International Journal of Electrical and Computer Engineering (IJECE)
Vol. 6, No. 3, June 2016, pp. 980 ~ 985
ISSN: 2088-8708, DOI: 10.11591/ijece.v6i3.7208

URL ATTACKS: Classification of URLs via Analysis and Learning

M. Rajesh[1], R. Abhilash[2], R. Praveen Kumar[3]
[1]Department of Computer Engineering, M.A.M School of Engineering, India
[2]Device Associate, Amazon Development Center, Chennai, India
[3]Cognizant Technology Solutions, Chennai, India

Article Info

Article history:

Received Dec 19, 2014
Revised Feb 1, 2016
Accepted Feb 16, 2016

Keyword:

HTML
Link Analysis
Social networks
Tiny URL
URL attacks

ABSTRACT

Social Networks such as Twitter, Facebook play a remarkable growth in recent years. The ratio of tweets or messages in the form of URLs increases day by day. As the number of URL increases, the probability of fabrication also gets increased using their HTML content as well as by the usage of tiny URLs. It is important to classify the URLs by means of some modern techniques. Conditional redirection method is used here by which the URLs get classified and also the target page that the user needs is achieved. Learning methods also introduced to differentiate the URLs and there by the fabrication is not possible. Also the classifiers will efficiently detect the suspicious URLs using link analysis algorithm.

Corresponding Author:

M. Rajesh,
Departement of Computer Science,
M.A.M School of Engineering,
Tamil Nadu, India.
Email: rajesh.manoharan89@gmail.com

1. INTRODUCTION

Social networking plays an important role in information sharing service for transferring of messages in the form of tweets or any other modes.When the social users need to share a URL with their close once then they formally use some of the shortening services.

The proliferation of social networking [1] lead to increase in spam activity. The spammers send unsolicited messages for various purposes. Hash tags and shortened URLs [2] like t.co are frequently abused by the spammers. Hash tags are used to denote the topic or latest trend and they are abused by the spammers. The ability to disguise URL destination has made twitter or other social networks as an attractive target for the spammers.

In the first study focusing on spam detection [3], we collect a number of users account. The users are considered as spammers by use of special methods and algorithms and to determine the false positive rate. Here we collect a specific number of users account such as in small environment like colleges or small scale industries to detect their spamming .This will act as the stand alone application for finding spam URLs.

2. RESEARCH METHOD

1. Conditional redirection scheme to ignore the suspicious URLs and there by fabrication is not possible anymore.
2. New features like learning concepts, classification and link analysis to differentiate the suspicious and unsuspicious URLs.
3. Data sets were taken that consist of URLs of suspicious and unsuspicious sites and they are classified by supervised learning methods.

The ultimate goal is to develop a conditional redirection to protect the suspicious URLs. The current visitor can be a normal browser .The normal browser will not know that the URL is being redirected to suspicious site and there by the user gets redirected to malicious page. Here the content of the suspicious URL is not retrieved, since they do not reveal their secrets to the normal browsers .So an analysis algorithm is needed for classification.

2.1. System details

The proposed system consists of following components: data collection, extraction, learning and classification [4] (Figure 1).

2.1.1. Data Collection

In this phase, the URL messages are collected from the public and made for URL redirections. The tweets always follow streaming APIs and look up for IP addresses. It simply blocks up the IP addresses if seems to be malicious and they are skipped off. It is known that the crawlers cannot reach the malicious URLs [5] when conditional redirection is used.

Table 1. Training data set

Phases	Label	Users
Training	Spam	104
	Non Spam	1483
Testing	Spam	104
	Non Spam	1548

2.1.2. Data extraction

This phase involves grouping of domains and extracting future vectors. The phase also monitors the message queue. If several URLs share the same IP address then they replace the sites to the one which is in the data set found to be benign.

2.1.3. Learning

Offline mode for supervised algorithm is used here to classify both URLs and also classification is made via rank basis (link analysis). For labeling, account status is used and so that URLs form suspended accounts are considered as suspicious where as from native accounts are considered as benign.

2.1.4. Classification

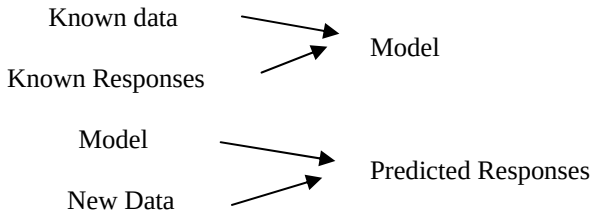
Input vectors [6] are used to classify the suspicious and unsuspicious URLs. LIBLINEAR methods were used earlier to implement this classifier. The classifier algorithms such as Ada Boost ,Naïve Bayes, Support Vector Classification(SVC) are compared and selected an link analysis based algorithm-*power iteration* that will classify the URLs effectively so that the false positive rate get decreased to a greater extent.

Figure 1. System components

The system components are described in Figure 1. In the data collection phase, tweets or mails with URL are collected for redirections that may be suspicious or unsuspicious. In extraction phase the domains that are identical are collected to classify them. Machine learning (supervised learning) is done in the next phase and classification is done at the termination level by link analysis algorithm.

2.2. Steps in machine learning with given data sets

Supervised learning (machine learning) which will take a known set of inputs and known responses , and build a model that generates reasonable predictions for the response to new data.



This method is based on prediction. Suppose if we take a real time example that the number of people will have heart attack within a year .This can be known by taking a trained data samples that consist of age, height, weight, blood pressure etc. So this will combine all the existing data into a model that can predict a person will have a heart attack within a year. Supervised learning splits into two broad categories:

Classification for responses that consist of two values, such as 'true' or 'false'. Classification algorithms apply to nominal data sets.Regression for responses that are considered as a real number, such as miles per gallon for a particular car. It is advised to create a regression model first, because they are often more computationally efficient.

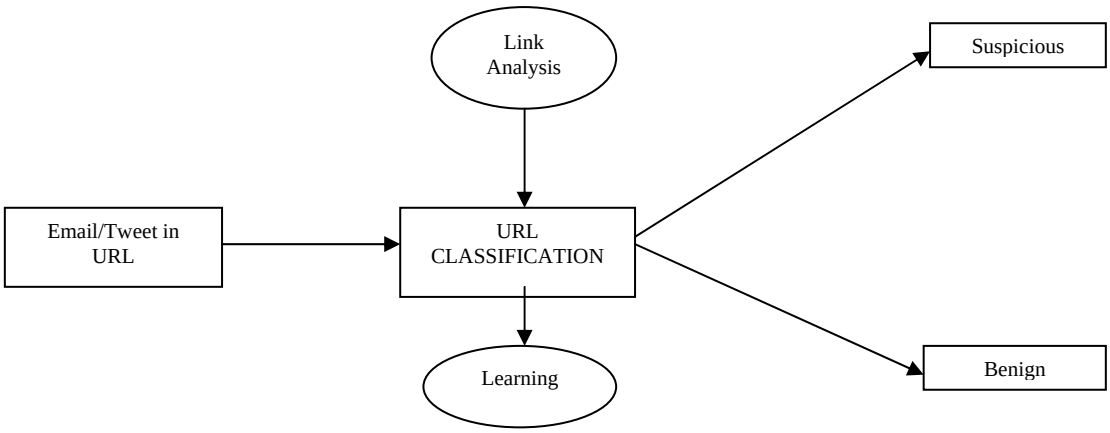


Figure2. URL Redirection scheme

The data set will consist of a number of user accounts and from which spam accounts were detected. The data sets have been separated into two: training and testing. The features are further classified as Phishing dataset and legitimate dataset. Taking 1000 phishing and 1000 legitimate url's into account, the percentage of legitimate URL's is clearly increased by using power iteration method.

Table 2. Legitimate and Phishing data in given data sets

Types	Legitimate	Phishing
IP Address	0%	0.04%
Hexadecimal Character	0%	0.01%
Suspicious symbol	0%	0.01%
Age of domain	35%	75%
Page rank feature	1.2%	88%

2.3. Algorithm- Power Iteration method:

Power iteration is based on Eigen value of a given matrix. The algorithm is mainly based on Eigen values and Eigen vectors which is also known as Von- Mises iteration. This method can be used when the sparse matrix is very large. It can compute only one Eigen value and lower convergence. The page rank iteration algorithm is given below by which the ranking equation is produced.

Algorithm: Power iteration

Initial Phase:
Generate data sets that comprise of URLs with suspicious and unsuspicious links

Classification Phase:
Initially consider page count as 1.
Increment the count as each user visit the page.
Attempt to compare with the datasets obtained
If successful, consider as normal
Else, consider as spam.

Power-Iterate(G)
P ←e/n
R ←1
Repeat
Pr ← (1-d)e+ dA^T Pr-1
Until || Pr-Pr-1||< ε
Return Pr

After satisfying all the conditions, the page rank equation is produced:

$$P = (1-d)e + dA^T P$$

Where, P is the principal eigen vector and R is the initial count.
e is the column vector and **d** is the damping factor.

3. RESULTS AND ANALYSIS

As discussed earlier we use power iteration method because it shows highest AUC and lowest FP (False Positive). AUC is an area under ROC curve. The area under the ROC curve (AUC) is a measure of how well a parameter can distinguish between two diagnostic groups (diseased/normal).In our case we use phishing and legitimate URL's. We compared various classifiers like L1 Regularized and L2 Regularized and also SVC (Simple vector classification) and comparison table is obtained using the power iteration method. Here we took 10000 sample tweets and found 156896 tweets are benign and 156,896 were malicious.

Table 3. Comparison with different classifiers

CLASSIFIER	AUC	ACCURACY%	FP%	FN%
L2R-LR	0.9000	91.11	1.56	6.54
L2-loss SVC	0.8995	90.79	1.49	6.54
Link Analysis	**0.9028**	**91.96**	**1.13**	**7.01**
SVC	0.8984	91.32	1.33	6.86

From the above table we can come to the conclusion that LINK ANALYSIS method will increase the accuracy level and thereby reducing the false positive rate.

3.1. Performance Analysis

Considering the performance aspect of any proposed and implemented algorithm is one of the main criteria to be considered during the research. In our research, performance analysis has been carried out for the implemented algorithm using the open source performance testing tool JMeter. Pages for which the algorithm has been implemented are fetched as an input to the JMeter. The following Table 4 and Figure 3 have been obtained as the result of performance testing.

Table 4. Performance Analysis

Label	Number Of Samples (Count)	Average Response Time (Ms)	Error%
Home Page	500	5446	0
Classification Page	500	4256	0
Total	1000	4851	0

Table 4 has been obtained as a result of performance testing from JMeter. Label indicates the pages for which the algorithm has been implemented. Number of Sample indicates, the number of virtually created users accessed the created page. Average Response time indicates the number of samples executed in particular instance. Error % indicates the amount of error occurred during the testing process.

Figure 3 shows the graphical representation of performance analysis, which has been obtained from the JMeter. In the Figure 3, X axis indicates the page which has been tested and Y axis indicates the Response time in Milliseconds. Bar value indicates the average value of the response time (5446 ms for Home Page and 4256 ms for Classification page).

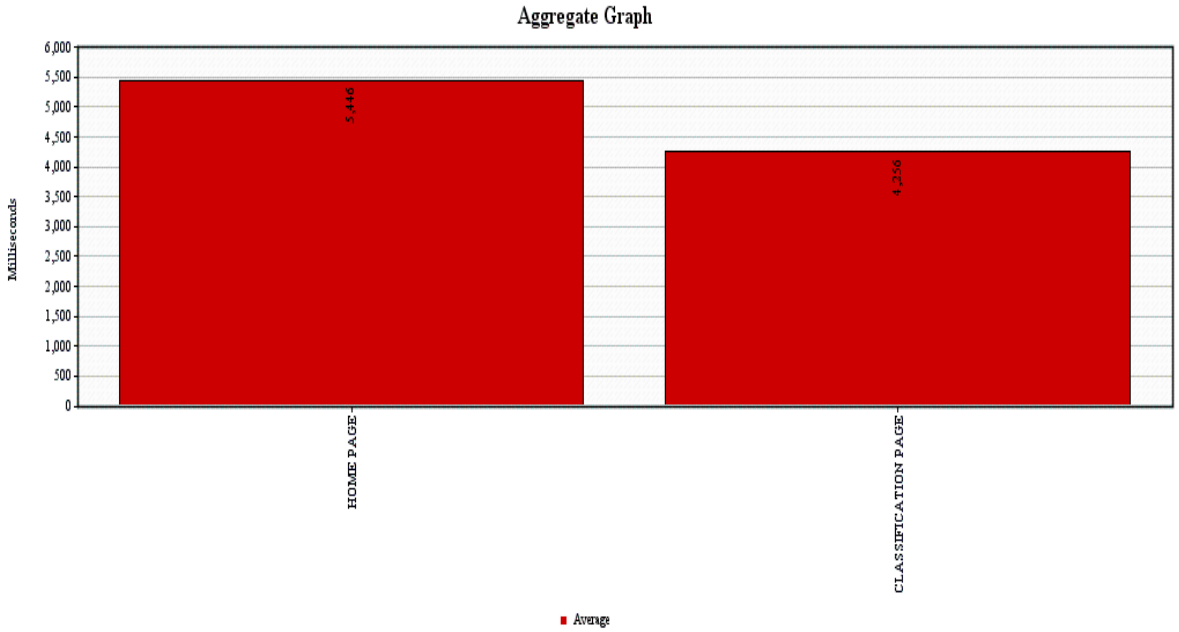


Figure 3. Graphical Representation of Performance Analysis

3.2. Inferences

From the Performance analysis made on the implemented algorithm, the following inferences were made.

1. The error % of performance analysis is 0 (Zero), this indicates that the application of the algorithm is functionally good.
2. The average response time of the implemented algorithm is 4851 ms for 1000 samples. From this we can infer that, for a single sample the response time is 4 seconds (less than 5 seconds). This indicates that the performance of the implemented algorithm holds good.
3. From the above two inference, It can be concluded that the implemented algorithm is functionally and non-functionally good.

3.3. Discussions

The main goal our research is to propose and implement an algorithm which is Simple, Scalable and Highly efficient (Rate of Detection). Apart from these criteria, considering the performance of the implemented algorithm is very important aspect of the research work. Performance analysis of the implemented algorithm has been discussed in the section 6 of this paper.

Here the implemented algorithm has been compared with the earlier research work made on in this area using the above narrated criteria. From our analysis the following Table 5 has been narrated which highlights the impact of the implemented algorithm is far better than the earlier research work which has been carried out. In Table 5 the value ‘Yes’, indicates that the criteria has been satisfied fully. ‘No’ indicates that the criteria have not been considered or not satisfied. Partial indicates that the criteria have been partially satisfied.

Table 5. Comparison of Implemented Algorithm Vs Earlier Research works

S.No	Research Work	Simple	Scalable	Efficiency	Performance
1	Dhanalkshmi Renganayakulu [1]	Partial	Partial	No	Yes
2	Jelena Isacenkova and Oliver Thonnard [2]	Yes	Partial	Partial	No
3	Kelin.F and Strohmaier.M [3]	Partial	Partial	Partial	Partial
4	Lee.S and Kim.J [4]	Partial	Partial	No	Yes
5	Nazpar Yazdanfar, Alex Thomo [5]	Yes	Partial	Partial	Yes
6	Stringhini.G, Kruegel.C and Vigna.C [6]	Yes	Yes	Partial	No
7	Song.J, Lee.S and Kim J [7]	Yes	Yes	Yes	No
8	Zachary Miller [8]	Yes	No	Partial	Partial
9	Our Work	Yes	Yes	Yes	Yes

4. CONCLUSION

Table 5 has been derived by our analysis, from which we can infer that the implemented algorithm is simple, efficient, high performance, and scalable comparing to the earlier research works. A conventional method seems to be ineffective in their conditional redirection that separates normal users from being redirected to suspicious page. Unlike the convectional systems, classification via analysis is robust. The system accuracy and performance seems to be high this method by referring the statistical Table 3. In the future, process has to be extended to handle dynamic redirections.

REFERENCES

[1] D. Renganayakulu, "Detecting Malicious URLs in E-mail," *AASRI Procedia Elsevier*, vol. 4, pp. 125-131, 2013.
[2] J. Isacenkova and O. Thonnard, "Inside Scam Jungle: a closer look at 419 scam email operations," *URASIP journal of information security*, 2014.
[3] F. Kelin and M. Strohmaier, "Short links under Attack: Geographical Analysis of Spam in URL Shortener Network," *Proc.23 ACM Conf.Hypertext and Social media (HT)*, 2012.
[4] S. Lee and J. Kim, "Warning Bird: A Near Real-Time Detection System for Suspicious URLs in Twitter Stream," *IEEE transactions on secure computing*, vol/issue: 10(3), 2013.
[5] N. Yazdanfar and A. Thomo, "Collaborative-Filtering for Recommending URLs to Twitter Users," *Procedia*, vol/issue: 19(3), pp. 412-419, 2013.
[6] G. Stringhini, *et al.*, "Detecting Spammers on Social Networks," *Proc.26th Ann. Computer Security Applications Conf. (ACSAC)*, 2010.
[7] J. Song, *et al.*, "Spam Filtering in Twitter Using Sender-Receiver Relationship," *Proc.14th International Symp. Recent Advances in Intrusion detection (RAID)*, 2011.
[8] Z. Miller, "Twitter spammer detection using data stream clustering," *Information sciences Elsevier*, vol. 260, pp. 64-73, 2013.