



Detection of Spam messages and Spam zombies in the Internet using Naïve Bayesian and SPRT

¹M.Vasu, ²K Munivara Prasad, ³Dr K Venugopal Rao

¹Dept. of CSE, SVEC, Tirupati, India

²Faculty, Dept. of CSE, SVEC, Tirupati, India

³Professor, Dept. of CSE, GNITS, Hyderabad, India

Abstract: Security attacks are one of the major threats on the Internet. The attacker uses the compromised systems or bots to launch various security attacks such as spamming and spreading malware, DDoS, and identity theft. Spamming refers the generation of huge volume of messages from the unauthorized persons in the internet by using the Bots. Identification of spam messages increases the performance of the server and detection of the spam source or spam zombies blocks the compromised systems in the internet from being transformed the spam messages permanently. In this paper we proposed Naïve Bayesian approach of the content based method for detecting the spam messages and used SPRT algorithm for identifying the compromised systems in the internet. We compared the results of our approach with existing key word based method and proved that the detection accuracy of spam messages with proposed method improves the detection accuracy of Compromised systems in the internet.

Keywords: Security attacks, Spam zombies, keyword based method, Naïve Bayesian approach.

I. Introduction

In today's computing world, internet plays an important role in our daily lives in almost every aspect. It is the place where we do lot of things just sitting at one place. Internet not only influences the people to do positive works but also influences the people to trouble others by posing many attacks. These attacks are posed by the attackers directly or indirectly.

Attacks are generally of two types, one of them is automatic attacks and the other type is manual attacks. Most of the successful attacks are from the automated generated code injected by the attackers. These are very dangerous some of them are Dos, DDos, E-mail Worms, Viruses, Worms, Trojan horses, phishing attacks etc..

Attackers control some machines to attack the target machine. These machines are called drones, zombies or compromised machines. Zombies search for the low level secured systems to infect them and can control them through their pre-defined commands to cause an (DDoS) attack. In spamming terminology those are called as spam zombies. It is given that spamming is the major security challenge in the email communication. Report of 2012 march says that more than 75% of all email traffic is occupied by the spam [10]. To detect these spam zombies is tough job for the system administrators.

Spamming is an important threat plaguing the internet from the past decades. More than 75% of traffic is spam and in that 0.4% was malicious [11]. It is done by controlling several hosts to send unwanted messages to some target machines. These compromised machines are called spam zombies. Normally spam is given as UBE/UCE i.e., Unsolicited Bulk or Commercial E-mail. Spam message is an unwanted message to the users because of these reasons. They occupy the network bandwidth, disk space, connection time, money. They could hide viruses inside spam message, can send pornography information and can tempt the users to send their money and the confidential details. E-mail spamming became the major platform for the attackers because of its unique behavior of low cost and high speed. It is given that spamming is the major resource for the attackers to get the incentives. They are earning around \$200 billion dollars per year. In other words it is the cheapest one to one means of communication available today. That is why spamming is attracting the most of the attackers day by day.

Spam filtering is a technique that classifies a message into two categories (good and spammed message). Filter can misclassify the message to generate false positives and false negatives. False positive is the misclassification of good message as a spam message and false negative is the misclassification of spam message as a good message. Effective spam filter aims to minimize the false positive percentage. There are many methods available to filter out the spam. Spam filter can detect the spam at 3 levels. Those are at Network level, user level and policy based technique.

Network level technique: It is very difficult and important to detect the spam at this level. The most important approaches are Domain verification and the Challenge Response Systems. The former technique uses to filter the messages by considering the sender, domain, route information obtained by the SMTP. Example is one cannot send the message to the incoming route in a network path. Whereas the later uses the technique that it probes the

sender by asking questions to ensure that the sending side is not an automated bot. By this technique huge number of messages that are sent from the automated bots can be reduced.

User level technique: the two popular methods used at this level are content based and the white and blacklists (parameter based).

Policy based Technique: This is not a technical approach. Through pricing e-mail, accusing the spammers by law some spamming can be reduced.

Spam zombie is a compromised machine that sends the spam messages to victim hosts. To detect the spam zombies at one level is not sufficient. So we are giving the idea to detect the spam at one level and the compromised machine at the other level. Related work regarding spam as well as machine detection is discussed in Section II. Proposed SPRT technique is discussed in Section III. Spam and Machine detection techniques are discussed in Section IV. We have given the experimental results in section V and concluded in Section IV.

II. Related Work

In related work, we are going to discuss the techniques that are used to detect zombies as well as spam classification. In spam classification, the existing technologies related to spam categorization are very complicated and gives poor results when compared to Naïve-Bayesian spam filter [12].

In zombie detection, first we discuss the spam zombie detection related work. [1] detects the email spam bots by aggregating the spam records and clustering them by similarities in the content of the email messages by shingling[2] algorithm. Yu.et.al discussed the framework called AutoRE which works by considering the URLs to categorize botnets [3]. Here AutoRE considers the URLs as signatures and regular expressions to create spam campaigns. But the above discussed techniques need to have the input of large number for observations.

In general spam botnet detection, many techniques came. Some of them are discussed here. Choi et.al proposed a technique to detect the bots based on the DNS queries generated by them [4]. Based on the similarity in the group activity of the DNS traffic the bots are detected in this paper. In [8] the botnets are detected based on the passive analysis on flow data. Xie et al. developed an effective tool named DBSpam to detect proxy-based spamming activities in a network relying on the packet symmetry property of such activities [5]. We intend to identify all types of compromised machines involved in spamming, not only the spam proxies that translate and forward upstream non-SMTP packets (for example, HTTP) into SMTP commands to downstream mail servers as in [5]. BotHunter uses the IDS trace to detect the bots by comparing the inbound intrusion alarms with the outbound communication patterns [6]. The proposed SPRT algorithm to detect spam zombies focuses on any spamming activity unlike BotHunter which depends on specifics of malware infection process. The algorithm (SPRT) we are using in this paper has been used in many areas [7]. Jun-Won Ho et.al is used SPRT algorithm to detect mobile replica node attacks in VANETS [9]

III. Proposed Work

A. Bots

The term bot is originated from the word robot. Bot is an automated system that can be controlled by a particular botmaster or a botnet to attack other machine. The targeted machine is called the target or victim machine. Attacker can install a malware to control the affected host. Bot may inject or send minimal traffic to the target machine but in turn it may be a part of the substantial traffic flow attacking to the target machine. Generally botmaster gains the control over an individual bot if the security levels of the bot are very low. Likewise bot groups also searches for the poor security entry points (hosts or victims) to destruct the other hosts.

What Bots do

- Uses to propagate the malware,
- Can consume the network traffic
- Harvest the usernames and the passwords
- Uses p2p networks to propagate the malware
- Uses spam messages to get the incentives

Types of the Bots:

AgoBot: It is the famous bot where more than 500 versions of this bot are available. It is developed in C++ and put source code of the written C++ code in GPL. It has the advantage of easy commands and scanners addition by extending the CCommandHandler and CScanner class and add your methods to them and hard to do reverse engineering. It is the protocol that uses other than the IRC. It is used for packet sniffing and sorting the traffic.

SDBot: It is developed in c language. Since it is written with poor coding many attackers attracted to use it.

B. Botnet

Botnet is a collection of computers that are controlled by a botmaster through some commands to forward malicious code, viruses, or spam. The compromised machines do not know that some other computer is controlling that system. Normally home based, less secured systems will be hosted for these purposes. Botmaster can completely consume the network since it attacks more number of high speed broad band connection computers. So that it causes the DDoS attack. Or it may send number of spam messages to victim machines to earn good incentives. Botmaster can send the commands through IRC channels or using other private tools to the

bots to execute their commands in the bots automatically. The simple commands can scan for the other bots to populate the botnet or can pose threats to the victim machines to fulfill its desire

C. Spam Messages

Spam messages are the special purpose messages to attract the users and to make the user to listen to their fraud words. Like normal message, it also contains the structure of header and body. Where header contains the details of the message and the body contains the content of the message including the subject. Many spam messages have some unique properties that distinguish with the other messages. Those are given below (for header portion): Empty To: field, missing To: field, more number of recipients in the Cc: field, no or suspected message ID's, Bcc: field exists, same but fake addresses usage in From: and To: fields etc. To detect the content of the spam messages many spam filtering algorithms are available those are given below.

D. Spam Filtering Methods

Spam Filtering Methods can be broadly classify into two types one is content based and the other is parameter based spam filtering.

Content based filtering:

These are the methods that are used to classify a message as either spam or normal by considering the body of the message. Content based algorithms plays major part because spammer can hide his identity and can enter into the victim's network easily. Later it has to pass through the spam filter which will be its last stage of his goal. Any spam message must be well written to read a person and understand the content easily. That is the content is the heart of the spam message. Some of the content based spams filtering algorithms are Bayesian, SVM, KNN, Naïve-Bayes etc.

Parameter based filtering:

These are the methods that are used to classify a message as either spam or non spam by considering the parameters of the message. Parameters of a message include From, To, Received by, Subject, IP address etc. By using parameter based filtering most of the spam messages will be filtered. Most of the messages will be filtered by checking the parameters of the message. Some of the parameter based filtering techniques are Blacklists, Whitelists, Challenge/Response methods etc.

Blacklists are the IP addresses/domain names/e-mail addresses of the real time spammers. In real world many black lists are available. These are called Realtime Black Lists (RBL). By comparing the RBL one can filter out the spammer messages undoubtedly. In the same way whitelists are the IP addresses/domain names/e-mail addresses of the trusted parties. To maintain these two techniques by an independent user is a tough job while most of the organizations follow them.

E. Content Based Spam filtering Methods

Content based spam filtering uses the content ie, the body of the message to decide a message as spam. These methods depend on the training of the previous messages.

Naïve-Bayesian Algorithm:

It is a probability based algorithm to decide a message as spam or normal message. Naive bayes approach is slight modification to the bayesian algorithm. It is the most effective spam filtering content based algorithm. More detailed description is given in 3.6.

K-Nearest Neighbors:

This method of classification is based on the distance measure among the messages. The distance can be measured based on the features between the messages by equations like Euclidean distance measurement. This method doesn't need training phase where the incoming messages will be directly measured with the available sample messages. So the time complexity of each message is of $O(n)$.

Bayesian Classifier:

One of the best spam filters available today is bayes classifier, which is based on the probabilistic method of classification. This is the assumption that the previous spam message features can be occurred in the future also. This basic principle is used to classify the message as either spam or normal. That means the previously classified spam message words can occur in the spam message more frequently than in the normal message. In this method each word probabilities are calculated in the training phase. After calculating the word probabilities, the message will be classified based on the combination of the words that are commonly presented in the training and in the given message.

In spam we have two only classes (SPAM or HAM). To know a message is in which class we use the probability distribution equation as

$$P\left(\frac{c}{x}\right) = \frac{p\left(\frac{x}{c}\right) \cdot p(c)}{p(x)}$$

Here $p(x/c)$ is the probability of the message with feature X from the class C. $p(c)$ is the a-priori probability of class C and $p(x)$ is the a-priori probability of message (X).

By calculating the equation, the probability of a message will be known. Normally spam message contains more probability (>0.8) than a normal message.

F. Naïve Bayesian spam filter

It is a spam filter that is based on the working principle of the Bayesian spam filter. Normal Bayesian with the independent features assumption in the calculation of conditional probability is called the Naïve Bayesian spam filter. Here the features in the text is considered independent among each other even though there is an inter dependency. For example, in a phrase “HURRY UP” the two words HURRY and UP is interdependent. That is after the occurrence of the word HURRY there are more chances to come the word UP than any other word. Fortunately the practical results are shown that the dependency will not play a major role in the decision of the spam classification. This is the main idea behind the naïve Bayesian spam filter.

G. SPRT (Sequential Probability Ratio Test)

It is a statistical approach of testing between two hypothesis one is null and the other is alternative hypothesis (H_0, H_1). Based on the observations, a one dimensional random walk will be moved between two boundaries (A, B). Whenever the variable value touches either of the boundaries the test is stopped and the corresponding result is considered according to the boundary. This can be illustrated as following

$$\Lambda_n \leq A \rightarrow \text{Accept } H_0 \text{ and stop the test}$$

$$\Lambda_n \geq B \rightarrow \text{Accept } H_1 \text{ and stop the test,}$$

$$A < \Lambda_n < B \rightarrow \text{Take additional observation and continue the test.}$$

The boundaries are calculated by the user-desired false positives (α) and the false negatives (β). False positive is the case where the algorithm accepts H_1 when H_0 is true. False negative is the case where the algorithm accepts H_0 when H_1 is true.

$$\Lambda_n = \ln \frac{\Pr(X_1, X_2, X_3, \dots, X_n | H_1)}{\Pr(X_1, X_2, X_3, \dots, X_n | H_0)}$$

Λ_n is the nth observation which is calculated in the below form:

$$\Lambda_n = \ln \frac{\prod_{i=1}^n \Pr(X_i | H_1)}{\prod_{i=1}^n \Pr(X_i | H_0)} = \sum_{i=1}^n \ln \frac{\Pr(X_i | H_1)}{\Pr(X_i | H_0)} = \sum_{i=1}^n Z_i$$

Here X_i is the Bernoulli variable, which are independent and identically distributed. We can denote the probability of an observation coming from the H_0 as θ_0 and H_1 as θ_1

$$\begin{aligned} \Pr(X_i = 1 | H_0) &= 1 - \Pr(X_i = 0 | H_0) = \theta_0 \\ \Pr(X_i = 1 | H_1) &= 1 - \Pr(X_i = 0 | H_1) = \theta_1 \end{aligned}$$

$$\sum_{i=1}^n Z_i \text{ Means if the observation is } X_i = 1, \ln\left(\frac{\theta_1}{\theta_0}\right) \text{ is added to the variable } \Lambda \text{ otherwise } \ln\left(\frac{1-\theta_1}{1-\theta_0}\right)$$

is added to the variable Λ . Now after adding each observation boundaries are checked. Whenever it crosses the boundaries, the test will be stopped.

According to the Wald's the boundaries calculated by the user-desired false positives and the false negatives as

$$A \geq \ln\left(\frac{\beta}{1-\alpha}\right), B \leq \ln\left(\frac{1-\beta}{\alpha}\right)$$

For practical purpose we consider the above equations as

$$A = \ln\left(\frac{\beta}{1-\alpha}\right), B = \ln\left(\frac{1-\beta}{\alpha}\right)$$

This consideration will be almost negligible because the with the true false positives and false negatives we can write the equation as

$$\alpha_1 \leq \frac{\alpha}{1-\beta}, \beta_1 \leq \frac{\beta}{1-\alpha}$$

And $\alpha_1 + \beta_1 \leq \alpha + \beta$

So true and user desired false positives and negatives are almost similar.

Another important computation that can be obtained from the SPRT test is the number of observations to reach a machine compromised or normal. We can compute the average number of observations by

$$E[N|H_1] = \frac{\beta \ln \frac{\beta}{1-\alpha} + (1-\beta) \ln \frac{1-\beta}{\alpha}}{\theta_1 \ln \frac{\theta_1}{\theta_0} + (1-\theta_1) \ln \frac{1-\theta_1}{1-\theta_0}}$$

$$E[N|H_0] = \frac{(1-\alpha) \ln \frac{\beta}{1-\alpha} + \alpha \ln \frac{1-\beta}{\alpha}}{\theta_1 \ln \frac{\theta_1}{\theta_0} + (1-\theta_1) \ln \frac{1-\theta_1}{1-\theta_0}}$$

The number of observations will depend on the user defined values of the four parameters $\theta_0, \theta_1, \alpha, \beta$. System administrators have to be very careful when providing these 4 parameter values.

IV. Spam & Zombie Detection

To detect a spam zombie, first we must detect the messages coming to the server are spam or normal messages. Later if we apply the SPRT algorithm, we can able to effectively stop the spam compromised machines. So here we will phase two steps in this process one is SPAM detection and the other is Zombie detection.

Phase-I (SPAM DETECTION):

In this phase we are going to discuss to spam filtering algorithms that we are comparing to detect compromised machine. Those are Keyword based and Naïve-bayesian.

Keyword based:

This is one of the machine learning techniques used to classify the messages. In this two phases are there. In training phase the text is tokenized, stemmed and the stop words will be removed. The spam word count will be calculated.

Based on the spam word, the weight of each word using TF-IDF is calculated with the below formula.

$$w_{ij} = tf_{ij} \cdot \log\left(\frac{N}{df_i}\right)$$

Here tf_{ij} means term frequency ith word from jth mail. N is the total messages count. df_i is the number of messages in which ith term occurred.

In testing phase, the tokens are taken same as in training. The tokens that are matched with the database are retrieved and message threshold is calculated to classify a message.

Naïve-Bayesian:

There are two phases in the naïve bayes filtering process. Those are the training phase and the classification phase. In training phase, the equal number of spam and normal messages is trained to get the probabilities of the each and every occurred word. These are stored for lateral retrieval purpose. In classification phase the message is made divided into tokens and spammicity and hammicity will be calculated based on the previously calculated word probabilities.

In training phase, Probability of a token is calculated by the formula

$$S_{token} = \frac{S_Count_{token}}{S_Count_{token} + H_Count_{token}}$$

Where S_{token} is the spammicity of a token, S_Count_{token} is the number of spam messages that contain this token. H_Count_{token} is the number of ham messages that contain this token.

In classification phase, the message total spammicity and hammicity are calculated by this formula

$$S_{message} = \prod_{i=1}^n S_{token_i}$$

Where $S_{message}$ is the spammicity of a message. hammicity can be calculated by the product of the hammicity of the each and every token.

$$H_{message} = \prod_{i=1}^n (1 - S_{token_i})$$

Here $H_{message}$ is the hammicity of a message. Hammicity can be directly calculated by using the spammicity of a message. Since spammicity and hammicity of a message are opposite in nature, the total probability will be 1 (because of two classes).

Algorithm:

Training phase:

- Take equal no. of spam and legitimate messages
- Tokenize each and every message
- Calculate the spammicity (S_{token}) and hammicity ($1-S_{token}$) of each token
- Store the results

Classification phase:

- Tokenize the message
- Retrieve the probabilities of the tokens
- Calculate the total spammicity of the message

Phase-II (ZOMBIE DETECTION):

Any message must pass through the server in a network. Each generated message will have sending machine Ip address. If we block the compromised machine we can stop the spam messages in that particular network.

In spam zombie detection, the above explained SPRT algorithm is used. Here the hypothesis is either compromised machine (H_0) or normal machine (H_1). The observations are the messages generated by the machines. The random walk of a variable (Λ) is based on the messages generated by machines (M).

In a network, a compromised machine will generate more number of spam messages than a normal machine. That is the probability of a spam message coming from a compromised machine is more than (the probability a spam message coming from) a normal machine.

$$\Pr(X_i = 1|H_1) > \Pr(X_i = 1|H_0)$$

Algorithm:

- 1: Record the IP address of a message sending machine.
- 2: Get the all 4 parameters ($\theta_0, \theta_1, \alpha, \beta$)
- 3: Let n be the new observation
- 4: Assume spam message as $X_n = 1$ and the normal message as $X_n = 0$
- 5: if ($X_n = 1$) //Spam msg
- 6: $\Lambda_n = \Lambda_n + \ln(\frac{\theta_1}{\theta_0})$
- 7: else //Non-spam msg
- 8: $\Lambda_n = \Lambda_n + \ln(\frac{1-\theta_1}{1-\theta_0})$
- 9: end if
- 10: if ($\Lambda_n \geq B$) then
- 11: machine is compromised
- 12: else if ($\Lambda_n \leq A$) then
- 13: machine is normal, take new observations
- 14: $\Lambda_n = 0$
- 15: else
- 16: Test continues with new observations.
- 17: end if

The explanation for the above algorithm is as follows. First IP address of a message sending machine is recorded. System administrator sets the 4 parameters according to his network conditions. Each message coming from the machine is an observation in the random walk over two boundaries A and B.

In that, if observation is a spam message ($X_i = 1$), is $\ln(\frac{\theta_1}{\theta_0})$ added to the variable Λ otherwise

$\ln(\frac{1-\theta_1}{1-\theta_0})$ is added to the variable Λ .

$\Lambda_n \leq A \rightarrow$ Accept the machine as Normal machine() and terminate the test,

$\Lambda_n \geq B \rightarrow$ Accept the machine as Compromised machine and terminate the test,

$A < \Lambda_n < B \rightarrow$ Take additional observation and continue the test.

Boundaries (A and B) are calculated with the user-desired false positives and false negatives will not affect the true false positives and false negatives. Because of the fractional values (ranges from 0.01 to 0.05) of the false positives and the false negatives we will not get errors in real time.

V. Experimental Results

To know the effectiveness of the two spam filters that are running with NB and Keyword based algorithms, we had run them by passing the same messages (Both spam and normal mails). The results are shown in the below table. TDR is the percentage of the total number of misclassified messages. FPR is the percentage of misclassified false positives. FNR is the percentage of misclassified false negatives. The above values of the table are the percentages. We tested the accuracy of the two spam filters by giving inputs in 3 cases. The results are showing that the Naïve Bayesian correctly classifying the messages than keyword based algorithm. Accuracy of the NB algorithm is increasing with the training. Since the spam messages are dependent on the local network. The more we get the spam mails the accuracy will be more. In the machine detection phase, the obtained messages that are passed through the spam filters are passed to the SPRT algorithm. This algorithm classified the machines as compromised or normal based the parameters given by us. Fortunately if we give the false positives and false negatives as 0.01 the algorithm had detected the compromised machine with only allowing 2 messages as spam. That is when a machine sends with 3 spam messages is identified as a compromised machine. To test the sample messages we have taken the LingSpam corpus of messages. We have got around 98% of accuracy, false positives as 0.01% and false negatives as 0.02%. To detect a compromised machine we (System administrator)

Input Method ⇨ ⇩	Case I: 50,000			Case II: 1,00,000			Case III: 3,00,000		
	TD	FP	FN	TD	FP	FN	TD	FP	FN
Keyword Based	9.6	18.54	1.14	11.9	21.79	2.35	10.86	20.17	1.28
Naïve-Bayesian	1.6	1.1	2	1.5	0.62	2.3	0.93	0.53	1.33

must pass 4 parameters. They are Normal machine spam messages sending probability (θ_0), compromised machine spam messages sending probability (θ_1), false positive and false negative rates. Normally the range of the 4 parameters is like this: θ_0 is 0.1 - 0.2, θ_1 is 0.8 - 0.9, false positives and negatives are around 0.05 - 0.001. θ_0 Value is from 0.1-0.2 means there are chances that 10-20 spam messages can come from normal machine. When passing through the 1st spam filter we observed that the messages that are misclassified are more in number and the impact on the machine detection is more. That is when false positive is occurred for 3 times there is chance to consider the machine as compromised. Since it has sent only ham messages the machine is considered as compromised is a blunder. So the results are showing that to confirm a machine as compromised or normal, it must first pass through a good spam filter that minimizes false positives and false negatives.

Table I: Comparison of Keyword based and Naïve-Bayesian Spam filters with different sample messages

T.D-Total Detection Rate
F.P-False Positive Rate
F.N-False Negative Rate

VI. Conclusion

Spam zombies are the major problem in the Internet. It is increasing day by day very rapidly. To detect any machine as compromised, we must first classify the messages coming to that particular machine correctly and then detect the machine. In this paper we have implemented the two spam filters based on keyword based and naïve-Bayesian algorithm to classify a message as spam or non-spam and we have shown that Naïve-Bayesian works efficiently and gives more results when we train more number. We proved that detecting a compromised machine depends on false positives and false negatives generated by the deployed spam filter with SPRT algorithm.

REFERENCES

- [1] L. Zhuang, J. Dunagan, D.R. Simon, H.J. Wang, I. Osipkov, G. Hulten, and J.D. Tygar, "Characterizing Botnets from Email Spam Records," Proc. First Usenix Workshop Large-Scale Exploits and Emergent Threats, Apr. 2008.
- [2] BRODER, A. Z., GLASSMAN, S. C., MANASSE, M. S., AND ZWEIG, G. Syntactic clustering of the web, WWW'97.
- [3] Y. Xie, F. Xu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, "Spamming Botnets: Signatures and Characteristics," Proc. ACM SIGCOMM, Aug. 2008.
- [4] Hyunsang Choi, Hanwoo Lee, Heejo Lee, Hyogon Kim, "Botnet Detection by Monitoring Group Activities in DNS Traffic", IEEE pp. 715-720, 2007
- [5] M. Xie, H. Yin, and H. Wang. An effective defense against email spam laundering. In ACM Conference on Computer and Communications Security, Alexandria, VA, October 30 - November 3 2006.

- [6] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. "BotHunter: Detecting malware infection through ids-driven dialog correlation". In Proc. 16th USENIX Security Symposium, Boston, MA, Aug. 2007.
- [7] G. B. Wetherill and K. D. Glazebrook, "Sequential Methods in Statistics. Chapman and Hall", 1986.
- [8] Anestis Karasaridis, Brian Rexroad, David Hoeftin, "Wide-scale Botnet Detection and Characterization", International Journal of Computer Applications, Volume 34 - Number 9 2011
- [9] Jun-Won Ho, Matthew Wright, Sajal K. Das, Senior Member, IEEE "Fast Detection of Mobile Replica Node Attacks in Wireless Sensor Networks Using Sequential Hypothesis Testing" IEEE TRANSACTIONS ON MOBILE COMPUTING, pp.767-782 VOL. 10, NO. 6, JUNE 2011.
- [10]http://www.securelist.com/en/analysis/204792226/Spam_report_March_2012.
- [11]http://www.securelist.com/en/analysis/204792154/Spam_report_November_2010.
- [12] Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos and Constantine D. Spyropoulos, "An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering with Personal E-mail Messages", In Proc of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp: 160-167, July 2000.