ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

IJIRSET

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Warning Bird Mail Alert Based Malicious URLs Blocker System in Twitter

T.Lakshmi[1], S.Parthiban[2]

[1]Department of Computer Science, Arunai College of Engineering, Thiruvannamalai, Tamilnadu,India

[2]Department of Computer Science, Arunai College of Engineering, Thiruvannamalai, Tamilnadu,India

ABSTRACT— Twitter is prone to malicious tweets containing URLs for spam, phishing, and malware distribution. Conventional Twitter spam detection schemes utilize account features such as the ratio of tweets containing URLs and the account creation date, or relation features in the Twitter graph. These detection schemes are ineffective against feature fabrications or consume much time and resources. Conventional suspicious URL detection schemes utilize several features including lexical features of URLs, URL redirection, HTML content, and dynamic behavior. However, evading techniques such as time-based evasion and crawler evasion exist. In this paper, we propose WARNINGBIRD, a suspicious URL detection system for Twitter. Our system investigates correlations of URL redirect chains extracted from several tweets. Because attackers have limited resources and usually reuse them, their URL redirect chains frequently share the same URLs. We develop methods to discover correlated URL redirect chains using the frequently shared URLs and to determine their suspiciousness. We collect numerous tweets from the Twitter public timeline and build a statistical classifier using them. Evaluation results show that our classifier accurately and efficiently detects suspicious URLs.WARNINGBIRD as a near real-time system for classifying suspicious URLs in the Twitter stream. In this project I proposed block the malicious URLs and provide mail alert for malicious URLs occur in the twitter stream.

KEYWORDS—Twitter, correlation, share URLs, spam, reciprocity, crawl

I. INTRODUCTION

Twitter is a micro blogging service less than three years old, command more than 41 million users as of July 2009 and is growing fast. Twitter users tweet about any topic within the 140-character limit also know as tweets and follow others to receive their tweets. Twitter it is a new medium of information sharing. We have crawled the entire Twitter site and obtained 41:7 million user profiles, 1:47 billion social relations, 4; 262 trending topics, and 106 million tweets. In its follower-following topology analysis we have found a non-power-law follower distribution, a short effective diameter, and low reciprocity, which all mark a deviation from known characteristics of human social networks. When a user Alice updates (or sends) a tweet, it will be distributed to all of her *followers* who have registered Alice as one of their friends. Instead of distributing a tweet to all of her followers, Alice can also send a tweet to a specific twitter user Bob by mentioning this user by including *@Bob* in the tweet. Unlike status updates, mentions can be sent to users who do not follow Alice. Twitter, we have ranked users by the number of followers and by Page Rank and found two rankings to be similar. When Twitter users want to share a URL with friends via tweets, they usually use URL shortening services to reduce the URL length since tweets can contain only a restricted number of characters. bit.ly and tinyurl.com are widely used services, and Twitter also provides a shortening service t.co.

Social networking sites have become one of the main ways for users to keep track and communicate with their friends online. Sites such as Face book, MySpace, and Twitter are consistently among the top 20 most-viewed web sites of the Internet.

All current Online Social Networks (OSNs) adopt the client-server architecture. The OSN service provider acts as the controlling entity. It stores and manages all the content in the system.OSN is using online spam filtering is deployed at the OSN service provider side Once deployed, it inspects every message before rendering the message to the intended recipients and makes immediate decision on whether or not the message under inspection should be dropped .If it is illegal message mean immediately dropped the message otherwise it is forward to the corresponding receiver.

OSN users form a huge social graph, where each node represents an individual user. In Face book-like OSNs, a social link would connect two nodes if the two corresponding users have mutually agreed to establish a social connection. Two users without a social link between them cannot directly interact with each other. Twitter-like OSNs impose looser restrictions, where a user can "follow" anyone to establish directed social link, so that he can receive all the updates. Recent studies suggest that the majority of spamming accounts in OSNs are compromised account. The below Fig. 1 Cumulative distribution of the social degree of spamming and legitimate accounts, respectively.

The popularity of Twitter, malicious users often try to find a way to attack it. The most common forms of Web attacks, Including spam, phishing, and malware distribution attacks, have also appeared on Twitter. Because tweets are short in length, attackers use shortened malicious URLs that redirect Twitter users to external attack servers.

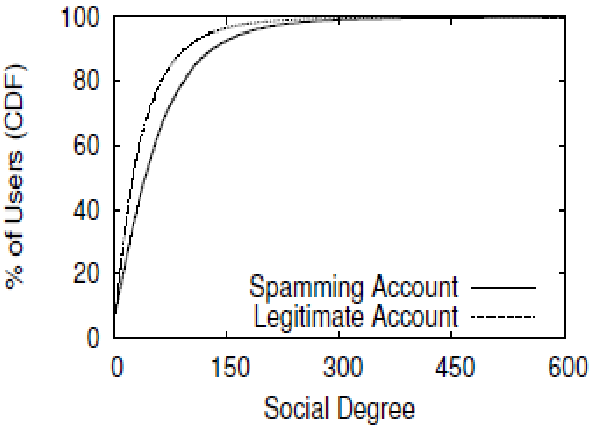


Fig. 1: Cumulative distribution of the social degree of Spamming and legitimate accounts, respectively.

To cope with malicious tweets, several Twitter spam detection schemes have been proposed. These schemes can be classified into account feature-based relation feature based and message feature based schemes. Account feature-based schemes use the distinguishing features of spam accounts such as the ratio of tweets containing URLs, the account creation date, and the number of followers and friends. However, malicious users can easily fabricate these account features. The relation feature-based schemes rely on more robust features that malicious users cannot easily fabricate such as the distance and connectivity apparent in the Twitter graph. Extracting these relation features from a Twitter graph, however, requires a significant amount of time and resources as a Twitter graph is tremendous in size. The message feature-based scheme focused on the lexical features of messages. However, spammers can easily change the shape of their messages.

A number of suspicious URL detection schemes have also been introduced. They use static or dynamic crawlers, and they may be executed in virtual machine honey pots, such as Capture-HPC, HoneyMonke, and Wepawet, to investigate newly observed URLs. These schemes classify URLs according to several features including lexical features of URLs, DNS information, URL redirections, and the HTML content of the landing pages. Nevertheless, malicious servers can bypass an investigation by selectively providing benign pages to crawlers. For instance, because static crawlers usually cannot handle JavaScript. or Flash, malicious servers can use them to deliver malicious content only to normal browsers. Malicious servers can also employ temporal behaviors—providing different content at different times to evade an investigation.

II. RELATED WORK

In this paper, we propose WARNINGBIRD, a suspicious URL detection system for Twitter. Instead of investigating the landing pages of individual URLs in each tweet, which may not be successfully fetched, we considered correlations of URL redirect chains extracted from a number of tweets. Because attacker's resources are generally limited and need to be reused, their URL redirect chains usually share the same URLs. We therefore created a method to detect correlated URL redirect chains using such frequently shared URLs. By analyzing the correlated URL redirect chains and their tweet context information, we discover several features that can be used to classify suspicious URLs. We collected a large number of tweets from the Twitter public timeline and trained a statistical classifier using the discovered features. The trained classifier is shown to be accurate and has low false positives and negatives.
The contributions of this paper are as follows:

- We present a new suspicious URL detection system for Twitter that is based on the correlations of URL redirect chains, which are difficult to fabricate. The system can find correlated URL redirect chains using the frequently shared URLs and determine their suspiciousness in almost real time.

- We introduce new features of suspicious URLs: some of which are newly discovered and while others are variations of previously discovered features.

- We present the results of investigation conducted on suspicious URLs that have been widely distributed through Twitter over several months.

- We also present the results of suspicious URLs that have been immediately informed the user through Mail Alert.

III. PROPOSED ALGORITHM

Here we present the proposed OFFLINE SUPERVISED LEARNING ALGORITHM **(OSLA)** supervised algorithms require categorized examples. After presenting these examples to the algorithm, adaptations are made to the configuration such that the different categories are recognized correctly in the future. With non-supervised learning, there is no explicit set of good and bad examples. In our project, we use an offline supervised learning algorithm, the feature vectors for training are relatively older than feature vectors for classification. To label the training vectors, we use the Twitter account status; URLs from suspended accounts are considered malicious whereas URLs from active accounts are considered benign. We periodically update our classifier using labeled training vectors.

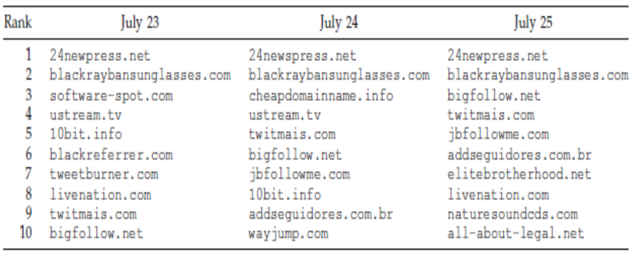
In this section discusses issues related to algorithm. Three Steps used in our offline supervised learning algorithm

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

A. *Frequent URL Redirect Chain*

We performed a simple investigation on three days' worth of tweet samples culled from July 23 to 25, 2011. We extracted frequent URL redirect chains from the sample data and ranked them according to their frequency after removing white listed domain names. Many suspicious sites, such as jbfollowme.com, which attempts to attract Justin Bieber's fans, proved to be highly ranked (Table 1).

TABLE 1
Domain names of frequent URL redirect chains on July 23–25, 2011



Rank	July 23	July 24	July 25
1	24newpress.net	24newspress.net	24newpress.net
2	blackraybansunglasses.com	blackraybansunglasses.com	blackraybansunglasses.com
3	software-spot.com	cheapdomainname.info	bigfollow.net
4	ustream.tv	ustream.tv	twitmais.com
5	10bit.info	twitmais.com	jbfollowme.com
6	blackreferrer.com	bigfollow.net	addseguidores.com.br
7	tweetburner.com	jbfollowme.com	elitebrotherhood.net
8	livenation.com	10bit.info	livenation.com
9	twitmais.com	addseguidores.com.br	naturesoundcds.com
10	bigfollow.net	wayjump.com	all-about-legal.net

B. *Reoccurrences of Redirect Chain*

We also discovered that suspicious URL redirect chains have frequently reoccur in the Twitter public timeline over several days. To verify the reoccurrence of suspicious URL redirect chains, we extracted the benign (posted by active accounts) and suspicious (posted by suspended accounts) URL redirect chains for each day in September 2011, and checked the average number of repetitions of the extracted URL redirect chains during the ensuing 60 days. Let *D* denotes a set of days in September 2011, **B**(*di*) denotes a set of benign entry point URLs on *di*, **S**(*di*) denotes a set of suspicious entry point URLs on *di* ∈ *D*, and **A**(*di*) denotes a set of all entry point URLs on *di*. For each *di,j* ∈ *{j* days later from *di* : 1 ≤ *j* ≤ 60*}*, we compute the following equations:

$$\sum_{d_i \in \mathcal{D}} \left(\frac{|\mathbf{B}(d_i) \cap \mathbf{A}(d_{i,j})|}{|\mathbf{B}(d_i)|} \right) / |\mathcal{D}|,$$
$$\sum_{d_i \in \mathcal{D}} \left(\frac{|\mathbf{S}(d_i) \cap \mathbf{A}(d_{i,j})|}{|\mathbf{S}(d_i)|} \right) / |\mathcal{D}|,$$

di,j 's are between September 2 and November 29, 2011

C. Check Whether same URLs is Posted

we can identify same URLs is used to posted same messages to all other users from same IP address then discover the meaningful characteristics of suspicious URLs. They use a number of different Twitter accounts and shortened URLs, or a number of domain names and IP addresses to cloak the same suspicious URLs. They also use long redirect chains to avoid investigation. Moreover, they appear more frequently in the Twitter public timeline than benign URLs over several days. These characteristics form the basis for the features we employ to classify suspicious URLs.

IV. SYSTEM DESIGN

Fig 2 our system consists of six components: data collection, feature extraction, training, classification, detecting suspicious URLs, and MailAlert.

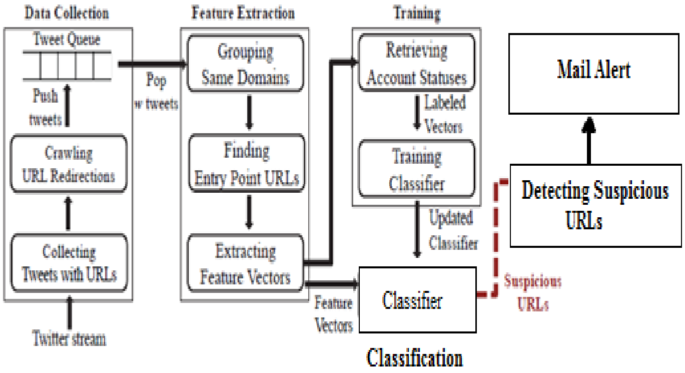


Fig. 2 System Architecture.

A. Data collection

It is important to notice that there is one important limitation imposed by the Twitter API. The number of requests could not exceed 350 per hour, which limits considerably the possibility to retrieve a large amount of samples, so we had to use several accounts to gather them. Our Java-based collecting method obtained, from the selected profiles, the users' ID and the timeline tweets, until having at least 100 genuine tweets. A genuine tweet is the tweet that is generated by the user itself (i.e., written by itself) and is not one re-tweet of another user's tweet.

The data collection component has two subcomponents: the collection of tweets with URLs and crawling for URL redirections. To collect tweets with URLs and their context information from the Twitter public timeline, this component uses Twitter Streaming APIs. Whenever this component obtains a tweet with a URL, it executes a crawling thread that follows all redirections of the URL and looks up the corresponding IP addresses. The crawling thread appends these retrieved URL and IP chains to the tweet information and pushes it into a tweet queue. As we have seen, our crawler cannot reach malicious landing URLs 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.

B. Feature Extraction

Our dataset contains the following features extracted from each of the profiles the tweets, time of publication, language, and geoposition and Twitter client. The first feature, the tweet, is the text published by the user, which gives us the possibility of determine a writing style, very characteristic of each individual. The time of publication helps determining the moments of the day in which the users interact in the social network. The language and geoposition also help filtering and determining the authorship because users have certain behaviors which can be extrapolated analyzing these features. Finally, despite being possible that users have several devices from where they tweet (e.g., PC, Smartphone or tablet), they usually choose to do it using their favorite Twitter client, which gives us another filtering mechanism.

The feature extraction component has three subcomponents: grouping of identical domains, finding entry point URLs, and extracting feature vectors. This component
Monitors the tweet queue to determine 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 several URLs share. at least one IP address, it replaces their domain names with a list of domains with which they are grouped.

For instance, when http://123.com/hello.html and http://xyz.com/hi.html share the same IP address, this replaces these URLs with http://[‘123.com’,‘xyz.com’]/hello.html and http://[‘123.com’,‘xyz.com’]/hi.html. This grouping process enables the detection of suspicious URLs that use several domain names to bypass the blacklisting.

Next, this 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 these tweets. It then discovers the most frequent URL in each URL redirect chain in the *w* tweets. The discovered URLs thus 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, the component finds URL redirect chains that contain the entry point URL, and extracts various features from these URL redirect chains along with 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 white listed domains to reduce false positive rates. White listed domains are not grouped with other domains and are not selected as entry point URLs.

C. Training

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

D. 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 an in-depth investigation.

E. Detecting Suspicious URL

In this module, we proposed a new suspicious URL detection system for Twitter, called WARNINGBIRD. Unlike the conventional systems, WARNINGBIRD is robust when protecting against conditional redirection, because it does not rely on the features of malicious landing pages that may not be reachable. Instead, it focuses on the correlations of multiple redirect chains that share the same redirection servers. We introduced new features on the basis of these correlations, implemented a near real-time classification system using these features, and evaluated the system's accuracy and performance.

F. Mail Alert

In this module, we enhance our system by providing mail alert system. Though the suspicious URLs are detected in an efficient way, it is unknown to the twitter users. Thus a Mail Alert system is generated for providing an alert before the usage of the malicious URLs.

V. DISCUSSIONS

URL shortened services such as bit.ly and others play a critical role on the web today. Using this shortened service attacker easily attack our server. Online algorithm is uses sliding window for achieving goal latency and detection coverage. But if it is small window fives immediate results but it cannot catch suspicious URLs otherwise it is large window size mean has goal detection coverage its latency bad .Online supervised learning algorithm consume more time and cost. we present offline supervised algorithm its overcome the above problem. The number of short URL is input to the algorithm. It first finds frequent URLs and then finds Entry point it is intermediate URLs that are associated with private redirection servers and finally check whether same URL is posted to other users from same IP. If three cases success mean classified it is normal URLs or suspicious URLs. It is normal URLs mean rendered it to corresponding receiver otherwise block it identity provide the mail alert to legitimate users.

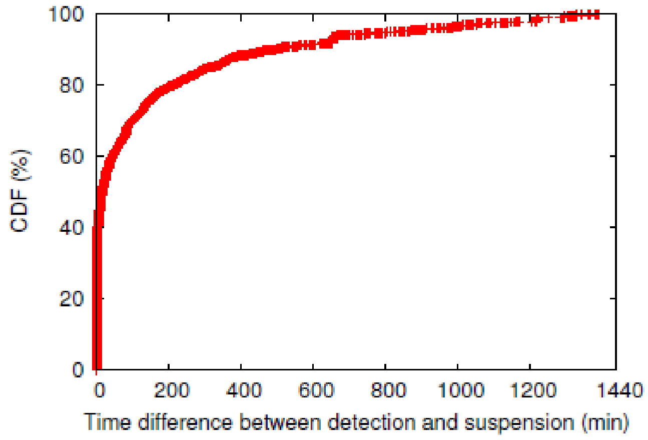


Fig. 3 Time difference between Warning Birds detection of suspicious accounts and Twitter's suspension within a day

We compare the efficiency of WARNINGBIRD with that of Twitter's detection system. For the comparison, we sampled 14,905 accounts detected by the real-time WARNINGBIRD between September 1 and October 22, 2011. To compare their efficiencies, we measured the time difference between WARNINGBIRD's detection and Twitter's suspension of the accounts. Among the sampled accounts, 5; 380 accounts were suspended within a day; 37:3% of them was suspended within a minute, another 42:5% of them was suspended within 200 minutes, and the remaining 20:7% of them was suspended within a day. (Fig 3).

VI. RESULT SET

Previous suspicious URL detection systems are weak at protecting against conditional redirection servers that distinguish investigators from normal browsers and redirect them to benign pages to cloak malicious landing pages its disadvantage is time consuming and less detection accuracy (Fig 4) our proposed system less time and high detection accuracy.(Fig 5)

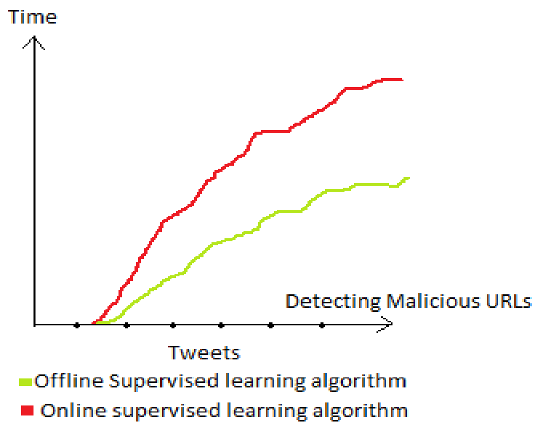


Fig. 4 Time consuming between offline and online supervised algorithm.

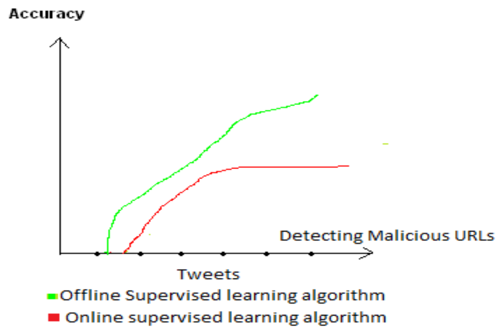


Fig. 5 Accuracy between offline and online supervised algorithm

VII. CONCLUSION

We proposed a new a new suspicious URL detection system for Twitter, called WARNINGBIRD. Unlike the conventional systems, WARNINGBIRD is robust when protecting against conditional redirection, because it does not rely on the features of malicious landing pages that may not be reachable. Instead, it focuses on the correlations of multiple redirect chains that share the same redirection servers. We introduced new features on the basis of these correlations implemented a near real-time classification system using these features, and evaluated the system's accuracy and performance. The evaluation results show that our system is highly accurate and can be deployed system to classify large samples of tweets from the Twitter public timeline. Using offline supervised learning algorithm to detect the suspicious URLs in Twitter stream then immediately block that URLs

and also provide alert to user through Mail. We present Malicious URLs blocker system provide high accuracy. Our main future objective is to extend these ideas to address to address dynamic and multiple redirections. We will also implement a distributed version of WARNINGBIRD to process all tweets from the Twitter public timeline.

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 social network or a news media?" in *Proc. WWW*, 2010.

[3] G. Stringhini, C. Kruegel, and G. Vigna, "Detecting spammers on social networks," in *Proc. ACSAC*, 2010.

[4] S. Chhabra, A. Aggarwal, F. Benevenuto, and P.Kumaraguru, "Phi.sh/$social: the phishing landscape through short URLs," in *Proc. CEAS*, 2011.

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

[6] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida, "Detecting spammers on Twitter," in *Proc. CEAS*, 2010.

[7] H. Gao, Y. Chen, K. Lee, D. Palsetia, and A. Choudhary, "Towards online spam filtering in social networks," in *Proc. NDSS*, 2012.