Malicious Web Page Identification using HTML Content-based Technique

Chiaw Lin Heng*, Kang Leng Chiew†, San Nah Sze‡, Wei King Tiong§
Faculty Computer and Information Technology,
University Malaysia Sarawak
94300, Kota Samarahan, Sarawak, Malaysia
*chiawlin.heng@gmail.com
†klchiew@fit.unimas.my
‡snsze@fit.unimas.my
§wktiong@fit.unimas.my

Abstract—The increase of the usage of website leads to the increase of the number of web page viewed. This causes the widespread of Internet criminal activities and website becomes the focus of the malicious attack. In this paper, a technique which is based on static analysis is proposed to classify malicious and benign web page. Support Vector Machines (SVM) is used as the classifier and is trained to classify the dataset. This technique obtained 82% of accuracy by extracting the significant feature of web pages to ensure the efficiency and effectiveness of the detection model. The features extracted by the proposed method are from HTML element, JavaScript and property of URL structure.

Index Terms—malicious detection; static analysis; machine learning; SVM;

I. INTRODUCTION

N owadays, the usage of website is getting wider due to the growing popularity of Internet. This phenomenon leads to cybercrime and computer security issues. One of the critical issues is the dramatically increase in the number of malicious web page. Malicious web content has turn into one of the instrument which is exploited by attacker to distribute malicious code. Attackers embed the malicious scripts into a web page and infect the victim’s device when the malicious script is executed. The malicious contents will be able to compromise the security of a system, obtain sensitive information without user’s permission and bring negative impacts to users and their devices. Therefore, malicious web page detection has become one of the major focuses among the Internet user community. Currently, two main approaches that used for malicious web page detection are dynamic analysis and static analysis.

Dynamic analysis performs analysis during the execution of a web page and checks for malicious evidence after the web page is executed in a control and monitor environment [1]. Dynamic analysis such as Honeyclient is effective in detecting malicious code and it usually produces a high accuracy in the malicious detection. However, dynamic analysis requires costly analysis to perform the detection as the process is relatively complicated. The associated script in a web page needs to be executed and examined whether the malicious activity is existed [1]. In order to avoid the downside of dynamic analysis, static analysis has been applied to malicious detection. Static analysis inspects and analyses the static property of a web page and determine the maliciousness of a web page. HTML contents are the fundamental elements used for the static aspect inspection and analysis of web page.

In this paper, a technique that classifies malicious and benign web page by analyzing web page property is proposed. The proposed method will analyse the property of a web page which includes the HTML elements, JavaScript and URL structure of the web page. Hence, the costly analysis process of dynamic is not necessary as the associated codes in the web page are not required to be executed and the analysis process will be faster as compared to dynamic analysis approach [1]. In addition, Support Vector Machines (SVM) is used as the classifier to classify and label a web page as malicious or benign.

In the next section, related prior works on malicious detection is discussed. We will discuss the general framework in Section III. Section IV will present the experimental results. Conclusion and some recommendations for future works are discussed in Section V.

II. RELATED WORKS

According to an investigation performed in July 2008, 90 websites are selected in CERNET1 to investigate the rate of malicious website. The results show that practically 32.2% of the selected websites contains malicious contents [2]. In order to protect user from malicious attacks, many static analysis based approaches have been proposed to detect malicious contents. These approaches provide useful methods and analysis for malicious detection and alert user from visiting malicious website. This section will present some of the common and related approaches.

1China Education and Research Network (CERNET) is a national education and research computer network in China. CERNET mainly provides services for universities, institutes and other non-profit organizations in China and it plays an important role in the research of advanced network technologies and applications.
A. Blacklisting

Blacklisting is a widely utilized detection approach. Blacklist consists of the listed URLs, domain names and IP addresses of malicious web pages. The listed elements are stored in a database and it consists of a control access mechanism. When a web page is about to be loaded, the web browser will query the blacklist and check the URL if it is listed in the blacklist [3]. This approach can prevent user from accessing malicious content and protect user’s device from malicious infection. According to [3], blacklist is commonly used in web browser such as Internet Explorer and Mozilla Firefox to protect user from malicious attack. Hence, blacklisting is considered as a simple and straightforward method in detecting malicious web page. However, the effectiveness of blacklist is limited by the rapid increasing new malicious web pages. If the URL of new malicious web page has not been included in the blacklist, the detection will fail. In order to maintain the effectiveness and usefulness of blacklist, regular maintenance is required. This required great effort and is costly [4].

B. URL Analysis

URL analysis approach analyses the structure of URL and determines whether the URL of the particular web page is malicious or benign. URL analysis performs interpretation by looking at the pattern, and domain name of URL to verify whether the given URL is a malicious web page. URL analysis can overcome the weakness of the blacklist because it does not depend on the pre-saved list of malicious URLs. The URL of a web page can be verified from the information retrieved from the pattern and structure of the URL. Nevertheless, certain malicious infections such as drive-by-download attack which injects malicious content into an innocuous web page are difficult to be detected by URL analysis [1]. Besides, less well-known website might be categorised as malicious website due to its low reputation [5]. URL analysis approach is also ineffective to detect malicious infections such as Cross-side Scripting (XSS). The malicious scripts used in XSS are typically developed in JavaScript and embedded in the HTML structure [6]. Attacker could exploit the vulnerability of a website and inject malicious scripts into a benign website which has legitimate URL. Sometime, the attacker can evade URL analysis by carefully crafting a URL that look similar to the legitimate URL.

C. Page Analysis

Page analysis approach inspects the properties of a web page based on its HTML source code. HTML source code with unusual code will be treated as suspicious and analysis will be performed to determine the maliciousness. This approach is more effective in detecting malicious content injection attack like XSS. Compared with blacklisting and URL analysis, more information can be retrieved by this technique as it is based on HTML source. In addition, this approach is also able to detect the HTML tags used for malicious attack.

Prophiler [1] is one of the methods which belong to page analysis approach. It is a fast and reliable. Prophiler utilises the HTML element, structure of URL and JavaScript code to differentiate malicious and benign web page. All together, Prophiler has considered 77 features from different classes of features. The classes include HTML, JavaScript, URL and host-based features. Prophiler statically interprets all the classes of features by applying several detection models. Based on the detection model, Prophiler will discard web page which is labelled as a benign web page. Web pages that are suspected as malicious web page will be further analysed by Wepawet2, a tool which consists of comprehensive detection [1]. By considering the features from different classes (HTML, JavaScript, URL and host-based), more information can be retrieved and inspected. However, the features in HTML class are mainly involved counting the number of suspicious HTML tag used but the properties and attribute of the tag are overlooked. Besides, this filter is partially relied on the costly dynamic analysis (i.e., Wepawet).

Another page analysis approach is proposed by Liang et al. [2]. The proposed method is based on the concept of abnormal visibility. Their approach analyses the display setting code that embedded in a web page to identify malicious content [2]. The abnormal visibility concept proposed by [2] is also effective in analysing JavaScript which is not included in most of the web page analysis system. The types of abnormal visibility fingerprints used in [2] includes:

1) Abnormal width or height of iframe tag
2) Abnormal display mode of iframe: none display style
3) Abnormal iframe generated by script

The maintenance of the abnormal visibility fingerprint is simple and straightforward because the number of abnormal visibility modes is relatively fewer [2]. Besides, the abnormal visibility fingerprints are more stable than the malicious codes signatures [2]. However, the abnormal visibility fingerprints considered are more emphasised on iframe related attributes. URL structure and other HTML tags which are exploitable by attacker have been neglected. Therefore, the information considered by this proposed method might be insufficient for an effective malicious detection.

III. METHODOLOGY

Motivated by Prophiler [1], the proposed method is divided into several stages, which are data pre-processing, feature extraction and classification. Figure 1 shows the framework of the proposed method, which illustrates the flow of the malicious detection process.

Data pre-processing is the first stage to process the collected web page. It helps to filter the collected data (web pages) and pre-process it to improve the quality of data. First,
the web page will be pre-processed to eliminate unnecessary and irrelevant information. Only HTML tags and JavaScript of the webpage will be retained and other unnecessary elements such as comments and main text contents in the web page source will be removed. Throughout this stage, data will be transformed into more structural and useful information for the following processes. The output of the data pre-processing will be the input for the next stage, feature extraction.

The main objective of this proposed method is to develop a detection model which is able to classify the malicious and benign web pages by analysing the HTML element, JavaScript and property of URL structure. The proposed method is enhanced from the filter developed in [1]. We make use of their most significance features and swap some less sensitive features with additional new features to improve the detection performance. These features can be categorised as follows:

A. HTML Features

**Number of element with small area.** This feature counts the number of HTML elements with small dimension. They include HTML tags like div, iframe and object which dimension is less than a determined threshold.

**Number of out-of-place element.** This feature counts the number of elements that is not placed in their natural position in the HTML document. Position of the tag elements such as iframe, frame, form, script, object and embed are checked according to the HTML Document Type Definition (DTD) specifications.

**Number of character in the web page.** This feature counts the number of characters in HTML document (i.e., the content, HTML tags and JavaScript).

**Presence of double document.** This feature checks the presence of double document such as html, head, title, and body elements. The presence of double document is considered abnormal in a web page.

**Presence of meta refresh tag.** This feature check the presence of a meta refresh tag that causes the refresh of a web page. This is very common in page redirecting to the exploit servers.

**Presence of abnormal display mode of iframe.** This feature checks the display mode of an iframe element which is set to “none”. Some iframe tags that embeds malicious code will become invisible if the display mode is set to none. This feature is selected from [2] as it is significant to the detection model.

B. JavaScript

**Keywords-to-word ratio.** This feature computes the ratio between the number of reserved words and other strings used in the JavaScript codes. The number of keywords (e.g., var, for and while) is limited while there are usually a large number of other operations like instantiations, arithmetical operations and function calls in malicious web page. Unlike malicious web page, JavaScript codes in benign web page usually has higher occurrence of keywords.

**Number of long strings.** This feature counts the number of long strings used in the script. A string is considered long if its length is beyond a determined threshold.

**Number of suspicious strings.** This feature records the number of suspicious string that is used as the variable or function name, for instance, “evil”, “shell”, “crypt” and etc.

C. URL

**Presence of sub-domain in URL.** This feature checks the presence of sub-domain in an URL.

**Presence of IP address in URL.** Malicious websites are usually hosted through Internet Protocol (IP) address instead of associating with a registered domain name. This feature records the presence of the IP address in an URL.

**Length of URL.** Some attacker will try to shorten the length of malicious website URL to evade URL analysis [7]. This feature counts the length of a given URL and compares to a determined threshold.

**Dot and special character.** This feature counts the number of a dot (.), at-sign (@) and double dash (//) in an URL and uses certain threshold to decide the URL is belongs to a malicious or benign web page [7]. This feature is selected from [7] as it has significant contribution to the detection model.

With the extracted features, the next step is to perform classification. Machine-learning technique is applied in this stage with Support Vector Machine (SVM) as the classifier.

![Fig. 1. Framework of the proposed method.](image-url)
Basically, classification process involves two modes: training and testing mode. The classification uses two non-overlapping datasets. Each dataset contains the extracted feature sets of both malicious and benign web pages. During the training, the classifier will be fed with the feature sets together with their labels (i.e., malicious and benign). The classifier will be trained to recognise the pattern of the feature sets corresponding to the label. The completion of training will produce a detection model. This detection model will be used to classify a given web page as malicious or benign. We use the second dataset to benchmark the overall performance in the testing mode.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, the effectiveness and performance of the proposed method is evaluated and discussed. Besides, we also present the performance comparison between the proposed method and the Prophiler [1].

We have collected a total of 1,000 web pages for the experiments. 500 malicious web pages are collected from [11], [12], [13], [14], [15], [16] and 500 benign web pages from [17] and [18]. Both datasets are collected from different sources to avoid bias and to ensure the wide coverage of different web pages. From the 1,000 collected web pages, 80% of them are used for training and 20% of them are used for testing. We use the SVM library implemented in [19] with the default setting (i.e., radial basis function is used as the kernel function and the values for parameter \( \gamma \) and \( C \) is set to 1/13 and 1.0, respectively. We use the following metrics to measure the experiment results. We abbreviate true positive, true negative, false positive and false negative as TP, TN, FP and FN, respectively. Table I shows the comparison results.

\[
\begin{align*}
\text{Accuracy rate} & = \frac{(TP + TN)}{(TP + TN + FP + FN)} \\
\text{Detection rate} & = \frac{TP}{(TP + FN)} \\
\text{False alarm rate} & = \frac{FP}{(FP + TN)} \\
\text{Error rate} & = \frac{(FP + FN)}{(TP + TN + FP + FN)}
\end{align*}
\]

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>COMPARISON OF TWO TECHNIQUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method</td>
<td>Prophiler</td>
</tr>
<tr>
<td>Accuracy rate</td>
<td>82.0%</td>
</tr>
<tr>
<td>Detection rate</td>
<td>76.0%</td>
</tr>
<tr>
<td>False alarm rate</td>
<td>12.0%</td>
</tr>
<tr>
<td>Error rate</td>
<td>18.0%</td>
</tr>
<tr>
<td>No. of features extracted</td>
<td>13</td>
</tr>
</tbody>
</table>

From the results presented in Table I, the proposed method achieved better performance compared to Prophiler. The proposed method can classify 82% of the web pages into the correct classes (malicious or benign) while Prophiler can only correctly classify 79.5%.

According to [9], the industry claims that it is necessary to obtain a 70% – 80% of detection rate on malicious codes with heuristic analysis. It seems that, both approaches are efficient comparing to the industry claims, as both techniques achieved 76.0% of detection rate. However, the proposed method is superior in term of computational resources as the proposed method only used 13 features whereas Prophiler used 16 features. The proposed method is able to use less features to accomplish the same detection rate as Prophiler.

The false alarm and error rate of the proposed method is also lower compared to the rates obtained by Prophiler. Hence, it shows that the proposed method has a more reliable detection performance. Therefore, the combination of significant features has increased the overall performance of the proposed detection model.

To identify the common strategy used in the malicious web page, we have carried out some statistical analysis and study on the features extracted by the proposed method. From the analysis, the most commonly used features in malicious web page are exploiting the out-of-place element, followed by double document element and long string in JavaScript. Figure 2 shows the distribution of features extracted from the 500 collected malicious web pages. In addition, as shown in Table II, malicious web page have the tendency of containing low keywords-to-words ratio in the embedded JavaScript and small number of character in their HTML source. We also found that malicious web pages tend not to specify any sub-domain in their URL.

![Fig. 2. Distribution of the features extracted from the 500 collected malicious pages.](image)

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>COMPARISON OF FEATURES IN MALICIOUS AND BENIGN WEB PAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malicious web page</td>
<td>Average of keywords-to-words ratio: 0.03695, Average number of character: 21,465, Absence of sub-domain in URL: 348 web pages</td>
</tr>
<tr>
<td>Benign web page</td>
<td>Average of keywords-to-words ratio: 0.04159, Average number of character: 48,809, Absence of sub-domain in URL: 82 web pages</td>
</tr>
</tbody>
</table>
V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a static analysis technique that extracts the web page properties (i.e., HTML elements, JavaScript and URL structure) to classify web page as malicious or benign. Throughout this research, a detection model is developed based on the extracted feature sets and classification approach. The detection model has achieved promising accuracy. The features extracted have significant contribution in distinguishing malicious and benign web pages. Furthermore, we have identified the features which are commonly exploited in the malicious web pages.

We plan to incorporate host-based feature in future because the reputation of a hosting center is also important. According to [10], malicious website is very likely to be hosted in a less reputable hosting center and registered through a disreputable registrar. Therefore, it will be very valuable to capture accurately this information. Analysing and extracting features from Cascading Style Sheet (CSS) is also a potential future direction.

ACKNOWLEDGMENT

The funding for this project is made possible through the research grant obtained from UNIMAS and the Ministry of Education, Malaysia under the Fundamental Research Grant Scheme 2/2013 [Grant No: FRGS/ICT07(01)/1057/2013(03)].

REFERENCES