ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

A Tree Based E-mail abstraction scheme for Spam Detection

U.D.S.V. Prasad #1, Dr K.Subramanyam#2, S.Jaya Prakash #3,

#1 Student, K.L.University,Vaddeswaram,Guntur(dt),

#2 Professor, K.L.University,Vaddeswaram,Guntur(dt),

#3 Student, K.L.University,Vaddeswaram,Guntur(dt),

#1 prasadudsv@gmail.com,#2 smkodukula@kluniversity.in, prakashsunkavalli@gmail.com #3,

Abstract: In recent years, the notion of collaborative spam filtering with near duplicate similarity matching scheme has been widely discussed. The primary idea of the similarity matching scheme for spam detection is to maintain a known spam database, formed by user feedback, to block Subsequent near-duplicate spam's. Prior approaches generate e-mail abstractions based mainly on hash-based content text. The improvement is limited since we map each subsequence in a node of an SpTree to a hash value. Therefore, the subsequences that have some prefix tags in common still can be differentiated with one comparison. Instead of mapping each subsequence in a node of anSpTree to a hash value using a binary hash function we propose to replace it with a special hash function, namely Simhash. The advantage of this over other hash functions is that it sets a minimum on the number of members that the two sets must share in order to match. This mitigates the effect of extremely common set members on data clusters.

Index Terms: *Spam detection, e-mail abstraction, near-duplicate matching*

1. INTRODUCTION

The amount of junk e-mail, commonly called spam, has skyrocketed in the recent past. Traditionally, spam was sent by single source mass mailers (spammers). This spam is relatively easy to screen by using lists of known spammers [5], and dropping all email that originates at a listed address. Although content-based filters provide some temporary relief from spam, they represent a cat-andmouse game between the spammers and the spam filter developers—every time the developers improve their filter's , the spammers create new ways of fooling them. Consequently, a content-independent method is needed. In the past, spammers used no distributed systems that in turn were susceptible to non distributed countermeasures. The advent of bonnets has given the spammers a distinct advantage, rendering many of the existing (no distributed) countermeasures ineffective.

To rectify this asymmetry, distributed countermeasures are needed. We
propose a distributed, content independent, spam classification system, called Trinity, that is specifically aimed at botnet generated spam and can be used in combination with existing spam classifiers. In order to be viable, Trinity must be easy to integrate within the existing e-mail infrastructure.

Specifically, it must not require changes to existing protocols or mail transfer agent (MTA), it can easily be hosted on the same host as the MTA, and it must work in combination with existing spam filtering mechanisms. Secondly, the system should provide the recipients with the option to bypass Trinity. Third, the system must be resilient to denial-of-service attacks, and must not have single point of failure modes, such as centralized servers. Trinity is based on the following observation: in order to be effective, bots must send a relatively large number of e-mails in a short amount of time.

This stems from the fact that most computers in a botnet are only up for a short period each day; in the evenings when the average user comes home and turns on their machine. Consequently, if an e-mail is received from an "unknown" source that has sent many e-mails in a short period of time, then the likelihood of this being spam is high.

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

Spam remains a serious problem today because spam emails continue to be a very profitable business for spammers. Spam email takes on various forms from adult content, selling products/services, pharmaceuticals to stock promotions, job offers, etc. These unsolicited email messages are a nuisance and can also be offensive to your end-users (particularly adult content). Email spammers are constantly pioneering new techniques to bypass email anti-spam filter solutions forcing organizations to invest in spam email prevention solutions and anti-spam filters that can keep up with the evolving approaches. As long as spam emails make money, spammers will continue to send out a barrage of unsolicited email messages.

2. RELATED WORK

Since the e-mail spam problem is increasingly serious nowadays, various techniques have been explored to relieve the problem. Based on what features of e-mails are being used, previous works on spam detection can be generally classified into three categories: 1) content-based methods,2) noncontent-based methods, and 3) others. Initially, researchers analyze email content text and model this problem as a binary text classification task.

Traditional SVMs [10] and improved SVMs [1] have been investigated. While above conventional machine learning techniques have reported excellent results with static data sets, one major disadvantage is that it is cost-prohibitive for large-scale applications to constantly retrain these methods with the latest information to adapt to the rapid evolving nature of spams. The spam detection of these methods on the email corpus with various language has been less studied yet. In addition, other classification techniques, including markov random field model [3], neural network [6] and logic regression [2], and certain specific features, such as URLs and images have also been taken into account for spam detection.

The other group attempts to exploit noncontent information such as e-mail header, e-mail social network [4], and e-mail traffic [5], [9] to filter spams. Collecting notorious and innocent sender addresses (or IP addresses) from e-mail header to create black list and white list is a commonly applied method initially. MailRank [4] examines the feasibility of rating sender addresses with algorithm PageRank in the email social network, and a modified version with update scheme is introduced. Since email header can be altered by spammers to conceal the identity, the main drawback of these methods is the hardness of correctly identifying each user. In [5], [9], the authors intend to analyze e-mail traffic flows to detect suspicious machines and abnormal email communication.

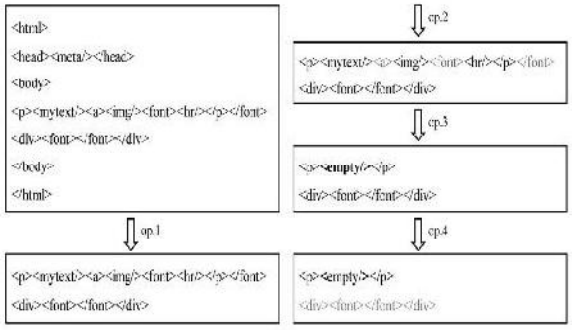

Fig. 1. An example of the preprocessing step in Tag Extraction Phase of procedure SAG.

Consider the example e-mail abstraction in Fig. 1 that has been produced through the execution of lines 1-5 in procedure SAG. The first operation of the preprocessing step is to remove tags which are in front of the <body> tag and which are in rear of the <=body> tag. With regard to operation 2, since there is no end tag of <a>, this tag is deleted. Besides, the tags <font> and < =font > are also deleted because the position of <=font> is incorrect. Note that we can utilize the stack data structure to determine Whether nonempty tags are mismatched. After that, empty tags are transformed to < empty=> in operation 3. Moreover, since mytext=> <img=> <hr=> appear consecutively, only one <empty=> tag is retained. Finally, <font> <=font> <=div> are removed due to the lack of content.

3. DESIGN OF SPTABLE AND SPTREES

One major focus of this work is to design the innovative data structure to facilitate the process of near-duplicate matching. SpTable and SpTrees (sp stands for spam) are proposed to store large amounts of the email abstractions of reported spams. Thus, if we distribute e-mail abstractions with different tag lengths into diverse SpTrees, the quantity of spams required to be matched will decrease. However, if each SpTree is only mapped to one single tag length, it is too much of a burden for a server to maintain such thousands of SpTrees. In view of this concern,

ISSN-2320-7884 (ONLINE)
ISSN-2321-0257 (PRINT)

each SpTree is designed to take charge of e-mail abstractions within a range of tag lengths.

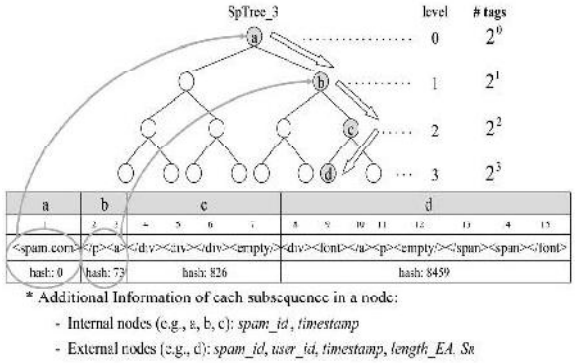

Fig. 2 The illustration of SpTree03 with an example e-mail abstraction

As shown in Fig. 2, the example e-mail abstraction is stored in a path from the root node a to the leaf node d. The primary goal of applying the tree data structure for storage is to reduce the number of tags required to be matched when processing from root to leaf. Since only subsequences along the matching path from root to leaf should be compared, the matching efficiency is substantially increased. Note that if each type of HTML tag. The branch direction of each SpTree is determined by a binary hash function. If the first tag of a subsequence is a start tag (e.g.,), this subsequence will be placed into the left child node. A subsequence whose first tag is an end tag (e.g., <=div>) will be placed into the right child node. Since most HTML tags are in pairs and the proposed e-mail abstraction is reordered in procedure SAG, subsequences are expected to be uniformly distributed.

4. PERFORMANCE ANALYSIS

The real spam data sets used in the experiments are from the e-mail servers of Computer Center in National Taiwan University, which has over 30,000 students. Since the ground truth of real e-mail streams is unavailable, spams are extracted from the well-known existing system, Spam Assassin Note that numerous related works have evaluated the proposed methods with static databases. However, to access the performance of spam detection system with near-duplicate matching scheme, real e-mail streams are more appropriate than static data sets. The processes of generating each digest in Digest and MultiDigest are identical. Although Sarafijanovic's work claims that using multiple digests can enhance the robustness of near-duplicate spam detection system, these two works have not been validated by real e-mail streams.

4.1 Accuracy Evaluation

The most important requirement for a spam detection system is the capability to resist malicious attack that evolves continuously. In procedure SAG of Cosdes, HTML tags are extracted and each paragraph of text is transformed to the newly created tag <mytext=>. If only these operations are performed, Cosdes will be the most efficient. We are displaying spam words list with accuracy present in the word compare to neighboring word in data.

5. CONCLUSION

In this paper, we propose Cosdes for Spam detection in email communication The primary idea of the similarity matching scheme for spam detection is to maintain a known spam database, formed by user feedback, to block Subsequent near duplicate spam's. Instead of mapping each subsequence in a node of anSpTree to a hash value using a binary hash function we propose to replace it with a special hash function, namely Simhash. The advantage of this over other hash functions is that it sets a minimum on the number of members that the two sets must share in order to match. This mitigates the effect of extremely common set members on data clusters.

6. REFERENCES

1) A. Kolcz and J. Alspector, "SVM-Based Filtering of Email Spam with Content- Specific Misclassification Costs," Proc. ICDM Workshop Text Mining, 2001.

2) Z. Wang, W. Josephson, Q. Lv, and K.L.M. Charikar, "Filtering Image Spam with Near-Duplicate Detection," Proc. Fourth Conf. Email and Anti-Spam (CEAS), 2007.

3) E. Damiani, S.D.C. di Vimercati, S. Paraboschi, and P. Samarati, "P2P-Based Collaborative Spam Detection and Filtering," Proc. Fourth IEEE Int'l Conf. Peer-to-Peer Computing, pp. 176-183, 2004.

4) Cadwell, L. (1997). "Bringing Reggio Emilia home:An innovative approach to early childhood education.". *Teachers College Press, New York*.

5) J. Hovold, "Naive Bayes Spam Filtering Using Word-Position-Based Attributes," Proc. Second Conf. Email and Anti-Spam (CEAS), 2005.