Malicious Web Page Detection and Content Analysis

Vishal Jagtap, Vaibhav Shinde, Pratik Sapre, Kartik Karande, Ketaki Bhoyar
Dr. D. Y. Patil Institute of Engineering,
Management and Research, Pune, India

ABSTRACT

The detection of malicious web pages is a complex engineering problem due to the dynamic nature of the information contained on the internet.Since the data stored on web-servers updates on a continuous basis, It is very hard to find and classify which links are malicious and which are not in real-time. Hence, brute-force checks (system-scans) and voting-based approaches (blacklisting) fail to capture the exhaustive list of malicious content on the internet. A machine learning based model is proposed which is able to classify the malicious links and content on the user's device. It can later be applied in the forms: a web application, Android, iOS mobile applications and also browser extension which is able to give you a report of that link which you want to open on a device. The whole system performs a complete scan on that link and generates a report.

Keywords

Malicious Web Page

1. INTRODUCTION

The objective is to provide a safe browsing environment for search where you will be safe from cyber fraud and phishing. In these attacks the users are made to click on attractive advertisements or are redirected to fraud websites where they unintentionally give the attackers access to their devices and their personal information. Phishing is considered as one of the major cyber menaces that pose a significant security threat in the present day world and is responsible for the loss of millions of dollars across the globe. Avoiding such attacks is the main scope of the system. Consider a web page as malicious if it contains data that can potentially exploit a client-side system by launching security attacks such as automated intrusion, phishing, spamming, click-jacking, drive-by-downloads, cross-site scripting, JavaScript Atul Choudhary et al.[4] obfuscation, misadvertising. Malicious web pages pose a serious threat to the security of distributed data-centres, information management systems, client-side systems, and any computing node in a network.The efforts to make robust detection systems are being actively pursued, However, the complexity and dynamic nature of the problem prevents the development of a persistent (long-term) solution.

The objective is to develop a system which detects a malicious webpage in real-time using machine learning techniques. The system acts as an interceptor between the server and web-browser, and filters every HTTP request. A predictive machine learning model is used to classify the webpage based on the extracted feature values. The prediction includes a binary classification as a malicious or benign tag.

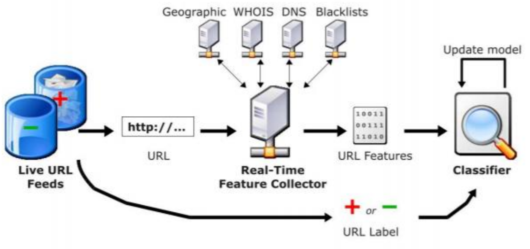


Fig 1: A machine - learning based web page classification system Reference for above figure Justin Ma et al.[16].paper

2. PROPOSED METHODOLOGY

An extensive dataset is required which includes a proportionate distribution of all classes of malicious web pages. Training an ensemble model on complex feature sets including lexical, source and host-based features is done. The Logistic Regression Classification model seems the most suitable which achieves an accuracy of 94%.

An ensemble of multiple detection systems are used namely, Lexical Analysis, Spam Detection, Page Ranking, and then scanning the links within the page give even better results.

2.1 Training dataset

There are many public datasets available on the internet, which contains the URL and the corresponding label. The URL string will be processed to remove any inconsistency generated due to URL encoding. The following public datasets are used:
1. Phishing Web Pages - Phishtank: https://www.phishtank.com/developer_info.php/

2. Malware Hosting Web Pages - DNS-BH: http://www.malwaredomains.com/

3. Non-Malicious Web Pages - A. Kaggle: https://www.kaggle.com/antonyj453/urldataset/

2.2 Feature vector generation and Data Pre-processing

The feature vector is generated by extracting web page features (static) in real-time through open-sourced third-party Application Programming Interfaces (APIs). This information is stored in a backend database and can be exported for data pre-processing. The values are cleaned according to the model requirement, and then further optimized for training of the machine learning model. The optimization includes feature selection, dimensionality reduction, and feature transformation.

3. FEATURE SET

3.1 Lexical Features

These features describe the statistical properties of URL String -

1. Length of the URL, Number of dots in URL, Number of hyphens, Directory Length etc.

2 Length of File, Total dots in file, Total delimiters in file etc.

3. Presence of IP address in Hostname, Length of Query string in URL, create_age(months), expiry_age(months), update_age(days), zip code.

4. Number of occurrences of symbols : [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]

5. Number of occurrences : [‘/’, ‘.’, ‘?’, ‘=’, ‘-’,’_’, ‘;’, ‘:', ‘(‘, ‘)’, ‘@’, ‘&’, ‘%’]

6. Presence of Security Sensitive words.

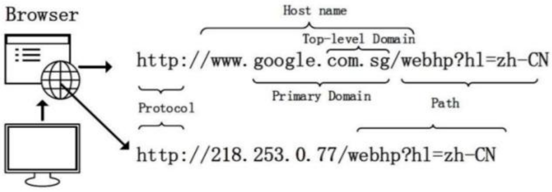

Browser
Host name
Top-level Domain
http://www.google.com.sg/webhp?hl=zh-CN
Protocol
Primary Domain
Path
http://218.253.0.77/webhp?hl=zh-CN

Fig.2 : Tokenization of URL String based on hierarchical levels of addressing

Motivated from a study by McGrath and Gupta et al. [8].

3.2 DNS Features

The DNS features are

1. Resolved IP count

2. Name server count

3. Name server IP count In

4. Malicious ASN ratio of resolved IPs

5. Malicious ASN ratio of name server IPs

4. MACHINE LEARNING MODELS

On training the data on the following machine learning models:

Table 1: Comparison of Classifiers based on Accuracy

	Decision Algorithm	**Accuracy**
1.	Random Forest Classifier	91.83
2.	Support Vector Machines (RBF)	83.111
3.	Logistic Regression	94.045

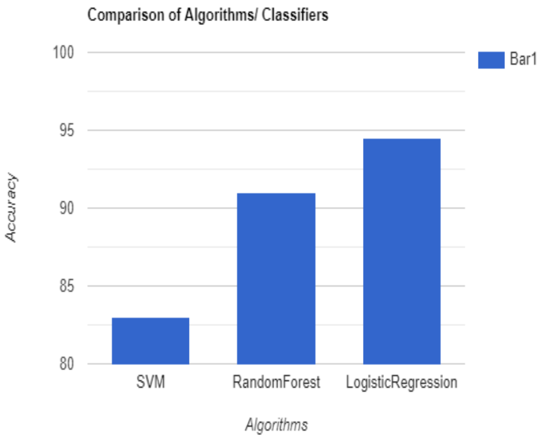
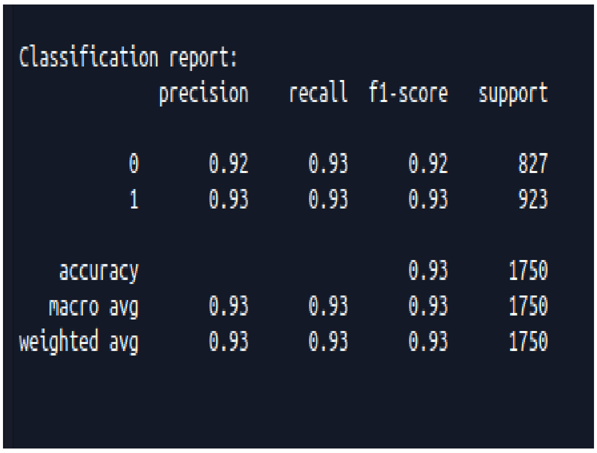


Fig. 3: Bar graph comparing accuracy of all algorithms



```
Classification report:
              precision    recall  f1-score   support

           0       0.92      0.93      0.92       827
           1       0.93      0.93      0.93       923

    accuracy                           0.93      1750
   macro avg       0.93      0.93      0.93      1750
weighted avg       0.93      0.93      0.93      1750
```

Fig. 4: Classification report of model with best accuracy.

Other than Accuracy on test data, metrics like precision, recall and f1-score were used to evaluate model performance. In use cases like these a higher recall is required i.e. Less number of False Negatives (low FNR) to avoid Malicious websites getting labelled as Safe as much as possible.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Logistic Regression algorithm has a recall of 93% (Fig.4) on the test data i.e. 93% websites are correctly labelled as ‘Malicious’ out of total malicious websites in test data. The f1-score is the harmonic mean of precision and recall. F1-score is high only if both precision and recall are high. The final model has a f1-score of 92%.

4.1 Logistic Regression

Logistic regression is a discriminative probabilistic model mainly used in which the output is binary. Logistic regression performs better than Naïve Bayes model when training size is close to infinity. Doyen Sahoo et al.[1]

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

Binary outcomes loss function aka binary categorical cross entropy (BCE)

4.2 Model Selection

Based on the distribution of features and metrics obtained, the model with optimal performance is: Logistic Regression Classifier (Lexical, DNS, Host features).

5. SYSTEM REQUIREMENTS AND SPECIFICATIONS

1. Hardware System

Google Collaboratory: https://colab.research.google.com Processor Specifications 1xTesla K80, 2496 CUDA Cores, Compute 3.7 GPU 1xSingle Core Hyper Threaded Xeon Processors@2.3Ghz CPU Memory 12 GB, GDDR5 VRAM 16 GB, 45 MB Cache, Turbo Boost Disk Storage 359 GB 2 TB

2. Software System

Python 3.X Programming Language, Open-Source Tools Operating System : Linux Tools : Anaconda, VS Code, Chrome, SublimeText Frameworks: Flask (Web Development): time, numpy, pandas, os, urllib, re, ipaddress, collections, math, bs4, socket, requests, tldextract, dns, sklearn, string, pathlib, json, warnings, matplotlib.

6. APPLICATIONS

The developed model is highly compact, efficient and is distributed as a REST API. Hence, it is platform independent and can be integrated in any Linux / Windows / OSX / Android / Web application with minimal reconfiguration. The model is production-ready and can be used in servers as well as client systems for detection of malicious webpages. Potential applications include a cross-platform extension, firewall based scanning system, and can be distributed as an open-source software for further API integrations.

7. LITERATURE SURVEY

The method consists of three stages training data collection, supervised learning with the training data, and malicious URL detection. The lexical and host-based features of Uniform Resource Locators contain a wealth of information for detecting malicious Web sites. Thus, to protect end users from visiting these sites, the system attempts to identify suspicious URLs by analyzing their lexical and host-based features.

The classification model of Ma et al. [13][14] can detect spam and phishing URLs. They described a method of URL classification using statistical methods on lexical and host based properties of malicious URLs. Their method detects both spam and phishing but cannot distinguish these two types of attacks.

8. CONCLUSION

A Machine-learning based real-time malicious web page detection application was developed. This application provides real-time classification of web pages without any significant external processing latency. The system runs with minimal overhead on memory resource utilization, processor utilization, and network utilization. It is platform independent and can be distributed as REST API. This API can be used into systems as web and mobile applications as well as into browser extensions.

9. REFERENCES

[1] Doyen Sahoo, Chenghao Liu, Steven C.H. Hoi, “Malicious URL Detection using Machine Learning: A Survey,”, School of Information Systems, Singapore Management University, Vol. 1, No. 1, Article . Publication date: August 2019.

[2] Immadisetti Naga Venkata Durga Naveen, Manamohana K, Rohit Verma, “Detection of Malicious URLs using Machine Learning Techniques,” International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8 Issue-4S2 March, 2019.

[3] Abubakr Sirageldin, Baharum B. Baharudin, and Low Tang Jung, “Malicious Web Page Detection: A Machine Learning Approach, ” Computer & Information Science Department, University Technology Petronas Bandar Seri Iskandar, 31750 Tronoh , 2014.

[4] Atul Choudhary, Manikrao Dhore “CIDT: Detection of Malicious Code Injection Attacks on Web Application, ” in International Journal of Computer Applications · August 2012

[5] Justin Ma, Lawrence K. Saul, Stefan Savage, Geoffrey M. Voelker, “Learning to Detect Malicious URLs,” ACM Trans.Intell. Syst. Technol. 2, 3, Article 30 (April 2011), 24 pages. DOI=10.1145/1961189.1961202 http://doi.acm.org/10.1145/1961189.1961202

[6] Y. Shi, G. Chen, and J. Li, “Malicious domain name detection based on extreme machine learning,” in Neural Processing Letters, vol.48, pp.1347-1357, 2018. DOI:10.1007/s11063- 017-9666-7

[7] Y. Hang, J. Hong, L. Cranor, “CANTINA: a content-based approach to detecting phishing web sites,” Proc. 16th International Conference on World Wide Web, pp.639-648, January, 2007. DOI:10.1145/1242572.1242659

[8] MCGRATH, D. K., AND GUPTA, M. Behind phishing: An examination of phisher modi operandi. In LEET: Proceedings of the USENIX Workshop on Large-Scale Exploits and Emergent Threats (2008)

[9] HOU, Y.-T., CHANG, Y., CHEN, T., LAIH, C.-S., AND CHEN, C.-M. Malicious web content detection by machine learning. Expert Systems with Applications (2010), 55–60.

[10] Parveen Rani, Er. Sukhpreet Singh: An Offline SEO (Search Engine Optimization) Based Algorithm to Calculate Web Page Rank According to Different Parameters, INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY Vol 9, No 1, July 15, 2013

[11] ZHANG, Y., HONG, J., AND CRANOR, L. CANTINA: A content-based approach to detecting phishing web sites. In WWW: Proceedings of the international conference on World Wide Web (2007).

[12] RAMACHANDRAN,A.,AND FEAMSTER, N. Understanding the network-level behavior of spammers. In SIGCOMM (2006).

[13] MA, J., SAUL, L. K., SAVAGE, S., AND VOELKER, G. M. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In KDD: Proceedings

of the international conference on Knowledge Discovery and Data mining (2009).

[14] Justin Ma, Lawrence K. Saul ,Stefan Geoffrey, M. Voelker, Identifying Suspicious URLs: An Application of Large-Scale Online Learning, Department of Computer Science & Engineering, UC San Diego (2009).

[15] MA, J., SAUL, L. K., SAVAGE, S., AND VOELKER, G. M. Identifying suspicious URLs: an application of large-scale online learning. In ICML: Proceedings of the International Conference on Machine Learning (2009).

[16] Justin Ma JTMA@CS.UCSD.EDU Lawrence K. Saul SAUL@CS.UCSD.EDU Stefan Savage SAVAGE@CS.UCSD.EDU Geoffrey M. Voelker “Identifying Suspicious URLs: An Application of Large-Scale Online Learning” Department of Computer Science & Engineering, UC San Diego — 9500 Gilman Drive, La Jolla, CA 92093-0404 year(2009)