

Associating Anti-Spam Method in a Secure Overlay Cloud Network

¹K. Phalguna Rao, ²Dr. Vinay Chavan

¹Dept. of CSE, Sri Vasavi Engineering College, Tadepalligudem, AP, India

²Dept. of Computer Science, S.K.Porwal College, Kampat, Nagpur MS, India

Abstract

Cloud computing is the newest term for the long-dreamed vision of computing as a utility. The cloud provides convenient, on-demand network access to a centralized pool of configurable computing resources that can be rapidly deployed with great efficiency and minimal management overhead. The industry leaders and customers have wide-ranging expectations for cloud computing in which security concerns remain a major aspect. Security services such as Intrusion Detection System (IDS), Anti-Virus software, Anti-Spam software and Distributed Denial of Service (DDoS) are used in general cloud-based security overlay network that acts as a transparent overlay network. We focus and analyze various Anti-Spam methods and have a comparative study among them to show their efficiency of providing security in the cloud overlay network.

Keywords

Cloud Computing, Security Services, Anti-Spam, Overlay Network

I. Introduction

Cloud computing offers dynamically scalable and virtualized resources as services over the Internet. Numerous conceptualizations of cloud computing alternatively emphasize utility computing, services oriented computing, or on-demand computing. Some of the benefits are scalability, simplicity of implementation, and lower capital expenditure. Although industry leaders and customers have wide-ranging expectations for cloud computing, privacy and security concerns remain a major impediment to widespread adoption.

A. Infrastructure-as-a-Service

When a vendor rents out infrastructure components on demand—such as servers, storage components, file systems, virtualization technologies, and network hardware—the vendor is delivering an IaaS service. The rented computational resources are usually the core components of an enterprise client's IT services infrastructure. Advances in virtualization technologies have facilitated this type of service, which an enterprise can use as a platform for building applications or to enable entire applications.

B. Platform-as-a-Service

PaaS delivers a computing platform and solution stack as a service. It facilitates the deployment of customer-created applications to the cloud using provider-supported frameworks. Although the consumers don't control the underlying cloud infrastructure, they have control over the deployed applications and hosting environment configurations. Google App Engine is the best example of a PaaS, where both individual users and enterprise clients can build software applications using different Google APIs, deploy them on Google's servers, and manage them through a dashboard.

C. Software-as-a-Service

In this type of a cloud computing solution, a provider's specialized software runs on a hardware cloud infrastructure and is accessible to the customer through a thin client interface such as a Web browser. Customers can configure application settings according to their specific needs. Some examples of online applications that SaaS providers employ are the Clarizen project management tool, Customer Relationship Management (CRM) software, Salesforce.com's human-resource applications, and Absolute Performance's project lifecycle management service.

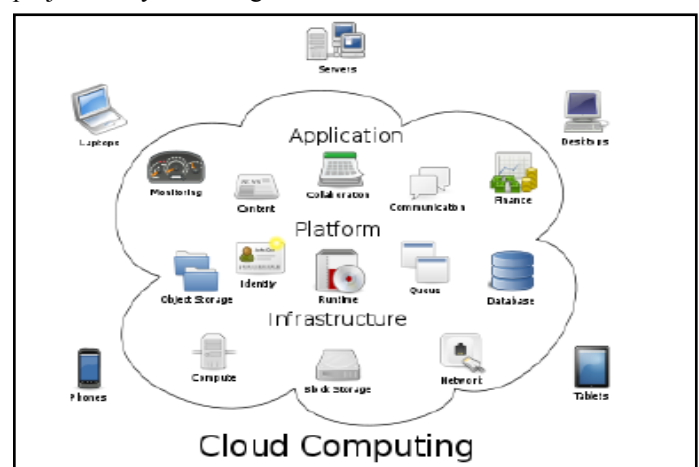


Fig. 1: Cloud Computing Services

D. Overlay Networks

The idea of an overlay network was first used in the deployment of the Internet over the existing telephone network. It can be defined as a virtual network that is built on top of an existing network. It consists of virtual nodes that are connected via virtual links. The main aim of overlay networks is to deploy a new network service that is not available in the underlying network. Cloud computing is mainly based on virtualization which enables multi-tenancy and scalable shared resources used on an on-demand basis by all tenants. Since overlay networks are also based on virtualized nodes for implementing network services, this fact allows combining both techniques to reap the benefits of each other together. This is particularly useful for the deployment of new transparent network security services over the current networks in order to enhance their protection.

E. Security

There is a high possibility of security threats in cloud IT infrastructures, because vendors store critical and confidential data in the cloud. In some cases, the clients require physically or virtually separated data and applications. Cloud providers can invest in better security controls through scale economies, but they can also develop standardized processes for regulatory compliance. Cloud providers endeavour to improve their offerings to meet clients' enterprise grade security needs, but this might not be sufficient in some key sectors. In a recent survey, 64 percent of respondents in the US federal government said security was

their topmost concern in cloud computing. The trust levels toward cloud computing in these sectors have, however, been improving. The launch of the federal cloud services portal for government agencies called Apps.gov is indicative of this shift. Vendors such as Google and Microsoft are close to obtaining accreditation for compliance with the Federal Information Security Management Act, making their cloud computing services acceptable for the public sector.

II. Related Work

A. Introduction to Spam

Spamming is the act of unsolicited (commercial) electronic messages in bulk, and the word Spam has become the synonym for such messages. This word is derived from spiced ham. Monty python's Flying circus used the term spam in the so called "spam sketch" as a synonym for frequent occurrence. Spam is adopted for unsolicited mass mail. Based on the origin of the word spam, all other e-mail is called ham. Official terminology for spam is unsolicited bulk mail or unsolicited commercial e-mail. Spam has severe negative effects on e-mail users. It consumes computer storage and network resources as well as human time and attention to dismiss unwanted messages.

B. Unsolicited Commercial E-Mail (UCE)

E-Mail containing commercial information that has been sent to a recipient who did not ask to receive it (or) unsolicited e-mail is advertising material sent by e-mail without the recipient either requesting such information or otherwise explicitly expressing an interest in the material advertised.

C. Unsolicited Bulk E-Mail (UBE)

Unsolicited Bulk Email, or UBE, is Internet mail ("email") that is sent to a group of recipients who have not requested it. A mail recipient may have at one time asked a sender for bulk email, but then later asked that sender not to send any more email or otherwise not have indicated a desire for such additional mail; hence any bulk email sent after that request was received is also UBE.

D. False Positives and False Negatives Spam

There are four scenarios for an outcome when a spam filter or another countermeasure operates on an email:

- The email is ham and the filter correctly identifies the email as a genuine mail.
- The email is spam and the filter correctly identifies it as such.
- The email is not spam and the filter wrongly identifies the email as spam. This is called a false positive.
- The email is spam and the filter wrongly identifies the email as a genuine mail. This is called a false negative.

E. Different Types of Anti Spam Methods Grey listing

Grey listing is designed to have minimal impact on users and require minimal maintenance. It have to be running on all the email servers published by a domain to be effective. Three pieces of information are used by grey listing.

- The IP address of the host attempting the delivery
- The envelope sender address
- The envelope recipient address

F. Sender Policy Framework

Sender Policy Framework (SPF) is a new system not yet standardized. But major email providers like Gmail and Hotmail has started to adapt it. SPF aims to prevent worms, viruses and spam from forging arbitrary email addresses as the envelope sender in SMTP.

G. Domain Keys

Domain Keys is a new system not yet standardized, but major email providers like Yahoo has started to adapt it. Domain Keys aims to prevent worms, viruses and spam from forging arbitrary sender email addresses. It also aims to detect if some of the other headers and the body are modified after an email has left the sender system.

H. Real Time Black Lists

Real Time Black Lists (RBL) are often distributed via DNS and contains IP addresses of the hosts that are thought to be insecure or send spam. Like the name says, the lists are updating in real time meaning that when an IP address is added, users of the black list are able to benefit from this almost instantly. Each host sending email is checked by the recipient via DNS to see if the host sending the email is black listed. If the host is black listed, the email can be considered spam.

I. Razor

Razor uses a distributed hash database and identifies spam by email content. A method which is related to Razor is described by Deepak and Parameswaran. With Razor, a hash is generated from the content and checked against dedicated Razor servers on the Internet. The Razor servers are updated by users who report the hash of spam they receive.

J. Distributed Checksum Clearinghouse

Distributed Checksum Clearinghouse (DCC) uses the same fuzzy hash algorithms as Razor. Unlike Razor it does not depend on user feedback. Every email server or email client that is DCC enabled reports the message it receives to DCC servers. The DCC servers then count how many similar email message that has been reported and sends this number back to the email server or client when asked. When a high number is sent back to the email server receiving an email, it is a bulk email and can be considered spam.

H. Spam Assassin

Spam assassin is a ruled based spam filter. It has an extensive set of built in rules to identify spam. Each rule is checked against an email to determine whether it is spam. When a rule matches the email, a number of points determined by the rule are added to a total sum for the email. When every ruled is checked the sum is compared to a user or default defined threshold. A number higher or equal than the threshold means that the message is spam and a number lower than the threshold means that the message is ham.

III. Analysis

In order to find out how effective spam filters and countermeasures work we present a algorithm for finding spam in the network

A. Methodology

The algorithm definition defines self to be equal to the collection of element in a feature space U.

U is represented by list of features which corresponds to the space of states of a system where

S=Subset of space that are considered as normal for the system.

R= Set of detectors generated.

B. $R \neq S \rightarrow R$ Fails to Match any String in S

This approach analyzes happening in Negative immune system by generating random detectors and discard those that match any element in the self set. Continually matching S for changes with detectors R against S if S ever matches R. Change is known to have occurred as detectors are not suppose to match any string in S. The higher the possibility of a spam, when the probability is higher in calculating the probability of occurrence of a given value. Let's make vector of the feature as the set of the system which is the system state space.

$$a^t = (a_1^t \text{-----} a_b^t) \in [u, 1]^b$$

Each state of the feature been represented by a set

$$U [0, 1]^b$$

This includes the feature vectors which correspond to all the probability state of the system.

U = Universal set of all the system.

0 and 1 represent system being a self or non-self.

This value represent degree of either it is a spam or not a spam. Where by 1 indicates that it is not a spam and 0 indicates that it is a spam. The intermediate value represent element with some degree of being a spam or non spam.

$$a_{\text{self}} : [0, 1]^b \rightarrow [0, 1]$$

$$a_{\text{self}} : (a) \{10 \text{ if } a \text{ is self } 1, \text{ if } a \text{ is non self } 0\}$$

The training data set is divided in to self detector and nonself detector (x,y) respectively using the suspicious spam extractor.

$$\text{Affinity}(x,y) \sum_{\text{match}} = (x,y) l, / x \neq 1$$

x = Affinity between memory-self x

y = Affinity between memory-non self y

l = length of the memory-self x and the memory non self y

Match (x,y)

$$\begin{cases} 1, \sum_{i=0}^n \text{much}(x \rightarrow x_i y) / l, / x \neq l \\ 0, \end{cases}$$

Match (x,y)

$$\begin{cases} 1, \sum_{i=0}^n \text{match}(y \rightarrow y_i x) / l, / y \neq l \\ 0, \end{cases}$$

Where, the Matching of x is the total match value between memory-self x

And the Matching of y is the total match value between memory-non-self y

The dataset is given as:

$$S = \{ \langle x, y \text{ affinity} \rangle \mid x, y \in \bigcup_{i=1}^n H, \text{affinity} \in N \}$$

Affinity is the ability of the self to match with both self and non-self

This exists when memory-self and memory-non self are generated.

The set is divided in to self training set and non-self training set. Let TR denotes training set

$$\text{TR}(x) = \begin{cases} 0; t = 0 \\ \text{TR}(t-1) + \text{TRnew}(t), t \geq 1 \end{cases}$$

$$\begin{aligned} \text{TRnew}(t) &= \left\{ X \in AG, \int_{\text{match}} (y \cdot x) \right\} \\ \text{TRnew}(t) &= \left\{ X \in AG, \int_{\text{match}} (x \cdot y) \right\} \end{aligned}$$

Therefore; for threshold value of x and y we have

$$\begin{aligned} \int_{\text{match}} (y) &= \begin{cases} 1, \int \text{affinity}(y) / l > a \\ 0, \end{cases} \\ \int_{\text{match}} (x) &= \begin{cases} 1, \int \text{affinity}(x) / l > a \\ 0, \end{cases} \end{aligned}$$

Represent the threshold value.

Also, ability of self matching self and non-self we have:

$$\int_{\text{affinity}} (x) = \max(y_1, y_2, \dots, y / l - l_x / + 1)$$

$$\int_{\text{affinity}} (y) = \max(x_1, x_2, \dots, x / l - l_y / + 1)$$

$$x_i = \sum_{j=1}^{\min(l, l_y)} \theta_{i,j}$$

$$y_i = \sum_{j=1}^{\min(l, l_x)} \theta_{i,j}$$

$$\theta_{ij} = \begin{cases} 1, x_i = y_{i+j}, 1 \leq i \leq l - l_y / + 1, 1 \leq j \leq j \\ 0, \end{cases}$$

$$\theta_{ij} = \begin{cases} 1, y_i = x_{i+j}, 1 \leq i \leq l - l_x / + 1, 1 \leq j \leq j \\ 0, \end{cases}$$

f_{match} is the match function of self and non-self. Affinity is the affinity function of self and non-self.

L is the length of self and non-self

If the affinity between self and non-self is greater than the threshold value, then returns to 1, otherwise it returns to 0;

The bigger the affinity, the higher the match value between memory-self, memory-non-self, self and non-self, and the more possibility of self or non-self were a spam

IV. Analytical Results

A. Anti Spam Methods Comparison

Comparing the results of the various methods gives valuable information about how methods can be combined to give a more accurate detection rate. Bogofilter, SPF and Domain Keys all had such a small spam detection ratio that they are left out of the comparison.

Table 2: Anti Spam Methods and Their Detected Spam Percentage

Anti spam method	Detected spam percentage
Razor	74.6%
Greylist	71.3%
All RBLs combined	69.5%
Spamassassin	66.8%
DCC	58.7%
Bogofilter	31.7%
SPF	3.1%
DomainKeys	0.1%

Table 2: The notation Indicates other Possible Values of Spam or Ham

Column name	Possible values
Id	unique ID
Date	when email first seen
relay_	ip IP address email
status_human	came from
status_greylist	spam, ham, don't
status_spamassassin	know
status_rbl_spamcop	spam, ham, white-
status_rbl_spamhaus	listed
status_rbl_ordb	decimal number
status_rbl_njabl	empty, in list
status_rbl_sorbs	empty, in list
status_rbl_dsbl_list	empty, in list
status_rbl_dsbl_multihop	empty, in list
status_rbl_dsbl_unconfirmed	empty, in list
status_spf	empty, in list
status_domainkeys	empty, in list
status_razor	empty, in list
status_dcc	fail, pass,
status_bogofilter	bad, good,
Message count	spam, ham
	spam, ham

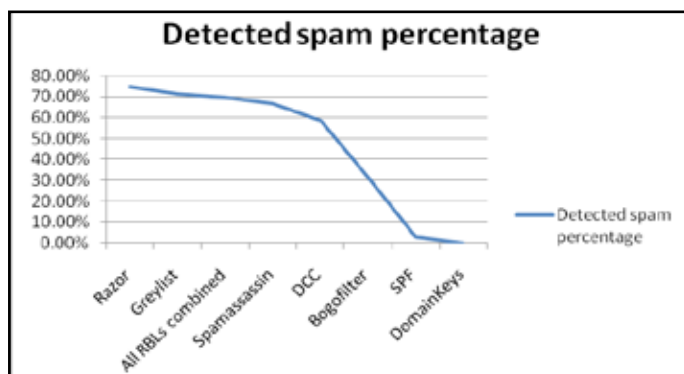


Fig. 2: Anti Spam Methods and Their Detected Spam Percentage

Table : Anti Spam Methods Compared with user Feedback.

Anti spam method	FNR	FPR
Razor	0.259	0.025
All RBLs combined	0.246	0.015
Spamassassin	0.573	0.031
DCC	0.553	0.288
Bogofilter	0.041	0.000
SPF	0.987	0.000
Domain Keys	1.000	0.118

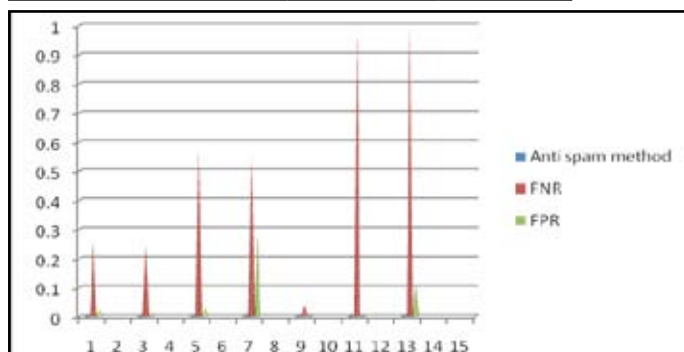


Fig. 3: Anti Spam Methods Compared With User

V. Conclusion

We have studied different anti spam solutions and the work that others have done regarding anti spam solutions. We have seen design implementation of various anti spam methods perform and compares to each other on e-mail data. The results have been analyzed and presented. The main methods in the experiment detected that about 37% to 75% spam had a low false positive ratio. Requiring several methods before deciding that an email is spam still gives a high spam percentage. This confirms regarding spam traffic on the Internet. Also, some methods claiming to have a low false positive ratio does. It is impossible to conclude that any method have a zero false negative ratio though.

References

- [1] W. Gansterer, M. Ilger, P. Lechner, R. Neumayer, J. Strau. "Anti Spam Methods – State-of-the-Art", University of Vienna, Austria March 2005
- [2] P. Deepak, Sandeep Parameswaran, "Spam filtering using spam mail communities", In Proceedings. The 2005 Symposium on Applications and the Internet, pp. 377–383, 2005.
- [3] Lov om behandling av personopplysninger. [Online] Available: <http://www.lovdato.no/all/nl-20000414-031.html>.
- [4] S. Mane, J. Srivastava, San-Yin Hwang, J. Vayghan, "Estimation of false negatives in classification", In Proceedings. Fourth IEEE International Conference on Data Mining, pp. 475–478, 2004.
- [5] R. Matsumoto, Du Zhang, Meiliu Lu., "Some empirical results on two spam detection methods", In Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, pp. 198–203, 2004.
- [6] Brian McWilliams. Spam Kings. O'Reilly Media, Inc., 2004.
- [7] Monty Python's Flying Circus, "Just the Words, Vol. 2, chapter 25, pp. 27–28. Methuen Publishing Ltd, 1989.
- [8] Cormac O'Brien, Carl Vogel, "Spam filters: bayes vs. chi-squared; letters vs. words", In ISICT '03: Proceedings of the 1st international symposium on Information and communication technologies, pp. 291–296. Trinity College Dublin, 2003.
- [9] T. Oda, T. White, "Increasing the accuracy of a spam-detecting artificial immune system", The 2003 Congress on Evolutionary Computation, 1, pp. 390–396, December 2003.
- [10] L. Pelletier, J. Almhana, V. Choulakian, "Adaptive filtering of spam", In Proceedings. Second Annual Conference on Communication Networks and Services Research, pp. 218–224, 2004.
- [11] S.L. Pfleeger, G. Bloom, "Canning spam: Proposed solutions to unwanted email", IEEE Security & Privacy, 3(2), pp. 40–47, Mar-Apr 2005.
- [12] Jeffrey Posluns, "Inside the Spam Cartel", Syngress Publishing, Inc., 2004.
- [13] Jon Postel, "RFC706: On the Junk Mail Problem", Technical report, Network Working Group, November 1975.
- [14] Vipul Ved Prakash, "Razor: spam should not be propagated beyond necessity", World Wide Web, 2005. [Online] Available: <http://www.razor.sourceforge.net>
- [15] Editor P. Resnick, "RFC2822: Internet Message Format", Technical report, QUALCOMM Incorporated, April 2001.
- [16] Matt Sergeant, "Internet-level spam detection and spamassassin 2.50", In Spam Conference at Cambridge MA, 2003.

- [17] Rhyolite Software, "Distributed checksum clearinghouse", World Wide Web, 2005. [Online] Available: <http://www.rhyolite.com/anti-spam/dcc/>
- [18] N. Soonthornphisaj, K. Chaikulseriwat, Piyanan Tang-On, "Anti-spam filtering: A centroid-based classification approach", 2002 6th International Conference on Signal Processing, 2, pp. 1096–1099, August 2002.
- [19] Brian Whitworth, Elizabeth Whitworth, "Spam and the social-technical gap", Computer, 37(10), pp. 38–45, 2004.
- [20] Meng Weng Wong. Spf overview. Linux J., 2004(120):2, 2004.
- [21] Yahoo! Inc. Domainkeys: Proving and protecting email sender identity. World Wide Web, 2005. [Online] Available: [http://antispam.yahoo.com /dom -ain keys](http://antispam.yahoo.com/dom-ain-keys) (Visited January 2005).
- [22] Kenichi Yoshida, Fuminori Adachi, Takashi Washio, Hiroshi Motoda, Teruaki Homma, Akihiro Nakashima, Hiromitsu Fujikawa, Katsuyuki Yamazaki, "Density-based spam detector", In KDD '04: Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 486–493. ACM Press, 2004.
- [23] Le Zhang, Jingbo Zhu, Tianshun Yao, "An evaluation of statistical spam filtering techniques", ACM Transactions on Asian Language Information Processing (TALIP), 3(4), pp. 243–269, 2004.