ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

A Distributed System for Detecting Phishing in Twitter Stream

Manju .C.Nair ,S.Prema (PhD)

Abstract: Phishing is a long-running problem that has taken a turn for the worse. Phishing tweets now so closely resemble legitimate ones, making it very difficult both for users and automated systems alike to tell them apart. As such, users end up clicking links embedded in phishing messages that take them to malicious sites, which directly or indirectly steal their personal information. Ease of information dissemination on Twitter and a large audience, makes it a popular medium to spread external content like articles, videos, and photographs by embedding URLs in tweets. However, these URLs may link to low quality content like malware, phishing websites or spam websites. Phishing attacks not only cause the leakage of personal information but also results in huge monetary loss. Recent statistics show that on an average, 8% tweets contain spam and other malicious content. In our existing system which does not focus on detecting phishing but on suspicious URLs in general. It uses correlated redirect chains of URLs on Twitter to detect phishing URLs. However, it may fail if the spammers use short redirect chain or multiple page-level redirects. This research paper describes a newly developed Trend Distributed methodology that correlates the format of tweets with sending agents to detect phishing messages. we propose a distributed methodology to automatically detect phishing tweets in realtime .We use machine learning classification techniques and detect phishing tweets with a higher accuracy than of existing systems . We have deployed our system for end-users by providing a new browser.

***Keywords*- suspicious urls, distributed system, twitter, classification.**

I. INTRODUCTION

Phishing is a long-running problem that has taken a turn for the worse. Phishing tweets now so closely resemble legitimate ones, making it very difficult both for users and automated systems alike to tell them apart. As such, users end up clicking links embedded in phishing messages that take them to malicious sites, which directly or indirectly steal their personal information. This research paper describes a newly developed Trend Distributed methodology that correlates the format of tweets with sending agents to detect phishing messages correlates the format of tweets with sending agents to detect phishing messages[15].

We demonstrate how we use “Distributed data analytics” to proactively identify phishing messages so we can protect our customers from today’s more sophisticated tweet threats. With the advent of online social media, phishers have started using social networks like Twitter, Facebook, and Foursquare to spread phishing scams. Twitter is an immensely popular micro-blogging network where people post short messages of 140 characters called tweets. It has over 100 million active users who post about 200 million tweets every day. Phishers have started using Twitter as a medium to spread phishing because of this vast information dissemination. Further, it is difficult to detect phishing on Twitter unlike emails because of the quick spread of phishing links in the network, short size of the content, and use of URL obfuscation to shorten the URL. Our Technique detects phishing on Twitter in real-time. We use Twitter specific features[1] along with URL features[1],[5] to detect whether a tweet posted with a URL is phishing or not.

Some of the Twitter specific features we use are tweet content and its characteristics like length, hash tags, and mentions. Other Twitter features used are the characteristics of the Twitter user posting the tweet such as age of the account, number of tweets, and the follower-followee ratio. These twitter specific features coupled with URL based features [1],[5] prove to be a strong mechanism to detect phishing tweets. We use machine learning classification techniques and detect phishing tweets with an accuracy of 92.52%. We have deployed our system for end-users by providing a new browser. This browser works in realtime and classifies a tweet as phishing or safe. In this research, we show that we are able to detect phishing tweets at zero hour with high accuracy which is much faster than existing system. To the best of our knowledge, this is the first realtime, comprehensive and usable system to detect phishing on Twitter. In our research, we propose a tool to automatically detect phishing tweets in realtime. This tool uses various features such as the properties of the suspicious URL [1], content of the tweet, attributes of the Twitter user posting the tweet and details about the phishing domains to effectively detect phishing tweets. This

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

tool decides whether a tweet is "phishing" or "safe" by employing machine learning techniques using a combination of the aforementioned features and features collected from various accounts in a distributed manner. Also, we have built a browser to provide realtime phishing detection to Twitter users. The browser extension protects the user from falling prey to phishing attacks by appending a red indicator to phishing tweets. Further, This tool is time efficient, taking an average of only few seconds to detect phishing tweets with high accuracy .Such low computation times make it ideal for real world use.

II. PROPOSED SYSTEM

Phishing is a harmful form of spam. Phishing attacks not only cause the leakage of personal information but also results in huge monetary loss. Hence it is important to build effective realtime phishing detection mechanisms for every OSM to protect its users. There exist browser based toolbars to detect phishing websites, but these toolbars require the user to click on suspected and possibly malicious URL. Thomas et al. proposed Monarch, a realtime malware and phishing detection system which crawls URLs submitted to a web service and assesses them in realtime to classify them as spam or legitimate. Monarch relies on features of the landing page which sometime may not be available. However, these solutions are not specific to Twitter. We believe that phishing detection in Twitter hosts a wide range of challenges specific to Twitter itself such as quick spread of information and the limitation of 140 characters in tweets.

In this we implement a distributed architecture for solving this problem. That is we collect these suspicious information from various warning bird tool (various system). Perform machine learning mechanism on this collected information to get phishing urls. Our modified architecture, this tool decides whether a tweet is "phishing" or "safe" by employing machine learning techniques using a combination of the aforementioned features and features collected from various accounts in a distributed manner. Also, we have built a browser to provide real-time phishing detection to Twitter users. The browser extension protects the user from falling prey to phishing attacks by appending a red indicator to phishing tweets. Further, this tool is time efficient, taking (an average of only few) seconds to detect phishing tweets with high accuracy. Such low computation times make it ideal for real world use.

Our major contributions of this research work are: Automatic realtime phishing detection mechanism for Twitter: There have been studies on phishing detection in emails and spam detection on Twitter, but, to the best of our knowledge, this is the only system which study on realtime detection and protection of phishing on Twitter. More efficient than plain blacklisting method: Our technique proves to be better than plain blacklist lookup which is the most common technique used for phishing detection. Develop a browser for real time detection system; provide a distributed architecture for detection of urls.

III. SYSTEM DETAILS

We implement a distributed architecture for solving this problem. That is we collect these suspicious information from various warning bird tool (various system). Perform machine learning mechanism on this collected information to get phishing URLs. The modules used are:

- Twitter Processing
- Link extract
- Calculate correlated URL features
- Calculate tweet features
- Find URL with features below or above threshold value
- Detecting suspicious URL

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

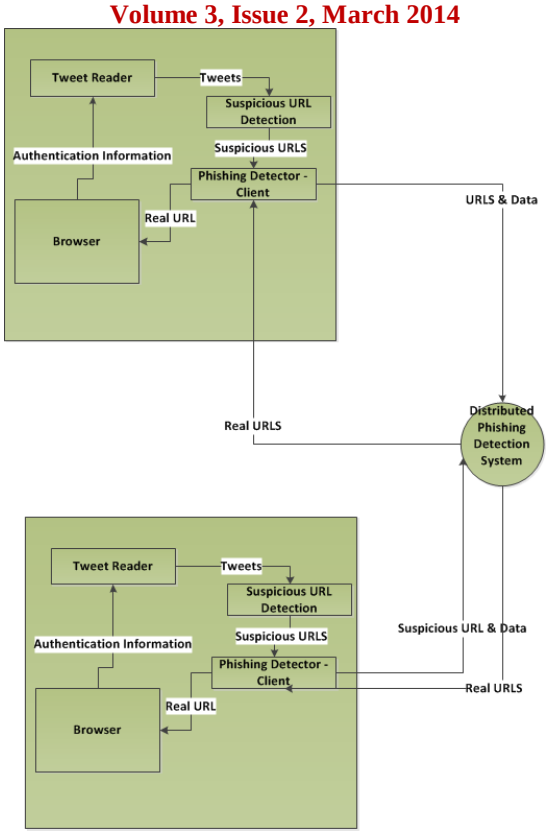

Fig. 1. Distributed system architecture

1. **Browser:** - Provide a web browser for viewing tweets and this act as a container for realtime detection of phishing tweets. When a user click on phishing tweet, this browser redirect to real site. Browser gets this redirection information from Distributed phishing detection system through phishing detector client.
2. **Tweet Reader**: - tweet reader reads from Twitter using Twitter API.
3. **Suspicious URL Detection**: - Block diagram for this system is shown below

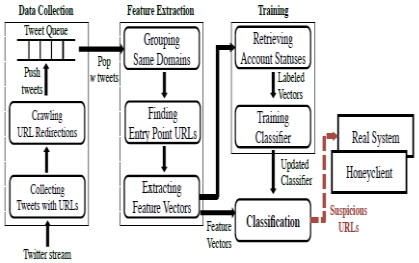

Fig2:System overview

When they use conditional redirections to evade crawlers. However, because our detection system does not rely on the features of landing URLs, it works independently of such crawler evasions.

Feature extraction: The feature extraction component has three subcomponents: grouping identical domains, finding entry point URLs, and extracting feature vectors. This component monitors the tweet queue to check whether a sufficient number of tweets have been collected. Specifically, our system uses a tweet window instead of individual tweets. When more than w tweets are collected (w is 10,000 in the current implementation), it pops w tweets from the tweet queue. First, for all URLs in the w tweets, this component checks whether they share the same IP addresses. If some URLs share at least one IP address, it replaces their domain names with a list of those with which they are grouped. For http://xyz.com/hi.html share the same IP address, this component replaces these URLs with http://['123.com','xyz.com']/hello.html and http://['123.com','xyz.com']/hi.html, respectively. This grouping process allows the detection of suspicious URLs that use several domain names to bypass. blacklisting.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

Next, the component tries to find the entry point URL for each of the w tweets. First, it measures the frequency with which each URL appears in the w tweets. It then discovers the most frequent URL in each URL redirect chain in the w tweets. The URLs thus discovered become the entry points for their redirect chains. If two or more URLs share the highest frequency in a URL chain, this component selects the URL nearest to the beginning of the chain as the entry point URL. Finally, for each entry point URL, this component finds URL redirect chains that contain the entry point URL, and extracts various features from these URL redirect chains and the related tweet information. These feature values are then turned into real-valued feature vectors. When we group domain names or find entry point URLs, we ignore whitelisted domains to reduce false-positive rates. Whitelisted domains are not grouped with other domains and are not selected as entry point URLs. Our whitelisted domain names include the Alexa Top 1000 sites, some famous URL shortening sites, and some domains that we have manually verified.

Training: The training component has two subcomponents: retrieval of account statuses and the training classifier. Because we use an offline supervised learning algorithm, the feature vectors for training are relatively old values than feature vectors for classification. To label the training vectors, We use the Twitter account status; URLs from suspended accounts are considered malicious and URLs from active accounts are considered benign. We periodically update our classifier by using labeled training vectors.

Classification: The classification component executes our classifier using input feature vectors to classify suspicious URLs. When the classifier returns a number of malicious feature vectors, this component flags the corresponding URLs and their tweet information as suspicious. These URLs, detected as suspicious, will be delivered to security experts or more sophisticated dynamic analysis environments for in-depth investigation.

IV. FEATURES

We introduce 12 features for classifying suspicious URLs on Twitter. These features can be classified as features derived from correlated URL redirect chains and features derived from the related tweet context information. We also describe how we normalize these feature values to real values between zero and one.

Features Derived from Correlated URL Redirect Chains/: URL redirect chain length: Attackers usually use long URL redirect chains to make investigations more difficult and avoid the dismantling of their servers. Therefore, when an entry point URL is malicious, its chain length may be longer than those of benign URLs. To normalize this feature, we choose an upper-bound value of 20, because most of the redirect chains we have seen over the four-month period have had fewer than 20 URLs in their chains. If the length of a redirect chain is l, this feature can be normalized as min (l, 20)/20. Frequency of entry point URL: The number of occurrences of the current entry point URL within a tweet window is important. Frequently appearing URLs that are not whitelisted are usually suspicious. When the window size is w and the number of occurrences is n, this feature can be normalized as n/w.
Position of entry point URL: Suspicious entry point URLs are not usually located at the end of a redirect chain, because they have to conditionally redirect visitors to different landing URLs. If the position of an entry point of a redirect chain of length l is p, this can be normalized as p/l.

Number of different initial URLs: The initial URL is the beginning URL that redirects visitors to the current entry point URL. Attackers usually use a large number of different initial URLs to make their malicious tweets, which redirect visitors to the same malicious URL, look different. If the number of different initial URLs redirecting visitors to an entry point URL that appears n times is i, this feature can be normalized as i/n.
Number of different landing URLs: If the current entry point URL redirects visitors to more than one landing URL, we can assume that the current entry point URL performs conditional redirection behaviors and may be suspicious. If an entry point URL that appears n times redirects visitors to λ different landing URLs, this feature can be normalized as λ/n.

Features Derived from Tweet Context Information
The features derived from the related tweet context information are variations of previously discovered features. Our variations focused on the similarity of tweets that share the same entry point URLs.
Number of different sources: Sources are applications that upload the current entry point URL to Twitter. Attackers usually use the same source application, because maintaining a number of different applications is difficult. Benign

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

users, however, usually use various Twitter applications, such as TweetDeck and Echofon. Therefore, the number of different sources may be small when the current entry point URL is suspicious. If the number of different sources of an entry point URL that occurs n times is s, this feature can be normalized as s/n.

Number of different Twitter accounts: The number of different Twitter accounts that upload the current entry point URL can be used to detect injudicious attackers who use a small number of Twitter accounts to distribute their malicious URLs. If the number of Twitter accounts uploading an entry point URL that occurs n times is α, this feature can be normalized as α/n.

Standard deviation of account creation date: Attackers usually create a large number of Twitter accounts within a relatively short time period. Therefore, if the creation dates of the accounts that upload the same entry point URL are similar, it might indicate that the current entry point URL is suspicious. We use the standard deviation of account creation date as a similarity measure. To normalize the standard deviation, we assume that the time difference between any account creation dates is less than or equal to one year. Therefore, this feature can be normalized as λ/n.

2. Features Derived from Tweet Context Information

The features derived from the related tweet context information are variations of previously discovered features. Our variations focused on the similarity of tweets that share the same entry point URLs.

Number of different sources: Sources are applications that upload the current entry point URL to Twitter. Attackers usually use the same source application, because maintaining a number of different applications is difficult. Benign users, however, usually use various Twitter applications, such as TweetDeck and Echofon. Therefore, the number point URL is suspicious. If the number of different sources of an entry point URL that occurs n times is s, this feature can be normalized as s/n.

Number of different Twitter accounts: The number of different Twitter accounts that upload the current entry point URL can be used to detect injudicious attackers who use a small number of Twitter accounts to distribute their malicious URLs. If the number of Twitter accounts uploading an entry point URL that occurs n times is α, this feature can be normalized as α/n. Standard deviation of account creation date: Attackers usually create a large number of Twitter accounts within a relatively short time period. Therefore, if the creation dates of the accounts that upload the same. entry point URL is similar; it might indicate that the current entry point URL is suspicious. We use the standard deviation of account creation date as a similarity measure. To normalize the standard deviation, we assume that the time difference between any account creation dates is less than or equal to one year. Therefore, this feature can be normalized as

$$\min\left(\frac{\text{std(a set of account creation date)}}{(1\text{ year})\sqrt{n}}, 1\right).$$

Standard deviation of the number of followers and number of friends: The numbers of followers and friends of attackers' accounts are usually similar, because attackers use certain programs to increase their numbers of followers and friends. We again use standard deviations to check for similarities in the numbers of followers and friends. To normalize the standard deviations, we assume that the number of followers and friends is usually less than or equal to 2,000, which is the restricted number of accounts Twitter allows one can to follow. Therefore, these features can be normalized as

$$\min\left(\frac{\text{std(\#followers or \#friends)}}{2000\sqrt{n}}, 1\right).$$

Standard deviation of the follower-friend ratio: We define the follower-friend ratio as below:

$$\frac{\min(\#\text{followers}, \#\text{friends})}{\max(\#\text{followers}, \#\text{friends})}.$$

Like the numbers of followers and friends, the follower friend ratios of attackers' accounts are similar. We use a normalized standard deviation to check the similarity as

$$\min\left(\frac{\text{std(a set of follower-friend ratios)}}{\sqrt{n}}, 1\right).$$

Because attackers' accounts usually have more friends than followers, the follower-friend ratios of malicious accounts are usually different from the follower-friend ratios of benign accounts. Attackers, however, can fabricate this ratio, because they can use Sybil followers or buy followers. Therefore, instead of using an individual

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

follower-friend ratio, we use the standard deviation of follower-friend ratios of accounts that post the same URLs and assume that fabricated ratios will be similar.

Tweet text similarity: The texts of tweets containing the same URL are usually similar (e.g., retweets). Therefore, if the texts are different, we can assume that those tweets are related to suspicious behaviors, because attackers usually want to change the appearance of malicious tweets that include the same malicious URL. We measure the similarity between tweet texts as

$$\sum_{t,u \in \text{a set of pairs in tweet texts}} \frac{J(t,u)}{|\text{a set of pairs in tweet texts}|},$$

Which is a famous measure that determines the similarity between two sets sets t and u, and is defined as below

$$J(t,u) = \frac{|t \cap u|}{|t \cup u|}.$$

We remove mentions, hash tags, retweets, and URLs from the texts when we measure their similarity, so that we only consider the text features. In addition to this features we introduce following feature

4. Phishing Detector Client

This is a client part, which collect suspicious information from Suspicious URL detector and send to Distributed Phishing Detector (Warning Bird) and Receive real pages of this suspicious page (if it is phishing)

5. Distributed Phishing Detection

Collect specious urls from different clients (Warning bird) and perform WHOIS test

WHOIS based Features

WHOIS is a query and response protocol which provides information such as ownership details, dates of domain creation / updation of the queried URL. We can identify tweets containing phishing URLs by identifying WHOIS based features that are common to phishing links. Most phishing campaigns register domains of websites from the same registrar, hence tracking the registrar may aid in detecting phishing. Further, most phishing urls are bought for a short period of one year as offenders need to keep constantly changing the url domain names to evade blacklists. Also, the phishing domains are usually created / updated just before they are tweeted. Thus, phishing links generally have low time interval between the domain creation / updation date and the tweet creation date. Therefore, we use WHOIS based features such as registrar's name, ownership period, time interval.

between domain creation / updation and tweet creation date to further enhance our phishing detection methodology. Based on above test, we detect whether a suspicious url is phishing or real. These sets of real pages are given to Phishing Detector Client.

V. DISCUSSION

Phishing is a harmful form of spam. Phishing attacks not only cause the leakage of personal information but also results in huge monetary loss. Hence it is important to build effective realtime phishing detection mechanisms for every OSM to protect its users. There exist browser based toolbars to detect phishing websites, but these toolbars require the user to click on suspected and possibly malicious URL.

Thomas et al. proposed Monarch, a realtime malware and phishing detection system which crawls URLs submitted to a web service and assesses them in realtime to classify them as spam or legitimate. Monarch relies on features of the landing page which sometime may not be available. However, these solutions are not specific to Twitter. We believe that phishing detection in Twitter hosts a wide range of challenges specific to Twitter itself such as quick spread of information and the limitation of 140 characters in tweets.

Kristof Schütt Konra Riec, Marius Kloft, Alexander proposed, EarlyBird: a detection method optimized for early identification of malicious behavior in JavaScript code[9]. The method uses machine learning techniques for jointly optimizing the accuracy and the time of detection. In an evaluation with hundreds of real attacks, EarlyBird precisely identifies malicious behavior while limiting the amount of malicious code that is executed by a factor of 2 (43%) on average.

M. Zubair Rafique and Juan Caballero purposed FIRMA[24], a tool that given a large pool of network traffic obtained by executing unlabeled malware binaries, generates a clustering of the malware binaries into families and a set of network signatures for each family. Compared with prior tools, FIRMA produces network signatures for each of the network behaviors of a family, regardless of the type of traffic the malware uses (e.g., HTTP, IRC, SMTP, TCP, UDP). We have implemented FIRMA and evaluated it on two recent datasets comprising nearly 16,000 unique malware binaries. Our results show that FIRMA's clustering has very high precision (100% on a

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

labeled dataset) and recall (97.7%). We compare FIRMA's signatures with manually generated ones, showing that they are as good (often better), while generated in a fraction of the time.

A dedicated solution proposed exclusively for Twitter by Lee et al. is the Warning Bird[1] system which does not focus on detecting phishing but on suspicious URLs in general. It uses correlated redirect chains of URLs on Twitter to detect phishing URLs. However, Warning Bird may fail if the spammers use short redirect chain or multiple page-level redirects. Though Warning Bird finds suspicious URLs on Twitter in real time, unlike Our System, it does not provide an end-user mechanism for users to use and protect themselves from malicious URLs.

VI. CONCLUSION

In this study, since our goal was to detect Phishing on Twitter and also build end-user solution for Twitter users which works in real time.For this we will develop a distributed architecture for detecting phishing in twitter stream. That is we collect these suspicious information from various systems. Perform machine learning mechanism on this collected information to get phishing urls. Also, we have built a browser to provide realtime phishing detection to Twitter users. The browser extension protects the user from falling prey to phishing attacks by appending a red indicator to phishing tweets.

REFERENCES

[1] S. Lee and J. Kim," WarningBird: A Near Real-Time Detection System for Suspicious URLs in Twitter Stream "IEEE transactions on dependable and secure computing, vol. 10, no. 3, may/june 2013.

[2] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting Spammers on Social Networks," Proc. 26th Ann. Computer Security Applications Conf. (ACSAC), 2010.

[3] D. Canali, M. Cova, G. Vigna, and C. Kruegel, "Prophiler: A Fast Filter for the Large-Scale Detection of Malicious Web Pages," Proc. 20th Int'l World Wide Web Conf. (WWW), 2011.

[4] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru, "Phi.sh/$oCiaL: the Phishing Landscape through Short URLs," Proc. Eighth Ann. Collaboration, Electronic Messaging, Anti-Abuse and Spam Conf. (CEAS), 2011.

[5] Alexander Neumann, Johannes Barnickel, "Security and Privacy Implications of URL Shortening Services" Ulrike Meyer IT Security Group RWTH Aachen University, 2010.

[6] J. Song, S. Lee, and J. Kim, "Spam Filtering in Twitter Using Sender-Receiver Relationship," Proc. 14th Int'l Symp. Recent Advances in Intrusion Detection (RAID), 2011.

[7] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary, "Towards Online Spam Filtering in Social Networks," Proc. 19th Network and Distributed System Security Symp. (NDSS), 2012.

[8] J. Zhang, C. Seifert, J.W. Stokes, and W. Lee, "ARROW: Generating Signatures to Detect Drive-By Downloads," Proc. 20th Int'l World Wide Web Conf. (WWW), 2011.

[9] Kristof Schütt , Konrad Riec,Marius Kloft, Alexander "Early detection of malicious behavior in javascript code,2009.

[10] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and Evaluation of a Real-Time URL Spam Filtering Service," Proc. IEEE Symp. Security and Privacy (S&P), 2011.

[11] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who Is Tweeting on Twitter: Human, Bot, or Cyborg?" Proc. 26th Ann. Computer Security Applications Conf. (ACSAC), 2010.

[12] Joshua S. White, Jeanna N. Matthews, John L. Stacy ,"A method for the automated detection of phising websites through both site characteristics and image analysis".

[13] L. Lu, V. Yegneswaran, P. Porras and W. Lee. Blade: An attack-agnostic approach for preventing drive-by malware infections. In Proc. ACM CCS, 2010.

[14] M. Aldwairi, R. Alsalman. "MALURLs: Malicious URLs Classification System". In Proceedings of the Annual International Conference on Information Theory and Applications (ITA), Singapore, Feb 2011.

[15] PhishAri: Automatic real time phishing detection on twitter.

[16] M. McCord, M. Chuah ,"spam detection on twitter using traditional classifiers" ,ATC'11 , Sept 2-4, 2011, Banff, Canada. Copyright 2011.

[17] Xia Hu†, Jiliang Tang†, Yanchao Zhang‡, Huan Liu† ,"Social spammer detection in micro blogging", Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence.

ISSN: 2319-5967
ISO 9001:2008 Certified
International Journal of Engineering Science and Innovative Technology (IJESIT)
Volume 3, Issue 2, March 2014

[18] Faraz Ahmed and Muhammad Abulaish,"A Generic Statistical Approach for Spam Detection in Online Social Networks" April 10, 2013.

[19] A. Kapravelos, M. Cova, C. Kruegel, and G. Vigna, "Escape from Monkey Island: Evading High-Interaction Honey clients," Proc. Eighth Conf. Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), 2011.

[20] M. Cova, C. Kruegel and G. Vigna. Detection and analysis of drive-by-download attacks and malicious javascript code. In Proc. WWW, 2010.

[21] Saeed Abu-Nimeh1, Dario Nappa2, Xinlei Wang2, and Suku Nair,"A comparison of machine learning techniques For phising detection" In Proceedings of the SIGCHI conference on Human Factors in computing systems, 2006.

[22] WANG Wei-Hong, LV Yin-Jun, CHEN Hui-Bing, FANG Zhao-Lin "A static malicious JavaScript code using svm" Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).

[23] Arun Kumar R , "Twitter Spamming: Techniques And Defence Approaches "International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11 (2012).

[24] M. Zubair Rafique and Juan Caballero,"Firma: malware clustering and n/w signature generation with mixed n/w behavior" In Proc. WWW, 2012.

[25] Roberto Perdiscia,b, Wenke Leea, and Nick Feamstera,"Behavioural clustering of HTTP based malware and signature generation using malicious network trace" In Net- work and Distributed System Security Symposium, 2009.

[26] A. Wang, "Don't Follow Me: Spam Detecting in Twitter," Proc. Int'l Conf. Security and Cryptography (SECRYPT), 2010.

AUTHOR BIOGRAPHY

Manju C.Nair , IInd year ME, Department of computer science Mahendra Institute of technology,Mallasumdram Tiruchenode,Tamil nadu,India.

S.Prema,AP In Department of computer science Mahendra Institute of technology,Mallasumdram Tiruchenode,Tamil nadu,India.