www.coast.osustech.edu.ng
COAST
JOURNAL OF THE FACULTY OF SCIENCE
Volume 2, Issue 2. November, 2020 ISSN: 2714-3716
Coast, J. Fac. Sci. 2 (2): 482 - 489

# DEVELOPMENT OF A MALICIOUS WEBSITE DETECTION SYSTEM USING WEB BROWSER EXTENSION

**[1]*Ikuomola, A. J., [1]Ogunbameru, A. and [2]Nwanze, M.N**

[1]Olusegun Agagu University of Science and Technology, Okitipupa, Ondo State, Nigeria
[2]Federal College of Education (Technical), Asaba, Delta State
*Corresponding Author's email: deroniknng@yahoo.com

**Abstract**

Malicious websites host unsolicited content and lure unsuspecting users to become victims of scams. It is imperative to detect and act on such threats in a timely manner. Traditionally, this detection was done mostly through the usage of blacklists features. However, blacklists lack the ability to detect newly generated malicious Uniform Resource Locators (URL). To improve the generality of malicious URL detectors, blacklist and machine learning techniques using feature extraction were explored in this design. Blacklist Feature takes lesser processing time and also relies on external data (list containing malicious websites) in detecting malicious websites while feature extraction method takes more time and does not rely on external data in detecting new malicious websites using the web browser extension. The system was implemented using JavaScript and MySQL. Certain malicious and benign websites were used to test run the system. The system consists of three major layers: Users, Web browser extension and Database. The web browser extension layer makes use of two techniques (Blacklist feature and feature extraction) to detect malicious website efficiently. The performance of the malicious websites detection system using both blacklist and feature extraction shows that it provides robust, secured and easier way to detect malicious website in real time.

**Keywords:** Blacklist, Feature Extraction, Malicious, Website

## Introduction

As the usage of internet continue to grow exponentially, the growth of malicious websites are also on the increase. These malicious websites were created for unlawful purposes, for example, spam-advertising, malware propagating and financial fraud through phishing and malvertising. As an example, on the aspect of malvertising activities, its growth had been recorded at 132% in 2016 over 2015(Arghire, 2017). It was also reported by Symantec (2016) that 76% of servers scanned were found to have vulnerabilities which could mean that it harbors malicious intention or codes. This would include luring unsuspecting users to a spoof sites and obtain key information by impersonating valid sites. Malicious websites host unsolicited content (spam, phishing, drive-by exploits, etc.) and lure unsuspecting users to become victims of scams (monetary loss, theft of private information, and malware installation) and cause losses of billions of

dollars every year.

These malicious websites not only steal or damage the information from users, but also allow hackers to regulate the computers. The malicious website becomes a platform helping diverse Internet crimes. As a result of this, detecting the malicious sites to stay away from the damage is of a significant priority. What's more, these websites often appear to be genuine websites. Sometimes it will request an unsuspecting user to install software that the machine seems to need. Another one is, a video website might request for a codec installation which in turns compromise the machine itself.

The major issue that occurs when internet users are making use of the web browser for obtaining information, handling official business, online shopping and lot more is that attacker targeted internet user by forging and writing misleading Uniform Resource Locators (URLs) and web-pages with various tricky skills which are difficult to identify even for the professionals, so as to steal money, manipulate devices and monitor or view user personal information. Currently, most of the malicious websites detecting technology is based on features of URLs or Web page elements.

According to the Internet Security report released by Symantec (2014), malicious Web pages are now the primary vector for malicious activities over the Internet, with this view, there is need to develop a system that can detect a malicious website in order to eradicate or minimize malicious internet activities. There have been many works on malicious websites detections efforts. The pioneering effort begins with the blacklist approach (Eshete, 2013). A blacklist is a list containing IP address information, website name or URLs of known malicious websites. The backlist provides credible validation of whether the site is malicious or not. Although, the credibility of the blacklist approach is high but the speed at which it is updated is quite slow. Furthermore, it needs an extended time period from finding, verifying and upgrading a fresh malicious website to the blacklist. Any new malicious website cannot be detected by blacklist-based system because the malicious websites most times have IP addresses and domain names different from the blacklisted malicious website. Beyond the reactive nature of malware detection, there are also significant works done in the proactive approaches of malware websites detection which includes blacklist, honey clients, machine learning techniques and webpage content. In other to improve on malicious URL detectors, blacklist and machine learning techniques, feature extraction were explored in this design. A web browser extension that detects malicious website was developed.

**Literature Review**

Rohit and Krishnaveni (2013) proposed a system that detects deep malicious web sites where the vast and hidden information is available. The system makes use of the deep web crawler for the detection of deep malicious websites as the normal crawler does not crawl into hidden websites. The deep web crawler uses the crawling infrastructure of search engines to retrieve URLs that are much more likely to be malicious than a random page on the web. In order words, this crawler increases the input to the URL stream. The deep web crawler also focuses on how to prepare appropriate queries to retrieve hidden pages. As vast amount of web pages lies in the deep or invisible web, these pages are typically only accessible by submitting queries to a database and regular crawlers are unable to access the deep web pages as they do not have links to the deep websites. Deep Web crawler is relatively slow because of the bulk downloading of web pages.

Zhou et al. (2013) proposed a system that detect malicious website and protect search engine. The system make use of static approach based on classification algorithm and extracted features for each webpage consisting of URL, HTTP response header, content, domain information, search engine, PageRank, WOT score and Alexa information. The features are firstly introduced to detect all kinds of malicious websites. The selected features greatly improve the classification performance and are robust to combat potential malicious websites. The system helps to protect search engine from malicious attack and also, the browser plug-in that was developed was a good safeguard for search engine poisoning. The system limitations are that: it cannot detect malicious website with hidden feature especially when the malware code is written with server scripting language (php, sql, asp) and also, it can't access all the malicious websites from public blacklist due to low memory.

Koo et al. (2013) proposed a system that made use of honey pot for malicious website detection. The system uses static content analysis performance to accelerate the detection, followed by actual browsing on web pages for in-depth probing using the client-side honey pot system. The client-side honey pot system uses 79 dynamic analysis processes where list and conditional constraints are incorporated to achieve optimal performance. The behavior recording module was implemented under a virtual machine (VM) environment, which sequentially simulates website browsing according to the order of the URLs requiring detection. The browsing history was recorded fully and the malicious packets in the network traffic were collected using Wireshark.

The system is a highly interactive web-based application. The system detection is time-consuming and any modifications to the system status may suggest an abnormality. Also, if the browser is controlled, this might cause abnormality if the system does not have adequate disk space.

Amrutkar et al. (2016) developed a security system for removing malicious web pages on mobile using KAYO. KAYO can classify and detect number of malicious mobile web pages in the wild that cannot be detected by existing techniques such as Google Safe Browsing and Virus Total. KAYO can detect new mobile websites threats by analyzing pages of the websites and extracting JavaScript, html and lexical features of the web page. The System uses 44 features set in detecting malicious website which make it very accurate. Its limitation is that the system can creeps into users' personal data which is unlawful.

Sahoo et al. (2017) proposed a system that made use of machine learning techniques to detect malicious URL. The Machine Learning approach uses a set of URLs as training data, and based the process on statistical properties. It also learns 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. The system saves already detected malicious website for future use. Although, the number of URLs available for training can be in the order of millions (or even billions) and as a result of this, the training time for traditional models may be too high to be practical. Also, the system does not detect strange features on the website and this might produce error in the result.

Ahmed et al. (2018) proposed a system that made use of blacklist feature for the detection

of malicious websites. The blacklist feature approach uses a list containing IP address information, website name or URLs of known malicious websites on the internet, whenever an internet user wants to make use of the web browser to surf the internet, in other to get information, the system checks if the websites that the user wants to visit is malicious or benign by checking the blacklist. If the website is not on the blacklist, the system gives a go ahead that the website is free from malware hence if the website is on the list a pop up is displayed on the screen that the website is malicious. Although, the system can work on all platforms but it update is quite slow as update is based on real time and also, it needs an extended time period from finding, verifying and upgrading.

Naga et al. (2019) proposed a system that makes use of machine-learning techniques to detect malicious URLs. A filter was designed for detecting malicious web pages and the filter consists of well-trained data sets that studies the behavior of the HTML, JavaScript, CSS and jQuery library and notifies the user of abnormal tags, however if the tags and behavior of the HTML, JavaScript, CSS and jQuery are okay, it is assumed that the URLs is free from malicious attack. C4.5 algorithm was used in performing evaluation on the data sets. The system was based on abnormal visibility, hence it can't detect an invisible abnormal tags, words, links, and so on and also it required an expert because if the machine learning is not trained with good features there would be an error in the result.

**Design Methodology**

**Design Considerations**

In this design, the following are incorporated into the malicious website detection using web browser extension:

(I) User Interface: A browser user interface (or BUI) is a method of interacting with an application, typically hosted on a remote device, via controls presented within a web browser. This is an alternative to providing controls via a separate application with a dedicated graphical user interface (GUI) or command-line interface (CLI). BUIs have become common for devices that have their own embedded microprocessor and network interface, such as a printer or network router. A web browser provides a much richer interface environment than a command-line interface.

(ii) Blacklists : A blacklist or block list is a basic access control mechanism that make use of table consisting list of malicious website, those websites on the list are denied access whenever an internet user is trying to enter any of the websites. Blacklists can be applied at various points in a security architecture, such as a host, web proxy, DNS servers, email server, firewall, directory servers or application authentication gateways.

(iii) Feature Extractor: Feature extractor transforms original data to a data set with a reduced number of variables, which contains the most discriminatory information. This will reduce the data dimensionality, remove redundant or irrelevant information, and transform it to a form more appropriate for subsequent classification.

(iv) Pop-up Window: A pop-up window is a type of window that opens without the user selecting "New Window" from a program's File menu. Pop-up windows are often generated by websites that include pop-up advertisements. These ads are produced with JavaScript code that is inserted into the HTML of a Web page. They typically appear when a user

visits a page or closes a window. Some pop-up ads show up in front of the main window, while others show up behind the main browser window.

**System Architecture**

The architecture of a malicious website detection using web browser extension is shown in Figure 1. The system comprises of three layers namely Users layer, Web browser extension layer and Database Layer. The web browser extension receives request from the user layer and the system check if the request (the website that the user wants to visit) matches with the information in the blacklist database. If the website matches with any of the website in the blacklist database, the web user receives response in form of alert/signal that the website is malicious and if the website does not match, the web browser extension proceed to the second approach to extract the HTML JavaScript and lexical content of the website to check for malicious code.

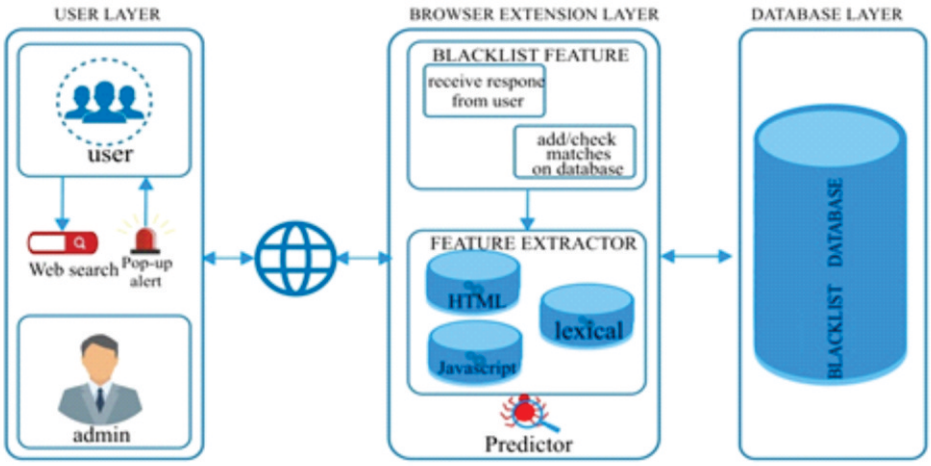



Figure 1: Three Layer architecture of a malicious website detection using web browser extension

**Component of the Architecture Design**

**(a) Users Layer:** The user Layer is an interface that allows the internet users to access the internet. It is made up of the web user and the web administrator. This layer allows web user to search for information by entering a web address and getting information on the predicted web browser extension. It also allows the web admin to update the blacklist database with new malicious website.

**(b) Browser Extension Layer:** The Extension Layer is the core of the architecture; the layer performs the prediction and determines whether the website the user is trying to explore is malicious or benign. The layer consists of two part which are:

**(i) The Blacklist Feature:** The blacklist feature is a trivial technique that identifies malicious URLs by interacting with the Blacklist database to check if the URLs the user wants to visit matches with any URLs on the blacklist database, the user would be alerted if the URLs matches and if there are no match, the second technique is considered

**(ii) The Feature Extractor:** The feature extractor is used for extracting the

appropriate feature of web pages which includes the web address, HTML and JavaScript contents of each webpage of the website. The blacklist feature is not enough to justify if the website is malicious or benign, however the feature extractor performs lexical, HTML and JavaScript analysis on the web pages in other to predict the state of the website (i.e. Lexical Analysis, HTML Feature Extractor and JavaScript Feature Extractor).

**(c) Database Layer:** The database layer has two components namely: the admin table which consists of the login information and the blacklist table which consist of the list of malicious URLs, it serves as the back end to the blacklist feature as it requests and receives response from the blacklist database.

**Implementation and Results**

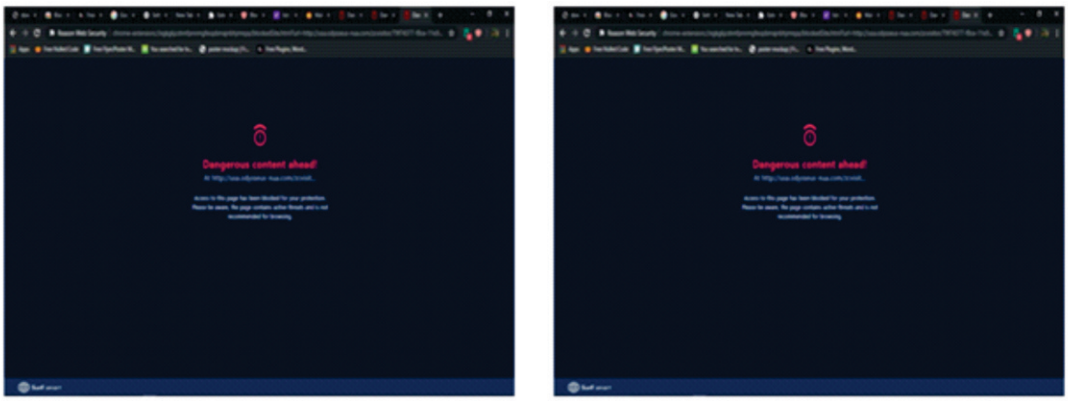
The malicious website detection using web browser extension was implemented using JavaScript programming language and MySQL was used to create a database which stores administrator Login information and the blacklist information which interacts with the blacklist feature.

**Preliminary Testing of the Malicious Website Detection System**

Figures 2 to 5 shows the users interface were malicious websites detection system was used on a Google chrome browser. Figure 2 shows the web browser extension interface where user can search for a web page.


Figure 2: Search page to request/load a website

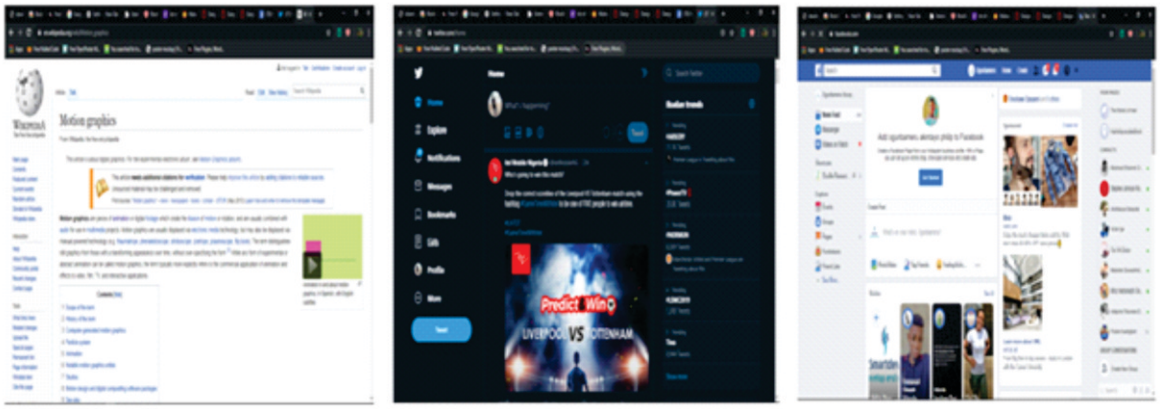
Figure 3(a-b) shows the web browser extension interface when malicious websites were detected.




(a) (b)

Figure 3: Malicious Websites

Figure 4(a-c) shows the web browser extension interface when a malicious free websites and benign websites were detected while figure 5 shows a search page listing detected malicious websites.



(a) (b) (c)

Figure 4: Benign web pages

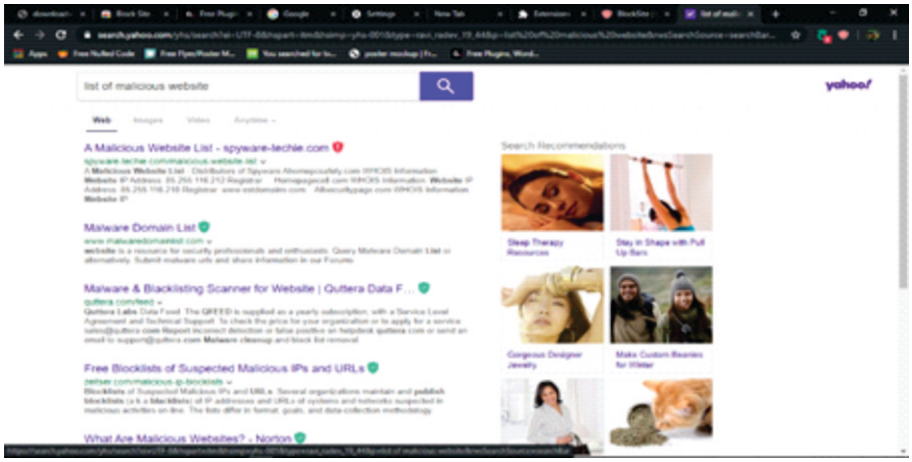



Figure 5: Search page listing detected malicious websites

**Table 1: Properties of Different Feature Representations for Malicious Website Detection**

| Features | | Category | Parameters | | | | |
|---|---|---|---|---|---|---|---|
| | | | Collection Difficulty | Risk | External Dependency | Collection Time | Processing Time |
| Blacklist | | Blacklist | Moderate | Low | Yes | Moderate | Low |
| Feature Extraction | Lexical | Traditionally | Easy | Low | No | Depends | Low |
| | | Advanced | Easy | Low | No | Low | High |
| | Content | HTML | Easy | High | No | Moderate | Moderate |
| | | JavaScript | Easy | High | No | Depends | High |

By analyzing the results in Table 1, Blacklist Feature takes lesser processing time and also rely on external data (list containing malicious websites) in detecting malicious websites while Feature extraction method takes more time and does not rely on external data in detecting new malicious websites using the web browser extension.

**Conclusion**

In this work, a malicious website detection system using web browser extension was designed. The system comprises of three layers namely Database Layer, Users layer and Web browser extension layer. With this system, malicious websites are easily detected by internet users. The most prominent feature of the system is that it can easily be installed on desktop browser and it does not take excess space when the system is running on users' browser. Many previous works make use of single technique which can be the blacklist feature or the feature extraction. In most cases, problem arise with the blacklist feature whereby the users are not sure if the website they are trying to visit is malicious or not as blacklist features needs regular update. This web browser extension is reliable as it makes use of two techniques to detect malicious website: blacklist and feature extraction. It also detects malicious website based on real time and the system is not time consuming unlike other detection system.

**References**

Aghire I. (2017). Malvertising Jumped 132% in 2016 over 2015. RiskIQ Research Press Release.

Ahmed A. A, Sunaidi H. and Quosthoni P. (2018). Malicious Website Detection. Proceeding of the 3rd ACM on International Workshop on Security and Privacy Analytics.7, 1- 4.

Amrutkar C., Kim Y. S. and Traynor P. (2016). Detecting Mobile Malicious Web pages in Real Time. IEEE Transactions on Mobile Computing.

Eshete B (2013). Effective analysis, characterization and detection of malicious web pages. In Proceedings of the 22nd International Conference on World Wide Web ACM, 355-360.

Symantec (2014). Internet Security Threat Report, Available online at https://its.ny.gov/sites/default/files/documents/...

Symantec (2016). Internet Security Threat Report, Available online at http://www.symantec.com/security_response/publications/threatreport.jsp

Koo T.M, Hung-Chang C. and Ya-Ting H. (2013). Malicious Website Detection Based on Honeypot Systems. 2nd International Conference on Advances in Computer Science and Engineering (CSE), 76-82.

Naga I, Venkata D.N, Manamohana K. and Rohit V. (2019). Detection of Malicious URLs using Machine Learning Techniques. *Inter J Innovative Tech Exploring Engrn.* 8(4:2): 368-393.

Rohit P.S. and Krishnaveni R. (2013). Deep Malicious Website Detection. *India Inter J Comp Math Sci* 2(4): 517-522.

Sahoo D, Liu C. and Steven C, H. (2017). Malicious URL Detection using Machine Learning: A Survey. *India Inter J Comp Math Sci* 7: 58-68.

Zhou H, Sun J. and Chen H. (2013). Malicious Website Detection and Search Engine. *J Adv Comp Network*, 1: 260-264.