Detection of Anomalous Mailing Behavior Using Novel Data Mining Approaches

Da-Wei Lin, Yi-Ming Chen
Department of Information Management,
National Central University, Chung-li, Taiwan 32054, Republic of China
s4364007@cc.ncu.edu.tw, cym@cc.ncu.edu.tw

Abstract

The paper presents a novel method for detecting anomalous mailing behavior based on data mining approaches. Known or unknown email viruses may cause anomalous behaviors. Such behavior can be measured by deviations from a user's normal behavior. Grouping and association analysis are used to establish a normal user profile. The building process is divided into two stages - first, group relation analysis and second, dependence relation analysis. Only group relationship analysis or both analyses may be selected, depending on the amount of data available to solve real problems.

Bulk amounts of SENDMAIL log data are analyzed and virus behavior simulated. Empirical results indicate that this method of detecting anomalous mailing behavior, based on data mining, is highly accurate. A prototype system has also been designed and constructed.

Keywords: anomalous behavior detection、mailing behavior、data mining、grouping

1. Introduction

Email viruses such as Melissa [11], Loveletter [8], Sircam [13] and Nimda [9], have posed great threats for businesses and individuals in recent years. Such viruses are distributed via the Internet as email attachments. Some viruses, such as the Nimda virus, use multiple propagation channels to distribute themselves through the Internet. Some viruses use a MIME bug in Microsoft software [2], enabling the system of a user who only previews the mail without opening the attachment to become infected by the virus. An email virus typically attaches itself as an attachment to be distributed to others, and waits to be opened by another user before distributing itself again. Email viruses can disseminate widely and quickly over the Internet in a short period. Previously, viruses spread slowly and locally, leaving sufficient time for them to be counteracted. However, email viruses now spread quickly and become destructive in a short period. Traditional anti-virus response mechanisms cannot combat this situation [34].

Most current anti-virus mechanisms use virus pattern matching to detect viruses [6][10][22]. They are limited in that they can only detect those viruses for which patterns have been collected. Namely, without a specific virus pattern, a new virus can be neither detected nor removed. Viruses and their variants may have different virus patterns. Virus generators [10] can be used to generate various viruses easily. Some new virus writers use polymorphous techniques [6]. Detecting all viruses is very difficult. Later detection implies a greater loss.

Updating a virus pattern code too slowly risks the loss of something. Users must thus react to viruses. A proactive method or mechanism must be developed to tackle the problem caused by known and unknown viruses.

Most virus-related research focuses directly on virus files, investigating specific virus patterns in the files, the structure of the virus file and the behavior of the virus, which are all related at a low level to operating system and hardware architecture. These considerations involve different characteristics in different environments. This study considers the detection of abnormal mailing behavior caused by viruses at the user-level. Each mailer's recipients are assumed to be related. Some people will always occur together as recipients of mails sent by a given mailer. Exactly how recipients of mails are related can be determined from historical mailing behavior. User-level behavior is a high-level view, making it uniform across all environments.

Mail delivery involves two information types when the SMTP protocol is used - protocol information and the mail message itself [26]. The protocol information is used to ensure mail delivery. The mail client must communicate protocol information to the mail server. The protocol information contains the mail sender, the recipients and other control information. The protocol information is invisible for destination recipients. The mail message [12] refers to mail sent to a destination, and is visible for destined recipients. The mail message contains the following information about the sender, recipients, subject, mail content and attachments. Additionally, the protocol information and the mail message contain information concerning the sender and recipients of the mail. The protocol information is verified, but the mail message (referred to as data) is sent through the local mail server without verification. A malicious program may vary the content of a mail message to evade detection by anti-virus software without affecting mail delivery. Using the content of a mail message to detect email viruses is extremely difficult.

Our previous work ever adopted the frequent itemsets method [1] to identify recipients who always receive mails simultaneously. For some mailers, although the itemsets are found, most mailers cannot be effectively located with minimum support constraint because mailing behavior is characterized by a large variance over time. The frequent itemsets having been identified only contain a small fraction of the recipients, thus reducing the likelihood of failure to judge the mailing behavior when the itemsets do not contain the recipients. The generation of frequent itemsets uses the intersection concept, but mailing behavior varies over time. Notably, minimum support can be reduced to locate many itemsets to encompass more recipients. However, doing so could cause many false alarms when attempting to detect abnormal mailing behavior. Therefore, the frequent itemsets mining method is inadequate for the mailing problem.

This paper presents a simple but effective method to detect anomalous email behavior caused by email viruses. Data can be reasonably explained concerning user-behavior. A profile of the user's normal behavior is established from his historical behavior. New behavior that deviates from this profile is considered to be abnormal. The user profile is built in two

stages. The first stage is referred to herein as group analysis, and builds user group relationship information. The second stage is referred to herein as dependence analysis, and determines further dependency relationships within a single group. Group relationship analysis provides a tolerance criterion, while the dependent relationship analysis provides a strict criterion. The order of the analysis must first determine the group relation, then the dependent relation. According to the sufficiency of the mailing data, only group relationship analysis or both analyses are applied.

Bulk amounts of SENDMAIL log data are analyzed and virus behavior simulated. Empirical results indicate that this method of detecting anomalous mailing behavior, based on data mining, is highly accurate.

The rest of the paper is organized as follows. Section 2 reviews related research; Section 3 elucidates the proposed method; Section 4 discusses the data analysis and virus simulation test; Section 5 describes the system design and implementation, and finally, conclusions are drawn and directions of future work given.

2. Related research

In this section we review research related viruses. Professionals and scholars have used the following methods to detect viruses:

Charlier [6] used general behavior patterns to detect computer viruses dynamically. His research relies on the assumption that viruses must modify host files to be distributed. This assumption is occasionally ineffective in detecting email viruses. Email viruses are mailed to recipients as attachments and directly distributed via mail channels after the attachment is opened. The viruses do not necessarily affect the host files. This method is based on DOS COM files, and its applicability to other viruses such as Macro virus remains doubtful.

Tesauro's research based on neural network applied only to boot viruses [31], has not been tested on other types of viruses such as macro viruses. CPU constraints, limitation of memory and disk space and the simplification of neural network reduce the detection rate and speed.

IBM and Symantec have designed a commercial grade system known as Digital Immune System [32][35], capable of detecting viruses, analyzing them and automatically creating a cure for unknown viruses. The system adopts a new technology, called Bloodhound [33], to detect various viruses. Generally, heuristic methods detect viruses by analyzing a program's structure, its behavior, and other attributes, rather than by searching for a signature. Better heuristic detection supports more effective combating of new viruses. The method is more intelligent than pattern matching only. The details of the technique are unknown since it is a commercial product. Heuristics imply can be used to detect unknown viruses but exist with some false alarms [9][34].

The Malicious Email Filter project applied data mining methods to construct detection models over known malicious executables [25]. These models can identify unknown malicious attachments, by assuming the presence of similar byte sequences in malicious

executables that differentiate them from benign programs. The method analyzes virus files and then extracts relevant byte sequences. It is relatively effective for detecting unknown malicious programs.

With reference to intrusion detection, most above research is primarily based on misuse detection. A few methods combine pattern matching with heuristics to enhance detection of unknown viruses. Other research belongs involves anomaly detection which may not be directly related to, but supports virus detection.

NIDES [19] and Emerald [24] applied a statistical approach to detect anomalies and a rule-based approach to detect known intrusions. The profile records only simple statistics such as frequency, mean and covariance. These statistical measures are designed for general applications. When these statistics are applied to a real environment, a system administrator must decide what measures and what thresholds are required to meet specific requirements.

Forrest [14] identified an important property: process system calls exhibit stable short-range sequences. This property can be used to build a definition of self that can be differentiated from non-self. The system call sequences are a continuous data stream that must be partitioned into fixed-length data patterns for subsequent calculation. The sequence of system calls is critical for determining abnormal behavior. With respect to the mailing behavior problem, only the relationship among recipients is considered here. The order of the recipients is irrelevant. Only the identities of the recipients present together in a mailing event are of interest. A mailing behavior is event-based and essentially variable in length. The characteristics of mailing problems differ from those of system calls. Forest's aimed to detect server anomalies using system call sequences, whereas this study investigates abnormal mailing behavior caused by a mail client, not by a server. Both targets are different.

Lee [23] adopted another approach to solve sequence system call problems. He used data mining to solve sequence problems concerning a system call. A data stream is partitioned into fixed-length patterns as in Forrest. Lee uses a rule-based reduction program to determine the rules embedded in the fixed-length patterns. Lee's method can effectively reduce the amount of data, but data characteristics of the problem are the same as in Forrest.

Ishibashi et al. [18] proposed a DNS traffic mining method to detect a mass-mailing worm. This works because before a virus sends mails to the targets, queries are sent to local DNS servers to find the appropriate mail server for the specified mail address. By using DNS traffic data with "a priori" knowledge about the signature query, they can detect a mass-mailing worm. The work is in progress and only preliminary results are so far available.

Gupta et al. [15] use a statistical anomaly detection to detect email viruses. Their approach looks for increases in mail traffic from clients to mail servers that exceed a threshold determined during a training period. Specifically, the statistics regarding send and deliver transitions in a state machine are maintained for both individual clients and the entire collection of clients within the network. In a series of simulation experiments they were able to detect stealthy viruses (e.g. polymorphic ones) with a low false positive rate.

Whyte et al. [36] proposed a method to detect malicious SMTP-based mass mailing

activity within a network. Contrary to other mass mailing detection techniques, their method relies strictly on the observation of DNS MX queries within the network. It is content independent and does not deal with the attachment. Some researchers, such as Ishibashi et al. [23], argue that the detection method is applicable in only some environments, however, it provides another approach to resolving mail virus problems.

Stolfo et al. [28] used a behavior-based anomaly detection method to detect mailing behavior violations. They developed two behavior-based models; user cliques and enclave cliques. The user cliques model profiles a user's naturally occurring communication groups, such as with colleagues, family members etc; user cliques can be inferred by looking at the email history of only a single user account, while enclave cliques consist of social groups that emerge as result of analyzing traffic flows among a group of user accounts. The recipients listed in a single email form a set referred to as a user clique. The sets are summarized by using the set operation "contain". Any subsets which are subsets of other sets or repeat sets are removed, as they do not contribute any more useful information. The final user clique is a set which cannot be subsumed by another set.

Their approach is very similar to ours. Both our research and theirs are based on the same data source: the recipient list of "history" mails for a given mailer. In our research, a user group relationship is constructed by using the union operation detailed in section 3. The groups constructed on our method are mutually exclusive of each other, while their groups are not. The number of sets constructed based on our method is less than theirs. Our processing stage is 2. Stage 1 is the group analysis stage, and stage 2 is the dependence analysis stage. If a mailing event is judged anomalous in stage 1, the stage 2 validation process is not needed. The measuring of the size of user cliques or groups based on each method shows that our method should has better performance than theirs.

In research [6][31][25][14][23] that considers the virus file itself (low-level program behavior) the characteristics are closely related to the operating system and the hardware architecture. They differ among systems. Some research [18][36] has relied on the observation of the DNS MX query which is a signal of malicious mass mailing. Like Stolfo et al. [28], in our approach we consider user behavior, taking a high level view, where meaning can be easily explained. In most of the above investigations anomalous behavior is determined from several records. A major advantage of the proposed method is that an anomaly can be judged on a per event basis, which makes it very useful, allowing anomalous behavior to be detected as soon as it occurs, so that some countermeasures against this anomalous behavior can be taken in real time. However, care must be taken not to generate too many false alarms, by the misinterpretation of normal behavior as anomalous. It is difficult to compare that the method presented here to the previous research, because some preconditions or parameters that must be considered are different for different approaches; no equivalent or statistically comparable datasets, to which all these techniques can be applied, exist. However, in Section 4 we compare the proposed method with a statistical method as a base test. In the following section the new method is explained.

3. Building a User Profile

This section explains the proposed approach to detecting anomalous mailing behaviors.

The proposed grouping method differs somewhat from clustering as used in data mining. Clustering in data mining groups objects into clusters according to attributes or elements [16], but the proposed method examines the elements of the object, and determines relationships among the elements of a group. Elements, rather than objects themselves, are considered. A mailing event is an object and the recipients of the mailing are elements. Accordingly, the group relation analysis focuses on the recipients, not on the events.

The following presents an example first and then defines some basic terms to illustrate the method.

David sends mail to four recipients, A, B, C and D, on Aug 20, 13:30:30.

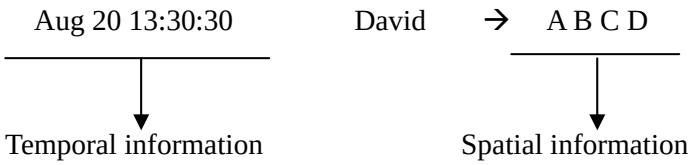

Figure 1 Mailing event

The mailing contains two kinds of information; temporal information (Aug 20, 13:30:30) and spatial information (A B C D).

Mailing event: the mailer generates a mail at a particular time. The mailing event includes information on mailing time, the sender and the receiver. Here, only information on the receiver is considered. The receiver may be a single or a few recipients. Mailing to multiple recipients differs from mailing to a single recipient many times but at different times. The former is a single event; the latter represents multiple events.

Mailing behavior pattern (the recipient list) is spatial information. Four recipients, A, B, C and D, specify a mailing behavior pattern. A mailing pattern is a collection of recipients extracted from a mailing event. Only the recipients are considered here. When “recipients” are referred to, a given mailer is assumed.

The proposed approach to constructing a user profile is described below. A user behavior profile is constructed in two stages - one for group analysis, the other for dependence analysis. The former determines which “group” relationship exists among all recipients for a given mailer, according to which recipients appear together in a mailing event of a given mailer. Dependence analysis investigates any dependency relationship within the same group, after group analysis is performed. Figure 2 depicts the flow for building a user profile.

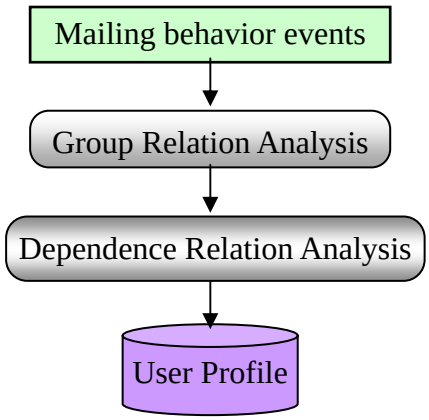

Figure 2 Flow for building a user profile

The two analyses are detailed below.

3.1 Group relationship analysis

What are the characteristics of a group? How a group relationship be established? These questions are formally answered below.

Let r be one recipient and let R = $\{r_1, r_2, \ldots, r_m\}$ be a set of all recipients for a given mailer. Let e be a set of some recipients and represent a mailing pattern. Where E = $\{e_1, e_2, \ldots, e_n\}$ is a set of all mailing patterns extracted from mailing events for a given mailer over a period. Let g be a set of some recipients similar to e, but represent one group and let G = $\{g_1, g_2, \ldots, g_k\}$ be a set of groups. Notably, the set of groups, G, can be derived using the later algorithm. The set G has the following two basic properties, universal and exclusive property, and group relationship property that can be used to validate mailing behavior.

- Universal Property (1)
  $\cup g = \cup e = R$
- Exclusive Property (2)
  $g_i \cap g_j = \varnothing \ \forall \ i \neq j$
- Group Relationship Property (3)
  $\forall \ e \in E \ \exists$ only one $g \Rightarrow e \subset g$

The universal property states that the final set of groups, G, includes all the recipients in the set of original mailing patterns, E. The exclusive property states that no group intersects another. Restated, any recipient exists only in a single group. The group relationship property is the most important when judging mailing behavior. Semantically, recipients of a mailing pattern are contained only in a single group.

The algorithm for building a set of group relations, G, derived from the set of historical mailing patterns, E, is described below.

```
                // E represents a set of mailing patterns for a given mailer
Let G = ∅       // G is a set and stores the group relation
While ( E ≠ ∅ )
begin
    Choose any pattern e from E
    Let g = e and drop e from E
    Repeat the following operation ∀ e_j ≠ e
    begin
      If ( g ∩ e_j ≠ ∅ )
      begin
        g = g ∪ e_j             // union operation
        drop e_j from E         // remove e_j from set E
      end
    end
    Add g as one element of set G
end
```

After set E becomes ∅, G is obtained. Assume set G has k elements, such that G ={g_1, g_2, … , g_k}. The final result of the operation is unique for given mailing behavior events. It is independent of the operational sequence of the mailing patterns in set E.

The above properties are proven as follows.

Proof of Universal Property

All recipients in set G originated in the mailing patterns in set E without loss any recipient, so all the recipients in all groups are the same as those in the original mailing patterns in set E.

Proof of Exclusive Property

Suppose there exist two groups, g_i, $g_j \in G$ and $g_i \cap g_j \neq \emptyset$. Then, there exist e_i, $e_j \subseteq E$, $e_i \subseteq g_i$, $e_j \subseteq g_j$ and $e_i \cap e_j \neq \emptyset$. Since $e_i \cap e_j \neq \emptyset$, e_i and e_j should both be members of a single group and not to different groups, which would violate the building algorithm of set G.

Proof of Group Relationship

$\forall$ e $\subseteq$ E, there exists one g $\in$ G, such that e $\subseteq$ g. Suppose there exists another group $g_i \in$ G, $\Rightarrow$ e $\subseteq g_i$. Then both g and g_i have a common element, e, violating the exclusive property. Eventually, the important property “A mailing pattern is contained only in a single group” is obtained.

An example is now presented and the group relation building procedure explained. Table 1 presents several mailing patterns for a given mailer. One mailing pattern is extracted from one mailing event. Field one is the mailing event ID, field two presents the mailing pattern (recipients), and A, B, C and D … represent the various recipients.

Table 1 Mailing events for a given mailer

Mailing event ID	Mailing Pattern(recipients)
0	[A]
1	[B]
2	[B C]
3	[D]
4	[E F]
5	[F]
6	[B G]
7	[B G]
8	[H]

Table 2 presents the result of group analysis.

Table 2 Groups of recipients of mails sent by a given mailer

Group ID	Group Pattern	Contributed by mailing event ID
I	[A]	0
II	[B C G]	1,2,6,7
III	[D]	3
IV	[E F]	4,5
V	[H]	8

Mailing event ID 0 constitutes group I; mailing event IDs 1,2,6 and 7 constitute group II; mailing event 3 constitutes group III, mail event IDs 4 and 5 constitute group IV and mailing event 8 constitutes group V. All recipients in Table 1 are also present in Table 2. No recipients are shown twice in Table 2. The crucial property that a behavior pattern in Table 1 applies only to one group in Table 2 is confirmed. Table 3 presents the correspondence between mailing patterns and groups.

Table 3 The correspondence between mailing events and groups for a given mailer

Mailing event ID	Mailing Pattern (recipients)	Group ID
0	[A]	I
1	[B]	II
2	[B C]	II
3	[D]	III
4	[E F]	IV
5	[F]	IV
6	[B G]	II
7	[B G]	II
8	[H]	V

3.2 Dependence relationship analysis

In the previous stage, the group relations were established. The group relation remains imprecise. The group relations obtained by merging the recipients of related mailing events

encompass some mailing behaviors that have not yet been observed. This section considers relations within a group, to reduce inaccuracy.

The dependence relationship is concisely described here. Suppose both A and B are recipients of E and belong to the same group g, in the above group analysis. If recipient A is involved in a mailing pattern e, then recipient B is also involved in the same mailing pattern e. Recipient A can be said to depend on recipients B, the following notation can be used to represent the dependency, A→B.

The concept behind the dependence relationship is the same as that behind the association rule in data mining [1][27]. Generally, the association rule is applied to a large volume of market data. Two constraints are imposed on association rule-mining problem - confidence and support. While confidence measures the strength of a rule, support specifies its statistical significance. The association rule is considered significant above a minimum support. However, the volume of mailing data is not as great as that of market data, so low support is acceptable. A high confidence level is considered here. The threshold confidence is set to 100% in order to prevent incorrect dependence rules that may cause false alarms from being generated.

Following the previous example, group II [B C G] in Table 2 is comprised of mailing events 1,2,6 and 7 from Table 1, and group ID II [E F] consists of mailing events 4 and 5 from Table 1. Dependence analysis focuses on multiple elements. Other groups in Table 2 have a single element. Single element groups are not considered.

From mailing events 1,2,6, and 7, the following mailing pattern is obtained.

B

B, C

B, G

B, G

The dependence rules C→B and G→B are determined: when a mailer sends mail to C, mail is also sent to B, but when a mailer sends mail to B, it need not also be sent to C. The dependence is not symmetric.

The following result was obtained using two-stage analysis, including group analysis and dependence analysis. This table presents the user profile for a given mailer.

Table 4 Mailing behavior profile for a given mailer

Group ID	Group relation	Dependence rule in the same group
I	[A]	
II	[B C G]	C→B, G→B
III	[D]	
IV	[E F]	E→F, F→E
V	[H]	

3.3 Judging abnormal mailing behavior

The validity of a mailing behavior is judged according to two rules:

Rule 1:

All recipients of one mailing pattern in mailing event should belong a single group; otherwise the behavior is anomalous.

Rule 2:

The relation among recipients in a mailing pattern should satisfy the dependence relationship; otherwise the behavior is anomalous.

Any mailing event in which the mailing pattern violates the above rules is regarded as anomalous. When any mailing behavior event violates the group relationship in the first stage, the dependence rule need not be checked in the second stage is not need, accelerating the processing of data.

3.4 Quickly building an effective user profile

Two problems must be addressed, when building a user profile. First, a user's mailing behaviors gradually change over time and thus will gradually deviate from the profile. The profile must be rebuilt after some period. Every time a profile is rebuilt, some old mailing data must be wastefully processed again. Second, old mailing behaviors that never recur disturb the effectiveness of the user profile over the long-run.

Hence, the concepts of the table of behavior pattern statistics and the cutting-time are introduced. These two concepts are useful in quickly rebuilding an effective user profile. The table of behavior pattern statistics helps to rebuild a user profile quickly. The cutting-time setting can help us to effectively filter old mailing behavior that never recurs over a long period.

Figure 3 depicts the flow of processing mailing events, according to the table of mailing behavior statistics and the cutting-time, which are detailed below.

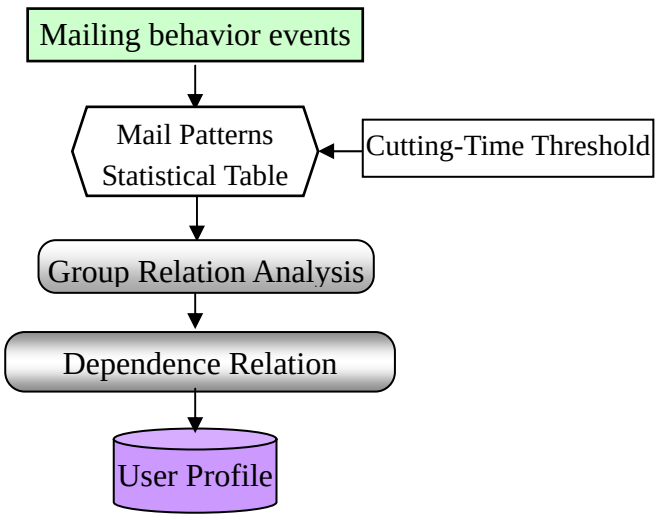

Figure 3 The flow of processing mailing behavior with table of mailing behavior statistics and the cutting-time

3.4.1 Table of behavior patterns statistics

Using the table of mailing behavior statistics as an intermediate step can help us to rebuild user profile quickly. The table records all the characteristics of mailing patterns that have occurred. A sample table of user behavior pattern statistics is presented and its construction is elucidated. The table includes four fields - field one, PID, is a unique ID number; field two, COUNT, counts the number of times that the mailing pattern has occurred until now; field three, LATEST, records the time that the pattern most recently occurred; field four, Behavior Pattern, records the actual behavior pattern.

Table 5 The table of mailing behavior patterns statistics

PID	COUNT	LATEST	Mailing Pattern
0	3	620	[A]
1	122	579	[B]
2	1	11	[C]
3	1	12	[D]
4	22	544	[E]
5	1	18	[F]
6	1	24	[B E]
7	5	564	[G]
8	14	90	[H]
9	14	559	[I]
10	6	182	[B I]

In Table 5, every mailing pattern is unique; that is, each pattern is only recorded once. For example, the mailing pattern [B I], whose PID is 10, is shown once. It occurred six times and most recently at 182 time units.

The algorithm for constructing the table is as follows:

```
While ( a new mailing event occurs )
begin
  if ( mailing pattern of new mailing event already exists in the table )
        add 1 to the COUNT field of that record
        refresh the time of LATEST field of the record
  else
        add the new mailing pattern as one new record to the table
        assign 1 to the COUNT field,
        assign time it occurred to the LATEST field
endwhile
```

The user profile can be quickly and easily derived using this table.

3.4.2 Cutting-Time

Old mailing behaviors that may never recur must be considered. Old mailing behaviors involved in the user profile may reduce the effectiveness of the user profile. The old mailing behaviors must be filtered out, using the cutting-time. Any mailing events older than the cutting-time threshold are discarded. Only mailing events that occurred after the cutting-time threshold are included in the profile.

In the following example, the cutting-time threshold is set to 100, and records for which the field, LATEST, includes a value smaller than 100, are discarded. The following table is obtained. The size of the table is effectively reduced. Section 4 discusses the effect of the cutting-time threshold. A feasible cutting-time threshold can be determined through testing, as considered in Section 4.

Table 6 the table of mailing pattern statistics with cutting-time threshold 100

PID	COUNT	LATEST	Mailing Pattern
0	3	620	[A]
1	122	579	[B]
2	22	544	[E]
3	5	564	[G]
4	14	559	[I]
5	6	182	[B I]

In summary, using a behavior patterns statistics table in an intermediate step enables the user behavior profile to be rebuilt quickly and easily. The cutting-time threshold helps to filter out older mailing behaviors that do not recur.

4. Experiment

This section verifies the effectiveness of the proposed method through data analysis. The experiment contains two parts: one to examine the profile and the mailing data, the other to determine the ability of the profile to differentiate normal behavior from viruses. Figure 4 shows the data-processing flow. The data preprocessing stage prepares mailing event data of each mailer. The user profile is built using the proposed method, group and dependence relation analysis. Two things must be considered. First, the profile must be built using clean data to avoid bias. Before the profile is built, the outlier detection method is applied to exclude mailers who generated suspicious mailing events. After the profile is built, the profile must be confirmed to encompass most user behavior. If the profile cannot encompass most behavior, it will cause many false positive alarms. The validation of the profile’s coverage does it. Eventually, related tests and simulations can be performed.

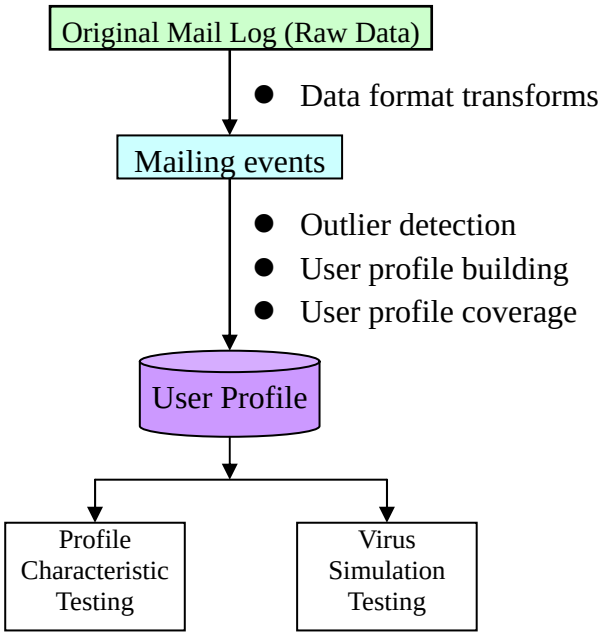

Figure 4 Data processing flow

The mail server software SENDMAIL used in the UNIX system at the NCU Computing Center generated original mail log data. Data concerning thousands of mailers were included, covering the period from 2001.02 to 2002.05. The volume of data was approximately 2.5GB. The mail log data records only the mail envelope information, and not the content of the mail. The mail envelope information includes the time when the mail was sent, the sender and the recipients. Information such as the sender's email address and the recipients' email addresses was transformed using an MD5 algorithm, as the log data obtained.

Data preprocessing

All mailers' data are mixed together in the original log files and one mailing event spans several records in the log file. In the data preprocessing stage, mailing event must be reconstructing by event for each mailer. Raw data (SENDMAIL log) are formatted into mailing events by mailer. Thereafter, the large log file is split into several files that each contains many mailing events for each mailer. The user profile can be built according to these personal mailing patterns files.

Outlier detection

The following guide was used to determine whether data were clean.

For a normal distribution, outliers can be considered to be observations that lie three or more standard deviations (i.e., ≥ 3) from the mean μ [4][21].

The variable of interest, in the mailing problem, is the number of recipients of a mailing event. The number of recipients becomes more suspicious as it deviates from the mean. For a given mailer, the mean and standard deviation of the number of recipients can be determined from the historical mailing events. If the number of recipients of a mailing event is further than three standard deviations from the mean, then the mailing event is considered abnormal.

Users' mailing behavior may not conform to the normal distribution completely; different users may have different distribution and generalizing a particular distribution to all mailers is difficult. Hence, the processing is simplified using the above guide. In this experiment, mailers whose mailing events are complete without outliers are chosen.

A user profile must encompass as much behavior as possible. When the mailing data cannot cover all of a user's behavior, a bias user profile is generated, and may generate many future false positive alarms. The coverage of a user profile is validated to check if user profile can cover a user's historical behavior. This also can be as a false positive validation. When the false positive alarm rate is low, it means user profile can effectively present the mailer's behavior. The following method is used to verify the profile. The mailing data is separated into the training data and the testing data. A user profile must be generated from training data and validated by testing data to confirm the completeness of the user profile. In this analysis, the ratio of the amount of training data to the amount of testing data is 5:1. The following modulo equation is used to divide the mailing data into two parts - RECORD_NO modulo 6. Applying the equation to divide the data yields training data and testing data that distribute over the same period, avoiding the effect of different periods. The threshold false positive rate can be determined. If the test is acceptable, training data and testing data are combined to build a profile again [20].

In the following, two kinds of analysis are performed. One analyzes user profile, examining the characteristics between the user profile and the mailing events. These include different period tests, cutting-time thresholds tests and sliding window tests. The other analysis simulates an email virus behavior, which checks the ability of a user profile to identify virus behavior.

4.1 Test over different periods

In this part of the analysis, the false alarm rate is performed for different test data periods. This and the following part of the analysis rely on more mailing events, so mailers with no less than 500 mailing events are used. Finally, 185 mail senders were considered. From the mailing data, the first 500 mailing events are assigned as training data, of which 50 consecutive mailing events are used as Testing Data Set 1(TEST1), 100 consecutive mailing events from are used as Testing Data Set 2(TEST2), 150 consecutive mailing events are used Testing Data Set 3(TEST3) and 200 consecutive mailing events are used Testing Data Set 4(TEST4). The training data and the testing data cover different time intervals and do not overlap. These testing data sets cover periods of different duration. Table 7 provides details concerning the training and testing data. Figure 8 presents the results.

Table 7 The distribution of the training and testing data

	Training Data	TEST1	TEST2	TEST3	TEST4
Record Range	1~300	301~350	301~400	301~450	301~500
Record Counts	300	50	100	150	200

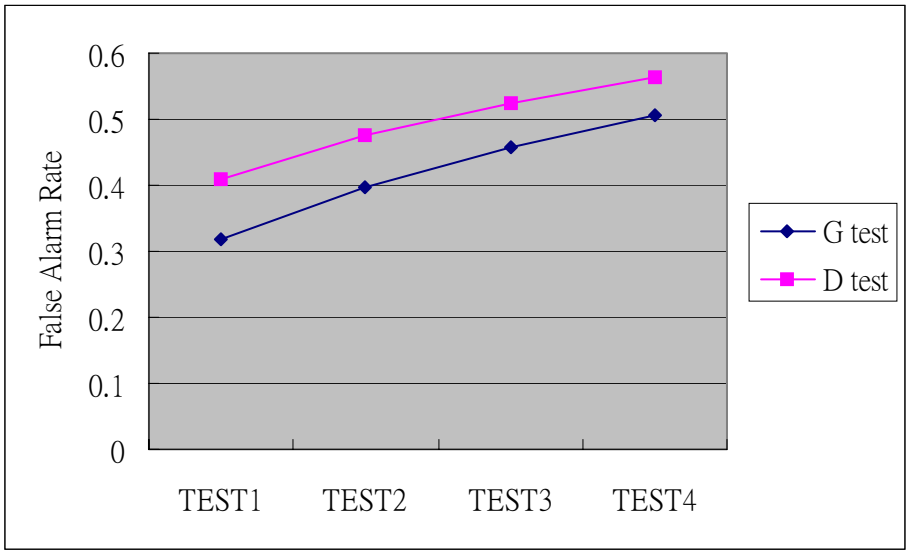

Figure 5 The average false alarm rate for different periods

Figure 5 shows that testing data that cover a longer period produce a higher false alarm rate. User mailing behavior may gradually change and bias the user profile, so the profile must be retrained when it becomes too old. Here, only the trend is considered without being controlled at a low false alarm rate.

4.2 Cutting-time threshold test

In this part of the analysis, the effect of cutting-time threshold is examined. The user mailing behavior changes slowly as time passes. Old behaviors that cannot recur must be filtered out to prevent these old mailing behaviors from being involved in the profile. Such old mailing behaviors generate biases. In this test, senders with no less than 500 mailing events are chosen. Five cutting-time thresholds 0, 50, 100, 150 and 200 are tested. A cutting-time threshold of 50 implies that first 50 records are cut and the others preserved. Table 8 details the training data and the testing data.

Table 8 Data distribution for training and testing data

Cutting-time Threshold	Training data set	Test data set
CT=0	1~400	401~500
CT=50	51-400	401~500
CT=100	101~400	401~500
CT=150	151~400	401~500
CT=200	201~400	401~500

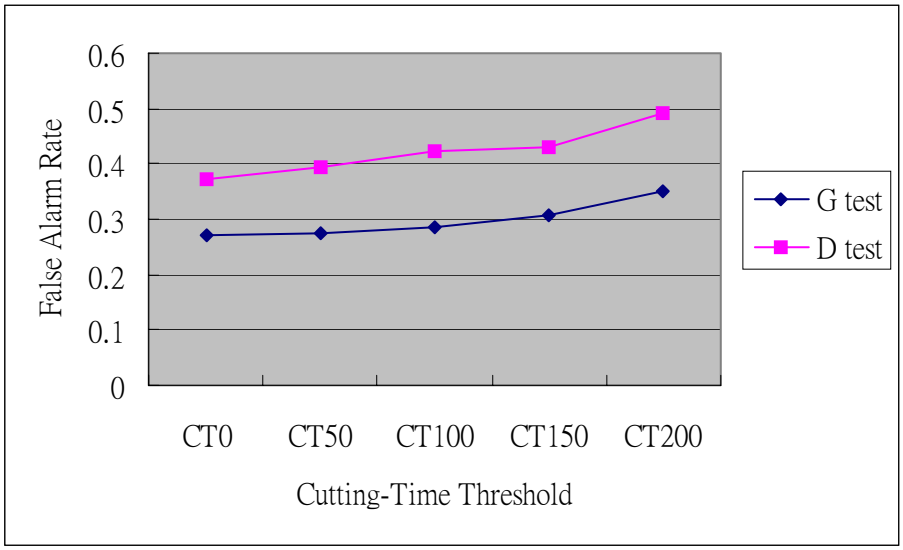

Figure 6 The average false alarm rate for different cutting-time thresholds

In Figure 6, the false alarm rate increases slightly with the cutting-time threshold. A change is apparent from CT150 to CT200, because of too much mailing data are filter, such that training data cannot cover all the mailer behavior. The profile with a higher cutting-time threshold better detects abnormal mailing behavior. The effect of the cutting-time threshold is shown without ensuring low rate of false alarms. Setting the threshold too high could generate more false alarms. An acceptable cutting-time threshold can be determined by testing for each mailer.

4.3 Sliding window test

Sliding window combines the concepts of the profile rebuilding over time and the cutting-time threshold. A sliding window means when the sliding window moves, training data changes and user profile must be retrained. In this analysis, the width of the sliding window training data keeps 300 mailing events. The width of testing data is 50 mailing events. The sliding window shifts 50 mailing events every time.

The following table depicts the relation between the sliding window data set and the base data set. The base data is for comparison with sliding window data. Figure 7 shows the result of a comparison of sliding window data set and the base set.

Table 9 Detail of the sliding window data set and the base data set

Sliding window		Base	
Training data set	Testing data set	Training data set	Testing data set
1~300	301~350(as TEST1)	1~300	301~350
51~350	351~400(as TEST2)	1~350	351~400
101~400	401~450(as TEST3)	1~400	401~450
151~450	451~500(as TEST4)	1~450	451~500

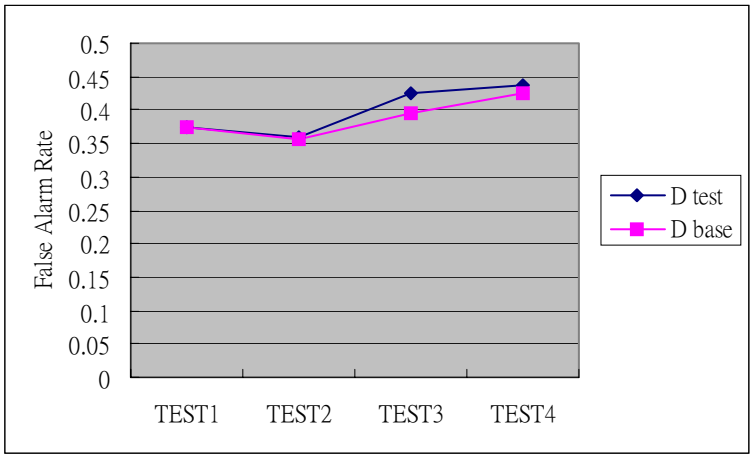

Figure 7 The average false alarm rates for the sliding window and the base data (only conditions for group and dependence analysis are shown, result of the group relation analysis is similar too)

Figure 7 reveals that, although the average false positive rate of the sliding windows is similar to the base data set, the profile generated by sliding windows is elite and sensitive when detecting abnormal mailing behavior. The characteristics of mailing data are shown here, without ensuring a low false positive rate. The time when the profile is rebuilt and the cutting-time threshold vary among mailers. Feasible values can be determined by testing for each mailer.

4.4 Email virus simulation testing

In this part, we want to verify the ability of the profile to differentiate from virus behavior. The profile is built based on the real data and the virus behavior is simulated. The simulation assumes that an email virus sends mail with malicious attachments to recipients in an “address book” or inbox folder, which assumption is reasonable for almost current email viruses. Most email viruses spread by mass mailings to all entries in the address book. In this simulation, the situation and also whether the user profile can differentiate between small and moderate numbers of recipients is established. The mailing behavior of 5% to 100% percent of the recipients in the address book is simulated. Recipients of historic mailing events were used to produce an address book because the mailer’s real address book was not available.

Mailers with a false positive rate under 10% are selected, according to group relation analysis. This simulation involved 765 mailers. Figure 8 shows the affect of ability of profile to discriminate between it and simulated behavior, according to group and dependence relation analysis. The proposed method is compared to the base test. The base test applies the following condition. The mailing behavior is assumed to be normal distribution, such that the average plus two standard deviations covers 97.7% of mailing events. If the number of recipients of the simulated mailing behavior exceeds the average plus two standard deviations, then it is referred to as abnormal, else as normal. The vertical axis is termed, “Discriminative capability” not “Detection rate”, because the behavior generated by simulation sometimes

coincides with normal behavior.

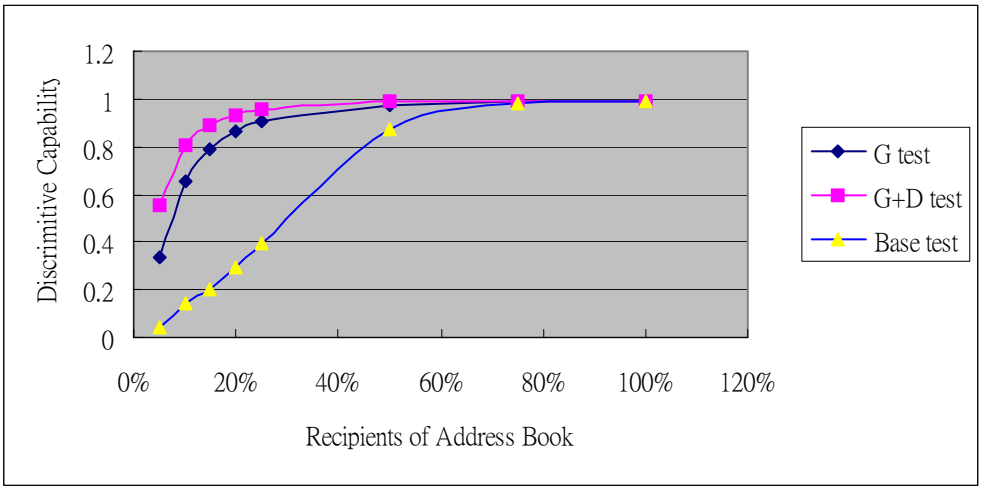

Figure 8 The discriminative capability from simulated behavior based on both analysis compared with base test

In Figure 8, the discriminative capability for the group and dependence relation analysis both is high than base test. The discriminative capability is high when the recipients of the mailing behavior are over 30% of the address book recipients for group analysis. For the dependence analysis, the differential ability rate is high when the recipients of the mailing behavior are over 20 % of the address book. For the majority of email viruses that generate mass mailing, these simulation results are satisfactory.

Using the above approach, only mailers who are not outliers are chosen. A real case may involve mailing data that could include a little abnormal behavior caused by viruses. Another approach is to remove only those mailing events that are outliers, and reserve other normal mailing events. Accordingly, abnormal data are assumed only to represent a very small fraction of the raw data, and so negligibly affect the mean and standard deviation. This assumption is reasonable for the virus problem. 3743 original mailers were considered. Allowing a false positive rate of no more than 10%, 1823 mailers were involved in the simulation. Figure 9 presents the results.

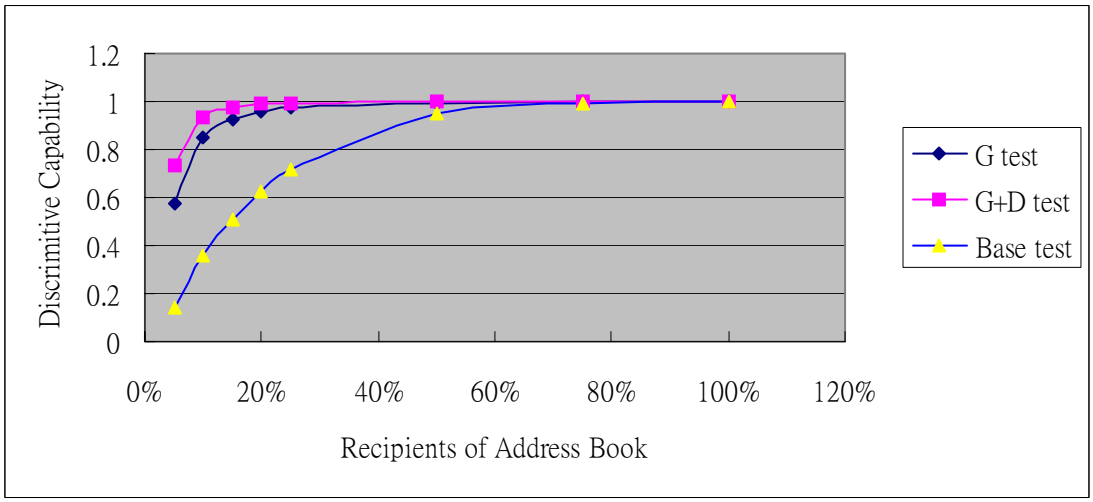

Fig. 9 Discriminative capability determined from simulated behavior based on 1823 mailers.

5. Discussion

The above analysis indicates that the discriminative capability obtained by both group and dependence relationship analyses exceeds that for group relationship analysis alone for a given sender, because both types of analyses performed together are stricter than group relationship analysis alone. Group relationship analysis can only determine whether one mailing behavior is contained in a group or not. Dependence relationship analysis explores further the relationship among recipients in a single group.

The group relationship analysis stage produces the largest possible sets of historical mailing behavior patterns, but the analyzed relationship is fuzzy and may include mailing behavior that has not yet occurred. When too few training data are available, this type of analysis can be performed to avoid generating many false alarms. When the training data suffice, dependence relationship analysis can be used to capture accurately many characteristics of relationships among the recipients.

A worst case and a best case are considered here. In the former, all recipients of a given mailer constitute a single group. In this situation, group analysis is not useful in validating mailing behavior. The dependence relationship must be determined. In the best case, every recipient constitutes a group alone. Only group analysis need be performed. Dependence analysis need not be performed, since it applies to groups with multiple recipients. In most situations, a user profile comprises some groups with multiple recipients and other groups with a single recipient.

An abnormal mailing event with fewer recipients is harder to detect. The proposed method fails to detect malicious mailing behavior that coincides with normal historical mailing behavior. This type of threat can be thought of as light and the system manager has more time to make a response. Epidemiologically, if email viruses are to spread widely and quickly, recipients cannot be too few. Our method works well for abnormal mailing behavior, when the number of recipients is greater than 30% of the recipients in the address book. The method is applicable where mass mailing is a significant threat. On this assumption, the results of the email virus simulation test are satisfactory.

The protocol information, which is a small amount of information, compared to the entire mail message, is processed for reasons of speed. A malicious program normally does not forge recipient information, because doing so would produce mail that could not be sent to its intended destination, defeating the purpose of the distribution. A malicious program may create some mail content such as subject, content or attachment filename, to evade detection by anti-virus software, but doing so does not affect the delivery of the mail, since delivery does not depend on internal information. Also, internal information is strongly related to personal and private matters. In summary, using envelope information has two benefits. First, the amount of data is small, so the processing time is short: the information can be processed in real time. Secondly, the data does not include sensitive information about the user that may raise issues of privacy.

Although some email viruses use a MIME bug to embed malicious code in the text

without attachment, with respect to mailing behavior, whether a virus involves attachments is unimportant.

6. System Design and Implementation

To apply our detection method, we have built and are now testing a prototype mail processing system, called the Mail Gate alarm system. The architecture of the system is illustrated in Figure 10.

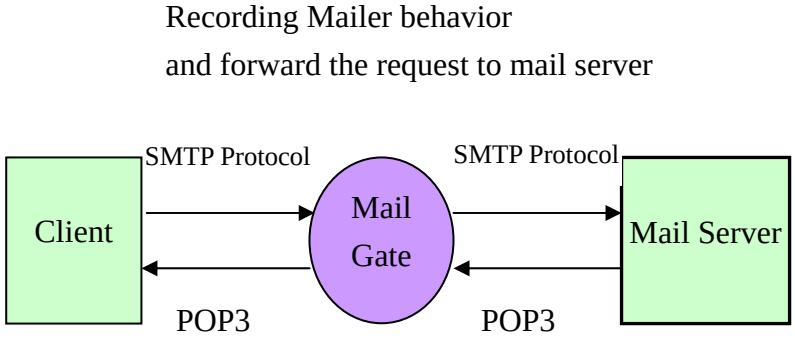

Figure 10 The architecture of the Mail Gate alarm system

Mail Gate is placed between the mail client and the mail server as an intermediate system. Originally, the mail client sent mail to the server but now sends mail through the Mail Gate, and the Mail Gate then records the necessary information before passing all information from the client to the intended mail server. The mail client, the Mail Gate and the mail server communicate with each other via the SMTP protocol.

The advantage of this architecture is that the Mail Gate can be easily integrated with the other mail system. Mail Gate is just arranged as an intermediate bridge. The mail client or mail server communicates with Mail Gate via STMP protocol, as it did originally. Such architecture provides the following advantages. The Mail Gate system is independent of the server and the client. Restated, Mail Gate can be integrated with any kind of the mail server system, such as the SENDMAIL or Exchange servers, and with any kind of the mail client, such as Outlook Express or Netscape. Mail Gate can also be built on various platforms. The system itself is lightweight. It accepts mail messages from the client, simply collecting and recording the envelope information, and then passes all the information to the mail server. Protection at the server-level also can help to complement a defense of desktop anti-virus solution [5].

The prototype Mail Gate email virus alarm system collects the recipients' information, rebuilds the profiles and judges mailing behavior. Whenever a new mailing behavior occurs, the pattern of the mailing event is recorded. From the bulk of data analysis, the size of mailing pattern is significantly less than that of the whole mail message (< 0.01). Additionally, the workload of the Mail Gate is less than that of the mail server since Mail Gate needs only to process a subset of SMTP commands and transmit the information to the mail server. The

profile is periodically rebuilt, and is generated quickly using a statistical table. The profile can also be rebuilt offline. Mailing behavior is judged in real time. The average number of groups in a user profile is 30, while the average number of dependence rules in a user profile is 160. Moreover, the amount of data in the profile is small for each mailer, accounting for the short processing time. Furthermore, dependence analysis is unnecessary if the behavior is computed as invalid in the group analysis stage.

The prototype Mail Gate email virus alarm system has been developed in a Linux system, running on a Pentium IV 1800 Mhz with 1024 MB of RAM. During the primitive test stage, when a particular mailing behavior is suspicious, a “warning message” is inserted into the mail head to advise the recipient to remain cautious of the mail. In our future work, we hope to redirect a suspicious mail to a local mail system for validation. If deemed normal, the mail can be resent. If the mail is deemed malicious, the system administrator automatically contacts the individual from whom the message originated.

7. Conclusion and Future Directions

This paper presents a novel method for detecting anomalous mailing behavior. Based on user behavior rather than a specific signature, the proposed method can be used to detect both known and unknown email viruses, as long as the new behavior violates historical mailing behavior. The method is based on the observation of abnormal mailing behavior not the direct detection of a virus or malicious code. The building of a user profile, using data mining, is divided into two stages - a group relation analysis that explores the “clique” relationships among recipients, and second, a dependence relation analysis that explores the “dependence” relationship that exists within a single group. Group relation analysis is less strict than dependence relation analysis. Only group relationship analysis or both analyses can be chosen in a real environment, according to the amount of available mailing data.

By combining the statistical table and the cutting-time threshold, we can quickly build an effective user profile. The behavior pattern table is continuously updated as new mail events occur. The building of the user profile does not rely on all original events but only on the table. Using a cutting-time threshold, old behavior that never recurs can be filtered from the profile building process.

The proposed method cannot detect viruses before they impact a system; the detection works after abnormal mailing behavior occurs. When the abnormal behavior caused by a known or unknown virus is detected, the proposed architecture can prevent the virus from spreading further at the Mail Gate although the sender has been infected. The method reveals the virus’s behavior. It is possible to fool the proposed system by imitating normal historical mailing behavior but normal mailing behavior can not generally be referred to as mass mailing, thus the proposed method has the effect to suppress the intent of mass mailing.

The authors are now constructing and testing a prototype system, called the Mail Gate email virus alarm system, using the proposed method. Server-level protection also provides some advantages over desktop anti-virus solutions in a corporation environment [2][29]. An

open source SMTP gateway virus filtering system [29] includes aMaViS that provides an interface between mail server and a virus scanning utility. The mail server can detect viruses, but the detection capacity still depends on the virus scanning utility. Another email virus alarm system developed at the University of Minnesota [2] uses a pattern-matching algorithm to seek viruses; this alarm system seems to be integrated into the mail server system.

A proactive method of detection is presented here. It complements traditional pattern-matching anti-virus mechanisms that usually catch known viruses. Combining the proposed anomaly-detection method with other methods can strengthen defenses against viruses, including previously unknown viruses, in a complete system of defense as described in [35] and [20].

Aggregating different mailers anomalies can facilitate the accurate detection at mail spread viruses since a mailing virus storm occurs concurrently over a short period of time. Additional data mining techniques using Bayesian network [17] or accumulating statistics can help to clarify the normal characteristics of mailing behavior. The issues at the temporal information, which has not yet been used in the user profile, will be discussed in our upcoming work.

Acknowledgement

The authors would like to thank to the anonymous reviewers for their suggestions and guidance, and also thank Mr. Ted Kony and Mrs. Debbie Nester for helping us to improve the English text.

References

[1] Agrawal, R., Imielinski, T. and Swami, A., “Mining Association Rules between Sets of Items in Large Databases”, In Proc. of the ACM SIGMOD Conference on Management of Data, Washington D.C., May 1993, pp. 207-216

[2] Automatic Execution of Embedded MIME Types, http://www.cert.org/advisories/CA-2001-06.html, 2001

[3] Automatic Virus Checks, http://www1.umn.edu/oit/newsletter/01/0601_itn/virus.html

[4] Barnett, V. and Lewis, T., Outliers in Statistical Data, 3rd edition, John Wiley, 1994

[5] Cella, Julio, “Antivirus at SMTP Gateways Level”, http://www.giac.org/certified_professionals/practicals/gsec/0846.php

[6] Charlier, B. Le, A. Mounji and Swimmer, Morton, “Dynamic detection and Classification of Computer viruses general behaviour patterns”, In Proc. of Fifth International Virus Bulletin Conference, Boston, September 20-22, 1995.

[7] Chess, David M., “Virus Verification and Removal Tools and Techniques”, November 18, 1991, http://www.research.ibm.com/antivirus/SciPapers/Chess/CHESS3/chess3.html

[8] Chien, Eric and Ewell, Brian, VBS.LoveLetter and variants, http://www.symantec.com/avcenter/venc/data/vbs.loveletter.a.html, 2001

[9] Chien, Eric, Nimda, http://www.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html, 2001

[10] Conquering Complex Viruses, http://enterprisesecurity.symantec.com/article.cfm?articleid=11&PID=4402422, 2000

[11] Elnitiarta, Raul K., Melissa, http://securityresponse.symantec.com/avcenter/venc/data/w97.melissa.a.html, 1999

[12] E-mail Explained, http://www.sendmail.org/email-explained.html

[13] Ferrie, Peter and Szor, Peter, Sircam, http://securityresponse.symantec.com/avcenter/venc/data/w32.sircam.worm@mm.html, 2001

[14] Forrest, S., Hofmeyr, S. A., Somayaji, A. and Longstaff, Thomas A., “A Sense of Self for Unix Processes”, In Proc. of the 1996 IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1996, pp120-128

[15] Gupta, A. and Sekar, R., “An approach for detecting self-propagating email using anomaly detection”, In Proc. of the Sixth International Symposium on Recent Advances in Intrusion Detection, September 2003.

[16] Han, Jiawei and Kamber, Micheline, Data mining: Concepts and Techniques, Morgan Kaufmann, 2001

[17] Heckerman, D., “A tutorial on learning with Bayesian networks”, Technical Report MSR-TR-95-06, Microsoft Research, March 1995.

[18] Ishibashi, Keisuke, Toyono, Tsuyoshi, Toyama, Katsuyasu, Ishino, Masahiro, Ohshima, Haruhiko, and Mizukoshi, Ichiro, “Detecting Mass-Mailing Worm Infected Hosts by Mining DNS Traffic Data”, In Proc. of the 2005 ACM SIGCOMM workshop on Mining network data, Philadelphia, Pennsylvania, USA , August 22-26, 2005, pp. 159-164

[19] Javitz, H. S. and Valdes, A., “The NIDES statistical component description and justification”, Technical report, Computer Science Laboratory, SRI International, Menlo Park, CA, March 1994

[20] Kennedy, R. L., Lee, Y., Roy, B.V., Reed, C. D. and Lippmann, R. P., “Solving Data Mining Problems through Pattern Recognition”, Prentice Hall, 1998

[21] Knorr, Edwin M. and Ng, Raymond T., “Algorithms for Mining Distance-Based Outliers in Large Datasets”, In Proc. of the 24th VLDB Conference, New York, USA, 1998

[22] Kumar, Sandeep and Spafford, Eugene H., “A Generic Virus Scanner in C++”, In Proc. of the 8th Computer Security Applications Conference, IEEE press, 1992

[23] Lee, Wenke and Stolfo, S. J., “Data mining approaches for intrusion detection”, In Proc. of the Seventh USENIX Security Symposium (SECURITY '98), San Antonio, TX, January 1998, pp 66-72

[24] Porras, Phillip A. and Neumann, Peter G., “EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances”, 20th National Information Systems Security Conference, October 1997, access from http://www.sdl.sri.com/projects/emerald/emerald-niss97.html

[25] Schultz, Mathew G., Eskin, Eleazar, Zadok, Erez, Bhattacharyya, Manasi, and Stolfo, Salvatore J., “MEF: Malicious Email Filter A UNIX Mail Filters that Detects Malicious Windows Executables”, In Proc. of USENIX Annual Technical Conference - FREENIX Track, Boston, Massachusetts, USA, June 25–30, 2001

[26] Simple Mail Transfer Protocol, http://www.sendmail.org/rfc/0821.html

[27] Srikant, R. and Agrawal, R., “Mining Generalized Association Rules”, In Proc. of the 21st international Conference on Very Large Data Bases, 1995, pp. 407-419

[28] Stolfo, Salvatore J., Hu, Chia-Wei, Li, Wei-Jen, Hershkop, Shlomo, Wang, Ke, and Nimeskern, Olivier. "Combining Behavior Models to Secure Email Systems", CU Tech Report, April 2003.

[29] Swab, Kevin, "SMTP Gateway Virus Filtering with Sendmail and AMaViS", http://www.sans.org/rr/whitepapers/email/576.php, 2001

[30] Tarala, James, Virii Generators: Understanding the threat, http://www.sans.org/rr/whitepapers/malicious/144.php, 2002

[31] Tesauro, G., Kephart, J. O. and Sorkin, G. B., “Neural Network for Computer Virus Recognition”, IEEE expert, Vol.11, No. 4, Aug 1996, pp. 5-6.

[32] The Digital Immune System, http://www.symantec.com/avcenter/reference/dis.tech.brief.pdf
[33] Understanding Heuristics: Symantec's Bloodhound Technology, http://www.symantec.com/avcenter/reference/heuristc.pdf
[34] White, Steve R., "Open Problems in Computer Virus Research", Virus Bulletin Conference, Munich, Germany, October 1998, access from http://www.research.ibm.com/antivirus/SciPapers/White/Problems/Problems.html
[35] White, Steve R., Morton Swimmer, Edward J. Pring et al., Anatomy of a Commercial-Grade Immune System, http://www.research.ibm.com/antivirus/SciPapers/White/Anatomy/anatomy.html
[36] Whyte, D., Oorschot, P.C. van, and Kranakis, E., "Addressing Malicious SMTP-based Mass-Mailing Activity Within an Enterprise Network.", Carleton University, SCS Technical Report, TR-05-06, May 2005.