International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:6, 2013

Utilizing Innovative Techniques to Improve Email Security

Amany M. Alshawi and Khaled Alduhaiman

***Abstract*—**This paper proposes a technique to protect against email bombing. The technique employs a statistical approach, Naïve Bayes (NB), and Neural Networks to show that it is possible to differentiate between good and bad traffic to protect against email bombing attacks. Neural networks and Naïve Bayes can be trained by utilizing many email messages that include both input and output data for legitimate and non-legitimate emails. The input to the model includes the contents of the body of the messages, the subject, and the headers. This information will be used to determine if the email is normal or an attack email. Preliminary tests suggest that Naïve Bayes can be trained to produce an accurate response to confirm which email represents an attack.

***Keywords*—**Email bombing, Legitimate email, Naïve Bayes, Neural networks, Non-legitimate email.

I. Introduction

IN the information age, rapid dissemination of data and a quick method to apprise others of the information available is possible by means of electronic mail. Electronic mail is a special type of document which contains text and other identifying information such as: from, to, and subject fields. Email has evolved as a convenient means of communication between various parties. It is a fast, efficient and an inexpensive method of reaching out to a large number of people at the same time. However, the average user is often overwhelmed by the amount of email sent and received on daily basis. A large part of the mail traffic is unsolicited bulk email or spam as it is popularly known. In addition, attackers might use email to disable critical servers or websites through the use of email bombing.

Email bombing, which is characterized by “abusers repeatedly sending an identical email message to a particular address [1]”, is one of the easiest ways to cause denial of service attacks (DoS). Email bombing tools are very simple, easy to configure, and are widely available on the Internet. The number of email attacks has increased rapidly and became a serious threat to the Internet community.

Amany Alshawi is with King AbdulAziz City for Science and Technology, Riyadh, Saudi Arabia (email: aalshawi@kacst.edu.sa).
Khaled LAduhaiman with RayanTech, Riyadh, Saudi Arabia (email: alduhaiman@gmail.com).

At this time there is no effective way of preventing an attack and no easy way to discriminate between normal and attack email. A huge amount of mail can suddenly fill up the recipient’s disk space on the server or, in some cases, even cause the server to stop functioning [2].

An attacker can download a small program to send one email repeatedly, as many as 10,000 times, to a single email address. Consequently, the target will be disabled and unable to receive additional email. There are several serious consequences of these email bomb attacks. They generate unwarranted expenses to both Internet Service Providers (ISPs) and the recipients. For example, when one 20 Kbyte message is sent 10,000 times to one user, the ISP needs 200 Mbytes for storage. It is not unusual for email bomb attackers to use third party hosts as relays causing additional innocent people to be affected [3].

This paper contributes to the solution of critical issues in the domain of email security. A review of the existing email security tools reveals an obvious lack of an adequate comprehensive solution. Preventive technologies, such as firewalls, have helped, but no single form of defense proved to be highly effective. A firewall can slow down known attacks, but it will not stop or detect threat of attacks. Most existing solutions involve rule-based solutions, such as Intrusion Detection Systems (IDSs), which monitor the system by looking for specific "signatures" of behavior. But they often do not provide enough information to detect malicious behavior. Most IDSs techniques have a high rate of false positive alarms, i.e., notification of an attack when, in actuality, none exists.

Although email classification can be viewed as a special case of text classification, the characteristics of documents and emails are different and as a consequence email classification poses certain challenges, not often encountered in text or classification. Some of those challenges are:

1. Each user’s mailbox is different and constantly evolving. Folder contents vary from time to time as new messages are added and old messages are deleted. A classification scheme that can adapt to varying folder characteristics is important.
2. Manual classification of emails is based on personal preferences and hence the criteria used may not be as simple as those used for text classification.
3. The information content of emails vary significantly, and other factors, such as the sender, group the email is addressed to, play an important role in classification. This

is in contrast to documents which are richer in content resulting in easier identification of topics or context.

4. The characteristics of folders may vary from dense (large number of emails) to relatively sparse. A classification system needs to perform reasonably even in the absence of a large training set.
5. Emails even within a folder may not be cohesive. That is, the contents may be disparate and not have many common words or a theme. We characterize these folders on a spectrum of homogeneous to heterogeneous. A folder may lose its homogeneity as it becomes dense making it difficult to associate appropriate central theme with the folder [4].

The remainder of this paper is organized as follows. Section II presents the related work in the areas of text and email classification. Section III gives an overview of the research methodology. Section IV explains the results of applying the proposed methodology and section V has conclusions and future work.

II. Related Work

A number of text classification techniques have been applied to the problem of email classification. Based on the mechanism used, email classification schemes can be categorized into: rule based classification, information retrieval based classification, and machine learning based classification techniques.

A. Rule Based Classification Systems

Use rules to classify mail messages into folders. Cohen uses the RIPPER learning algorithm to induce "keyword spotting rules" for email classification. RIPPER is a propositional learner capable of handling large datasets. Cohen argues that keyword spotting is useful as it induces an understandable description of the email. The RIPPER system is compared with a traditional IR method based on the TF-IDF weighting scheme and both show similar accuracy. i-ems is a rule based classification system that learns rules based only on sender information and keywords. Ishmail is another rule-based classifier integrated with the Emacs mail program Rmail. Although rules are easy for people to understand, managing a rule set may not be so. As the number and characteristics of incoming mails change, the rules in the rule set may have to be modified. This puts a burden on the user, to review and update the rule set from time to time, which often involving a complete re-writing of rules. Most of the email managers allow users to set rules for classifying email to folders. These rules have to be specified manually and can use words from various categories. The main problem is in the manual specification and management of these rules which can become cumbersome [5]-[6]-[7].

B. Information Retrieval Based Classification

Segal and Kephart use the TF-IDF classifier as the means for classification in SwiftFile, which is implemented as an add-on to Lotus Notes. It predicts three likely destination folders for every incoming mail message. The TF-IDF classifier performs well even in the absence of large training data, and the classifier accuracy remains reasonable as the amount of training data increases, adding to the heterogeneity of a folder. The classifier learns incrementally with every new message that is added or deleted from a folder, eliminating the need for re-training from scratch [8].

C. Machine Learning Based Classification

Various machine learning based classification systems have been developed. The iFile system by Rennie uses the Naive Bayes approach for training, providing good accuracy, and for performing iterative learning. The Naive Bayesian probabilistic classifier has also been used to filter junk mail as shown by Sahami et.al. The Reagent mail classifier by Boone uses the TF-IDF measure to extract useful features from the mail and then predicts the actions to be performed using the trained data and a set of keywords. It uses the nearest neighbor classifier and a neural network for prediction purposes and compares the results obtained with the standard IR, TF-IDF algorithm. Mail Agent Interface (Magi) by Payne and Edwards uses the symbolic rule induction system CN2 to induce a user-profile from observations of user interactions. The system suggests actions such as `delete', `forward' and so on for each new mail message based on the training; hence results for multi-class categorization are difficult to assess [9]-[10]-[11].

III. Method

A. Data Collection

The study used the Ling-Spam corpus consisting of 2412 legitimate messages and 481 non-legitimate messages. This means there are 84% normal emails and 16% Spam emails. The data was divided into two subsets: the training data set and test data set. Each of the classification algorithms used the training set to generate rule sets to discriminate between legitimate and non-legitimate emails. Finally, the test data set of unseen examples was used to check the accuracy of the classifiers [12].

Cross-validation was used to divide the data into multiple data sets. The reason of using cross-validation was to avoid bias in data collection. It divides the data into m subsamples. Each one was predicted via a different classification method using the remaining sub-samples (m-1).

Classification has been the subject of much research in the machine learning society. One approach is Bayesian classification, a method that has become gradually more popular lately in part due to recent developments in learning with Bayesian belief networks [4]. The simplest Bayesian classifier is the widely used Naive Bayes. It significantly simplifies learning by assuming that features are independent given class, that is,

$$P(x,c) = \prod_{i=1}^{n} P(x_i, c)$$ Where $X = (x_1, x_2, ...x_n)$ is a

feature vector and c is a class.

Despite the fact that feature independence is generally a weak assumption, Naïve Bayes is surprisingly successful in practice [2]. Naïve Bayes has been applied successfully in text classification, medical diagnosis, and computer performance management.

B. Definition of Terms

- Words, W is an ordered collection of words i.e., W={w1, w2…,wn}.
- A classifier is a machine in the mathematical sense that deterministic-ally returns a class Ci in C= {0, 1} *where* 0 means that the class is Non-Legitimate (*NL)* and 1 that the class is Legitimate *(L),* given a training data.

C. Notations

- *m*= a message
- *M*= a set of all possible messages
- *L*= a set of all possible Legitimate messages
- *NL*= a set of all possible Non-Legitimate messages

Then *M=L + NL*

D. Features Selection

It is impossible to include all of the features that were selected as good candidates; however it is much easier to select some of them. There are many techniques designed to select the best features and Mutual Information Gain is among the best choices. The fact that the features are statistically independent can also be an advantage in the context of feature selection. To pick among all possible word attributes, we capture the mutual information (MI) of each candidate attribute X with the category class C:

$$MI(X;C) = \sum_{x,c} \mathrm{P(x,\ c)} \times \log \frac{\mathrm{P(x,\ c)}}{\mathrm{P(x)} \times \mathrm{P(c)}}$$

where *x* in {0, 1} and *c* in {legitimate"1", non-legitimate "0"}.

The attributes with the m highest *MI*-scores were selected from the messages that were collected during the experiment phase.

IV. Results

A. Transformation of a Message m into a Word Vector (w)

The experiment starts by choosing a set of 14 "key" words, *W={w1, w2…,w14},* which occur with different likelihood in ***L*** or ***NL*** messages.

Then given a message m, define the corresponding word *W(m)* as a 14 elements vector whose elements are 1, or 0, according whether or not the words w1, w2…,wn appear in m or not, for example:

$$W(m) =< 1,0,0,1,1,0,1,0,1,0,0,0,1,1 > \quad (1)$$

B. Bayes' Theorem

Naïve Bayesian Classifier (NBC) or Naïve Bayes (NB) is a simplified form of Bayes' rule that assumes independence of the observations. Research demonstrated that NBC has competitive performance in comparison with other learning algorithms if the normal distribution assumption holds [2].

Suppose the vector is the one in equation (1), then the first item of information in this vector is that m contains a number of keywords.

First Evidence Item = *E1 = {m* contains key words *w1}*

$$P(H \mid E) = \frac{P(E \mid H) \times P(H)}{P(E)} \quad (2)$$

The hypothesis, *H,* is that our message *m* is Non-Legitimate i.e. *m=NL.*

$$P(m = NL \mid E_1) = \frac{P_0(m = NL) \times P(E_1 \mid m = NL)}{P(E_1)} \quad (3)$$

$P(m = NL \mid E_1)$ is the Posterior probability that our message is Non-Legitimate, given that *m* contains *w1*

$P(m = NL)$ is the prior probability of *m=NL,* it's equal to the fraction of all messages that are Non-Legitimate.

$P(E_1 \mid m = NL)$, is the fraction of the Non-Legitimate messages that contain keyword *w1*

Similarly, we can calculate the probability that message *m* is Legitimate,. i.e. *m*=L

$$P(m = L \mid E_1) = \frac{P_0(m = L) \times P(E_1 \mid m = L)}{P(E_1)} \quad (4)$$

Since we must have:

$P(m = L \mid E_1) + P(m = LN \mid E_1) = 1$, we can conclude that :

$$P(E_1) = P(E_1 \mid m = NL) \times P_0(m = NL) + P(E_1 \mid m = L) \times P_0(m = L) \quad (5)$$

By substituting (5) into (4), we get:

$$P(m = NL \mid E_1) = \frac{P_0(m = NL) \times P(E_1 \mid m = NL)}{P(E_1 \mid m = NL) \times P_0(m = NL) + P(E_1 \mid m = L \times P_0(m = L)} \quad (6)$$

C. Evaluation Methods

1. Method I

- *TANL* is the Total Actual Non-Legitimate and *TAL* is the Total Actual Legitimate denotes the number of legitimate and non-legitimate messages in the validation data.
- *TCNL* is the Total Correct Non-Legitimate and *TCL* is the

Total Correct Legitimate denotes the number of messages that are correctly classified as legitimate and non-legitimate in the validation data.

Two different measures were used to evaluate the performance of Naïve Bayes technique:

- *FPR* False Positive Rate is the percentage of legitimate messages that are misclassified as non-legitimate.

$$FPR = 1 - (\frac{TCL}{TAL})$$

- *FNR* False Negative Rate is the percentage of non-legitimate messages that are misclassified as legitimate.

$$FNR = 1 - (\frac{TCNL}{TANL})$$

2. Method II

Let

- *LN* the number of legitimate emails that are classified as non-legitimate
- *NL* the number of non-legitimate emails that are classified as legitimate.
- *LL* the number of legitimate emails that are classified as legitimate
- *NN* the number of non-legitimate emails that are classified as non-legitimate.

Non-legitimate Recall NR and Non-legitimate Precision NP could be calculated as:

$$NR = \frac{NN}{(NN + NL)}$$

$$NP = \frac{NN}{(NN + LNL)}$$

D.Neural Networks (NN)

NN were used as a common method for email classification. The classification procedure consists of three steps, data preprocessing, data training, and testing. The data preprocessing refers to the feature selection involving selecting a set of features which is more informative in the task while removing irrelevant or redundant features. For the email classification, feature selection is formulated into the problem of identifying the most relevant word features within a set of text documents for a given text learning task. For the data training, the selected features from the data preprocessing step were fed into the NN, and an email classifier was generated through the NN. For the testing, the email classifier was used to verify the efficiency of NN. As shown in Fig. 1, a neural network usually consists of three layers or groups: the output layer, a hidden layer, and the input layer.

Neural networks were used to bench mark the results of the Naïve Bayesian Classifier. A total of 600 legitimate and non-legitimate emails were used. 400 of them were classified with class 1 if the email is legitimate and 0 otherwise. Table I shows the comparison between Naïve Bayes and Neural Networks.

TABLE I
FALSE POSITIVE AND FALSE NEGATIVE RATE

	Naïve Bayes	Neural Networks
FPR	2.60%	2.60%
FNR	17.39%	19.57%

Knowing the fraction *Po(m=NL)* of all messages that are Non-Legitimate, and given the information that a particular message, m, contains or does not contain specific keywords, *Wi*, it is possible to calculate the Updated or Posterior probability that *m* is Non-Legitimate.

V. CONCLUSION AND FUTURE WORK

In this paper, two classifiers, Naïve Bayesian and Neural Networks were tested to filter spams from the dataset of emails. All the emails were classified as legitimate (1) or not-legitimate (0). That was the characteristic of the dataset of email for spam filtering. Naïve Bayesian classifier showed better results compared with Neural Networks. From this experiment, we can conclude that a simple classifier can provide better classification result for spam mail filtering. In the near future, there is a plan to incorporate other techniques like different ways of feature selection and classification using ontology. Also, classified result could be used in the semantic web by creating a modularized ontology based on classified result. There are many different mining and classification algorithms, and parameter settings in each algorithm. Experimental results in this paper are based on the default settings. Extensive experiments with different settings are also applicable.

REFERENCES

[1] Carnegie Mellon University. The Computer Emergency Response Team. (2002, March) Internet *Denial of Service Attacks and the Federal Response*. http://www.cert.org

[2] Langley, P. 1992. Systematic and nonsystematic search strategies. In *Proceedings of the First International Conference on Artificial Intelligence Planning Systems*, pp. 145-152 College Park, Maryland. Morgan Kaufmann.

[3] Lindberg, G. "Anti-Spam Recommend-ations for SMTP MTAs" Internet RFC 2505, February 1999.

[4] M. Aery and S. Chakravarthy. eMailSift: Adapting Graph Mining Techniques for Email Classification: Technical Report. University of Texas at Arlington. 2004.

[5] J. Catlett. Megainduction: A test °ight. *International Conference on Machine Learning*, 1991.

[6] W. W. Cohen. Learning rules that classify e-mail. *Proceedings of AAAI-1996 Spring Symposium on Machine Learning in Information Access*, pp. 145- 155.

[7] J. Herman and C. Isbell. Ishmail: Immediate identification of important information, AT&T labs. 1995.

[8] J. D. M.Rennie. Application of machine learning to e-mail filtering. Proceedings of KDD-2000 Text Mining Workshop, Boston Aug, 2000.

[9] T. Payne and P. Edwards. Interface agents that learn: An investigation of learning issues in a mail agent interface. Applied Artificial Intelligence, pp 132-144. 1997.

[10] J. Rissanen. Stochastic complexity in statistical enquiry. World Publishing Company, 1989.

[11] M. Sahami, D. Heckerman, and E. Horovitz. A Bayesian approach to filtering junk e-mail. AAAI-98 Workshop on Learning for Text Categorization, 1998.
[12] Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C.D. Spyropoulos and P. Stamatopoulos, "Learning to Filter Spam E-Mail: A Comparison of a Naive Bayesian and a Memory-Based Approach". In H. Zaragoza, P. Gallinari, and M. Rajman (Eds.), Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000), Lyon, France, pp. 1-13, 2000.