Implementation of A System To Detect Malicious URLs for Twitter Users

Nupur S. Gawale [*1]
[#]PG Student,
Department of Computer Engineering,
R. C. Patel Institute of Technology, Shirpur,
MS, India
nupurgawale14@gmail.com

Nitin N. Patil [*2]
[*]Head and Associate Professor,
Department of Computer Engineering,
R. C. Patel Institute of Technology, Shirpur,
MS, India
er_nitinpatil@rediffmail.com

***Abstract*- Over the last few years, there is tremendous use of online social networking sites. It's also providing opportunities for hackers to enter easily in network and do their unauthorized activities. There are many notable social networking websites like Twitter, Facebook and Google+ etc. These are popularly practiced by numerous people to become linked up with each other and partake their daily happenings through it. Here we focus on twitter for an experiment which is more popular for micro-blogging and its community interact through publishing text-based posts of 140 characters known as tweets. By considering this popularity of tweeter hacker's use of short Uniform Resource Locator (URL), as a result it disseminates viruses on user accounts. Our study is based on examining the malicious content or their short URLs and protect the user from unauthorized activities. We introduce such a system which provides the security to multiple users of twitter. Besides, they get some alert mails. Our goal is to download URLs in real time from multiple accounts. Then we get entry points of correlated URLs. Crawler browser marks the suspicious URL. This system finds such malicious URLs by using five features like initial URL, similar text, friend follower ratio and relative URLs. Then alert mail is sent to users, which is added to the host.**

***Index Terms* —API keys, classifier, Conditional redirect, crawler, Suspicious URL, Twitter .**

I. INTRODUCTION

Social networking websites are significant pieces of everyday's life from the last few years. People take help of social networking sites where they are easily associated with their family members and billions of people from the whole globe. But those social situations are not just only used to partake in their sentiments, but some people use it for doing their unauthorized activities by which public gets disturbed or hack their system and access their individual data.

So they put up some malicious parts in the form of URLs for doing such activities. Hence, real time detection system which is based on correlation URLs redirect chain which makes the exact classification between suspicious and non-suspicious URLs. This implemented system is not established on account information, profile based information. It focuses on conditional redirection. Now, System gives the accurate result for a single user. It downloads the tweets from user account. Then that database or URLs proceed for the next performance, and so, finally gives classification. As we have these social sites for users, through which they share their everyday's information. But there are about 200 or 300 million accounts on the twitter. Today, using this developing system we can be able to provide protection to all accounts of twitter. With this challenge we produced a system which makes a final classification between suspicious URLs and non-suspicious URLs not only for a single user account but also for multiple users of twitter. Because of this, it contributes to our development system and generates more accurate outcomes. As we receive the knowledge of keywords or abbreviation for communication on twitter for single user, same is used for multiple users as well. In planning this scheme we just add accounts simultaneously. Hence we are making arrangement for multiple users like (1) We firstly perform add accounts by making a dynamic pin from the host (2) After that it authorize user using his/her twitter id and password and make the seven digit dynamic pin number and email id of that particular user, then finally add that user account on our development system [1]. Like a short here, we require email id of the user because after final solution the system also gives the facility, by sending suspicious URLs report to the user's email address. This is also very important as a user point of view, because it alerts the users about suspicious URLs before clicking on any other URL. Once we add the multiple accounts on twitter then, to detect the fishy and non-suspicious URLs among them. The scheme is performing like (1) It takes in the tweets from multiples accounts simultaneously in actual time (2) now all that tweets foreword to finding correlation of URL redirect chain using crawler browser (3) then looking for the domain grouping as well as entry point URL (4) again thread master module compared this entry point URL with existing database (5) and at the end evaluating features we get last results, i.e. suspicious and non-suspicious URLs.

II. LITERATURE SURVEY

Till now a lot of work is done on security related to twitter. Besides, the researchers have also focused on twitter with multiple accounts. For this problem they worked using some technology and scheduling algorithm. But they were not able to accomplish with the multiple accounts.

J. Funasaka et al. [2] proposed the system to download the information from multiple accounts simultaneously. Only the system required technology and high bandwidth. Any information downloaded

978-1-4799-6272-3/15/$31.00(c)2015 IEEE

from the server is temporarily stored. Also for storing information there is restricted. System not providing buffering to add database every time. For download particular information about user sometime is used proxy server. This system requires every time to add particular account and afterwards that it can download information of that user account.

Z. Shao et al. [3] proposed a system for downloading tweet from the background and foreground scheduler. This system, downloaded only recently database. This database cannot distinguish duplicate and original URLs. Besides the scheme used the round robin scheduler approach. After comparing both approaches, the foreground and background scheduler requires a lot less network resources to download tweets from multiple account. Only they cannot able to detect the harmful URLs among that collected tweets and URLs.

I. Qasem et al. [4] developed the alert message on client about malware, virus attacks. They used manual detection. Their communication is also in traditional ways. It means only using messages and telephone calls, but because of today's life this is really old ways to assure the user account and user data. The system can't be able to check multiple.

C. Yang et al. [5] manifested the empirical analysis and fresh features like graph based feature and neighbor based feature. The system gathers up the database statically and from that it finds relationship among users using graph based. Also in which bi-directional link ratio. It means when two accounts follow with each other then it consider bi-directional link between each other. Spammer follows a big act of legitimate reports, simply cannot push them to follow back. Then from this spammer bi-directional links has low. And the other side, the legitimate account follows their all relatives, friend and co-staff member who will also watch this user back. It means legitimate account bi-directional links have more compared to spammer. Only from this analysis system not able to provide a resolution to this problem.

W. Lee et al. [6] proposed the drive download detection system. System downloading tweets from multiple invoices, but the system demands to find exact results from accumulating a database. Attackers always tried to discover multiple ways to distribute their malicious content on user account without knowing them such type of attack is called drive by download attacks. Attackers to spread their malicious URLs or tweets they used their malware distribution network i.e. MDN. The system needs to identify this malware distribution networks central server. Because central server is usually shared by all drive download attackers. To find this central server they used regularized expression based signatures which are generated. But to locate this server and signatures is not an easy job.

L. Ballard et. al [7] constructed the web malware detection system. Malicious web content which spread on user accounts. But the researchers are highly motivated and quickly take hold of the technology that attempted to protect user from malicious content. This detection system worked on or helps the millions of users. Firstly, it contains a static database or URLs then uses the most prevalent web malware detection system like virtual machine client then browser emulator client also classification domain grouping and antivirus engines. But as a result, none of these schemes are effective for finding exact solutions.

III. METHODOLOGY

Our system is real time and thus we are able to connect to the account which already exists on twitter server for proper authentication and to gather the information from twitter.

Hence we have registered such account on twitter. By the quote of that account, twitter provides some master key called as consumer key by the reference of dev.twitter.com. Then whenever a new user wants to register on twitter through our system, then he/she gets seven digit pin numbers from that authorize account. That pin number access the permission to particular accounts and every time new users get dynamic pin. This is basic reason to add multiple users one by one in our database. But every user receives its own access secret key. This access secret key is differentiating that user from among existing users.

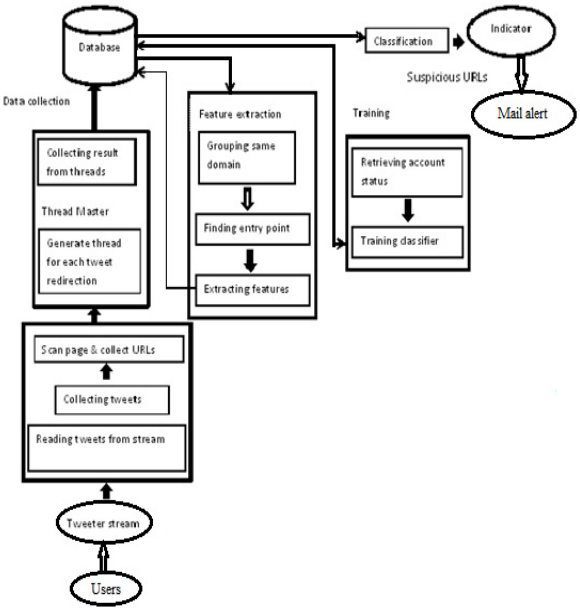

Fig. 1. System Architecture

Let's consider system diagram as shown in the Fig. 1 above. Here we collected data from multiple accounts on twitter. This information is collected in real time to create more precise outcomes. The system starts to download tweets with URLs from every user. They mention that access secret key. It's starting to download tweets from multiple accounts simultaneously and stored in the database which are increasing every time. Once that URL collected then that forwarded for converting from the short URL into long URL. In this way correlation of the URLs redirect chain we get which extracted from tweets. Then that redirect URL needs to control entry point where they collect same URLs which having the same domain grouping. These frequently shared URL from among all URLs is connected to crawler browser. Then crawler browser has the ability to detect malicious content [8]. The next piece is a thread master module, that detect the URL which is alike to that entry point URL or URL which having any intercourse to that entry point URL. And ultimately it is observed by the classification by which we get the

suspicious and benign URLs. Then last module is training in which, non-suspicious URLs are compared with suspicious URLs using features.

Here we considered top 5 features like initial URLs, similar tweet texts, a number of accounts, similar follower friend ratio, and bit of landing URLs etc. In which first feature is initial URLs means the initial landing page or URL checked. Because the many times initial pages are same of that entry point URL. Then second feature is similar tweet text again in that same URL has the same text with a different URL, and also some URL with text both are same. Then side by side feature is account date means at which date tweets were downloaded from accounts. Afterward that the friend, follower ratio in this characteristic, the friend and follower of that every account should consider. And last feature is relative URLs, in which the URLs which are related to suspicious URLs. All this features applying in training module. And compare the non-suspicious URLs with the suspicious URLs. Ultimately, at the end it moves over the classification of suspicious and non-suspicious URLs.

By the acknowledgment of this system generates the reports and alert mail to a user report. This arrangement is divided into two components for generating more precise results. We discuss this as traces:

A. Client

In this constituent the user who receives the information close to shady and non-suspicious URLs from his/her account. This module is performed by the users. Here the user which is registered user with the twitter account. This user needs to obtain the access key which is generated after user login through this module by a scheme. The logged in user displays his/her twitter wall of this module. Here user can post the tweets, take in their tweet, search the other users and tweet also get the current trending topics. That's why every time our database up to date. The URLs are proceeding to crawler for performing redirection.

B. Server

This is the main part of our system. Today, here we desire to add multiple users and for that first we will configure login. New user receive dynamic pin by using get pin operation. At the same time user needs to enter his/her email address which becomes the master key through which, user login into twitter. In this way simultaneously, we add accounts in the system. Once the accounts adding operation is over, then our system is dynamically downloaded the tweets from every account at the same time which is added in the database. All the downloaded data is stored into database. That's why the database is up to date and increased. Then server performs the operation of fetching the database from collected one. The server can also download URLs or tweets from the register tweeter account. This server part collects that data from the crawler, always which is endlessly playing in the background of the system. It is just like any antivirus software which always matches data on our personal information processing system.

We adopt the same concept and the same way our host is continuously active. Here server knows about the URLs entry point and the domain grouping. If any user account is not properly working, then the server has the ability to get changes into particular account. So the server is main central part of the our system. The server names which are malicious URLs and non-suspicious URLs then, it forwards for training. At last training just again comparing non-suspicious URLs with the suspicious and its relative URLs using f-score features. It evaluates f-score of all features like initial URLs, similar tweet text, similar friend followers ratio, relative URLs and account date and its user. At the end, we make the classification between suspicious URLs and non-suspicious URLs for multiple accounts.

IV. RESULTS AND DISCUSSION

Here dynamically crawling at real time is done and the suspicious URLs are found. Thus the probability of getting more accurate results is increased. Besides the generated reports and graphs for multiple accounts illustrates the same. The most important is that we experiment independently of window size from user point of view. These results are shown in Table I below. Generally out of 1000 URLs, we need 497 ms, to process a single URL. By increasing the number of crawling threads, we can treat grater than 40% of the tweet samples. Then we redirect 492 chains and we acquire two of them sustaining the same field name. Take only that URLs which are having a same domain group name, next we perform feature extraction and categorization of the URLs.

TABLE I. Time to classify a single URL

Component	Avg. Running time (ms)
Redirect chain crawling	492
A domain grouping	2
Feature extraction	1.3
Classification	1.7
TOTAL	**497**

Now Fig. 2 have dates on the X-axis, it is dependent on the user. Along the Y- axis tweet downloads from multiple accounts. Using this graph we can understand at which dates tweets are downloaded from accounts.

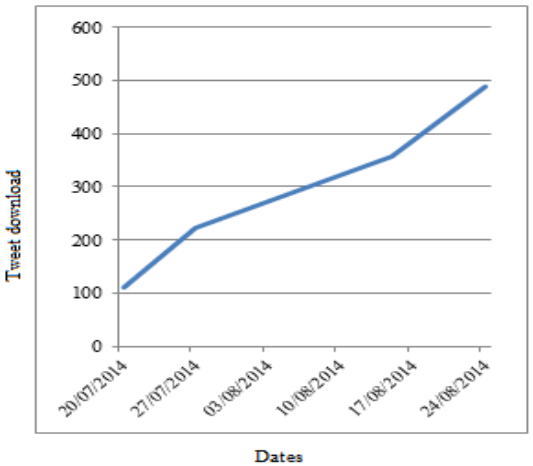

Fig. 2. Date Vs. Tweet Download

Then Fig. 3 shows users on the X-axis. Along the Y-axis tweet counts from multiple accounts.

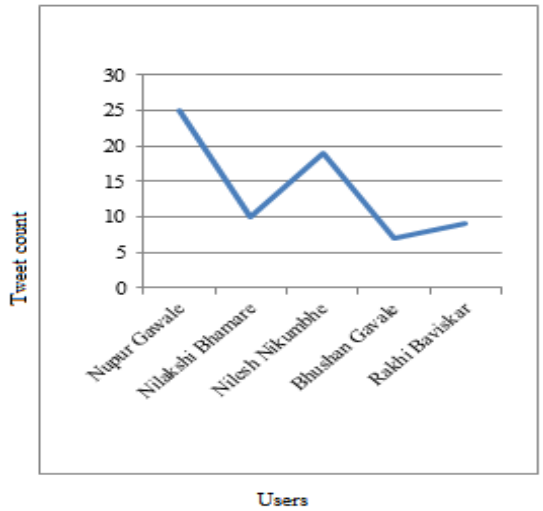

Fig. 3. Users Vs. Tweet Count

Then Fig. 4 shows users on the X-axis. Along the Y-axis friends count from multiple accounts.

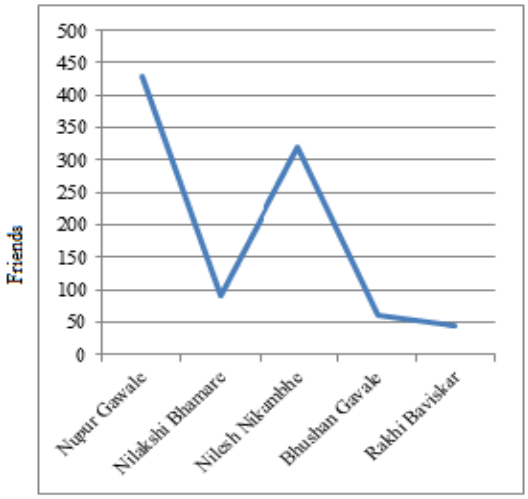

Fig. 4. User Vs. Friends

Using this graph we understand how many friends of every account or following since friends are active accounts or employment.

Now Fig. 5 shows users on the X-axis. Along the Y-axis followers of multiple accounts.

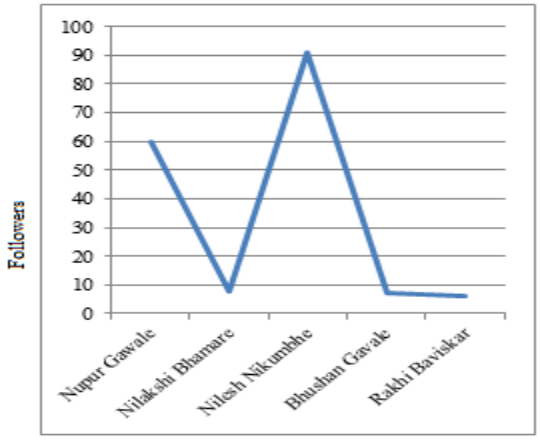

Fig. 5. Users Vs. Followers

Fig. 6 shows friends on the X-axis. Along the Y-axis the followers of multiple users, who have same sequence as shown in the Fig. 5 above.

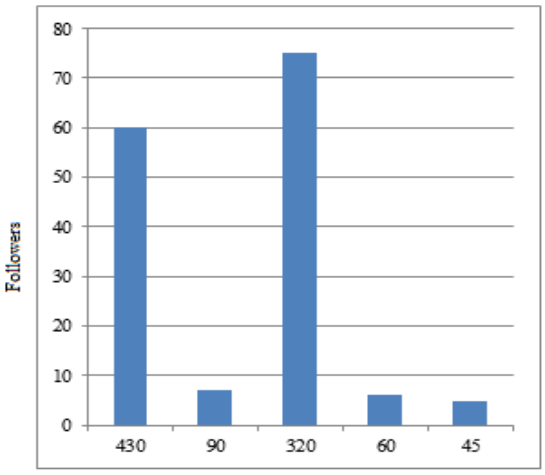

Fig. 6. Friends Vs. Followers

By observing all these graphs our system gives comparatively better results for multiple users.

The purpose of our system is to detect the suspicious URLs from multiple accounts simultaneously in real time manner. The other goal is alert by e-mail and the reason behind that to protect and aware the user before suffering from malicious attempts. Once any particular client is added, then no need to add same client next time as the system automatically downloads or gets data from all accounts which are added already. The final benefit of this evolving system is whenever we clicked any URL on clientwall and if that URL is suspicious or related to that, and so it generates warning message for user otherwise it spreads out the URL. In this manner the discussed system gives more precise solutions for multiple users.

V. FUTURE WORK

This system gives protection to multiple twitter user from malicious content simultaneously and in real time. It presents the exact classification between suspicious and non-suspicious URLs. Hence 94 % accuracy proves to be a better one from the security point of view. In future for improved outcome, multiple accounts can be added using single email id. Scope can be expanded for other popular social networking websites like Facebook, Google+ etc.

ACKNOLEDGMENT

We are thankful to all who helped us directly or indirectly throughout our targeted work.

REFERENCES

[1] Twitter Developers, "Streaming API," https:// dev.twitter.com/ docs/ streaming-api.

[2] J.Funasaka, "Implementation issues of parallel downloading methods for a proxy system." Distributed Computing Systems Workshops, pp. 58–64, 2005.

[3] Z.Shao, "A qos framework for heavy-tailed traffic over the wireless internet." MILCOM.Proceedings, pp.1201–1205.

[4] I.Qasem, "Leveraging online social networks for a real-time malware alerting system." Local Computer Networks (LCN), pp. 272–275, 2013.

[5] C. Yang, R. Harkreader, and G. Gu, "Die free or live hard? empirical evaluation and new design for fighting evolving Twitter spammers," in *Proc. RAID*, 2011.

[6] J. Zhang, C. Seifert, J. W. Stokes and W. Lee, "ARROW: Generating signatures to detect drive-by downloads," in proceedings of the 20th international conference on World Wide Web, ACM, pp.187-196, 2011.

[7] M. A. Rajab, L. Ballard, N. Jagpal, P. Mavrommatis, D. Nojiri, N. Provos, and L. Schmidt, "Trends in circumventing web malware detection," Google Tech. Rep., 2011.

[8] S. Lee and J. Kim, "Warningbird: Detecting suspicious URLs in Twitter stream," in *Proc.* Network and Distributed System Security (*NDSS)*, vol.10, 2012.

[9] C. Whittaker, B. Ryner and M. Nazif, "Large-scale automatic classification of phising pages," in *Proc. NDSS*, 2010.

[10] K. Thomas, C. Grier, J. Ma, V. Paxson and D. Song, "Design and evaluation of a real-time URL spam filtering service," in proceedings Security and Privacy (SP), IEEE Symposium on, pp.447-462, 2011.

[11] S. Chhabra, A. Aggarwal, F. Benevenuto and P. Kumaraguru, "Phi.sh/$oCiaL: the phishing landscape through short URLs," in proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference CEAS, ACM, pp.92-101, 2011.

[12] "The t.co URL wrapper," https:// dev.twitter.com/ docs/ t.co-URL-wrapper.

[13] Twitter Help Center, "The Twitter rules," https:// support.twitter.com/ articles/ 18311-the-twitter-rules.

[14] C. Y. R. Harkreader, J. Zhang, S. Shin and G. Gu, "Analyzing spammers' social networks for fun and profit-a case study of cyber-criminal ecosystem on Twitter," in Proceedings of the 21st international conference on World Wide Web, ACM, pp.71-80, 2012.

[15] F. Klien and M. Strohmaier, "Short links under attack: geographical analysis of spam in a URL shortener network," in Proceedings of the 23rd ACM conference on Hypertext and social media, ACM, pp.83-88, 2012.

[16] S. Ghosh, B. Vishwanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K.P. Gummadi, "Understanding and combating link farming in the twitter social network," in Proceedings of the 21st international conference on World Wide Web, ACM, pp.61-70, 2012.

[17] A. Wang, "Don't follow me: Spam detecting in Twitter," in *Proc.SECRYPT*, 2010.

[18] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: Social honeypots + machine learning," in Proc. 33rd Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, 2010.

[19] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: The underground on 140 characters or less," in proceedings of the 17th ACM conference on Computer and communications security, pp.27-37, 2010.

[20] D. Antoniades, I. Polakis, G. Kontaxis, E. Athanasopoulos, S. Ioannidis, E. P. Markatos, and T. Karagiannis, "Web: The web of short URLs," in proceedings of the 20th international conference on World Wide Web, pp.715-724, 2011.

[21] "Identifying suspicious URLs: An application of large scale online learning," in Proc. of the International Conference on Machine Learning, ICML, 2009.

[22] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in proceedings of the 19th international conference on World wide web, ACM, pp.591-600, April 2010.

[23] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Who is tweeting on Twitter: Human, bot, or cyborg?" in Proceedings of the 26th annual computer security applications conference, ACM, pp.21-30, 2010.

[24] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in proceedings of the 26th Annual Computer Security Applications Conference *(ACSAC)* pp.1-9, 2010.

[25] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on Twitter," in *Proc.* Collaboration, electronic messaging, anti-abuse and spam conference (CEAS), vol. 6, July 2010.