The models used for this study are based upon volume and velocity statistics of a user's email rate and an analysis of the user's (social) *cliques* revealed in the person's email behavior. We show by way of simulation that virus propagations are detectable since viruses may emit emails at rates different than human behavior suggests is normal, and email is directed to groups of recipients in ways that violate the users' typical communications with their social groups.

Categories and Subject Descriptors: K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; C.2.0 [**Computer Communication Networks**]: General—*Security and protection (e.g., firewalls)*

General Terms: Security, Theory, Algorithms

Additional Key Words and Phrases: Email virus propagations, behavior profiling, anomaly detection

1. INTRODUCTION

This article describes the detection capabilities of the Email Mining Toolkit (EMT). EMT provides the means of loading, parsing, and analyzing email logs in a wide range of formats. Many tools and techniques have been available from

This article was guest-edited by Professor Daniel Zeng and Professor Hsinchun Chen, both of the University of Arizona in Tucson (Zeng@bpa.arizona.edu; hchen@eller.arizona.edu).

Authors' address: Department of Computer Science, 450 Computer Science Building, Columbia University, New York, NY 10027; email: {sai,shlomo,Charlie,wei-jen,on2005,kewang}@columbia.edu.

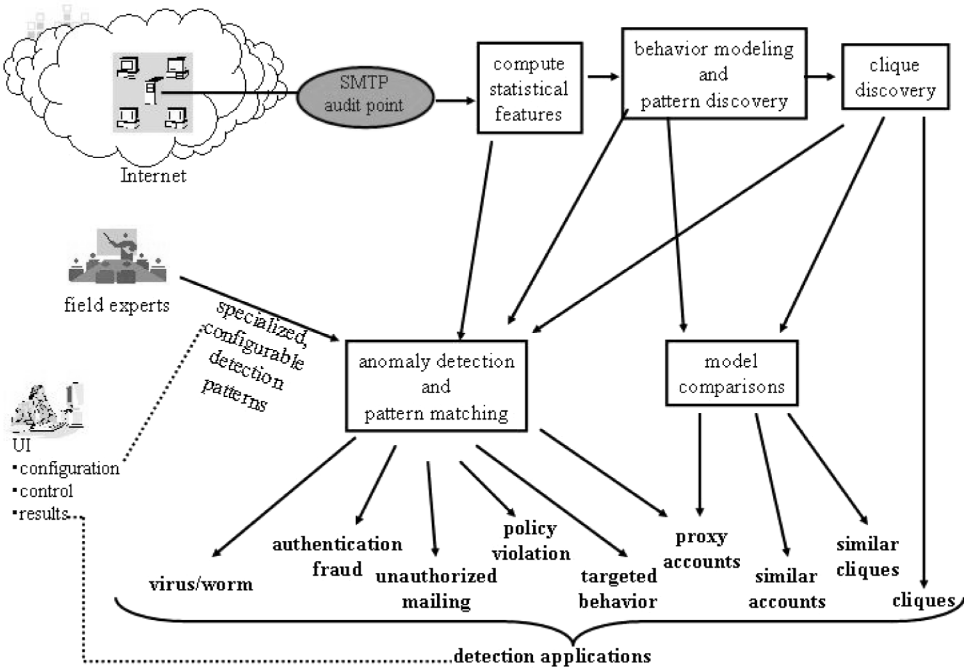


Fig. 1. Overview of email behavior modeling, architecture, and applications.

the fields of information retrieval and natural language processing for analyzing documents of various sorts, including emails. EMT, however, extends these kinds of content-based analyses with an entirely new set of analyses that model "user behavior." EMT models the behavior of individual user email accounts, or groups of accounts, including the "social cliques" revealed by a user's email behavior. EMT's design has been driven by the core security application to detect virus propagations, "spambot" activity, and security policy violations. However, the technology also provides critical intelligence gathering and forensic analysis, capabilities for agencies to analyze disparate Internet data sources for the detection of malicious users, attackers, and other targets of interest. The multiple uses of EMT are graphically displayed in Figure 1. For example, one target application for intelligence gathering supported by EMT is the identification of likely "proxy email accounts," or email accounts that exhibit similar behavior and thus may be used by a single person attempting to avoid detection. Although EMT has been designed specifically for email analysis, the principles of its operation are equally relevant to other Internet audit sources.

This data mining technology has been proven to automatically compute or create both signature-based misuse detection and anomaly detection-based misuse discovery. The application of this technology to diverse Internet objects and events (e.g., email and Web transactions) allows for a broad range of behavior-based analyses including the detection of proxy email accounts and groups of user accounts that communicate with one another, including possibly covert group activities.

Behavior-based misuse detection can provide important new assistance for counterterrorism intelligence by automatically detecting certain patterns across user accounts that are indicative of covert, malicious, or counterintelligence activities. Moreover, behavior-based detection provides workbench functionalities to interactively assist an intelligence agent with targeted investigations and offline forensics analyses.

Intelligence officers have a myriad of tasks and problems confronting them each day. The sheer volume of source materials requires a means of honing in on those sources of maximal value to their mission. A variety of techniques can be applied drawing upon the research and technology developed in the field of information retrieval. There is, however, an additional source of information available that can be used to aid even the simplest task of rank ordering and sorting documents for inspection: behavior models associated with the documents can be used to identify and group sources in interesting new ways. This is demonstrated by EMT as it applies a variety of data mining techniques for profiling and behavior modeling of email sources.

Behavior-based misuse detection is more robust against standard knowledge-based techniques. Behavior-based detection has the capabilities to detect new patterns (i.e., patterns that have not been previously observed), provide early warning alerts to users and analysts based upon detection of changes in behavior, and automatically adapt to both normal and misuse behavior. By applying statistical techniques over actual system and user account behavior measurements, automatically generated models are tuned to the particular source material. This process, in turn, avoids the human bias that is intrinsic when misuse signatures, patterns, and other knowledge-based models are designed by hand, as is the norm.

Despite this, no general infrastructure has been developed for the systematic application of behavior-based detection across a broad set of detection and intelligence analysis tasks such as fraudulent Internet activities, virus detection, intrusion detection, and user account profiling. Today's Internet security systems are specialized to apply a small range of techniques, usually knowledge-based, to an individual misuse detection problem, such as intrusion, virus, or SPAM detection. Behavior-based detection technology as proposed herein will likely provide a quantum leap in security and in intelligence analysis in both offline and online task environments.

Table I enumerates a range of behavior-based Internet applications. These applications cover a set of detection, security, and marketing applications that exist within the government, commercial, and private sectors. Each of these applications is within the capabilities of behavior-based techniques by applying data mining algorithms over appropriate audit data sources.

The diversity of these applications also spans national security concerns in ways that may not be immediately apparent. Consider spam detection, for example. Spam continues to be a scourge upon users of the Internet, consuming enormous bandwidth and reducing the productivity of corporate email users. However, spam may also present a national security issue when employed by terrorists in blended attacks that combine physical with cyber attack events, as a "force multiplier." Imagine a terror attack upon some critical

Table I. General Behavior-Based Analysis Internet Applications

Application	Description and Variations:
Internet fraud detection	Unauthorized outgoing email, unauthenticated email, unauthorized transactions
Malicious email detection	Spam, viruses, worms
Intrusion detection	Network-based, host-based, application-based
User community discovery	Closely connected users
Behavior pattern discovery	Account-based, community-based
Analyst workbench	Interactive forensic analysis
Account proxy discovery	Accounts used by same user
Collaborative filtering	Website recommendations, purchase recommendations
Policy violation detection	ISP or enclave security policies
Web- and spam-bot detection	Statistics/knowledge gathering, site maintenance, search-engine spider

infrastructure of the U.S. followed by a flooding of spam emails warming citizens that schools and children may be targeted next. The psychological operations (or psychops) of terrorism would be greatly enhanced by spam, and hence spam detection and filtering may ultimately be considered a national security priority. (Indeed, Web sites like Spammimic provide the means of encoding secret messages to look like spam, allowing nefarious broadcast secret messages to go unnoticed.)

EMT is an offline system applied to email files gathered from server logs or client email programs. EMT computes information about email flows from and to email accounts, aggregates statistical information from groups of accounts, and analyzes content fields of emails. The EMT system provides temporal statistical feature computations and behavior-based modeling techniques, through an interactive user interface, to enable targeted intelligence investigations and semimanual forensic analysis of email files.[1] EMT provides the following functionalities, interactively:

- Querying a database (warehouse) of email data and computed feature values, including ordering and sorting emails on the basis of content analysis (n-gram analysis [Damashek 1995] keyword spotting, and classification of email supported by an integrated supervised learning feature using Naïve Bayes classifier-trained on user-selected features) [John and Langley 1995; Mitchell 1997], and historical features that profile user groups by statistically measuring behavior characteristics;
- Creation of user models that group users according to features such as typical emailing patterns (as represented by histograms over different selectable statistics) and email communities (including the "social cliques" revealed in email exchanges between email accounts;
- Application of statistical models to email data to detect abnormal or unusual email events;

[1] EMT does not employ the means to analyze image content, but techniques such as those described in Niblack et al. [1993] and Smith [1997] could conceivably be embedded in EMT.

—Analysis of the rank order or relative importance of individuals in an organization based upon their membership in many groups and the "response rates" to their emails exhibited by their coworkers.

To demonstrate the power of the behavior-modeling techniques of EMT and the variety of functions and features it provides, this article details a common email detection problem familiar to all users: virus detection. Detecting viral and spam emails is important in its own right. The economic loses due to worm and virus email propagations are estimated to have been $13 billion in 2003 alone. Viruses and spam emails consume bandwidth, deny service, and damage systems. Most approaches to virus and spam detection are based on content filtering techniques. Here we demonstrate the power of behavior-based techniques for virus detection in particular.

The rest of this article is organized as follows. Section 2 defines the virus detection problem and introduces how behavior-based methods may be employed to detect the early onset of a viral propagation. Section 3 describes the related research on anomaly detection in intrusion detection systems from which this work was originally conceived. Section 4 provides an overview of EMT's features and details three different kinds of behavior models supported by EMT and their use in virus detection. The first model we discus is a user's (social) cliques revealed in the person's email behavior, and an independent test of this model shows that viral emails violate a user's email clique behavior. We then detail a frequency-based model of the user's typical recipients, and likewise a simulated test showing how this model may detect viral email propagations. We also present an experiment where these two models are combined for better detection performance and reduced false alarm rates. Section 4 concludes with another statistical model, a cumulative distribution over outbound email recipients, and a final performance evaluation of the correlation of all three models producing our best results. Section 5 concludes the article with a discussion of future research and suggests how other EMT features have interesting uses in forensics and intelligence.

2. DETECTING VIRUS BEHAVIOR

By way of demonstrating the power of EMT's behavior-based models, the rest of this article is devoted to a detailed analysis and experimental evaluation of virus detection. Indeed, email is the dominant method of choice for the propagation of viruses and worms. Typically, a virus will extract email addresses in an infected computer and send copies of itself to some or all of these addresses. These addresses may be obtained from many sources, such as a computer's address book, socket-layer sniffing, or a locally stored email archive. Virus scanners are signature-based, which means they use a collection of byte sequences or embedded strings to identify known malicious attachments. If a virus scanner's signature database does not contain a signature for a malicious program, the virus scanner is unable to detect or protect against that malicious program. In general, to mitigate against this *false negative problem*, virus scanners require frequent updating of signature databases; otherwise the scanners become useless in detecting new attacks. Similarly, security patches protect systems only when they have been written, distributed, and applied to host systems in response to known attacks. Until then, systems remain vulnerable and attacks are able to propagate widely and rapidly over the Internet.

For example, the SoBig.F propagation that occurred in the late summer of 2003 spread rapidly across the Internet using a high-speed spam-based propagation strategy. It took several days before an effective signature was available for distribution to locally update virus scanners to stop this virus. During this period of time, no signature-based filters were available and SoBig.F flooded the Internet, causing remarkable damage and expense. It is such new viral attacks that are the subject matter of this work.

Furthermore, virus writers have demonstrated their continual cleverness by thwarting virus scanners with strategies that defy signature-based detection. Stopping a polymorphic virus that uses several points of entry, and that also "morph" the contents of the virus in various ways, can be a daunting task using traditional signature-based virus scanning methods alone.

Our core premise is that viral propagations fundamentally behave differently than typical human user email behavior. Thus, the idea is to create a detector that models a user's email behavior, and then to apply this model to the email flow of a user's account to detect abnormal or anomalous email flows that may indicate a viral propagation has been initiated targeting the user as a new victim. Much prior work on anomaly detection systems has been reported in the literature to solve the false negative problem of signature-based detection systems in intrusion detection systems. Here we apply this methodology to email. We believe EMT demonstrates a way to raise the bar of protection in detecting and extinguishing viral propagations as early as possible until new signatures are developed and deployed. Consider the following observations.

First, viral email propagations involve an email sent to or from a victim email account with either an attachment or with something equivalent to an HTML page in the text body. In the former case, the user will have to run the executable that launches a virus directly, or invoke a program that uses the seemingly innocent data file that exploits the weakness of the program that makes use of it. In the latter case, the user may simply click on an innocent appearing URL that may start the download and execution of "malware."

Second, it is highly unlikely a virus will propagate itself with only one or a few emails. This is because usually viruses are designed to infect as many computers as possible in a short period of time. Otherwise, they would be stopped long before they had a chance to inflict damage on many systems. Creating many copies ensures the virus will propagate quickly and widely. We conjecture that the frequency of emissions of emails during a viral propagation will be substantially different than the victim user's typical email rate (both inbound and outbound).

Finally, a virus is ignorant of its victim's behavior, in the sense that it does not know the relationship between a user and those with whom the user communicates. For example, a user would be unlikely to send an email, or many copies of an email, to a large number of recipients among the user's separate social cliques. Instead, a virus might use simple hard-coded rules in deciding whom to propagate to, violating the user's typical behavior in sending emails to his/her social cliques. These observations suggest that viral propagations may be detected by profiling email behavior and using the user's behavior models to detect the onset of a "randomly guided" virus propagation.

Behavior-based detection is not a new concept. Credit card fraud detection [Stolfo et al. 1999] is perhaps the best example of a widely deployed security system that depends upon profiling behavior of credit card users. We posit that a similar approach directed toward "email transactions" will provide comparable broad-based security without requiring a complete overhaul of email protocols and server implementations.

By measuring the behavior of individual email users over time using different statistics and profiling techniques, and the probabilities associated with these statistics, we wish to correlate as much evidence as possible from multiple models to accurately detect errant or malicious email while minimizing false alarms.

Three types of behavior-based models are examined in detail: *user cliques*, the *Hellinger distance*, and *cumulative distribution* models. The user clique model profiles a user's communication groups that naturally occur in her or his email communication history (for example, colleagues, family members, friends, etc). These clique models also provide important information to rank order the relative importance of individuals in an organization. For example, a person who is a member of many separate cliques indicates that that person has many compartmentalized relationships with subgroups within the organization.

The Hellinger distance model profiles the distribution of the frequency of communication of the user, and the variability of that frequency, between a user and his/her correspondents. (Interestingly, the analysis we have performed on the email archives of many volunteer email users has revealed that email communication behavior follows a Zipf distribution, the same distribution that models the naturally occurring frequency distribution of words in natural language.) The recipient frequency analysis also identifies the relative importance of various email users. By extending the analysis to compute the "response rates" to a user's typical recipients, one can learn the relative rank ordering of various people. Those to whom a user responds immediately to be are likely important people in the organization. A virus would also not know this relative ranking and may behave quite differently than the victim user.

The cumulative distribution model profiles the (daily) rate at which a user sends emails to distinct parties in sequential order. Again, a virus would generally not know this statistic and so would violate the user's typical behavior while propagating itself to new victims.

These three models are more or less orthogonal to each other and are combined together to form a correlated model that yields very good virus detection performance, purely on the basis of behavior, not content analysis.

We describe a number of experiments using an email archive collected from 15 volunteers (1 faculty member, 2 research associates, and 12 graduate students) representing about 8–9% of the Columbia University Computer Science Department population of users. This archive was acquired late in 2002 by providing a script to capture all of the users' emails covering the time frame of 2001 and 2002 while hashing and compressing the body of each message, retaining all other exact information in each email. Approximately 88,000 email messages and 45,000 attachments were acquired from many volunteers. However, for the experiments reported in this article, we selected a subset of the archives of 15 specific users who stored all of their emails without any filtering. We conducted independent tests of each statistical model introduced and describe the portion of the archive used in each such experiment. The experiments we report were performed by injecting the sample of the archive with simulated viral emails using the virus's propagation strategies described above. The viruses were not run on a network to avoid potential damage (and to avoid violating security policy).

To measure and compare the detection rate of the combined behavior models, there is no baseline to study other than typical COTS virus scanners. We take the point of view that a virus scanner will have a 100% true positive (TP) rate and 0% false positive (FP) rate for any virus for which a signature exists; but it will also exhibit a 0% TP rate for any "new" virus for which a signature has not yet been developed and deployed. It is these "new" viruses that cause damage, and that we used under simulation to test the performance of EMT. Of particular importance here is the tradeoff between EMT's TP rate (detecting new viruses) and its FP rate, that is, the percentage of emails deemed viral by EMT but which are actually nonviral. We demonstrate this performance using ROC curves and evaluate the "annoyance rate" EMT may exhibit in generating false alarms.

The results show that EMT's behavior models are an effective detection system. Its best performance in detecting inbound viral propagations over all users was 99% TP and 0.38% FP, while its best performance for detecting outbound viral propagations from an account was 99% TP and 0.9% FP. EMT also exhibited its worst performance for inbound viral detection at 70% TP and 0.38% FP (outbound detection was 60% TP and 0.9% FP) if the viral propagation was a very slow, stealthy propagation with one viral email delivered every 5 days. Thus, fast propagations were easy to detect by observing anomalous email flows that were inconsistent with a user's normal email behavior. Slow and stealthy propagations were, however, hard to detect.

3. RELATED REASEARCH

EMT is a data mining and profiling system applied to email data to detect anomalous email behavior. Our prior work on the Malicious Email Tracking (MET) system focused on modeling the behavior of attachments, and attachment flows in email [Bhattacharyya et al. 2002] among participating sites either within an enclave or across sites within an enterprise. A precursor project, the Malicious Email Filter [Schultz et al. 2001], focused on the detection of likely malicious attachments in those flows. The concept behind MET was to measure the statistics of attachment flows across a mail server and to detect viral propagation as an anomaly (e.g., a "burst" as a "high host saturation") in this attachment flow. Thus, MET was best viewed as an anomaly detector for flows.

Anomaly detection systems were first proposed by Denning [1987] for intrusion detection, and later implemented in NIDES [Javitz and Valdes 1993] to model normal network behavior in order to detect deviant behavior that may correspond to an attack against a network computer system. Lee et al. [1997] described a framework and system for auditing and data mining and feature selection for intrusion detection. This framework consisted of classification, link analysis, and sequence analysis for constructing intrusion detection models. [Lee at al. 1998, 1999].

A variety of other work has appeared in the literature detailing alternative algorithms to establish normal profiles, applied to a variety of different audit sources, some specific to user commands for masquerade detection [Schonlau et al. 2001], others specific to network protocols and LAN traffic for detecting denial of service attacks [Lee and Xiang 2001; Matthew et al. 2001; Taylor and Alves-Foss 2001] or Trojan execution, or to application file-system or system call-level data for malware detection [Hershkop et al. 2003; Hofmeyr et al. 1998; Warrender et al. 1999], to name a few.

A variety of different modeling approaches have been described in the literature to compute baseline profiles. These include probabilistic or statistical distributions over temporal data [Apap et al. 2002; Eskin 2000; Ye 2000; Lane and Brodley 1999; Wagner and Soto 2002], supervised machine learning [Ghosh et al. 1999; Lee et al. 1997], and unsupervised cluster-based algorithms [Eskin et al. 2002]. Some approaches consider the correlation of multiple models [Ghosh et al. 1999; Warrender et al. 1999].

In general, when an audit source is a stream or temporally ordered data, a variety of models may be defined for an audit source and a detector may be computed to generate an alarm if a violation is observed based upon volume and velocity statistics. Volume statistics represent the amount of data observed per unit of time, while velocity statistics model the changes in frequency of the data over time. In our EMT work, for example, we compute volume statistics, such as the "number of distinct recipients of emails" and the "cumulative number of emails with attachments" sent sequentially. EMT also computes the Hellinger distance of the recipient frequency as an example of velocity statistics. These two kinds of statistics represent one aspect of a user's behavior profile and are used to detect the abnormal behavior indicative of virus and spam emails.

We are not aware of any prior work devoted to anomaly detection applied to email audit streams other than MET. However, recent work by Newman et al. [2002], at HP [Williamson 2002], and by social scientists at Columbia University [Watts 2003] has analyzed email account connectivity for various purposes. In the case of Newman et al. they considered email accounts linked in a graph as defined by address books to measure network density specifically to provide guidance on address book management. They noted that viral propagations will spread fast among accounts whose address books are deemed "dense" from a graph theoretic point of view. The HP and Columbia social science work is similar to our work on cliques. In these two pieces of work, the communication density and flow within an organization is studied to understand the effectiveness of communication within an organization. In the case of the Columbia social science work, the authors sought to answer the question whether six levels of indirection indeed separate any two people within email communication.

4. EMT MODELING FEATURES

EMT [Stolfo et al. 2003] is useful for report generation and summarizing email archives, as well as for detecting email security violations when incorporated into a real-time violation detection system, such as MET [Bhattacharyya et al. 2002]. EMT contains a large collection of (statistical) modeling features that may be combined for various detection tasks.

EMT is implemented in Java, providing a GUI implementing an interface to an underlying database application. The data can reside in any SQL RDBMS. EMT is also provided with a set of parsers written in Java that can read email files from a variety of formats (mbox, nsmail, Outlook, and Lotus are all supported) and inserts data into the underlying database [MYSQL 2002]. Each row of this database is a detailed record of an email to which a variety of statistical analyses may be applied. Most of EMT's statistical models are computed by SQL commands against this database. Thus, EMT has been designed for scalability to large email archives, and generality to other communication mediums. A version of EMT that analyzes instant messaging traffic has also been implemented.

For this article, we focus primarily on testing three behavior models computed by EMT to detect the onset of viral propagations. We first describe EMT's analysis of group communication behavior.

4.1 Group Communication Models: Cliques

In order to study email flows between groups of users, EMT computes a set of *cliques* in an email archive. We seek to identify clusters or groups of related email accounts that participate with each other in common email communications, and then use this information to identify unusual email behavior that violates typical group behavior. For example, intuitively it is unlikely that a user would send a distinct message to a spouse, a boss, "drinking buddies," and church elders all appearing together as recipients of the same message (whether delivered in one email, or a series of emails). Of course this is possible, but it is rather unlikely. A virus attacking a user's address book at random would surely not know these social relationships and the typical communication pattern of the victim. Hence it would violate the users' group behavior profile if it propagated itself in violation of the user's *social cliques*.

Clique violations may also indicate email security policy violations internal to a secured enclave. For example, members of the legal department of a company might be expected to exchange many Word attachments containing patent applications. It would be highly unusual, and probably unwise, if members of the marketing department, and human resource services likewise received these attachments. We can infer the composition of related groups by analyzing normal email flows to compute the naturally occurring cliques, and use the learned cliques to alert when emails violate that clique behavior. This may be particularly important in intelligence applications that seek to discover violations of "compartmentalized" communications.

Conceptually, two broad types of cliques can be extracted from user email archives: *user cliques* and *enclave cliques*. In simple terms, user cliques can be computed by analyzing the email history of only a single user account, while enclave cliques are social groups that emerge as a result of analyzing traffic flows among a group of user accounts within an enclave. In this article, we utilize only user clique models, leaving the analysis of enclave cliques to a future article.

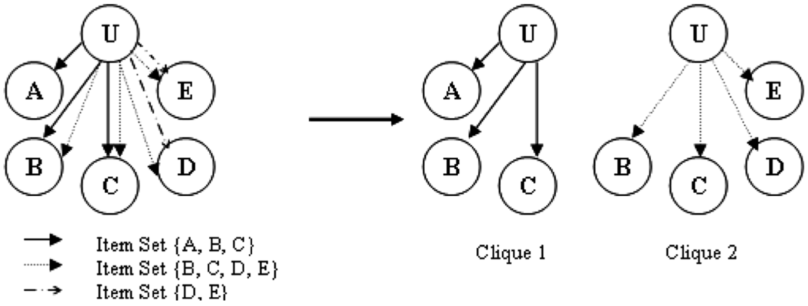


Fig. 2. Three item sets from account U: {A, B, C}, {B, C, D, E} and {D, E}. The first two sets share two nodes and the last set is subsumed by the second set. The resulting user cliques are {A, B ,C} and {B, C, D, E}.

4.1.1 *User Cliques.* We model the collection of recipients in a single email as a set, and summarize these sets and their dynamics. This information is used to detect abnormal emails that violate a user's clique behavior.

Formally, email communication can be captured by a directed graph $G(V, E)$ with the set of nodes V corresponding to individual email accounts and a set of edges E corresponding to email messages. A directed edge e_{12} exists if $v_1$sends an email to v_2, where v_1 and v_2 are nodes in V. Viewed in this way, cliques are a certain pattern in this graph that we characterize and use as norms of communication behavior. (EMT also provides an enclave clique feature that implements the Bron-Kerbosch clique finding algorithm [Bron and Kerbosch 1973] to compute all connected components in this graph.)

Aside from the graphical view, the user clique model is best described in terms of item sets. An item set is a set of items associated with a transaction, such as a single purchase at a supermarket. The goal of analyzing item sets is to extract useful association rules of how items appear together. This problem has been studied in the data mining and database community and is of great commercial interest for its wide range of applications and the potential predictive value that can be derived [Agrawal et al. 1993].

In the context of mining email, an email can be viewed as a transaction that involves multiple accounts, including a sender (in the FROM field) and recipient(s) in the (TO, CC, and BCC fields). If we discover the rules governing the coappearance of these addresses, we can the use them to detect emails that violate these patterns. Suspicious emails may then be examined further by other models to confirm or deny that they are malicious.

The recipient list of a single email can be viewed as a clique associated with the FROM account. However, using this set (or item set) directly is problematic for two reasons. First, a single user account would contain a large number of such sets and enumerating them for real-time reporting or detection tasks would be undesirable. Second, some of these sets are duplicates or subsets of one another and it would be difficult to use them directly for any purpose. For these reasons, we define a *user clique* as a set of recipients that cannot be subsumed by another set. Thus, we compute the most frequent email item sets that are not subsumed by another larger item set (see Figure 2). Naturally, a single user will have a relatively small number of user cliques. As an example, suppose a user has in his/her sent-folder four emails with the following recipient lists: {A, B, C}, {A, B, C}, {A, B}, and {A, B, D}. The user cliques belonging to this user would be {A, B, C} and {A, B, D}. Note that duplicate user cliques are removed, as having then does not contribute useful information.

Once these sets are derived offline by analyzing a user's "profile" period, we inspect each email sent from the user's account in a subsequent "test" period of time to determine if there is a *clique violation*—that is, if the recipient list is inconsistent with the user's cliques. An email sent from a user is regarded as inconsistent with the user's cliques if its recipient list is not a subset of any user cliques belonging to that user.

The usefulness of this model depends not only on how quickly new groups of recipients form over time but also on how it is combined with other models. Installing a monitoring tool using this model on a new account or an account that is constantly communicating with new groups may cause too many false alarms and thus render the model useless. However, this very behavior is indicative of user email usage patterns and thus can be turned into a feature that characterizes user behavior.

Although the dynamics of clique formation [Davis 2003] (and expiration) are implemented in EMT, in the present article we shall ignore the dynamics of clique formation to explore the utility of the base user clique model. Computing the set of "static cliques" is sufficiently informative for the purposes at hand; this model provides useful evidence of a viral propagation launched from a user's account.

Notice that if a user ever sends a single broadcast email to everyone in her/his address book, there would be only one user clique remaining in the model for that user. This would render the model almost useless for the user in question for the virus detection task because no clique violation is possible as long as a user does not communicate with someone new. In practice, however, this scenario is highly unlikely to happen. We illustrate this point by examining the communication patterns of 15 users in our database. We show that most of the time, a user will send a single email to less than 10% of the people in his/her address book. For an account with a small address book, a single email could cover 20%, 30%, or an even higher percentage of the address book. As we can see from Table II, the probability of an email covering a given range of percentages of an address book decreases quickly as the percentage range increases. In fact, none of the 15 users ever sent a broadcast email to everyone in his/her address book.

Although a single user might produce a broadcast message to all addresses in her/his address book (eg., when changing contact information), intuitively, and as the data shows in Table II, such events are so rare that EMT ought to detect these cases as clique violations. This is accomplished by modeling the frequency of clique communications. Even if the model falsely labels such an event as a viral propagation, the annoyance factor associated with this rare event can be comfortably ignored and does not invalidate the core thesis that viruses will violate a user's clique behavior, as we demonstrate experimentally in the next section. For users who may broadcast every one of their email messages to all addresses in their address book, the clique model may have little value.

Table II. Percentages of an Address Book Covered by A Single Email; Data Provided by 15 Distinct, Volunteer Email Users

User	No. of Distinct Addresses	≤ 10%	10–20%	20–30%	30–40%	40–50%	50–60%	60–70%	70–80%	80–90%	≥ 90%
1	324	1	0	0	0	0	0	0	0	0	0
2	1308	1	0	0	0	0	0	0	0	0	0
3	38	0.46	0.49	0.04	0	0	0	0	0	0	0
4	144	0.96	0.01	0.01	0.00	0	0	0	0	0	0
5	26	0	0.74	0.04	0.09	0.06	0.02	0.01	0.02	0	0
6	105	0.95	0.04	0.01	0	0	0	0	0	0	0
7	64	0.98	0.01	0	0	0	0	0	0	0	0
8	92	0.95	0.05	0	0	0	0	0	0	0	0
9	43	0.70	0.15	0.11	0.04	0.01	0	0	0	0	0
10	24	0.54	0.12	0.25	0.05	0.02	0.01	0	0	0	0
11	75	0.91	0.09	0.01	0	0	0	0	0	0	0
12	1231	1	0	0	0	0	0	0	0	0	0
13	231	1	0	0	0	0	0	0	0	0	0
14	368	1.00	0.00	0	0	0	0	0	0	0	0
15	568	1	0	0	0	0	0	0	0	0	0
Avg	291	0.83	0.11	0.03	0.01	0.00	0.00	0.01	0.01	0	0

4.1.2 *Test of Simulated Viruses.* Here we evaluate the utility of user clique violations (independent of other modeling techniques) for viral propagation detection. We simulate viruses by inserting "dummy" emails into an email archive following a propagation strategy that has been observed from numerous real viruses seen in the wild. The first 80% of emails sent from each account is deemed the profile period used for deriving user cliques associated with that account. The remaining 20% of the emails are used during the testing phase where the dummy emails simulating the propagation are inserted.

For this simulation, it is not critical exactly when and how often viral emails are sent out. That is, we ignore the propagation rate entirely; determining whether or not a recipient set violates existing user cliques is independent of the timing of the email in question. However, during the simulation/test phase, user cliques are updated on a daily basis and the timing of email is affected slightly. Such effects are still more or less negligible, as having viral emails that are sent later in time is tantamount to having a longer training phase and a shorter test phase.

In terms of modeling attack strategies, we tested the effectiveness of the user clique violation model against various sizes of a viral email recipient list. For illustrative purposes, we assume that a virus would fetch email addresses from the address book of an infected user to propagate itself. In reality, email addresses could be obtained via others means, such as scanning the inbox, sent folder, and email archives. Without loss of generality, the simulation has the virus propagating itself to recipients chosen at random. However, the usefulness of user-clique violation detection in practice depends on how a virus obtains the target email addresses. For example, a virus obtaining addresses from an inbox and replying to respective senders and everyone else in the message might not be detected easily, depending upon how compatible the addresses

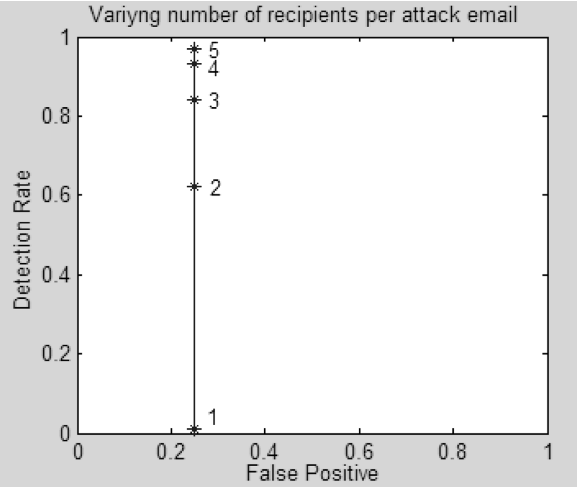


Fig. 3. Test of simulated viruses. Parameter: varying number of email recipients per attack email.

are with existing user cliques. (This implies that the virus would imitate or mimic the user's behavior; avoiding this mimicry attack involves other security mechanisms and is the subject of ongoing work to be reported in a future article.)

Herein lies the reason for false positives produced by this model. The other models we explore below mitigate these mistakes by modeling the user's email frequency distribution.

As we can see from the ROC curve in Figure 3, the false positive rate is invariant with respect to the size of the recipient list. This is expected, as this rate is defined as the number of false positives over the number of normal emails, and both of these quantities do not vary with respect to how viral emails are sent under our simulation setting. It is interesting to note that the true positive detection increases dramatically as the size of the recipient lists in a viral email grows from 1 to 2 to 3 and then approaches 100% gradually as the list size further increases. This result is intuitive; we should not expect that there would be many user clique violations if a virus sends an email to only one recipient at a time. The fact that this number is not zero, as one might have thought, deserves some mention. This could happen because certain email addresses appear in an address book before any email is sent to them.

While a virus may try to thwart our detection effort by sending itself to one address at a time, it will inevitably have to send many separate emails to achieve the same propagation speed. In doing so, it is likely a different level of threshold would be triggered by another model that is tuned to the user's outbound email frequency. Thus we combine the user clique detection model with other methods of detection, such as Hellinger distance described in the next section, to mitigate this error. Alternatively, as demonstrated below in the section on the *Backward/Forward Scanning algorithm*, we may delay email transmission to gather evidence of clique violations among a sequential set of similar or equivalent emails indicative of a propagation. The TP and FP detection rates dramatically improve under this strategy as well.

4.2 Nonstationary User Profiles

Most email accounts follow certain trends, which can be modeled by an underlying distribution. As a practical example, many people will typically email a few addresses very frequently, while emailing many others infrequently. Day-to-day interactions with a limited number of peers usually results in some predefined groups of emails being sent. Other contacts communicated with on a less than daily basis have a more infrequent email exchange behavior. These patterns can be learned through an analysis of a user's email archive over a set of sequential emails. For some users, 500 emails may occur over months, for others over days. The duration of these email transmissions is not material for the profile we now consider.

Almost every user of an email system develops a unique pattern of email emission to a specific list of recipients, each having their own frequency of occurrence (with respect to the number of emails). Modeling every user's idiosyncrasies enables the system to detect malicious or anomalous activity in the account. This is similar to what happens in credit card fraud detection, where current behavior violates some past behavior patterns.

It is important to note that a user's email pattern is not static. The frequency distribution computed by EMT accommodates the user's change in frequency that may occur during the profile period, whether the user goes on vacation, is out sick, or is in a flurry of activity to meet a deadline for submission. These changes are measured and modeled as we describe next.

4.2.1 *Profile of a User.* We analyze the user account's activity in terms of recipient frequency. Figure 4 displays the frequency at which the user sends emails to all the recipients communicated to in the past. Each point on the x-axis represents one recipient and the corresponding height of the bar measures the frequency of emails sent to this recipient as a percentage. (The display is an actual distribution from a volunteer email account. All others have been found to follow the same type of distribution.)

This bar chart is sorted in decreasing order, and usually appears as a nice convex curve with a strong skew: a long low tail on the right side, and a very thin spike at the start on the left side. This frequency bar chart can be modeled with either a Zipf distribution, or a DGX (discrete Gaussian exponential) distribution, which is a generalized version of the Zipf distribution. This family of distributions characterize some specific human behavioral patterns, such as word frequencies in written texts or URL frequencies in Internet browsing [Bi et al. 2001]. In brief, its main trait is that a few objects receive a large part of the flow, while many objects receive a very small part of the flow.

The rank-frequency version of Zipf's law states that$f(r)1/r$, where $f(r)$ is the occurrence frequency versus the rank r, in logarithmic-logarithmic scales. The generalized Zipf distribution is defined as$f(r) \propto (1/r)^{\theta}$, where the log-log plot can be linear with any slope. Our tests indicate that the log-log plots are concave, and thus require the use of the DGX distribution for a better fit [Bi et al. 2001].

We also analyze the number of distinct recipients and attachments. However, we use "records" instead of emails. For example, if a user sends one email, to

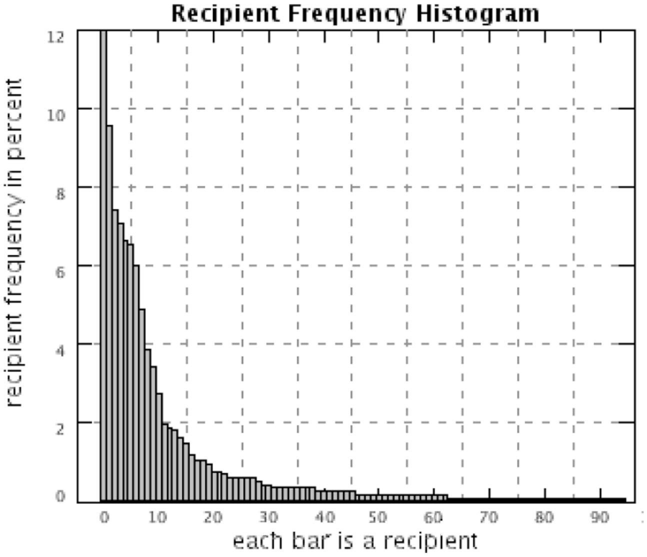


Fig. 4. Recipient frequency distribution. Notice it follows a Zipf-like distribution.

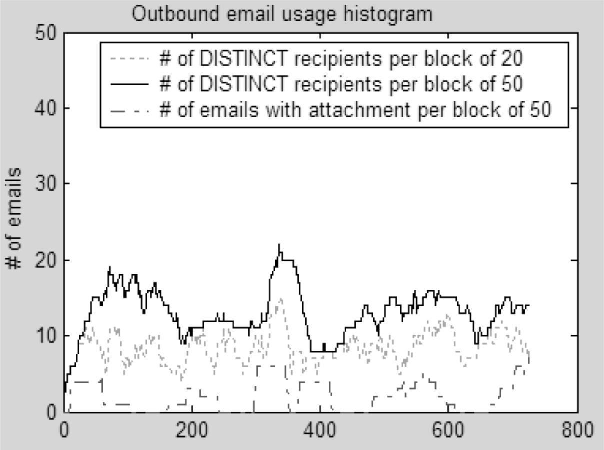


Fig. 5. Plots showing variability of recipients and attachments for one account.

B, and carbon copies that email to C, D, and E, there are four records in the database recording these communication events. Because a virus may send to several victims via one email (by Carbon Copying everyone), or via several emails (one by one), by using "records" we consider both cases within the model.

Figure 5 contains several curves that visualize the *variability* of the user's emission of emails. The statistics calculated are the *number of distinct recipients* and the *number of messages with attachments*. The first type of curve uses a rolling window of 50 (or 20) records to calculate the number of distinct recipients. These values are ordered by time. For example, in the data shown in Figure 5, the user has 750 records, and all of them are sorted by time. At location 200 in the chart, the value of the curve, with the rolling window size of 50, is 10 (see the highest plot). This means that in the past 50 records there are 10 different recipients of the user's outbound email. What this analysis means is that the higher this plot approaches 50, the wider the range of recipients the selected user sends emails to over time. (The user is thus conducting many conversations with many people.) On the other hand, if the metric is low, it means that the user predominantly sends messages to a small group of people.

We have also plotted a curve in Figure 5 (the middle dashed line) using 20 as the window size instead of 50. This metric has a faster reaction to anomalous behavior, while the previous one using blocks of 50 shows the longer-term behavior. The short-term profile can be used as the first level of alert, the longer-term one acting to confirm any detected anomalous frequency change.

Another type of curve shown in Figure 5 is the number of messages with attachment(s) per block of 50 records (the lowest dashed line). It shows the average ratio of emails with attachments versus emails without attachments, and any sudden spike of emails sent with attachments will be detected on the plot as a significant spike. The profile displays a fingerprint of a specific user's email frequency behavior. The most common malicious intrusion can be detected very quickly by these metrics. For instance, a Melissa-type virus would be detected since the curves will increase rapidly to 50, 20, and 50, respectively.

4.2.2 *Chi Square Test of Recipient Frequency.* We test the hypothesis that the recipient frequencies are identical over two different time frames by a chi square test. Obviously, recipient frequencies *are not constant* over a long time horizon, as users will add new recipients and drop old ones. It can be informative for behavioral modeling, though, to analyze the variability of frequencies over two near time frames.

We compare two time periods of activity for the same user. The idea is to treat the first period as the true distribution corresponding to the user under normal behavior, while the second time period is used to evaluate whether or not the user's frequencies have changed, providing evidence that perhaps a malicious activity is taking place. Generally, we operate under the usual 1/5–4/5 ratio between testing and training sets. For example, we use 1000 records as a training-testing set; the 200 most recent records are selected as the testing range, while the previous 800 are designated the training range. Note that the testing range represents a user's new incoming or outgoing emails, and the training range represents the previous normal behavior used to generate the profile.

Assuming that the observed frequencies corresponding to the first, longer time frame window are the true underlying frequencies, the chi square statistic enables us to evaluate how likely the observed frequencies from the second time frame are drawn from that same distribution [Hogg and Craig 1994]. The chi square formula is

$$Q = \sum_{i=1}^{k} (X(i) - np(i))/np(i),$$

where $X(i)$is the number of observations for recipient i in the testing range, $p(i)$is the true frequency calculated from the training range, n is the number of

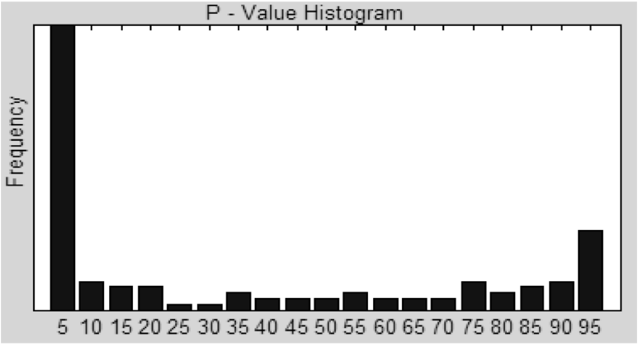


Fig. 6. *P*-value plot.

observations in the testing range, and k is the number of recipients during the training period. There are $k - 1$ degrees of freedom.

The p-value represents the probability that the frequencies in both time frames come from the same distribution. In order to get an idea of the variability of the frequencies under real conditions, we used a sample of 37,556 records from eight users. We ran two batches of calculations. First, we used a training period size of 400 records and a testing period size of 100 records; for each user, we started at the first record, calculated the p-value, then shifted the two windows by steps of 10 records until the end of the log was reached, each time calculating the p-value. Second, we reproduced the same experiment, but with a training period size of 800 records and a testing period size of 200 record. We thus collected a total of 7,947 p-values; the histogram is shown in Figure 6.

Under the hypothesis that the frequencies are constant, the histogram would be expected to be a flat line. On the contrary, the histogram in Figure 6 is characterized by a very large concentration of p-values between 0 and 5%, and a large (but less large) concentration between 95 and 100%, while p-values in the range of 5 to 95% are underrepresented. Intuitively, most of the time, frequencies change significantly (in a statistical sense) between two consecutive time frames; this is why 60% of the p-values are below 5% (as a low p-value indicates a very high chance that the frequencies have changed between two time frames). Email users tend to modify their recipient frequencies quite often (at least the eight volunteers) whose records we examined did). On the other hand, there are nonnegligible times when those frequencies stay very stable (as 13% of the p-values are above 95%, indicating strong stability). As the frequencies were found to be so variable under normal circumstances, the chi square test itself could not be used to reliably detect an abnormal email behavior. Instead we utilized the Hellinger distance metric, a related metric that evaluates changes in frequency over two frequency distributions.

4.2.3 *Hellinger Distance.* Our first tests using the chi square statistic revealed that the frequencies cannot be assumed to be constant between two consecutive time frames for a given user. We postulate, though, that what is specific to every user is how variable their frequency changes are over time. We model this user behavior by calculating a measure between two frequency

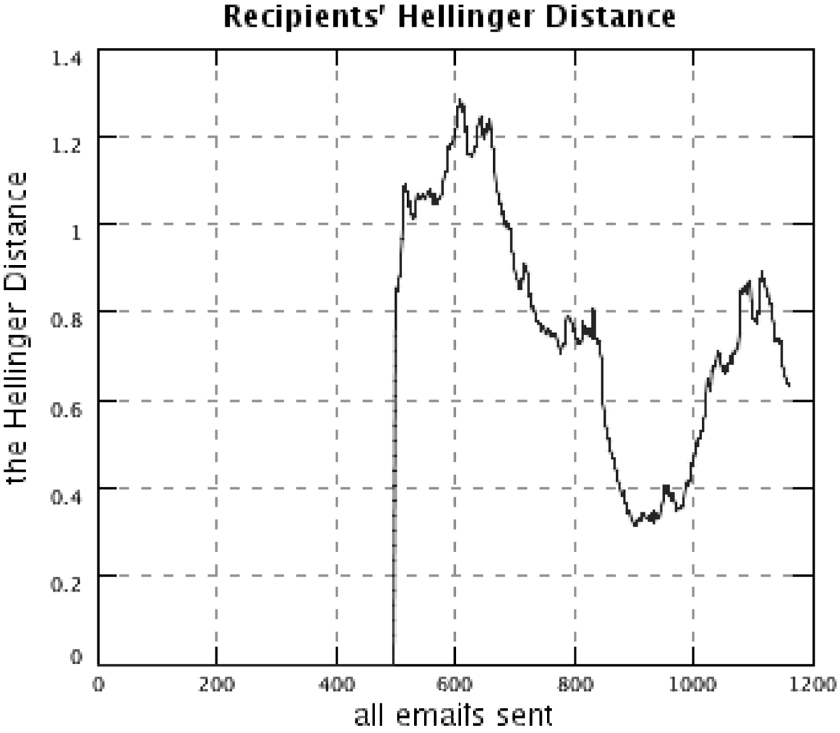


Fig. 7. The Hellinger distance plot of an email user.

tables. We use the *Hellinger distance* for this purpose, as this metric is effective in comparing two probability distributions of frequencies. It is defined as

$$HD(f_p[], f_t[]) = \sum_{i=0}^{n-1} \left(\sqrt{f_p[i]} - \sqrt{f_t[i]} \right)^2,$$

where $f_p[]$ is the array of normalized frequencies for the training set (profiling period), $f_t[]$ is the array of normalized frequencies for the testing set, and n is the total number of distinct recipients/senders observed during both periods. We define the *Hellinger testing window size* as the range of emails that are tested, while the *training range size* is a multiple of that, usually 4. The arrays of frequencies is defined as

$$f_p[i] = N(i)_p / ws_p, \quad \text{and} \quad f_t[i] = N(i)_t / ws_t$$

where ws_p is the Hellinger training window size, and ws_t is the Hellinger testing window size, and $N(i)_p$ and $N(i)_t$ are the number of times that the current recipient (in the case for outbound traffic) or sender (for inbound traffic) of the emails appears in the range ws_p and ws_t for the profiling period p and testing period t, respectively, of emails being evaluated. This is computed for both inbound and outbound email traffic. Figure 7 displays an example for a user from our group of volunteers who provided their email archive.

The Hellinger distance plot shows the distance between training and testing sets plotted over the entire email history of the user. For example, if a user has

2500 outbound records and the window size is 100, the plots starts at the 500th record and measures the distance between the frequencies corresponding to the first 400 records, versus the emails corresponding to the next 100 records; these two windows of 400 (training) and 100 (testing) records, respectively, are then rolled forward over the entire email history of the user, by steps of one record. At each step, a Hellinger distance is calculated between the given training window of 400 records and the corresponding testing window of 100 records.

What this plot tells us is that when a burst in email activity occurs, the recipient frequencies have changed significantly. This statistic provides evidence of either a highly variable user or a possible viral propagation.

4.2.4 *Evaluation Techniques Using Simulated Viruses and Threshold Settings.* As real email data with real embedded viral emails are very difficult to obtain [Schonlau et al. 2001] (and dangerous and possibly illegal to generate), we injected "dummy" viruses into a real email log file as described above. A set of parameters introduced randomness in the process, in order to mimic real conditions and explore boundary conditions: the time at which the virus starts, the number of corrupted emails sent by the virus, and the propagation rate of the virus.

For testing purposes, all the recipients of such "dummy" corrupted records were picked randomly from the address list of a selected user. In reality, where addresses are obtained and how they are combined can be a crucial issue for a virus to successfully propagate itself without being detected. The simulated recipient list of the virus can be set to be all distinct addresses, as most viruses seem to do. But not all viruses would send an email only once to each target recipient account. In our simulation, each "dummy" record contained one attachment, but no information about the attachment was provided or used. (Recall, our focus here was to demonstrate the value of behavior models, as an adjunct to content-based analyses.) For our purposes, we did not need to know the content of the message, its size, and the size and content of the attachments. So these techniques may be general enough that they encompass polymorphic viruses as well (where content analysis or scanners may fail).

The experiments used a combination of three plots: Hellinger distance, "number of distinct recipients," and the "number of attachments." Figure 8 displays plots detailing the profile of one user in our archive. Our intuition was that when a virus is executed, its propagation will cause each plot to increase, that is, it will not "simulate" the user's real frequency distribution. We used a threshold logic to detect "abnormal" growth of these plots.

We used two types of thresholds to determine when a "burst" occurred, a threshold proportional to the standard deviation of the plots, and a threshold based upon the changing "trend" revealed by the Hellinger distance, both conditioned on a window size of prior plotted values. An email was deemed viral if any model deemed it to be viral according to the threshold settings.

We believe it may be necessary to calibrate the threshold settings on a per user basis. For this study, we did not implement specific "user calibration"; rather we aimed at establishing a baseline of performance over all users to reveal whether user specific thresholds might be needed.

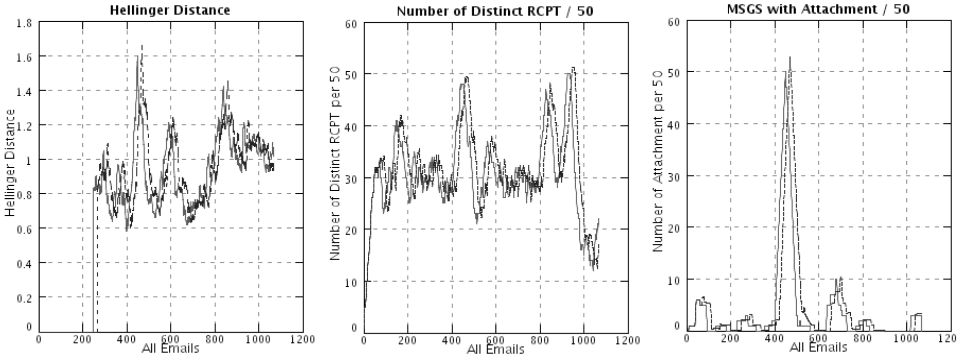


Fig. 8. Three evaluation models. Solid lines are calculated values, dashed lines are thresholds.

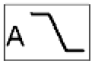
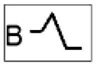
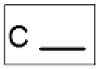
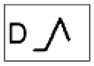
The first threshold was the value of the plot, at some point, adjusting it by a factor proportional to the standard deviation of the average value of the plot calculated from the previous n values of the plot. Thus, the threshold was dynamic, essentially proportional to 1 standard deviation from the mean for the recent user's behavior. The threshold value is defined as $T[i] = (V[j] + \alpha * st(j, j - n))$, where $T[i]$ is the threshold value at location i, $V[j]$is the value of the plots at location j, α is a constant that is set to 0.1, the function *st(a, b)* returns the standard deviation in the range $[a, b]$, n is the window size, and$j = i - shift$. We needed a *shift* value to calculate the threshold using the prior range of data. Without the *shift* value, the threshold would be always higher than the original value since $T[i]$ is always greater than $V[i]$.

The second threshold was developed by observing the trends of the three plotted statistics. When both the "number of distinct recipients" and the "number of attachments" plots grew (the values increased), and when the *slope* of the "Hellinger distance" model grew, this range of emails was marked as suspicious. This means that the "slope" acted as a threshold for the Hellinger distance model, but was used only when both plots (number of recipients and number of emails with attachments) exceeded their threshold. Hellinger distance thus served as a "confirmation" of the other two models.

In Figure 8, the test "dummy" emails simulating viral emails are injected at location 300. In the two rightmost curves, we see a "burst." However, in the Hellinger distance plot on the left, it's not confirmed as a "burst" since in this period of time the user's changing behavior is not abnormal. Recall the Hellinger function. The Hellinger distance expresses the change in the user's behavior. There are four trends that the Hellinger metric may reveal, as shown in Table III. The columns show the user behavior indicating the number of recipients that the user usually sends to.

The curves in Table III graphically represent different user behaviors during periods of time when viral emails may appear. In region A, the user is changing his/her behavior rapidly, and a viral propagation with many recipients would be more difficult to detect. These would be detected toward the end of the period when the slope of the Hellinger plot changes to zero. In region B, when the user's behavior is stable, viral propagations are more noticeable. In region D, a stable

Table III. User Behavior and Hellinger Distance

	Lots of different recipients	Few different recipients
High Hellinger Distance		
Low Helllinger Distance		

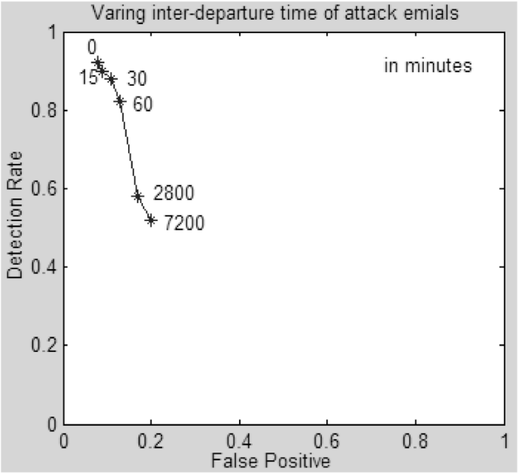


Fig. 9. Varying interdeparture time.

user behavior provides the means of detecting a viral propagation more easily. However, in region C, nothing can be found easily. A stable user who sends to lots of different recipients is the best victim of a virus. This situation means the user's normal behavior is akin to a virus propagation! He or she always sends emails to all the people he or she knows.

4.2.5 *Results of Hellinger Test on Simulated Viruses.* The dataset used for this independent test was an archive of 15 users, totaling 20,301 emails. The parameters that were randomly generated at each simulation were the time of the injected viral emails and the list of recipients (taken from the address list of each selected user). The parameters that were controlled were the propagation rate, the number of corrupted records sent, and the window size (of the Hellinger distance metric). In total, about 500,000 simulations were performed.

As expected, a slower propagation rate (longer interdeparture time) made detection harder. Each email record corresponding to a virus email inserted into the archive became less "noticeable" among the entire email flow. As can be seen in Figure 9, the performance got worse when the interdeparture time increased; that is to say, slow and stealthy propagations are hard to detect.

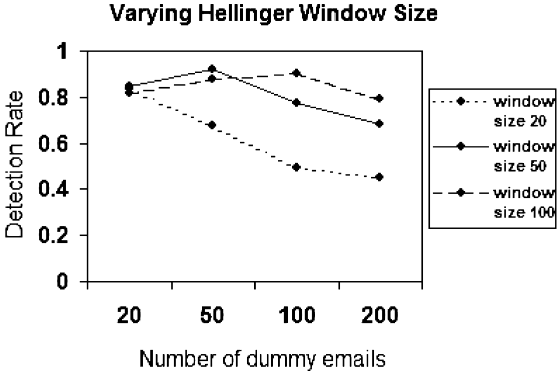


Fig. 10. Increasing Hellinger window-size produces lower detection rates.

The Hellinger window size is the most important parameter. We plot the TP rate as a function of window size in Figure 10 to evaluate the sensitivity of the model to this parameter. In this test, the interdeparture time was 30 min. The performance was best when the window size was the same as the number of dummy records inserted into the archive. The reason is that, for example, when the window size is 50 and there are 20 injected viral records (the number of injected viral records is less than the window size), these records do not occupy a very significant portion of the 50 records. The model may not determine that they are suspicious. On the other hand, if there are 100 injected records and the first 50 are not detected, these 50 dummy records will be treated as normal records in the next round of Hellinger training. As a consequence, the system will likely model the first 50 as normal and not be able to detect any abnormality. (This is a consequence of any "unsupervised" learning-based approach. Without supervised or cleaned training data, whatever may be present in the training data, such as viral emails, could be considered normal and hence becomes part of the normal model.)

In summary, we achieved very reasonable results with the Hellinger distance model. However, there are still three problems. First, we assumed that we had enough normal records before and after the inserted viral emails to develop sufficient statistics modeling the user and detecting the propagation. We also assumed in this simulation that we could analyze all the records (both dummy and normal) at the same time, which was not practical. We cannot block a user's email for a long time, for instance, for a few hours. However, we may store a record of the emails and detect the propagation after the fact, but perhaps still in sufficient time to forewarn the recipients that they likely have received a viral email in their inbox from the recently detected infected victim.

Second, it's difficult to optimize the Hellinger window size, as it depends on the viral strategy used. In practice, we can overcome this by blocking all outgoing emails once we detect a virus. The question then is: how can we detect the first instance of the virus propagation?

Third, the false positive rate was about 15%, which cannot be reduced in this model easily. Thus, to achieve a better detector, this method has to be used in combination with other models. The first two issues will be addressed in the next section.

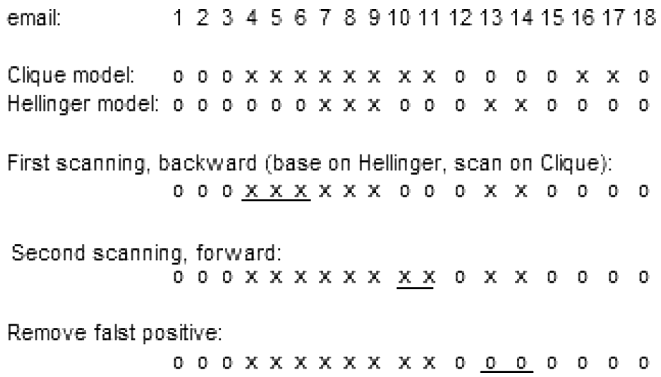
4.3 Combining User Clique and Hellinger Distance

The Hellinger distance model is the result of analyzing the aggregate behavior of a sequence of emails. As such, it would not react immediately when a viral email appeared. Similarly, it would keep setting off alarms for a short while after a batch of viral emails had already been sent out. On the other hand, the user cliques model could detect a suspicious viral email upon its first appearance. It is worth mentioning that every time an email with a new address appears in the user's inbox, the user clique model will treat it as a violation. In short, Hellinger analyzes the trend of users' behavior by analyzing a buffer of email records in terms of their recent behavior, while the user clique method is oriented toward detection of individual viral emails at that moment in time when they are sent or received. Ideally, combining these models may achieve better overall detection performance.

4.3.1 *Backward/Forward Scanning Algorithm.* The intuitive reasoning here is quite simple. When sufficient evidence for a viral propagation has been detected, that is, when an email has an alert generated by both models (clique violation and a substantial change in the user's email emission), it is highly likely that prior and subsequent emails will be part of the virus propagation. We seek to detect these other emails by searching a set of buffered emails (or their record of emission), inspecting the model outputs for each. We search prior emails for evidence of being part of the onset of a viral propagation. This evidence is simply whether *any one of the EMT models has deemed it a violation*. We also search forward in time and test emails until we find an email that violated no model. Intuitively, therefore, the propagation has terminated, or the user has sent legitimate emails during the propagation. We apply this technique for both inbound traffic (the optimal case to prevent infection) and the outbound case when an infection has succeeded but we wish to limit the viral spread as quickly as possible.

The most straightforward method to combine the user clique and Hellinger distance models is to "intersect" their alert outputs. Depending upon the threshold settings, a close examination shows that they have different distributions of false positives. For example, the user clique model may generate false positives on email numbers 1, 3, and 5, while Hellinger may generate false positives on email numbers 2, 4, and 6. If we take the intersection, we can eliminate most false positives. However, a lower false positive rate may be achieved at the expense of a lower TP detection rate.

We propose an alternative strategy we call the *Backward/Forward Scanning algorithm*. Emails are assumed to be buffered before they are actually sent out or, as we mentioned, a record of all sent emails are kept for analysis, including instances of the virus that have escaped without early detection. These records, however, inform us as to where those viral emails were sent so new victims may be warned. This is a key feature introduced in the MET system.



```
email:           1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Clique model:    o o o x x x x x x x  x  o  o  o  o  x  x  o
Hellinger model: o o o o o o x x x o  o  o  x  x  o  o  o  o

First scanning, backward (base on Hellinger, scan on Clique):
                 o o o x x x x x x o  o  o  x  x  o  o  o  o

 Second scanning, forward:
                 o o o x x x x x x x  x  o  x  x  o  o  o  o

Remove falst positive:
                 o o o x x x x x x x  x  o  o  o  o  o  o  o
```

Fig. 11. Depiction of the Backward/Forward Scanning algorithm.

Such rate limiting or buffering of email could be hidden and unknown to the user. Email may be viewed as a *store-and-forward* technology (at least one hop through the server). However, an egress "*store for a while, then forward*" strategy for email delivery has a practical advantage. As far as the user is concerned, the email is sent from client to server and is delivered by the underlying communication system at some arbitrary future time. Thus, the strategy of buffering and holding emails for some period of time allows sufficient statistics to be computed by the models and also benefits mitigation strategies to quarantine viral emails before their delivery, limiting exposure to the enterprise or enclave.

Alternatively, a record of the recently delivered emails may also benefit early detection and mitigation strategies. When the system sees an alert triggered by both the Hellinger distance model and the user clique model, it will examine all adjacent emails more closely, those preceding it and those newly sent by the client. Namely, it will trace (scan) all buffered emails forward and backward (or their record of delivery), starting from the common trigger. The trace attempts to find all sequential emails deemed suspicious by the user clique model and will end once a harmless email, as viewed by user cliques, is encountered. The system then marks all those emails found along the trace as suspicious.

Figure 11 is a graphical view of this Backward/Forward Scanning algorithm. Each email in the sequence is denoted by "x" or "o," depending on whether or not there is an alert associated with it. In this example, we have 18 emails, labeled from 1 to 18. These emails are buffered (stored) and analyzed by both models. The alerts generated by the user clique model are in the first row. The suspicious emails with alerts are 4 to 11, 16, and 17. The alerts generated by the Hellinger distance model are in the second row, and the suspicious emails are 7, 8, 9, 13, and #14.

The algorithm proceeds as follows. In the first step, we find the first alert triggered by both models. In the case shown in Figure 1, email 7 is detected. We then inspect the model outputs for each of the adjacent emails prior to 7 and find 4, 5, and 6 have model outputs that triggered alerts by the user clique model, but not by the Hellinger distance model. We thus generate alerts for each of these as the first set of outputs by this buffer scanning method.

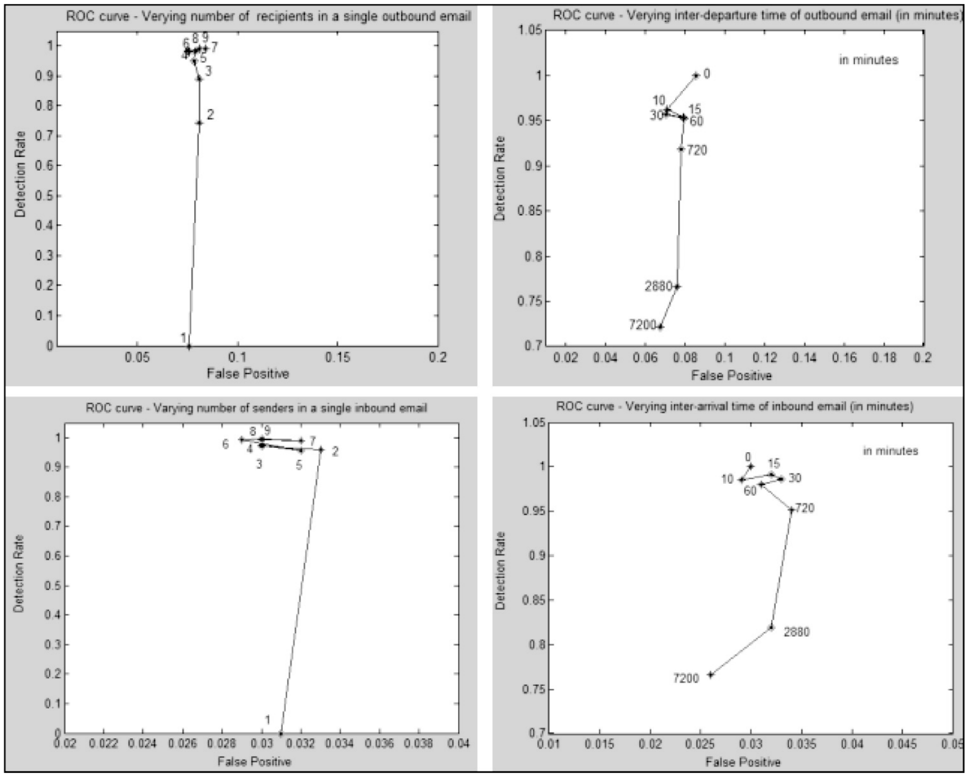


Fig. 12. Average results of all tests. Upper left: outbound test; lower left; Inbound test; both varied the number of recipients of each email. Upper right: outbound test; lower right, Inbound test; varying interdeparture time in minutes.

In the second step, we scan forward, those emails occurring after 7. Here we find emails 8 and 9 have alerts triggered by both models, but 10 does not. We check the model outputs for those immediately following emails occurring after 10 for which one model has generated an alert. Stepping forward, we capture 10 and 11, but terminate at email 12, which has generated no alerts for either model. Finally, we clear all the alerts that are not triggered by both models after 11, deeming these as false positives. We next report results using this strategy.

4.3.2 *Tests of Simulated Virus Propagations.* The dataset for this independent test, both for inbound and outbound emails, included 53, 163 emails from 15 users, and 8% of these emails contained attachments.

We treated the first 80% of the email data as each user's normal behavior (training data) and injected a batch of "dummy records" into the last 20% (testing data). We tested 100 times for each user, generated the dummy emails at a random location (time), and took the average of the results over all runs for the three different users. The summary results of this test are displayed in the ROC curves depicted in Figure 12. (Space limitations do not permit the inclusion of all source data generating these plots.) Naturally, only emails with attachments were tested by the EMT models since only these can contain viruses. Thus, for clarity, the TP rate is the percentage of dummy emails deemed correctly to be viral, and the FP rate is the percentage of "normal" user emails with attachments that were mislabeled by EMT as viral.

During the test phase, we trained and tested the email data on a daily basis. For example, on the first day of the eleventh month, we put all the data (on that day) into the buffer. Then we used the training data to test whether the emails were suspicious depicting abnormal behaviors. We then moved forward on a day-by-day basis for testing. After an individual test day, we assumed the user would review and confirm the normal and viral data. Then we updated the database, which was training data (i.e., we added the normal emails for the next daily test and dropped viral emails).

The parameters that were controlled were the *propagation rate* and the *number of recipients in a single dummy email*. The first parameter was one of the most important issues in the Hellinger simulation. The second parameter was more pertinent to user cliques. Having more recipients in a single email makes it easier for the user clique model to detect a violation. Both inbound and outbound emails were tested. However, we divided the results, because a user's behavior of inbound and outbound emails may be different.

Another important issue was the Hellinger window size. Since it is impossible a priori to choose a single and perfect Hellinger window size in the general case for all users, we changed it by evaluating the size of data (records) each day for each user. The window size is meaningless if it is too small or too large. If it's too small, each email has too much of an influence on the statistics and each may look like a virus. If it's too large, the training data would not be enough to establish sufficient statistics and a small number of virus emails could easily go undetected. In our simulation, we set the window size to the average number of daily emails sent by each user, bounded below by 20 and above by 100. Optimally calibrating this parameter for each user is the subject of ongoing research. We first tested and measured the *outbound email* from a user account to detect the onset of a viral propagation from an early victim. Varying the number of recipients in a single virus email yielded a very interesting result. The upper-left plot of Figure 21 displays the average of the results. The TP rate increased with the size of the recipient list in a simulated viral email, rapidly approaching 100%. This means a virus email is easy to detect if it propagates itself to many email addresses (for example, nine) and sends them in a single email or at the same time. We found that, with just three recipients in a single email, the average TP rate was about 90%. The reason FP hovers around 8% and is almost invariant with respect to virus strategy is rooted in the definition of FP, the number of false positives divided by the total number of nonviral emails with attachments. Only the numerator depends on the properties of the emails being tested by the models. In addition, the alarm was triggered due to clique violations and Hellinger violation. The same false alerts were always triggered, regardless of the viral propagation strategy. We found that each user model exhibited a different TP rate and FP rate.

The first test revealed encouraging results. However, this was because we set a high propagation rate in our simulated "dummy" emails. The interdeparture time used was uniformly distributed between 0 and 10 min in this test. The next test varied this propagation rate.

In this next, independent test, the number of recipients in a single email was set to four. The detection rate got worse when the interdeparture time increased (i.e., the virus stealthily propagated at a very slow rate, see the upper-right plot of Figure 12). If this happens in the real world, once we detect the first virus (with long interdeparture time), we would likely have enough time to mitigate its effects, since it propagates slowly. Thus the issue here again is how best to detect the first virus in a new propagation.

We next consider the results achieved by EMT for *inbound email* the optimal case of preventing viral propagations from entering an enclave or attacking a victim for in the first place. The parameters for the inbound test of EMT were the same as the outbound test. The results are displayed in the lower plots of Figure 12.

Here we aimed to detect an inbound email with an attachment from a sender that is unusual. Notice, in the lower-left plot of Figure 12 inbound emails with a single recipient have a very low TP rate. For the user data on hand, it was *not* unusual that inbound emails had only one recipient. When the number of recipients increased, and clique violations appeared aud the detection rate naturally grew, but with a fairly stable FP rate. Notice too in the lower-right plot of Figure 12 that the rate of receipt of viral emails affected detection performance. Fast arrival times are easy to detect for inbound viral propagations. Slow rates decrease performance markedly. We may mitigate these FP rates in the same fashion as for outbound traffic. By inspecting sequences of inbound emails destined to a user over a period of time, we increase the likelihood of detecting the inbound viral traffic. Moreover, because each user may have a distinct behavior from other users and a user's behavior may change rapidly over time, we can also measure the dynamic activity of users by using statistics capturing cyclic interaction patterns [Davis 2003] to achieve better performance.

4.4 Improving the Detection of Changes in Frequency

The Hellinger distance model performed fairly well, with an impressive TP rate. However, the FP rate may render the approach too frustrating for users who are accustomed to seeing no false alarms generated by their virus scanners. Combining the Hellinger model with the clique violation model improved the results, yet the false positives remained too high. So we explore the addition of a third model, the *cumulative distribution of emitted emails by a user in a sequence of emails*. Here we restrict the statistic to those emails with attachments that appear in the user's archive.

4.4.1 *Cumulative Attachment Distribution.* Suppose we have a period over T days, from day 1, 2, ..., T, and let N_i be the number of emails with attachments on day i and U_i be the cumulative number of emails with attachments until day i. Thus

$$U_i = \sum_{j=1}^{i} N_j.$$

The idea here is that a user will emit emails with attachments at a relatively low and constant (human-oriented) daily rate. If we take a long period of time (for example, 3 months), the slope may be a positive constant. The introduction of a viral propagation would manifest as a significant discontinuity in this plot. Because in the real email data in our database, most users don't send emails with attachments each day, we compute the slope of several days (testing days), and test the slope with a previous longer period (the longer term profiling training days).

Assume we have t testing days from day i to day $i+t$, and r training days from day j to day $j+r$. We detect this discontinuity of the user emission rate behavior over t days by comparing the slopes of the cumulative distribution over r days:

$$\alpha * (U_{i+t} - U_i)/t > (U_{j+r} - U_j)/r,$$

where α is the tolerance parameter for the change of the slope. The threshold α is set by the following intuition. Assume a user's normal behavior of sending emails with attachments will not be more than 1 standard deviation above the mean of the prior d days, that is, let V_i=U_i+ *standard deviation in previous d days*, where V_i is the number of emails with attachments of the user's normal trend on day i. For example, if on average, a user sends one email with an attachment every day, we expect that the user will send one with perhaps a few more email(s) with attachments on the next day, denoted V_i. Then, to compute the boundary of normal behavior, we compute

$$\alpha = ((V_i - U_{i-d})/d)/((U_i - U_{i-d})/d).$$

After computing this ratio for each users' email in our database, α ranged between 1.1 and 1.5. Using this formula, d can be set to any value, of course. In the experiments reported here, d was set to 5 and α was set to 1.2, which we shall see are acceptable values.

If the calculated value of the data violated the inequality in our daily testing (day $i+t$), we said emails on this day were *suspicious*. Then we used this fact to confirm the alerts that might have been generated by the Hellinger and clique models. If either of them had issued alerts on day $i+t$, we verified these alerts.

In Figure 13, the blue (upper) lines are the cumulative number of emails plotted day by day, and the red (lower) lines are likewise the cumulative number of emails with attachments (a strict subset of all emails emitted by the user). The left plot is a user's normal email cumulative distribution. In the right plot, after we added some dummy simulated viral emails, the lower line (closest to the x-axis) displays an apparent discontinuity or a burst identified by the circled area.

4.4.2 *Combing All Three Models.* Next we tested the application of the three combined models—Hellinger distance, violations of the cumulative distribution, and clique violations—using the same Backward/Forward Scanning algorithm described earlier. The dataset for this test was the same used in the previous tests, which included 53,163 emails from 15 users; approximately 8% of these emails contained attachments. The detailed performance results are

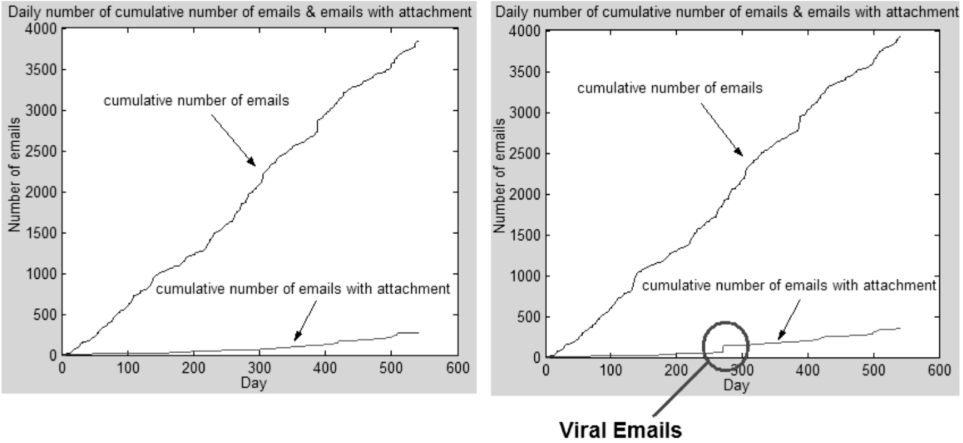


Fig. 13. Cumulative distribution analysis of emails with attachments.

provided in the average ROC plots displayed in Figure 14. The false positive rate has now substantially dropped down to 0.9%. (Note, that the two right-side plots of Figure 14 are scaled by a factor of 10^3). Some false positives remain simply because sometimes users formed new cliques, and these situations caused false positives, which we cannot entirely avoid without modeling the dynamics of clique formation. (That remains as future research, with preliminary results reported in Davis [2003].)

4.5 EMT Examples

Here we exhibit some real examples of the virus simulation test of EMT. Table IV displays a "dummy" injected viral email detected by the EMT models; the generated alerts are labeled "Y," while "N" denotes normal. In the example, we can see that this user communicated with many different organizations. A virus searched the address book, picked up recipients randomly, and sent itself four times. Each of the viral emails included four recipients, and the propagation rate (interdeparture time) was a randomly chosen schedule, from 0 to 30 min. The Hellinger model considers each sender-recipient pair as an individual event. An email sent to four recipients will be four records. The Hellinger model detected the anomalous email record after the second email (fifth record). These emails were all on the same day, and the cumulative attachment distribution indicates an unusual jump in the rate of emission on this day. So all of the emails with attachments for this user on that day had alerts issued. Note, too, that all of the emails violated the user's normal cliques. We can also see that the Backward/Forward Scanning algorithm can recover the first viral email missed by the Hellinger model.

Table V, however, displays a false positive. In this case the user uncharacteristically sent an email with an attachment to different domains never before seen in the user's history. Thus, a clique violation occurred, confirmed by a high rate of email emissions on that day, causing a false alarm..

Virus writers are constantly devising new ways of beating detection algorithms. The behavior-based models presented in this article provide an

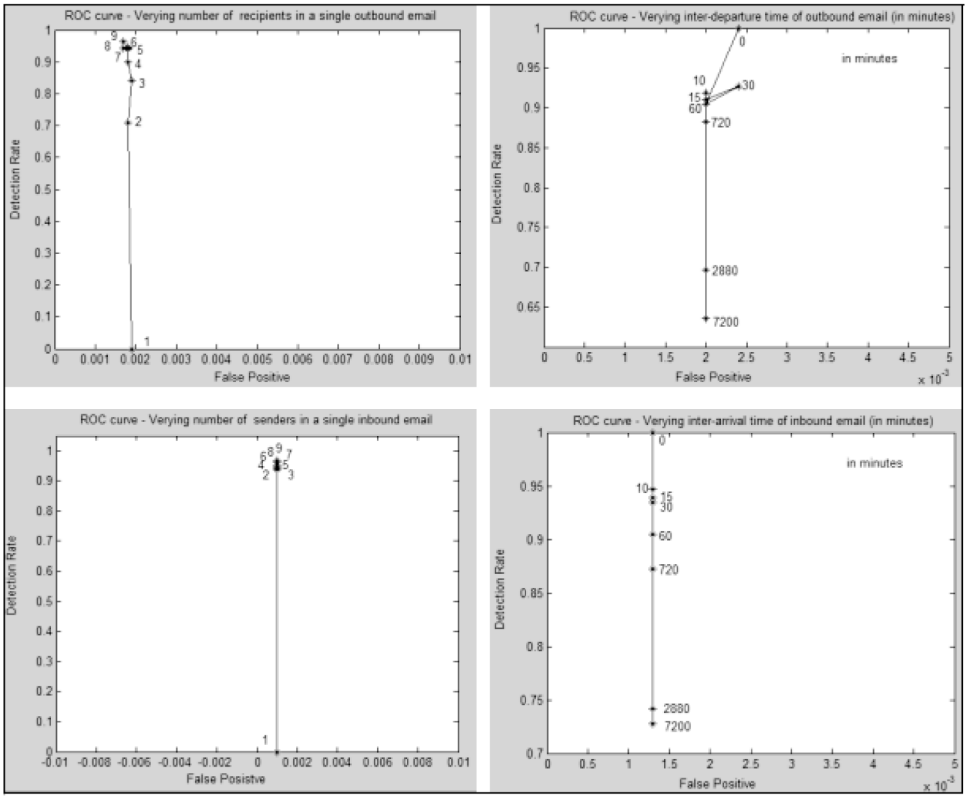


Fig. 14. Average results. Upper left: outbound; lower left: inbound; varying number of recipients in each email. Upper right: outbound; lower right: inbound; varying interarrival time in minutes.

Table IV. Real Injected Viral Emails (He: Hellinger; Cu: cumulative analysis; Cl: clique model.)

	He	Cu	Cl	Combination	MailRef	Time	Recipient
Email 1	N	Y	Y	Y	NA	2002-10-09 08:12:40	B1@baka.org
	N	Y	Y	Y	NA	2002-10-09 08:12:40	CU1@cs.columbia.edu
	N	Y	Y	Y	NA	2002-10-09 08:12:40	P1@pingnet.com
	N	Y	Y	Y	NA	2002-10-09 08:12:40	P2@pingnet.com
Email 2	Y	Y	Y	Y	NA	2002-10-09 08:32:12	A1@allianttech.com
	Y	Y	Y	Y	NA	2002-10-09 08:32:12	CU2@columbia.edu
	Y	Y	Y	Y	NA	2002-10-09 08:32:12	Ta1@tamashunas.com
	Y	Y	Y	Y	NA	2002-10-09 08:32:12	CU3@cs.columbia.edu
Email 3	Y	Y	Y	Y	NA	2002-10-09 08:33:59	CU4@cs.columbia.edu
	Y	Y	Y	Y	NA	2002-10-09 08:33:59	L1@lucent.com
	Y	Y	Y	Y	NA	2002-10-09 08:33:59	N2@nic.com
	Y	Y	Y	Y	NA	2002-10-09 08:33:59	P3@pingnet.com
Email 3	Y	Y	Y	Y	NA	2002-10-09 08:54:41	Ot1@outpost.tanis.org
	Y	Y	Y	Y	NA	2002-10-09 08:54:41	CU5@cs.columbia.edu
	Y	Y	Y	Y	NA	2002-10-09 08:54:41	Acm1@acm.org
	Y	Y	Y	Y	NA	2002-10-09 08:54:41	A2@allianttech.com

Table V. A Real False Positive Email

He	Cu	Cl	Combination	MailRef	Time	Recipient
N	Y	Y	Y	1024036921	2002-06-14 02:42:01.0	Student1@cs.ucsb.edu
N	Y	Y	Y	1024036921	2002-06-14 02:42:01.0	Student2@cs.ucsb.edu
Y	Y	Y	Y	1024036921	2002-06-14 02:42:01.0	Employee1@ibm.com
Y	Y	Y	Y	1024036921	2002-06-14 02:42:01.0	CU1@columbia.edu
Y	Y	Y	Y	1024036921	2002-06-14 02:42:01.0	CU2@columbia.edu
Y	Y	Y	Y	1024036921	2002-06-14 02:42:01.0	CU3@columbia.edu
N	Y	Y	Y	1024036921	2002-06-14 02:42:01.0	CU4@cs.columbia.edu
N	Y	Y	Y	1024036921	2002-06-14 02:42:01.0	User1@icir.org

alternative to content-based detection methods and can serve as a foundation for detecting new viruses. There are numerous avenues of research these techniques suggest. Several are described in our concluding remarks.

5. CONCLUDING REMARKS

We have introduced in this article several email behavior-based methods using principled statistical analysis techniques, and have described how these notions can be used in detecting viral email propagations. These methods complement traditional signature-based approaches to virus detection, and are aimed at detecting new viruses for which signatures have not yet been developed and deployed. This is the maximal period of vulnerability when a new virus does its damage.

The methods described are representative of a range of behavior-based analyses that may be performed for a wide range of general Internet applications. Virus detection provides a good exemplar application since it is familiar to all users of the Internet.

In general, we find that fast and broad-based viral propagations sent to many victims are very easy to detect using behavior-based techniques without content-based analyses. Stealthy and slow-moving propagations remain a challenge. (We hope the next generation of viruses do not implement this tactic, but they probably will.)

In particular, we have defined user cliques and user email frequency behavior profiles. Three specific modeling techniques were combined—user cliques, Hellinger distance, and the daily cumulative distribution of emails—to achieve high detection rates with remarkably good FP rates. Tests on outbound traffic indicate that using EMT's combined models, a high detection rate can be achieved: *95% or more in general cases, with an FP rate ranging from about 0.38% in the best case to as high as 9% in the worst cases of very slow and stealthy propagations.* Tests on inbound traffic show similar results.

The FP rate would translate to a different daily false alarm rate depending upon the user's email emission rate. In the general case, where one user's data exhibited one email per day with attachments, the outbound FP rate of 2% suggests that that user would have received one false alarm every 45 days for the outbound email. The FP rate of 1% of the user's inbound email suggests a false alarm once every 90 days.

There are several areas of new research that can provide substantial improvement in several respects. We chose to use threshold settings for the models based upon a static prior period of time (the window size in Hellinger, and the d days in the cumulative distribution) and did not attempt to incorporate user-specific calibration. As noted in prior work, the choice of such parameters has a very big impact on detection performance (see Tau and Maxion [2002]). Calibrating a detector and setting parameters specific to a user would logically improve individual detection performance for each user. Our current research includes strategies and techniques to best calibrate the detector for each user.

Furthermore, detection in our research so for has been performed on a "per user profile" basis; we do not yet have performance results at the "enclave" level. This is particularly interesting in that shared statistics among a group of users would naturally inform a model more precisely about the onset of a viral propagation within an organization served by a single mail server. Any infected user would naturally propagate to members of his/her own organization (those the user frequently communicates with) and the combined statistics among multiple users would tend to favor early detection for all users. These shared statistics would make a viral propagation appear faster moving than would otherwise be seen by an individual victim. Furthermore, it is sensible that there would be a higher likelihood of detecting clique violations at the enclave level where more email traffic may be inspected.

The models used here for clique violation are not only specific to a user, they are also static. We chose to analyze a user's historical cliques without modeling the dynamics of clique formation and expiration. Clearly, conversations with different groups of folks would tend to be revealed by considering the dynamics of the interactions between the parties to a conversation. This is the subject matter of recent work by Kleinberg [2002], who considered the onset of new "content," rather than new viral propagations, by stochastically modeling the flow of subject lines in email streams. Our current work is focused on modeling dynamic clique formations [Davis 2003], which would logically improve the performance of clique violation models that consider shorter-term statistics.

The focus of this article has been on viral propagation detection as an example of the power of behavior-based computer security. This concept is applicable to a far wider range of problems, including spam detection, security policy violations, and a host of other detection tasks of value to forensic analysis, evidence gathering, and intelligence. It is important to note that testing EMT in a laboratory environment only suggests what its performance may be on specific tasks and source material. The behavior models are naturally specific to a site or particular accounts, and thus performance will vary depending upon the quality of data available for modeling and the parameter settings and thresholds employed.

EMT is designed to be as flexible as possible so an analyst can effectively explore the space of models and parameters appropriate for their mission. An analyst simply has to take it for a test spin. For this reason, we have recently embarked on a collaboration with a police department of a major U.S. city and its criminal intelligence division. EMT will be used by detectives and honed to tasks specific to their needs. We expect in the future to be able to report on the productivity gains EMT may offer when deployed for these law enforcement and intelligence applications.

REFERENCES

AGRAWAL, R., IMIELINSKI, T., AND SWAMI, A. 1993. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on the Management of Data*. pp. 207–216.

APAP, F., ANDREW HONIG, A., SHLOMO HERSHKOP, S., ELEAZAR ESKIN, E., AND STOLFO, S. J. 2002. Detecting malicious software by monitoring anomalous windows registry accesses. In *Proceedings of the Fifth International Symposium on Recent Advances in Intrusion Detection* (RAID-2002, Zurich, Switzerland, Oct.). 16–18.

BHATTACHARYYA, M., HERSHKOP, S., ESKIN, E., AND STOLFO, S. J. 2002. MET: An experimental system for malicious email tracking. In *Proceedings of the 2002 New Security Paradigms Workshop* (NSPW-2002), Virginia Beach, VA, Sept.).

BI, Z., FALOUSTOS, C., AND KORN, F. 2001. The DGX distribution for mining massive, skewed data. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 17–26.

BRON, C. AND KERBOSCH, J. 1973. Finding all cliques of an undirected graph. *Commun. ACM 16*, 9, 575–577.

DAMASHEK, M. 1995. Gauging similarity with n-grams: Language independent categorization of text. In *Science, 267*, 843–848.

DAVIS, P. T. 2003. Finding friends and enemies through the analysis of clique dynamics. Tech. rep., Computer Science Department, Columbia University, New York, NY.

DENNING, D. E. 1987. An intrusion-detection model. *IEEE Trans. Softw. Eng., SE-13*, 222–232.

ESKIN, E. 2000. Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the 17th International Conference on Machine Learning* (ICML-2000).

ESKIN, E., ARNOLD, A., PRERAU, M., PORTNOY, L., AND STOLFO, S. J. 2002. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. *Data Mining for Security Applications.(Jajodia, Barbara, Eds.),* Kluwer, Norwell, MA.

GHOSH, A. K., SCHWARTZBARD, A., AND SCHATZ, M. 1999. Learning Program Behavior Profiles for Intrusion Detection. In *Proceedings of the Workshop Intrusion Detection and Network Monitoring* 1999. 51–62.

HERSHKOP, S., FERSTER, R., BUI, L. H., WANG, K., AND STOLFO, S. J. 2003. Host-based anomaly detection by wrapping file system accesses. Tech. rep. Columbia University, New York, NY. Go online to `http://www.cs.columbia.edu/ids/publications/`.

HOFMEYR, S. A., FORREST, S., AND SOMAYAJI, A. 1998. Intrusion detection using sequences of system calls. *J. Comput. Secur. 6*, 151–180.

HOGG, R. V. AND CRAIG, A. T. 1994. *Introduction to Mathematical Statistics*, Prentice Hall, Englewood Cliffs, N.J., 293–301.

JAVITZ, H. S. AND VALDES, A. 1993. The NIDES Statistical Component: Description and Justification. Tech. rep. SRI International, Menlo Park, CA.

JOHN, G. H. AND LANGLEY, P. 1995. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*. 338–345.

KLEINBERG, J. 2002. Bursty and hierarchical structure in streams. In *Proceedings 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 91–101.

LANE, T. AND BRODLEY, C. E. 1999. Temporal sequence learning and data reduction for anomaly detection. *ACM Trans. Inform. Syst. Secur.*, *2*, 295–331.

LEE, W. AND STOLFO, S. 1999. A framework for constructing features and models for intrusion detection systems. In *Proceedings of the 1999 IEEE Symposium on Computer Security and Privacy* and *Proceedings of the 8th ACM SICKDD International Conference on Knowledge Discovery and Data Mining.*

LEE, W., STOLFO, S., AND CHAN, P. 1997. Learning patterns from Unix process execution traces for intrusion detection. In *Proceedings of the AAAI Workshop: AI Approaches to Fraud Detection and Risk Management* (July).

LEE, W., STOLFO, S., AND MOK, K. 1998. Mining audit data to build intrusion detection models. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining* (KDD '98), New York, NY, Aug.)

LEE, W., STOLFO, S. J., AND MOK, K. 1999. Mining in a data-flow environments: Experiences in intrusion detection. In *Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining* (KDD–99).

LEE, W. AND XIANG, D. 2001. Information-theoretic measures for anomaly detection. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy* (May).

MAHONEY, M. V. AND CHAN, P. K. 2001. Detecting novel attacks by identifying anomalous network packet headers. Tech. rep. Florida Institute of Technology, Melbourne, FL. CS-2001-2.

MITCHELL, T. M. 1997. *Machine Learning*, McGraw-Hill, New York, NY, 180–183.

MYSQL. 2002. Go online to `www.mysql.org`.

NEWMAN, M. E., FORREST, S., AND BALTHRUP, J. 2002. Email networks and the spread of computer viruses. *Phys. Rev. E 66*, 3 (Sept.).

NIBLACK, W., ET AL. 1993. The QBIC project: Querying images by content using color, texture, and shape. In *Proceedings of the SPIE* (Feb.).

SCHONLAU, M., DUMOUCHEL, W., JU, W., KARR, A. F., THEUS, M., AND VARDI, Y. 2001. Computer intrusion detecting masquerades. *Statist. Sci. 16*, 1, 1–17.

SCHULTZ, M. G., ESKIN, E., AND STOLFO, S. J. 2001. Malicious email filter—A UNIX mail filter that detects malicious windows executables. In *Proceedings of USENIX Annual Technical Conference—FREENIX Track* (Boston, MA).

SMITH, J. R. 1997. Integrated spatial and feature image systems: Retrieval, compression and analysis. Ph. D. deissertation. Columbia University, New York, NY.

STOLFO, S. J., HERSHKOP, S., WANG, K., NIMESKERN, D., AND HU, C.-W. 2003. Behavior profiling of email. In *Proceedings of the 1st NSF/NIJ Symposium on Intelligence & Security Informatics* (ISI 2003, Tucson, AZ).

STOLFO, S. J., CHAN, P., AND PRODROMIDIS, A. 1999. Distributed data mining in credit card fraud detection, *IEEE Intell. Syst. 14*, 6, 67–74.

TAN, K. M. C. AND MAXION, R. A. 2002. Why 6? Defining the operational limits of stide, an anomaly-based intrusion detector. In *Proceedings of the IEEE Symposium on Security and Privacy.* IEEE Computer Society Press, Los Alamitos, CA, 188–201.

TAYLOR, C. AND ALVES-FOSS, J. 2001. NATE: Network analysis of anomalous traffic events, a low-cost approach. In *Proceedings of the New Security Paradigms Workshop*. 89–96.

WAGNER, D. AND SOTO, P. 2002. Mimicry attacks on host-based intrusion detection systems. In *Proceedings of the 9th ACM Conference on Computer and Communications Security* (CCS, Washington, DC). 255–264.

WARRENDER, C., FORREST, S., AND PEARLMUTTER, B. 1999. Detecting intrusions using system calls: Alternative data models. In *Proceedings of the IEEE Symposium Security and Privacy.*

WATTS, D. J. 2003. *Six Degrees: The Science of a Connected Age.* W.W. Norton & Company, New York, NY.

WILLIAMSON, M. M. 2002. Throttling viruses: Restricting propagation to defeat malicious mobile code. In *Proceedings of the ACSAC Security Conference* (Las Vegas, NV).

YE, N. 2000. A markov chain model of temporal behavior for anomaly detection. In *Proceedings of the 2000 IEEE Workshop on Information Assurance and Security* (U. S. Military Academy, West Point, NY).

Received October 2003; revised April 2004, July 2004; accepted July 2004