ISSN No: 2309-4893
International Journal of Advanced Engineering and Global Technology
Vol-2, Issue-8, August 2014

A Distributed System for Detecting Phishing and Mail Alert based Malicious Tweet URLs Blocker in a Twitter Stream

S.Mamatha
Department of CSE
ASCET, India

C.Rajendra
Department of CSE
ACSET, India

Abstract: Twitter is a hugely well-liked famous social network where people exchanges messages of 140 characters called tweets. Because of short content size, and use of URL, it is difficult to detect phishing on Twitter unlike emails. Ease of information exchange large audience makes Twitter as a popular medium to spread external content like articles, videos, and photographs by embedding URLs in tweets. In our existing system Warning Bird 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 paper describes a newly developed Distributed methodology that correlates the format of tweets with sending agents to detect phishing tweets automatically in real time by using Link Guard algorithm .We use machine learning classification techniques and detect phishing tweets with a higher accuracy than of existing systems . Also in this paper I proposed to block the malicious URLs and provide mail alert for malicious URLs occur in the twitter stream Using dynamic trained classifier.

Keywords: Phishing, URL Obfuscation, LinkGuard algorithm, URL redirection, Suspicious URL.

1. INTRODUCTION

Twitter, Facebook, Google+ is a well-known social networking and information sharing service that allows users to exchange messages.Registered users can read and post tweets, but unregistered users can only read them. When a user sends a tweet, it will be distributed to all of her followers who have registered as one of their friends.

Detecting phishing on social network is a challenge because of large volume of data - social network allow users to easily share their opinions and interests which results into large volumes of data and Hence, make it difficult to mine and analyze

Limited space: social network often impose character limitation (such as Twitter's 140character limit) on the content due to which users use Shorthand notations. Such shorthandNotation is difficult to parse since the text is usually not well-formed.

Shortened URLs: researchers have observed that more than half of the phishing URLs are shortened to obfuscate the target URL and to hide Twitter.

Phishing is a problem which occurred in most social networking sites. The most common methods used today for the detection and analysis of phishing web sites are:

Manual view and report services such as Phishtank.com [7]. This is by far the most common method which first, requires actual users to find, identify, and report suspicious web-pages and second requires, additional people to verify the status of reported pages.

Correlating links in known SPAM email to phishing sites [5]**.** Industry standard SPAM email detection techniques are used to identify SPAM email and then links from those emails are examined. Typical examination includes looking for URL redirection, or a variety of DNS tricks that might signify a phishing site. The emphasis is typically on examining characteristics of the URL itself. Unless a URL is malformed or some other SPAM-like characteristic is identified, phishing sites are often not identified in this way.

Crawler classification of websites
Pages themselves are examined for suspicious characteristics like misspelled words, link obfuscation, right-click menu disabling, DNS redirection, etc. [6]. This method is similar to our own in that it places a strong emphasis on fetching and analyzing the actual page rather than just the URL. However, it looks for suspicious content in individual pages where we focus on identifying groups of pages that would "look" the same to an unsuspecting user.
Whenever twitter users clicking links embedded in phishing messages that redirect them to malicious sites.

A URL is said to be redirected, if a client requests a re-source located at a specific URL, but the client's final destination at the end of the request is a different URL Based on the implementation techniques, URL redirections are classified into
1) Server-Side redirections,
2) JavaScript redirections, and
3) META redirections.

Server-side redirections: Server-side redirection occurs when a client requests a resource and the server issues a di-rective in the form of HTTP status codes which makes the client request through a different URL. HTTP reply status codes of type 3xx as well as some 4xx with a location field in the header imply that the client has to redirect to a different URL. For example, request for http://www.google.netreturns a status code 302 redirecting the request tohttp://www.google.com.

JavaScript redirections: JavaScript redirections are initiated at the client side through statements like `window.Location=some URL these instructions are inserted into the script sections of the HTML page sent and when the client JavaScript engine executes these statements, it is redirected to the URL specified within the script. Un-like server-side redirections, a web page is actually loaded partially or completely into the client browser before the redirection occurs.

META redirections: Another client side redirection is based the META tags located in the HEAD section of the HTML page. By setting the associated http-equiv attribute to refresh and the content attribute to a target URL, a redirection would be triggered at the client browser to this target URL. The redirection happens after a browser finishes parsing a HTML page, then the META refresh action is triggered to load content from the target URL.

This research paper describes a newly developed Trend Distributed methodology that correlates the format of tweets with sending agents to detect phishing messages [4] we demonstrate how we use "Distributed data analytics" to proactively identify phishing messages so we can protect our Tweet threats. 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 Twitterspecificfeatures [1] such as tweet content and its characteristics like length, hash tags, and mentions Along with URL features [1],[2] to detect whether a tweet posted with a URL is phishing or not.

We use machine learning classification techniques and detect phishing tweets with an accuracy of 92.52%. We have built our system by providing a new browser to the end-users. 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. Tweets contain texts and http links. Earlierdetection mechanisms were not efficient, but time consuming.The features used for the purpose of detection could be fabricated easily by an attacker. Therefore the problem is to provide security to twitter users by identifying malicious URLs from benign ones in real time. So many techniques for detecting spam [3] and suspicious URLs were introduced.

2. LITERATURE SURVEY

H.Kwak, C.Lee, H.Park, and S.Moon proposed a work in 2010 [1]which mainly focuses on Twitter, a social networking service ,more than 41 million users as of July 2009 and is growing fast. Twitter users tweet about any topic within the 140-character limit. Twitter offers an Application Programming Interface (API) that is easy to crawl and collect data. Twitter tracks phrases, words, and hash tags that are most often mentioned and post them under the title of "trending topics" regularly. A hash tag is a convention among Twitter users to create and follow a thread of discussion by prefixing a word with a '#' character.Inorder to identify influential on Twitter.

Juan Chen and chuanxiongguo described online detection of phishing attacks and prevention of phishing attacks [8]

Dhanalakshmi, prabhu and chellapan had described how to detect the phishing websites and to protect secure transaction [9]

3. PROPOSED SYSTEM

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 afore mentioned 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

We introduced a new end-host based anti-phishing algorithm, which we call Link Guard, by utilizing the generic characteristics of the hyperlinks provided by the Anti-Phishing Working Group (APWG)In detecting phishing attacks .it not only detects known but also unknown phishing attacks. We have implemented Link Guard in Windows XP. Our experiments verified that Link Guard is effective to detect and prevent both known and unknown phishing attacks with minimal false negatives

SHA-1 is an algorithm which is previously implemented. SHA refers “secure hash algorithm”. It is a cryptographic hash function which was designed by national security agency at the time of implementation SHA-1 is more secure but it slow down in execution as it includes many rounds and has collisions. Because of these drawbacks we had implemented an Efficient Link Guard Algorithm.

Link Guard works by analyzing the differences between actual links and visual links. It calculates the similarities of URI with a known trusted site; it contains some important systems they are:

Communication: Input process can be collected and send all the information to analyzer.

Database: It stores the user input URL’S Blacklist and White list.

Analyzer: Link Guard is the main component which is applied on link guard algorithm, all the data provided by Communication and database sends all results to alter and then to logger modules.

Alerter: Analyzer sends a warning message as soon as it receives from the analyzer and alerts the user, the Analyzer receives the reactions of the user.

Logger: All the related information is stored.

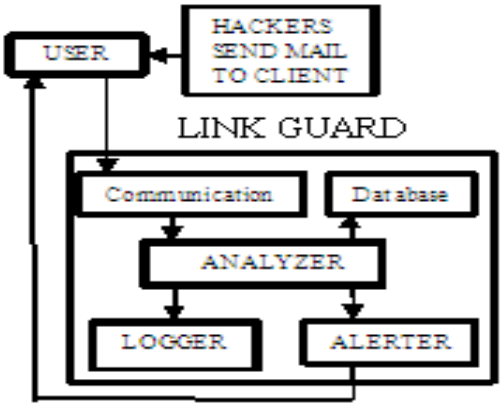

Fig: 1 .Link guard algorithm architecture

The Link Guard work is to examine the differences between the actual link and visual link. The following terms are used in the algorithm.
a _link: actual link;
v _ link: visual link;
a _ dns: actual DNS name;
v_ dns: visual DNS name;
sender_ dns: sender DNS names.
int Link Guard (a_ link, v_ link)

```
{
a_ dns=GetDNSName(a_ link);
v_ dns=GetDNSName (v_ link);
if ((a_ dns and v_ dns are both empty) and
(a_ dns ! = a_ dns))
return phishing;
if (a_ dns is dotted decimal)
return possible_ phishing;
if (v_ link or a_ link is encoded)
{
a_link2=decode(a_ link);
v_link2=decode(v_ link);
returnLinkGuard (v_link2, a_link2);
}
if (v_ dns is NULL)
returnAnalyzeDNS(a_ link);
}
intAnalyzeDNS (actual_ link)
{
if (actual_ dns in blacklist)
return PHISHING;
if (actual_ dns in whitelist)
return NOTPHISHING;
return Pattern Matching actual_ link);
}
Int Pattern Matching(actual _link)
{
if(sender_ dns and actual_ dns are different)
return POSSIBLE_PHISHING;
for (each item prev _dns in seed_ set)
{
bv=Similarity (prev_ dns, actual _link);
if (bv==true)
return POSSIBLE_PHISHING;
}
return NO_PHISHING;
}
float similarity(str, actual_ link)
{
if (str is part of actual_ link)
return=true;
int maxlen=the maximum string
lengths of str and actual_ dns;
int min change=the minimum number of
changes needed to transform str to
actual_dns;
if(thresh<(maxlen- minchange)/maxlen<1)
return true
return false;
}
```

In this efficient link guard algorithm firstly, we have to find out the DNS names from visual link and actual link. Secondly, it compares both visual and actual DNS names, if these names are not similar then it is phishing attack for line 3 and 5(Group1).We are having ipaddress which is said to be dotted decimal ip address, it directly used in actual_dns, then it is possible of phishing attack in lines 6 and 7(Group2).In this algorithm it checks character wise so that it can be easily find phishing attack When we don't have any destination information in visual link, link guard immediately calls and analyze DNS, which is used to analyze the actual dns. If the actual dns name is having black list then it is confirm that it is phishing attack. In the same manner if it in white list, then it is not a phishing attack. If it is not white list or black list then it invokes pattern matching.

The unknown attacks are handled by a special feature called pattern matching. In actual link we have DNS or ipaddress and in visual link we don't have DNS or the destination of ip address because visual link is used future analysis. To overcome this problem we are having two methods. Firstly, phishers retrieve the data or information from email address which is similar to legal information. Visual link and actual links looks similar but actual link contains legal information and visual link contains tricky information.

Normally hackers use legal DNS name in the sender email address to trick the

user. Secondly, we will think that all the inputted address or the data that are surfed by the user in the internet are trust worthy and we store these data in the seed set. Similarity procedure is checking the similar names with one or more names in the Seed set, by invoking this procedure pattern matching checks whether the actual DNS names is different from senders address.

Our offline supervised learning algorithm uses 3 steps for blocking and sending mail alerts in twitter.

- Frequent URL with similar domain names and from same IP address.
- Reoccurrences of redirect chains in URLs (entry points)

- Check whether same URL is posted to other users (followers) from same IP.

Support Vector Machines (SVM) is pretty much the standard classifier which is used for any general purpose classification

- Day X.svm (where *X* is an integer from 0 to 120) -- The data for day *X* in SVM-light format. A label of +1 corresponds to a malicious URL and -1 corresponds to a benign URL.
- Feature Types - A text file list of feature indices corresponds to real-valued features.

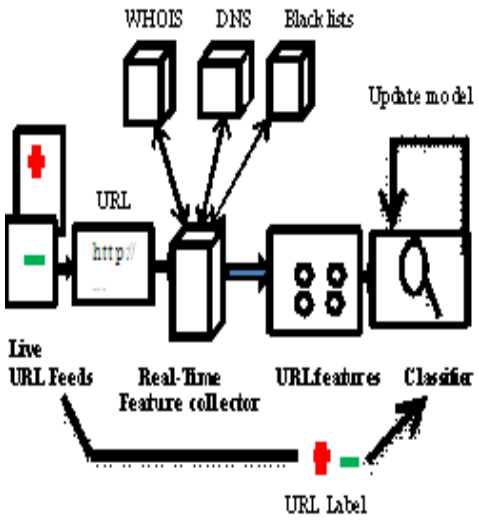

Fig.2machine learning techniques to detect malicious URLs

4. SYSTEM DETAILS

In this we implement a distributed architecture to detect phishing URLs.

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.

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.

Browser:

Web browser viewing tweets and this act as a container for real time 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.

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)

Tweet Reader:

Tweet reader reads from Twitter using Twitter

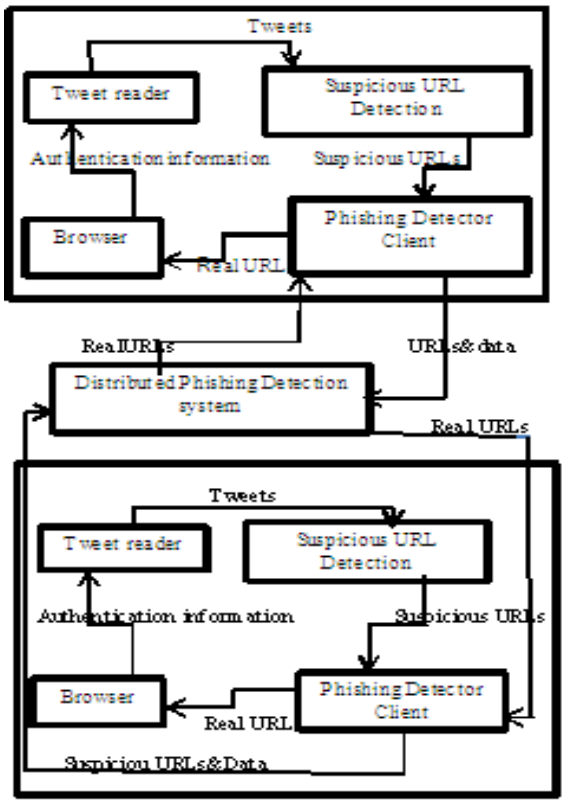

Fig.3Distributed system architecture For Suspicious URL Detection

Modules used for blocking and sending mail alerts in twitter:

- Data collection
- Feature extraction
- Training and classification
- Blocking
- Mail Alert

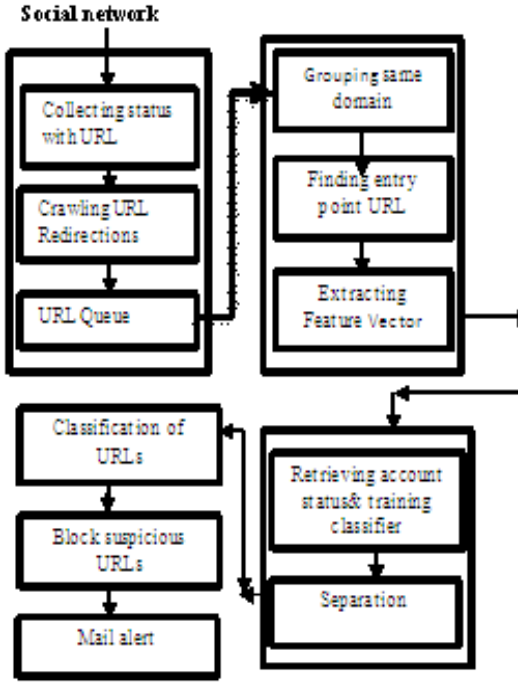

Figure.4 blocking malicious URLs and sending Mail alerts in twitter

A.Data collection

This module has two subcomponents:

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

Twitter Streaming APIs helps to collect tweets with URLs and their context information from the Twitter public timeline 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.

B. Feature extraction

The feature extraction factor has three subcomponents:

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

Our dataset contains the following features extracted from each of the profiles the tweets, time of publication, language, geo position 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 geo position also help filtering and determining the authorship because users have certain behaviors which can be extrapolated analyzing these features.

Machine learning is the tool where people have developed various methods to classify. Classifiers may or may not need training data but machine learning classifiers, Support Vector Machine need it. Classifiers require training data and hence these methods fall under the category of supervised classification

C. Training

The training factor has two subcomponents:

- Retrieval of account statuses
- Training of the classifier

Because we use an offline supervised learning algorithm, the feature vectors for classification are relatively newer than feature vectors for training. 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 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.

The classifier used is SVM Light can train SVMs with cost models and example dependent costs handles several hundred-thousands of training examples

E. Blocking

This module used to block the suspicious URLs which are detected by above modules it obtains the corresponding memory page from the target. The specification for URLs limits the allowed characters in aRequest-URI to only a subset of the ASCII character set.This means that the query parameters of a request-URIbeyond this subset should be encoded. Because a maliciouspayload may be embedded in the request-URI as a requestparameter

F. Mail Alert

In this module, we enhance our system by providing mailalert system. Though the suspicious URLs are detected in an efficient way, it is unknown to the twitter users. Careful analysis of the characteristics of suspicious URLs is done by offline supervised learning algorithm After detecting the suspicious URLs it will go for further investigation for conforming theURLs is suspicious, if it is suspicious then alert message is passed to the secondary mail

5. Implementation

Our goal is to develop a suspicious URL detection system for Twitter that is robust enough to protect against conditional redirections [1] (Fig5) shows the implementation framework of our proposed system. Thus shows how the benignURL and the malicious URL are classified which leads to the detection of attacker and block the malicious URL and prevents system disaster

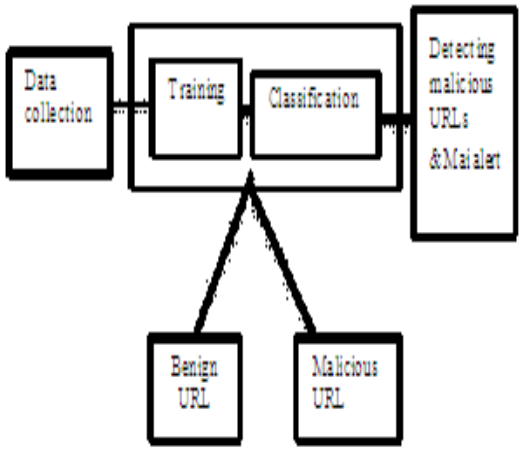

Fig 5.Implementation framework

1. Execution Setup for Link Guard Algorithm

- Intel R core TM i3 processor-2310M CPU @2.10ghz
- main memory RAM 3GB
- operating system 64bits
- java development lit jdk 1.7.0

6. RESULTS AND DISCUSSIONS

When we compare with both link guard algorithm and SHA algorithm. SHA algorithm has more time complexity and it has less secured because it is having more storage capacity. Link guard algorithm has more secure and has less rounds.

Attributes	Existing algorithm	Proposed link guard algorithm
Security	90%	94%
Number of phishing attacks	5	3
Generation time	4 msec	3 msec

7. CONCLUSION

In this study, since our goal was to detect Phishing on Twitter and also Provide 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. Perform machine learning mechanism on this collected information from various systems to get phishing URL. Also we have built a browser to provide realtime phishing detection to twitter users and it protects the user from falling prey to phishing attacks by appending a red indicator to phishing tweets. Using offlinesupervised learning algorithm to detect the suspicious URLs then immediately block that URLs and also provide alert to the twitter users through Mail. We present Malicious URLs blocker system provide high accuracy. Using link guard algorithm which is a character based detecting many attacks by using APWG (anti phishing working group). Link guard is used for detecting the phishing attacks and also malicious links in web pages through hyperlinks. Link guard is implemented for windows XP, it can detect up to 96% of Unknown phishing attacks our future work includes further extending the Link Guard algorithm, so that it can handle CSS (cross site scripting) attacks.

REFERENCES

[1] S. Lee and J. Kim," Warning Bird: 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] Alexander Neumann, Johannes Barnickel, "Security and Privacy Implications of URL Shortening Services" Ulrike Meyer IT Security Group RWTH Aachen University, 2010.

[3] 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.

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

[5] Tyler Moore, Richard Clayton, and Henry Stern. 2009. Temporal correlations between spam and phishing websites. In *Proceedings of the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, Spyware, worms, and more* (LEET'09). USENIX Association, Berkeley, CA, USA.

[6] Maher Aburrous, M. A. Hossain, Keshav Dahal, Fadi Thabtah, "Predicting Phishing Websites Using Classification Mining Techniques with Experimental Case Studies, "Information Technology: New Generations, International Conference on IT, 2010; 176-181

[7] Report a Phishing website, http://www.phishtank.com

[8] Ollman, G.(2004) The phishing Guie-Understanding and Preventing , White paper , Next Generation Security software Ltd.

[9] Neil Chou, Robert Ledesma, Yuka Teraguchi, D anBoneh, and John C.Mitchell. Client-side defense against web-based identity theft.Proc. NDSS 2004,2004.