Mail Alert based Suspicious and Malicious Tweet Urls Blocker System in Twitter

Priya.U
PG Scholar
Department of computer science and engineering
Sri Shakthi Institute of Engineering and Technology

R. Vidhyaprakash
Assistant professor
Department of computer science and engineering
Sri Shakthi Institute of Engineering and Technology

Abstract - With the advent of online public media, phishers using public networks like Twitter, Facebook, and Foursquare to spread phishing scams. Twitter is an hugely well-liked micro-blogging network where people place petite messages of 140 characters called tweets. It has over 100 million dynamic users who place about 200 million tweets everyday. In progress with Twitter, phisher use it as a medium to spread phishing because of this vast information dissemination. Because of short content size, and use of URL, it is difficult to detect phishing on Twitter unlike emails. Our technique, PhishAri, detects phishing on Twitter in realtime. We use Twitter explicit features along with URL features to sense whether a tweet posted with a URL is phishing or not. Some of the Twitter explicit 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 relocation the tweet such as age of the report, number of tweets, and the follower ratio. These twitter specific features coupled with URL based features prove to be a strong mechanism to detect phishing tweets. We use instrumental learning classification techniques and detect phishing tweets.

Key terms: Suspicious URL, Twitter, URL redirection, conditional redirection, classification, mail alert.

I. INTRODUCTION

Twitter is a famous social networking and information sharing service that allows users to exchange messages with their friends up to 140-character, also called as tweets, When a user Alice sends a tweet, it will be spread to all of her followers who have registered Alice as one of their friends. Alice can send a tweet to a exact twitter user Bob by mentioning this user by including @Bob in the tweet, instead of sending a tweet to all of her followers. Unlike status updates, reference to someone can be sent to users who do not follow Alice. Through tweets, twitter user wish to supply URL with friends in order to reduce the length of URL by using URL shortening service because only the restricted numbers of characters are used in tweets. bit.ly and tinyurl.com are widely used services, and a shortening service can also provided by Twitter. Malevolent users can try to discover a method to attack twitter because of its reputation. The most general forms of Web attacks such as spam, fiddle, phishing, and malware distribution attacks, have also spread over Twitter, because length of the tweet is short. Attackers use short length malicious URLs that redirect twitter users to outside attack servers. To manage with malicious tweets, numerous spam detecting system can be used in Twitter. These schemes can be classified into description feature-based, relative feature-based, and communication feature-based system. Description feature-based system can use the unique features of spam accounts such as the proportion of tweets containing URLs, the description creation date, and the number of supporters and associates though, malicious users can easily construct these description features. The relative feature-based schemes rely on more vigorous features that malicious users cannot easily construct such as the distance and connectivity apparent in the Twitter graph. Extracting these relative features from a Twitter graph, however, requires a major amount of time and resources as a Twitter graph is tremendous in size. The communication feature-based schemes pay attention on the lexical features of messages, though spammers can easily change the shape of their messages.

II. PROPOSED SYSTEM

Attackers use short length malicious URLs that redirect twitter users to outside attack servers. To manage with malicious tweets, numerous spams detecting system can be used in Twitter. These schemes can be classified into description feature-based, relative feature-based, and communication feature-based system. Description feature-based system can use the unique features of spam accounts such as the proportion of tweets containing URLs, the description creation date, and the number of supporters and associates though; malicious users can easily construct these description features. The relative feature-based schemes rely on more vigorous features that malicious users cannot easily construct such as the distance and connectivity apparent in the Twitter graph. Extracting these relative features from a Twitter graph, however, requires a major amount of time and resources as a Twitter graph is tremendous in size. The communication feature-based schemes pay attention on the lexical features of messages, though spammers can easily change the shape of their messages. A number of suspicious URL detection schemes have also been introduced. Disadvantages of this system are i) Malicious servers can bypass an investigation by selectively providing benign pages to crawlers. ii) For instance, because static crawlers usually cannot handle JavaScript or Flash, malicious servers can use them to deliver malicious content only to normal browsers. iii) A recent technical report from Google has also discussed techniques for evading current Web malware detection systems. iv) Malicious servers can also employ temporal behaviors— providing different content at different times—to evade an investigation

Our technique, PhishAri, detects phishing on Twitter in realtime. We use Twitter explicit features along with URL features to sense whether a tweet posted with a URL is phishing or not. Some of the Twitter explicit 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 relocation the tweet such as age of the report, number of tweets, and the follower ratio. These twitter specific features coupled with URL based features prove to be a strong mechanism to detect phishing tweets. We use instrumental learning classification techniques and detect phishing tweets.

Our system consists of six components: Data collection, Feature extraction**,** Training, Classification, Detecting Suspicious URL, Mail Alert.

A.data collection -

The data collection factor has two subcomponents:

- The gathering of tweets with URLs and
- Crawling for URL redirections

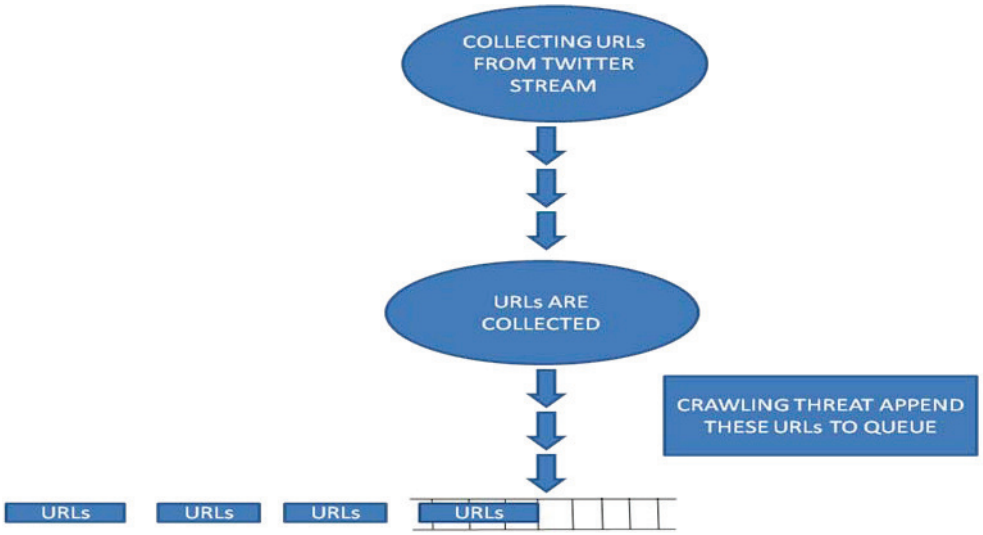

Figure1. Structure of data collection

Collection of tweets with URLs and their background information from the Twitter public timeline, this factor uses Twitter Streaming APIs. Whenever the data collection factor obtains a tweet with URL, it executes a crawling thread that follows all redirections of the URL and looks up the parallel IP addresses. The crawling thread then pushes the tweet information into a tweet queue along with these retrieved URL and IP chains. As we have seen, our crawler cannot get to malevolent landing URLs when they use conditional redirections to escape crawlers. However, because our finding system does not rely on the features of landing URLs, it works separately of such crawler evasions.

B. feature extraction -

The feature extraction factor has three subcomponents:

- Grouping of identical domains
- Finding entry point URLs
- Extracting feature vectors

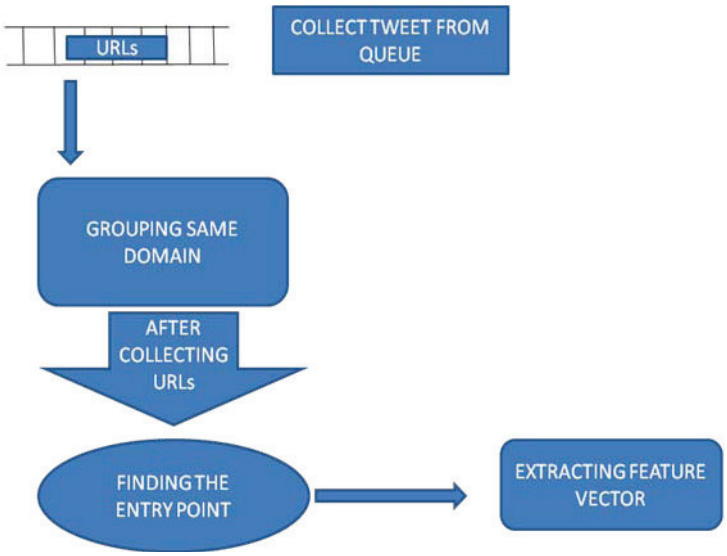

Figure2. Structure of feature extraction

This component monitors the tweet queue to determine whether a sufficient number of tweets have been composed. instead of using individual tweets., our system uses a tweet window. 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 a number of URLs share no less than one IP address, it replaces their field names with a list of fields with which they are grouped.

For instance, when http://abc.com/hello.html and http://pqr.com/hi.html share the same IPAddress, this component replaces these URLs with http://[‘abc.com’,‘pqr.com’]/hello.html and http://[‘abc.com’,‘pqr.com’]/hi.html, respectively. This grouping procedure gives (someone) the authority for the detection of suspicious URLs that use several domain names to bypass the blacklisting.

Next, this factor tries to discover the doorway of URL for each of the *w* tweets. First, it measures the frequency with which each URL appears in these tweets.The most frequent URL is discovered in each URL redirect chain in the *w* tweets. The discovered URLs thus become the entry points for their redirect chains. If more than one URLs share the highest frequency in a URL chain, this factor selects the entry point URL which is nearer to the beginning of the chain.

Finally, for each entry point URL, the factor 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 Whitelisted domains are not grouped with other domains and are not selected as entry point URLs. Our whitelisted field names include the Alexa Top 1000 sites, some famous URL shortening sites, and some domains that we have manually verified.

C. training -

The training factor has two subcomponents:

- Retrieval of account statuses
- Training of the classifier.

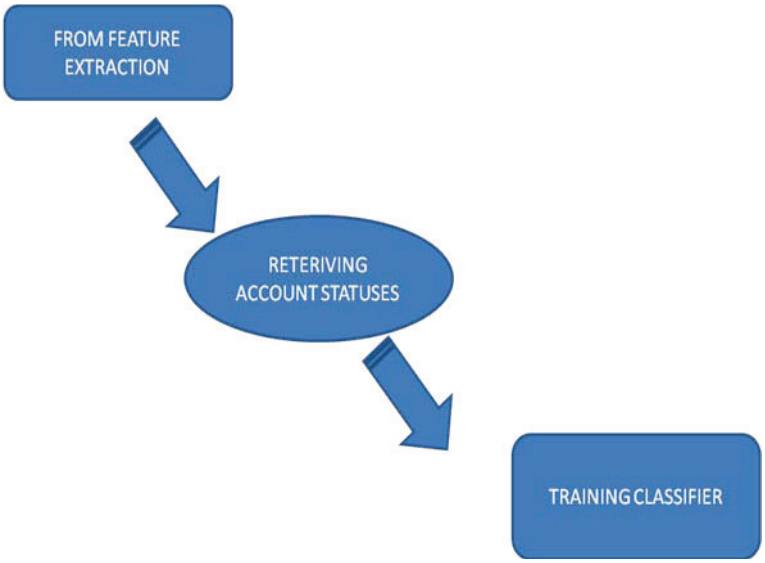

Figure3. Structure of training

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 -

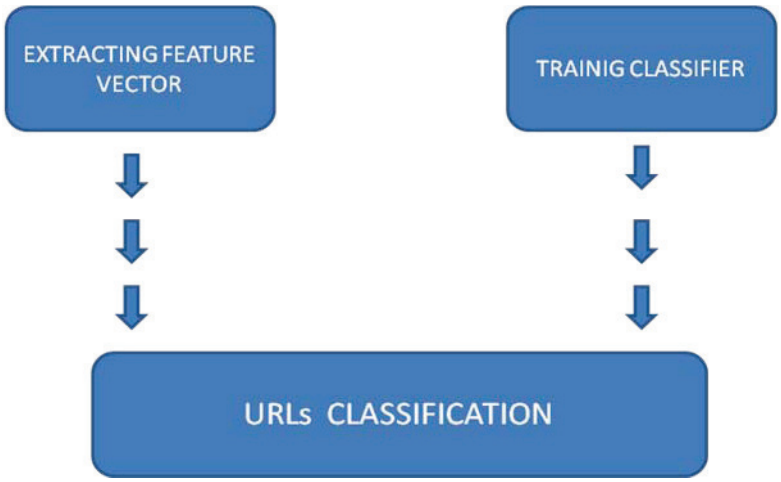

Figure4. Structure for classification

The classification factor executes our classifier using input feature vectors to classify suspicious URLs. Number of malicious feature can be returned by classifier. This factor then flags, that the particular URLs and their tweet information is suspicious. For an in-depth investigation, the detected suspicious URLs will be moved to security experts or more sophisticated dynamic analysis environments.

E. detecting suspicious url -

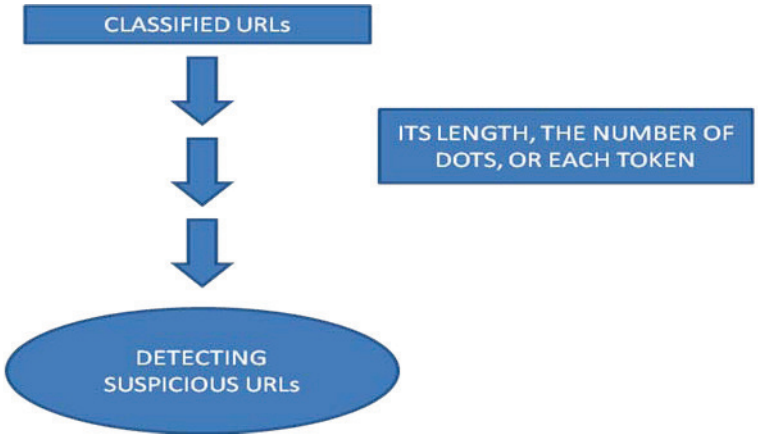

Figure5. Structure for suspicious URLs detection

Many suspicious URL detection systems have been used and it can be classified into either static or dynamic detection systems. Lexical features of a URL such as its length, the number of dots, or each token it has, and also considers underlying DNS and WHOIS information can be focused by some trivial static detection systems. The most difficult static detection system, such as Prophiler, Drive-by download attacks can be detected by additionally extracted features from HTML content and JavaScript codes. However, suspicious URLs with dynamic content cannot be detected by static detection systems by using such unintelligible JavaScript, Flash, and ActiveX content. Therefore, we need dynamic detection systems. For in-depth study of suspicious URLs dynamic detection systems uses virtual machines and instrumented Web browsers. Yet, all of these detection systems may still fail to detect suspicious sites with conditional behaviors.

F. mail alert -

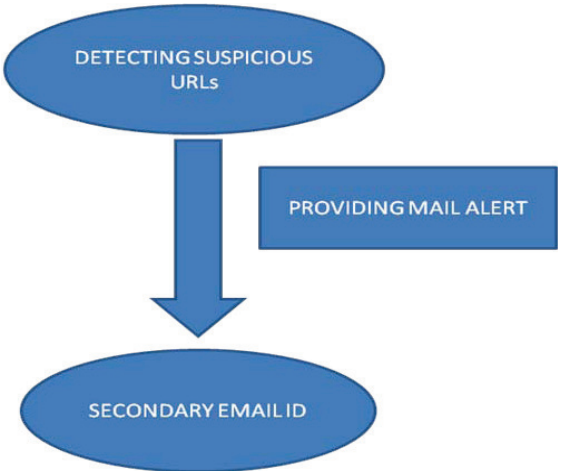

Figure6. Structure for mail alert

Phishing emails sent by the phishers to seize the information of the end user is identified by using the link Guard algorithm .Careful analysis of the characteristics of phishing hyperlinks is done by Link Guard and also each end user is implemented with this algorithm. After identify the phishing email, the end user cannot send any message to such mail. Detection and prevention for known and unknown phishing attacks because Link Guard is characteristics based. After detecting the suspicious URLs it will go for further investigation for conforming the URLs is suspicious, if it is suspicious then alert message is passed to the secondary mail.

III. CONCLUSION

Conventional suspicious URL detection systems are ineffective in their protection against conditional redirection servers that distinguish investigators from normal browsers and redirect them to benign pages to cloak malicious landing pages. In this study, we built PhishAri – an effective mechanism to detect phishing on Twitter. Our methodology exploits not just the traditional phishing detection features which are based on the URL and the suspicious landing page, but also Twitter specific and WHOIS based features. We use a combination of URL based and Twitter based features which help in an effective and realtime detection of phishing on Twitter. In the future, we will extend our system to address dynamic and multiple redirections.

REFERENCES

[1] M. Jakobsson and S. Myers, Eds., Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft. Wiley-Interscience, 2006.

[2] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru, “Phi.sh/social: the phishing landscape through short urls,” in Collaboration, Electronic messaging, Anti-Abuse and Spam Conference. ACM, 2011.

[3] Anupama Aggarwal, Ashwin Rajadesingan, ”PhishAri:Automatic Realtime Phishing Detection on Twitter,” in collaboration spam report and intelligence report,2012.

[4] S. Lee and J. Kim, “WarningBird: Detecting suspicious URLs in Twitter stream,” in Proc. NDSS, 2012.

[5] Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M.Voelker,”Beyond Blacklists: Learning to Detect Malicious Web Sitesfrom Suspicious URLs”, ACM, 2007.

[6] Jonghyuk Song1, Sangho Lee1 and Jong Kim,”Spam Filtering in Twitter using Sender-Receiver Relationship”, ACM, 2011.

[7] M. Cova, C. Kruegel, and G. Vigna. Detection and Analysis of Drive-by Download Attacks and Malicious JavaScript Code. In Proceedings of the International World Wide Web Conference (WWW), 2010.

[8] B. Feinstein and D. Peck. Caffeine Monkey: Automated Collection, Detection and Analysis of Malicious JavaScript. In Proceedings of the Black Hat Security Conference, 2007.

[9] S. Garera, N. Provos, M. Chew, and A. D. Rubin. A Framework for Detection and Measurement of Phishing Attacks. In Proceedings of the Workshop on Rapid Malcode (WORM), 2007.

[10] D. Antoniades, I. Polakis, G. Kontaxis, E. Athanasopoulos, S. Ioannidis, E. P. Markatos, and T. Karagiannis. we.b: the web of short urls. In Proceedings of the 20th international conference on World wide web, WWW ’11, pages 715–724, New York, NY, USA, 2011. ACM.

[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’11, New York, NY, USA, 2011. ACM.

[12] 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, CCS ’10, New York, NY, USA, 2010.

[13] T. Inoue, F. Toriumi, Y. Shirai, and S.-i. Minato. Great east japan earthquake viewed from a url shortener. In Proceedings of the Special Workshop on Internet and Disasters, SWID’11, New York, NY, USA, 2011. ACM.

[14] WILSON, C., BOE, B., SALA, A., PUTTASWAMY, K. P., AND ZHAO, B. Y. User interactions in social networks and their implications. In *Proceedings of the ACM European conference on Computer systems* (2009).

[15] G. Brown, T. Howe, M. Ihbe, A. Prakash, andK. Borders. Social networks and context-aware spam.In ACM Conference on Supportive Cooperative Work, 2008.