ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 2, February 2019

Classification of URL into Malicious or Benign using Machine Learning Approach

Deebanchakkarawarthi G[1], Parthan AS[2], Sachin Lal[3], Surya A[4]

Assistant Professor, Department of Computer Engineering, JCT College of Engineering and Technology, Coimbatore, Tamilnadu, India[1]

B.E. Student, Department of Computer Engineering, JCT College of Engineering and Technology, Coimbatore, Tamilnadu, India[2,3,4]

Abstract: Now days everything is digitalized and it's difficult to secure our data. Web security is a major issue because everything is connected through internet. The common way of launching an internet attack is by using malicious URLs. Hackers or Intruders leaks billions of confidential data every year by using malicious websites The traditional way of detecting these kinds of malicious URLs or websites is by the use of a web database called Blacklists. There are many URL shortening methods and Domain Generation Algorithms are available which make it difficult to detect the newly generated malicious URLs. To increase the efficiency and avoid database dependency we proposed the Machine learning approach to detect the malicious URLs. In machine learning approach there are so many algorithms available for the classification and feature extraction from that we select the best method that is Random forest which will gave us more accurate result than the past ones. So our proposed system will increase the efficiency and gave accurate prediction whether the URL entered is malicious or benign by using a well-defined dataset. It is also possible to implement the machine learning application in a proxy server or any network traffic controller system.

Keywords: Malicious URL; Random forest; Machine learning; Blacklist; Detection; prediction; URL; benign

I. INTRODUCTION

Web-based malware attacks become one in every of the foremost serious threats that require to be addressed desperately. Many approaches that have attracted attention as promising ways in which of defence work such as malware embrace using numerous blacklists. However, these standard approaches usually fail to observe new attacks because of the flexibility of malicious websites. Thus, it's tough to take care of up-to-date blacklists with information concerning new malicious web sites. Malicious address detection plays an important role for several cyber-security applications, and clearly machine learning approaches square measure a promising direction. In combination with privacy constraints on knowledge sets of actual user traffic, its troublesome for researchers and merchandise developers to gauge anti-malware solutions against massive scale knowledge sets of realistic net traffic. Machine learning technique [1] area unit employed in order to classify the online traffics into malicious and benign URLs. The appearance of recent communication technologies has had tremendous impact with in the growth and promotion of companies spamming across several applications as well as online-banking, e-commerce, and social networking. In fact, in today's age it's nearly obligatory to possess a web presence to run an eminent venture. As a result, the importance of the world wide net has unendingly been increasing. Sadly the technological advertisements return in addition to new subtle techniques to attack and scam user. Such attacks embrace malicious websites that sell counterfeit merchandise, monetary fraud by tricking users into revealing sensitive information that eventually cause thieving of cash or identity, or perhaps putting in malware within the users system. There square measure a large type of techniques to implement such attacks, like specific hacking tries, Derive-by exploits, Denial of service [2], Distributed denial of service [1] and lots of others. Concentrating the variability of attacks, doubtless new attack varieties, and also the unnumbered contexts within which such attacks will seems, it's arduous to style-strong systems to discover cyber security breaches. The limitations of traditional security management technologies are becoming more and more serious given this exponential growth of new security threats, rapid changes of new IT technologies, and significant shortage of security professionals. Most of these attacking techniques are realized through spreading compromised URLs [1].

II. PROPOSED ALGORITHM

The machine learning approach of detecting malicious URLs contain mainly three phases they are feature extraction, classification and prediction. Machine Learning approaches, use a set of URLs as training data, and based on the

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 2, February 2019

statistical properties, learn a prediction function to classify a URL as malicious or benign. This gives them the ability to generalize to new URLs unlike blacklisting methods. The primary requirement for training a machine learning model is the presence of training data. In the context of malicious URL detection, this would correspond to a set of large number of URLs. Machine learning can broadly be classified into supervised, unsupervised, and semi-supervised, which correspond to having the labels for the training data, not having the labels, and having labels for limited fraction of training data. Labels correspond to the knowledge that a URL is malicious or benign [1].

A. *Description of the Proposed Algorithm (Random forest):*

Aim of the proposed algorithm is to increase the accuracy and efficiency of the system. Random forest is supervised learning algorithm which is mainly use for classification of URLs. It creates a forest and makes it somehow random. The "forest" it builds, is an ensemble of Decision Trees, most of the time trained with the "bagging" method. The general idea of the bagging method is that a combination of learning models increases the overall result.To say it in simple words: Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

The random forests algorithm (for classification) is as follows:

1. Draw n-tree bootstrap samples from the original data.
2. For each of the bootstrap samples, grow an unpruned classification tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample m-try of the predictors and choose the best split from among those variables. (Bagging can be thought of as the special case of random forests obtained when m-try = p, the number of predictors.)
3. Predict new data by aggregating the predictions of the n-tree trees (i.e., majority votes for classification, average for regression).

An estimate of the error rate can be obtained, based on the training data, by the following:

1. At each bootstrap iteration, predict the data not in the bootstrap sample (what Breiman calls "out-of-bag", or OOB, data)using the tree grown with the bootstrap sample.
2. Aggregate the OOB predictions. (On the average, each data point would be out-of-bag around 36% of the times, so aggregate these predictions.) Calculate the error rate, and call it the OOB estimate of error rate.

III. PSEUDO CODE

Precondition: A training set *S:* = (*x*1; *y*1); : : : ; (*xn*; *yn*), features *F*, and numberOftrees in forest *B*.
Step 1: **function** RandomForest(*S*,*F*)
Step 2:*H* ←∅
Step3:**for** *i* 2 1; : : : ; *B* **do**
Step 4:*S* (*i*) ←A bootstrap sample from *S*
Step 5:*hi* ←RandomizedTreeLearn(*S* (*i*); *F*)
Step 6:*H* ←*H* U {*hi*}
Step 7:**end for**
Step 8:**return** *H*
Step 9:**end function**
Step 10:**function** RandomizedTreeLearn(*S* ,*F*)
Step 11: At each node:
Step 12: *f* very small subset of *F*
Step 13: Split on best feature in *f*
Step 14: **return** The learned tree

IV. METHODOLOGY AND ARCHITECTURE

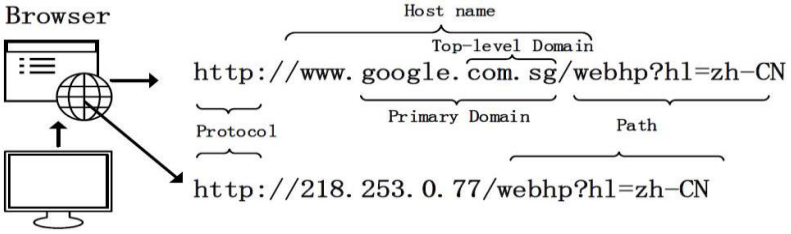

Fig.1: Example URL

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 8, Issue 2, February 2019

The proposed system consists of a well-defined training model which contains dataset, feature extraction, classifiers and predictors, which will give an accurate predicted result and update (online update) the result in the new dataset.
Let us consider an example URL data,

Supervised machine learning requires examples of the form f1,f2,...fn: C in which n features are distilled from a problem instance, and provided to the learner along with the class label C. In our system, tokens derived from the URL serve as binary features: for each token ti present in a training URL, fi(u) = 1 if ti is present in the URL u and 0 otherwise, where i ranges from 1 to |T|, the number of unique tokens seen during training. Thus, it is critical to select meaningful tokens to represent the URL. A simple and effective approach is to segment a given URL into its components as given by the URI protocol (e.g., scheme :// host / path / document . extension ? query # fragment). We can further break these components at non-alphanumeric characters and at URI-escaped entities (e.g., '%20') to create smaller tokens. Such baseline segmentation is straightforward to implement and typically results in 4-7 tokens for subsequent classifier induction [18].

Dataset collection: Collection of datasets used to analyse and train the machine learning algorithm [1]. Along with dataset various features of a URL is studied and collected. The dataset will be of .csv file type.It contains various columns that includes Name of URL, Good or bad, Creation date, Verified time,Hosted from, Server name

Feature extraction: As stated earlier, the success of a machine learning model critically depends on the quality of the training data, which hinges on the quality of feature representation. Feature extraction consists of Lexical features, Host based features, Content based features, HTML features and JavaScript features [1].

Classifiers: A URL classification block is proposed in this architecture so as to provide a mechanism for successfully and clearly classify the URLs. The various parameters considered for the proposed classification includes- type of URL, features, datasets, learning approaches, models and attack types. The classification of URLs on basis of 'type' parameter involves two types - benign and malicious URLs. The malicious URLs are further categorized on basis of attack types of malicious URLs. The variety of attack types are: Spamming, phishing, malware, attack page, Gumblar, sql injection, Fastflux [1] and denial of service etc.

Predictor: A predictor will receive input from the classifiers and analyze the features of the URL, now the predictor got an idea about the nature of the URL and now they are ready to predict whether the given URL is malicious or benign. The symbol' 0' indicate that the URL is benign and the symbol '1' indicate that the URL is malicious.

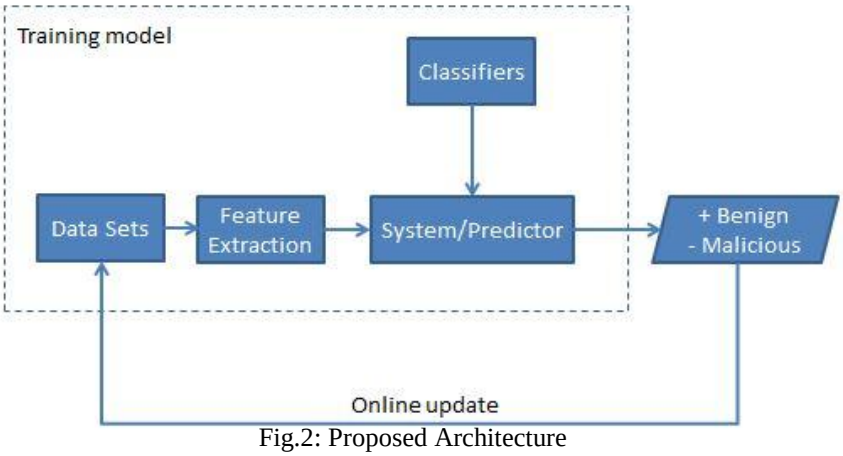

Fig.2: Proposed Architecture

V. CONCLUSION AND FUTURE WORK

The output of the proposed work shows improved accuracy and efficiency than the existing one. By using random forest algorithm the efficiency and accuracy of the classification and feature extraction of URL is improved and the proposed system shows an accuracy of 97%, which is more than the existing one.The main advantage of the proposed system is that the blacklist dependency is removed so that we don't keep a separate database for the prediction. The proposed system is easy and efficient to manage and it will automatically predict the malicious URL based on the training data. The proposed system Provide more security against obfuscating and bypassing techniques.New URL shortening techniques and DGA are stronger and stronger now a day so we need a system which will help us to

continuously monitor and secure our browsing. In feature work we can implement the proposed model in a proxy server or a network traffic controller for the cyber security application.

REFERENCES

[1] Doyen Sahoo, Chenghao Liu, and Steven C.H. Hoi, "Malicious URL Detection using Machine Learning:A Survey",. IEEE, 2017
[2] Johann Vierthaler, Roman Kruszelnicki, Julian Schütte, "WebEye Automated Collection of Malicious HTTP Traffic", November 30, 2017
[3] A. Astorino, A. Chiarello, M. Gaudioso, and A. Piccolo, "Malicious url detection via spherical classification," Neural Computing and Applications, pp. 1–7, 2016.
[4] M. Kuyama, Y. Kakizaki, and R. Sasaki, "Method for detecting a malicious domain by using whois and dns features," in The Third International Conference on Digital Security and Forensics (DigitalSec2016), 2016, p. 74.
[5] J. Hong, "The state of phishing attacks," Communications of the ACM, vol. 55, no. 1, pp. 74–81, 2012.
[6] B. Liang, J. Huang, F. Liu, D. Wang, D. Dong, and Z. Liang, "Malicious web pages detection based on abnormal visibility recognition," in E-Business and Information System Security, 2009. EBISS'09.
[7] S. Garera, N. Provos, M. Chew, and A. D. Rubin, "A framework for detection and measurement of phishing attacks," in Proceedings of the 2007 ACM workshop on Recurring malcode. ACM, 2007, pp. 1–8.
[8] S. Chhabra, A. Aggarwal, F. Benevenuto, and P. Kumaraguru, "Phi.sh/$ ocial: the phishing landscape through short urls," in Proceedingsof the 8th Annual Collaboration, Electronic messaging, Anti-Abuse andSpam Conference. ACM, 2011, pp. 92 101.
[9] Y. Alshboul, R. Nepali, and Y. Wang, "Detecting malicious short urlson twitter," 2015.
[10] D. K. McGrath and M. Gupta, "Behind phishing: An examination ofphisher modi operandi." LEET, vol. 8, p. 4, 2008.
[11] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists:learning to detect malicious web sites from suspicious urls," inProceedings of the 15th ACM SIGKDD international conference onKnowledge discovery and data mining. ACM, 2009, pp. 1245–1254.
[12] ——, "Learning to detect malicious urls," ACM Transactions onIntelligent Systems and Technology (TIST), vol. 2, no. 3, p. 30, 2011.
[13] S. Purkait, "Phishing counter measures and their effectiveness–literature review," Information Management & Computer Security,vol. 20, no. 5, pp. 382–420, 2012.
[14] A. Blum, B. Wardman, T. Solorio, and G. Warner. Lexical feature based phishingurl detection using online learning. In the 3rd Workshop on Artificial Intelligenceand Security, pages 54–60, 2010.
[15] D. Canali, M. Cova, G. Vigna, and C. Krugel. Prophiler: A fast filter for thelarge-scale detection of malicious web pages. In Proceedings of the International World Wide Web Conference, pages 197–206, March 2011.
[16] BirhanuEshete, Adolfo Villafiorita, and KomministWeldemariam. Binspect:Holistic analysis and detection of malicious web pages. Security and Privacy inCommunication Networks, pages 149–166, 2013.
[17] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-JenLin. LIBLINEAR: A library for large linear classification. Journal of MachineLearning Research, 9:1871–1874, 2008.
[18] Min-Yen Kan Hoang Oanh Nguyen Thi Department of Computer Science, School of Computing. "Fast webpage classification using URL features"