Real Time Detection of Suspicious URLs on Social Networking Sites Twitter

Jyoti D. Halwar
Department of Computer Science
Dr.D.Y.Patil College of Engineering
Ambi, Pune, India

Sandeep U. Kadam
Department of Information Technology
Dr. D.Y.Patil College Of Engineering,
Ambi, Pune, India

ABSTRACT
Twitter, FACEBOOK are very famous social networking site utilized by billions of individuals to exchange the data to one another. To communicate with one another over the long separation it is used. At the same time it additionally attracts the attacker in doing diverse assaults or get the data being imparted by their clients. Twitter users can send the messages to one another as tweets, that tweets have the size impediment of greatest 140 characters. So to share the large information addresses to that pages is used by providing links of that pages. And for this purpose URL shorting is used. Attackers send the suspicious URLS in tweets and move the clients to malignant pages. These URLs can also be shared on Facebook with followers and friends. This paper introduces a Near REAL TIME APPLICATION to detect the suspicious URLs which are shared on twitters public timeline. This application collects the tweets, extracts the features correlated to the URL redirect chain, and with the help of training classifier it classifies the URLs as Suspicious and Benign.

Keywords
Suspicions URL, Twitter, URL redirection, conditionnel redirection, classification.

1. INTRODUCTION
In today's world social networking sites are becoming very important part of daily life of human being. The sites like Facebook, twitter, Myspace, google plus and so on are utilized by a huge number of individuals to speak with one another however they're such a great amount of far from each other. Just as a result of this on-line social networking website individuals can send their data, feature, sound or even pages too. Due to these things there is a great increase in business opportunities for developer, programmer and network providers. Internet connectivity suppliers are plays an important role in social Networking sites. They are joined with the client through some interface.

Now in case of Twitter, It permits the clients to send their messages as tweets which have a size limit of greatest 140 characters. In twitter assume 2 clients Alice and Bob are conveyed then Alice is tweeting his content inside the type of tweet. As Bob is friend of Alice with the goal that Bob acquiring all the post of Alice on his window. It implies that Bob is adherent of Alice. He will see the all post of Alice. Afresh instead of bringing on to all or any the post there is also post may be creating to just 1 specific individual by specifying his or her name by utilizing @. Thusly twitter is working.

Presently, on this social site have a few advantages furthermore their some disservice. Without Network support this locales are not living up to expectations legitimately. Thus, inside the system huge measure of individuals are offering data among themselves. The a Few sorts of people are expected to urge the illumination or it implies that they hacked the client individual and classified data and those people we have the capacity say that wrongdoer. In this way, in the system regular mixtures of system assaults are there.

Consider the example, Alice and Bob are friends, and if the Alice sends account details to Bob then at that time third party may get that information then appallingly extensive loss of the every Alice and Bob must be urged to survive. In this way, that inside the system security is to a great degree essential for secret data. This fundamental risky and high issue is inside the system. Still the research is going on to solve these issues. Some work is already carried out by some people to solve these issues. Along with theft of data, some other kinds of attacks are carried out by attackers. Attacker may land the normal users to malicious pages or May carried out Dos kind of attacks, interruption in services to the users or any other kind of interruption. Consequently there are various courses that of assaulting spam, phishing and malware and so on. So security is the main issue in Social networking sites.

Twitter is vulnerable to suspicious tweets containing URLs for spam, phishing, and malware circulation. Conventional Twitter spam identification plans use account choices like the extent connection of tweets containing URLs furthermore the record creation date, or connection alternatives inside the Twitter graph. These detection schemes are not able to give the expected performance because these schemes consume considerable amount of time and assets. Average suspicious URL detection schemes use numerous choices and in addition lexical alternatives of URLs, URL redirection, HTML substance, and element conduct.

This paper shows a REAL TIME APPLICATION, a suspicious URL detection framework for Twitter. This application considers relationships of URL redirect chains extracted from numerous tweets. This system gathers different tweets from the Twitters public timeline, Find the tweets containing URLs. Also it finds the redirection of URL and correlation between URL redirect chains. By Using a mathematical classifiers this system detects the Suspicious URLs from Twitters public timeline.

This system is a near real time system to detect the suspicious URLs from twitters public timeline.

2. EXISTING METHODS

Various suspicious URL detection schemes have been proposed. They can be grouped into either static or dynamic schemes. Some lightweight static detection frameworks concentrate on the lexical peculiarities of a URL, for example, its length, the quantity of spots, or every token it has, furthermore consider basic DNS and WHOIS data. Canali et al. proposed application Prophiler, considers characteristics of HTML content and JavaScript codes to locate drive-by download assaults. But some of the static Suspicious URL detection schemes can't recognize suspicious URLS with element substance, for example, jumbled JavaScript, Flash, and ActiveX content. So we require dynamic detection schemes that utilize virtual machines and instrumented Web programs for inside and out examination of suspicious URLs.

Some focuses on detecting spam accounts from no spam accounts by considering Collected tweets , account details like account creation dates, number of friends and followers, URL degrees, and tweet content similitudes. Yang et al. concentrated on relations between spam node and their neighboring nodes, for example, a bi-directional connection degree and betweenness centrality, on the grounds that spam node generally can't create solid associations with their neighboring nodes. They additionally presented different schemes focused around timing and robotization. Melody et al. considered the relations between spam senders and collectors, for example, the most brief ways and least cut, on the grounds that spam nodes normally can't secure vigorous associations with their victimized person nodes. The extraction of these vigorous schemes, notwithstanding, is time and asset expending. Record and connection gimmick based plans can't catch spam messages from bargained records, on the grounds that the traded off records have favorable schemes. To tackle this issue, Gao et al. proposed a spam location plan utilizing message-based schemes. They concentrated on the syntactic closeness of spam messages.

Zhang et al. Proposed a solution called ARROW: Generating signatures to detect drive-by downloads, which, relies on logs of HTTP traces to detect central servers. ARROW's HTTP traces are redirect chains between malicious landing pages and malware binaries. This system is not real time system and also not able to detect the web attacks. System in this paper can handle conditional redirection as well as multi redirection and it is near real time system.

Proposed system

This system detects the suspicious URLS in Social Networking Site Twitter, by collecting the large number of tweets from Twitters public timeline and finding the correlation between URL redirect chains & their context information. It also discovers the several features from correlation between URL redirect chains and classifies the URLs as Suspicious and Non Suspicious.

3. SYSTEM ARCHITECTURE

This system is decomposed into following modules

A. Data Collection
   - Collecting tweets URL
   - Crawling Redirect URL.

B. Feature Extraction
   - Grouping Same Domain
   - Feature Vector

C. Training

D. Classifications

3.1 Data collection

This module collect the tweets with URLS from twitter public timeline by using twitters Streaming API. This module works is in 2 parts first collecting the tweets and secondly crawling the URL redirection. At whatever point this module gets a tweet with a URL, it triggers a crawling thread that takes after all redirections of the URL and finds IP addresses. This creeping thread pushes these recovered URL and IP fastens to the tweet data and pushes it into a tweet queue. As we have seen, our crawler can't achieve malignant arriving URLS when they utilize restrictive redirections to sidestep crawlers. Notwithstanding, in light of the fact that this system does not depend on the characteristics of landing URLS, it meets expectations freely of such crawler avoidances.

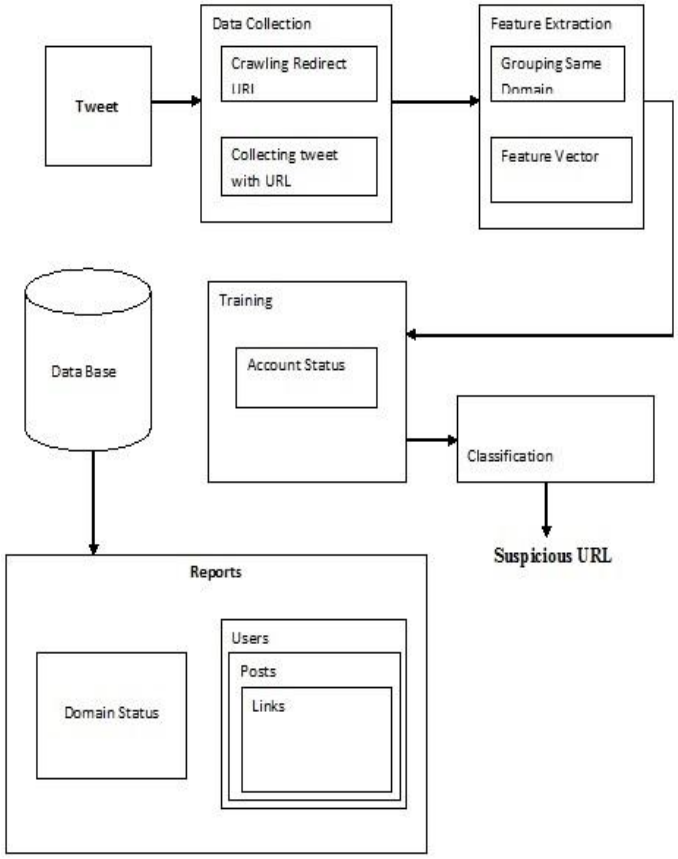

Fig 1: System Architecture

3.2 Feature Extraction

This module works in 3 submodules: grouping of identical domains, finding entry point URLS, and extracting feature vectors.

First it observes the tweet queue for sufficient number of tweets. Once there are sufficient number of tweets are available in queue it pops X tweets from the tweet queue by using tweet window instead of individual tweet .Next this component observers whether they have common IP addresses. For the URLs sharing Same IP it replaces domain names. Due to which we can detect suspicious URLS that use several domain names to bypass the blacklisting. Next, it searches the entry point URL for each of the x tweet and calculates the frequency with which each URL appears in these tweets .then it find most frequent URLs and Extract its features and stored into feature vectors.

3.3 Training:

This component is developed by using offline supervised learning algorithm, by using twitters account status labelling to training vectors is done. URLs are classified as suspicious and non-suspicious from blocked and active accounts respectively. Classifier is updated time to time by using labelled training vectors.

3.4 Classification:

This module triggers classifier using input feature vectors to distribute suspicious URLs. When a number of malicious feature vectors are returned to classifier, respective URLs and their tweets are marked as malicious.

4. MATHEMATICAL MODEL

Let Y=fs(Φ) be a solution for finding the suspicious URLs from twitter's public time.

In general it is represented by program set Fi where,

fs={ S, D, Fs, DD, NDD, Ffrnds|Φ}
S = Output i.e Suspicious URL
D = Input i.e All set of URLS
Fs = Σdiϵ D([(|B(di)| ∩ |A(di)|]/|B(di)|)
Σdiϵ D([(|S(di)| ∩ |A(di)|]/|S(di)|)
DD =deterministic data
For all accepted URL, system is able to detect suspicious URL considered as deterministic data.

NDD =Nondeterministic data
for all accepted URL, system is not able to detect suspicious URL considered as non Deterministic data.
Ffrnds= Friend function using shared memory
Set of function used to find the correlation of URL chain.

5. FEATURES

Features As attackers are having limited resources hence thy are reusing them, also some attackers are having different accounts and different numbers of same friends and followers so considering these things these features are derived to classify the URLs as a SUSPICIOUS AND BENIGN and grouped as features derived from correlated URL redirect chain and "Features Derived from Tweet Context Information".

5.1 Features derived from correlated URL redirect chains

- URL redirect chain length
- Frequency of entry point URL
- Position of entry point URL
- Number of different initial URLS
- Number of different landing URL

5.2 Features derived from tweet context information

- Number of different sources
- Number of different Twitter accounts
- Tweet text similarity:
- Similarity in the number of followers and number of Friends
- Similarity in the follower-friend ratio
  min(number of followers, number of friends)/ max(number of followers, number of friends)

6. TECHNICAL DETAILS OF THE SYSTEM

This system is developed by using various JAVA frameworks. The use of these frameworks is done main for real time performance, security, and distributed collection of tweets. After the collection of data to classify the tweets as suspicious and benign this application is developed by focusing on the following three points:

- Reoccurrences of redirect chains in URLs (entry points)
- Check whether same URL is posted to other users (followers) from same IP.
- Frequent URL with similar domain names and from same IP address.

To develop this system following frameworks are used

- Spring MVC for UI and JAVA Development,
- Rest API for communication between Web Application, Storm Real-time multithreaded distributed framework and Desktop Distributed Application,
- Storm for near real time, multithreaded, distributed computation system. This is used to collect Real-time tweets.

7. EXPERIMENTAL RESULTS

7.1 Tweet with URLs collection

As this system is implemented with Java Framework Strom to collect the numerous tweets from twitters public timeline, there is no limit on number of tweets collected. After collecting the tweets system separates the tweets containing URLs. This graph shows the number of tweets with URLs against the total number of tweets collected on date 23rd march 2015.

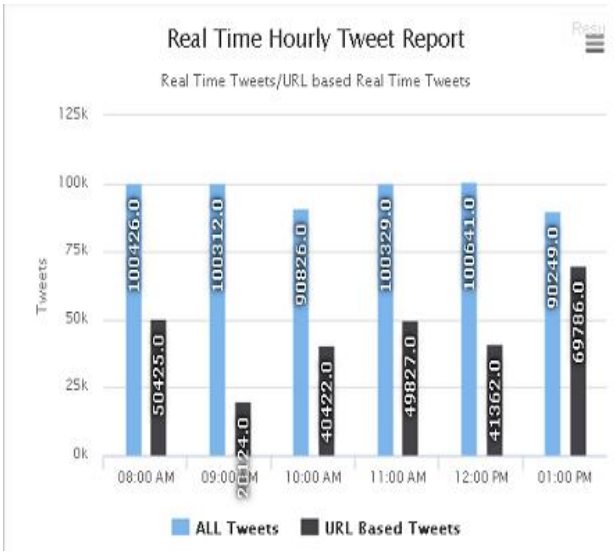

Fig 2: Real time Hourly tweet report

7.2 Reoccurrences of URL redirect chain.

Attackers are having limited resources so most of the time they have to reuse them. So to reuse the same resource again and again attackers are sharing same URL redirect chain repeatedly. So to find the Suspicious URLs this case is also considered. This graph shoes the number of different URL

redirect chain that are reoccurred from the collected number of URL's Redirect chain

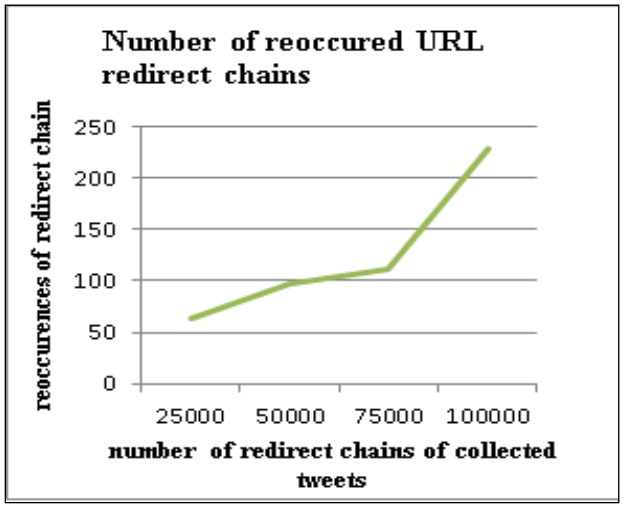

Fig 3: Number of reoccurred URL redirect chain

7.3 Running time evaluation

Running time of each component is evaluated against different sizes of tweet window. Data collection, feature extraction, training and classification give the different running time for different size of window containing different number of tweets.

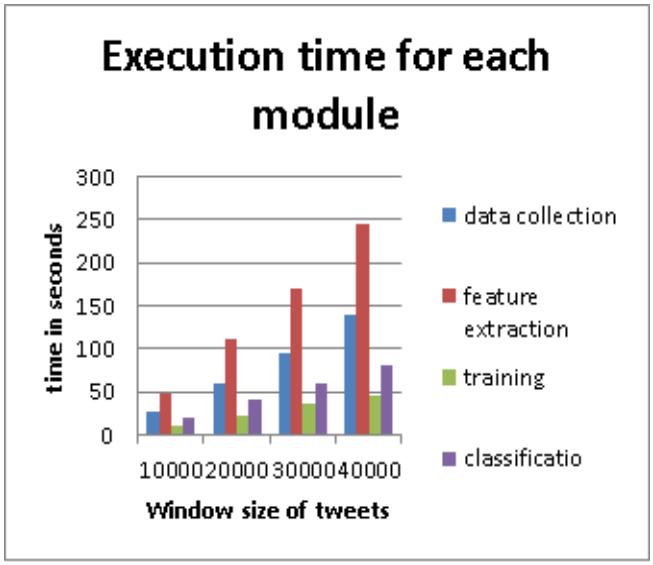

Fig 4: Execution time of each component

8. CONCLUSION

Existing suspicious URL Detection methods cannot handle conditional redirection servers that recognize crawler from typical programs and side-track them to non-suspicious pages to shroud malevolent points of arrival. Also existing system have the limitation on collecting the number of tweets as well as they cannot handle multi redirection. This paper, `presents another suspicious URL Detection framework for Twitter. Dissimilar to the past frameworks, this system is powerful when securing against restrictive redirection, on the grounds that it doesn't depend on the peculiarities of noxious points of arrival that may not be reachable. Rather, it concentrates on the connections of various side-track chains that impart redirection servers. This system not only handles the static HTTP redirections but also Dynamic redirection. Also this system can handle unlimited number of tweets from all the tweets so distributed version needs to be implemented to handle all the tweets from twitters public timeline. In future system will be implemented with some modification so that it can be adapted to other services like FACEBOOK, Google plus where the same shortened URLs are shared. Also current feature extraction method can be fabricated by attacker by altering the values of some features, example number of friends and followers etc,.so the strongest feature extraction method will be implemented.

9. REFERENCES

[1] S. Lee and J. Kim, "WarningBird: Detecting suspicious URLS in Twitter stream," in Proc. NDSS, 2012.

[2] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a socialnetwork or a news media?" in Proc. WWW, 2010.

[3] Antoniades, I. Polakis, G. Kontaxis, E. Athanasopoulos, S. Ioannidis, E. P. Markatos, and T. Karagiannis, "we.b: The web of short URLS," in Proc. WWW, 2011.

[4] Klien and M. Strohmaier, "Short links under attack: geographical analysis of spam in a URL shortener network," in Proc. ACM HT, 2012.

[5] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design andevaluation of a real-time URL spam filtering service," in Proc. IEEE S&P, 2011.

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

[7] M. Cova, C. Kruegel, and G. Vigna, "Detection and analysis of drive-by-download attacks and malicious JavaScript code," in Proc. WWW, 2010.

[8] J. Zhang, C. Seifert, J. W. Stokes, and W. Lee, "ARROW: Generatingsignatures to detect drive-by downloads," in Proc. WWW, 2011.

[9] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLS," in Proc. ACM KDD, 2009.