IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

A Survey on Current Malicious JavaScript Behavior of infected Web Content in Detection of Malicious Web pages

A Survey on Current Malicious JavaScript Behavior of infected Web Content in Detection of Malicious Web pages

Wan Nurulsafawati Wan Manan, Mohd Nizam Mohmad Kahar and Noorlin Mohd Ali

Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, 26300 Kuantan Pahang Malaysia

E-mail: safawati@ump.edu.my

Abstract. In recent years, the advance growth of cybercrime has become an urgent issue to the security authorities. With the improvement of web technologies enable attackers to launch the web-based attacks and other malicious code easily without having prior expert knowledge. Recently, JavaScript has become the most common attack construction language as it is the primary browser scripting language which allow developer to develop sophisticated client-side interfaces for web application. This lead to the growth of malicious websites and as main platform for distributing malware or malicious script to the user's computer when the user access to these webpages. Initial act and detection on such threats early in a timely manner is vital in order to reduce the damages which have caused billions of dollars lost every year. A number of approaches have been proposed to detect malicious web pages. However, the efficient detection of malicious web pages previously has generated many false alarm by the use of sophisticated obfuscation techniques in benign JavaScript code in web pages. Therefore, in this paper, a thoroughly survey and detailed understanding of malicious JavaScript code features will be provided, which have been collected from the web content. We conduct a thorough analysis and studies on the usage of different JavaScript features and JavaScript detection technique systematically and present the most important features of malicious threats in web pages. Then the analysis will be presented along with different dimensions (features representation, detection techniques analysis, and sample of malicious script).

1. Introduction

The dramatic rise of the Internet technology has given various impact to our daily lives. This technology become the most popular medium of communication and information collection. Web based attack is happening daily where, JavaScript has become the most common language used by attackers. From the G Data Security report, most of the common malware threats are launch by the attacker coming from web sites and not only from the executable file [1].

As reported from the previous years, the most remarkable security breach is, Ransomware attack, which caused billions of dollar lost and identity breach or stealing user's money. Attackers have used many techniques to launch malicious attack through websites. Figure 1 below illustrates some examples of attack classification.

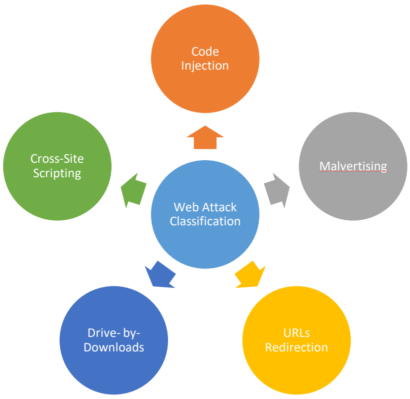


Figure 1: Malicious web attacks classification.

Malicious web crime can be categorized into several types of attacks and drive by downloads are become the most common surreptitious attack on the Internet [19]. While, cross sites scripting (XSS), code injection, URL redirection, malvertising, and many more become the most predominant attack and threats on web pages nowadays.

JavaScript is the most common programming language used in web development. This scripting language is a lightweight and object-based client-side scripting language which add many features and effect to the end user experience. This common web language was used by the web developer to develop interactive content for a web pages [1] and making the web user experience the dynamic and better control of web interaction. Besides, HTML and CSS are the common construction language which used in the web browser in developing the interactivity in web pages.

Detecting and filtering web pages URLs based on a blacklist is a simple and powerful technique. However it is difficult to notice malicious threats on websites and unknown malicious URLs on the web. Besides, it is difficult to keep up to date with newly created malicious URLs for blacklisting [2]. This is because a new URL can be generated easily and a massive amount of web content is generated every day by both legitimate and malicious users.

In this research, we have categorized and suggested a set of JavaScript features from the thoroughly literature studies of other researcher. Features which have been considered in this paper aim at describing a classes of web pages or web content components that are frequently used attackers for conducting attack in website. Besides, these proposed features was collected based on various comparisons from different researchers' point of view thus making the features expected to correctly classify the malicious websites with a high level of precision rate.

This paper is organized as follows. We presented the related works on previous research in detecting malicious websites in section two. In the section 3, we have discussed the characteristic and features representation in identifying malicious website. Last section conclude the proposed idea and future works.

2. Related works

This section surveys related approaches malicious JavaScript detection and classification, related applications of statistical method, machine learning and other techniques.

2.1. Malicious web pages detection

Variety of approaches have been proposed by previous researchers in detecting the malicious JavaScript in web pages. In authors [4] a novel approach was proposed to detect the malicious JavaScript Code. They have developed a system which used a number of JavaScript features and combining the anomaly detection with emulation. This machine learning classifier was used to classify whether the web pages is benign or malicious.

R.Wang et al [5], suggested a hybrid analysis technique to detect malicious web pages. According to the researcher detecting the malicious activity in web pages using static analysis only was become difficult. Therefore, researcher has combined the dynamic analysis in order to achieve high performance result. In their studies, static analysis utilized the classification algorithm in machine learning while dynamic analysis method checked the web pages that are directly from the web browser to determine whether they contain malicious shell codes during execution.

Another work by Canali et al [6] has designed a fast and reliable filter, which also use static web page analysis based on machine learning for detecting malicious web pages. The features being used in the analysis were derived from the HTML, JavaScript code, and corresponding URL. The author also claim that the system was faster than the where processing time about 0.27 s/page

In [7] authors have proposed five features that capture different characteristics of script for detecting malicious JavaScript. They are, execution time, external referenced domains and calls to JavaScript functions. From the experiment results, it depicted that a combination of selected features is able to detect malicious JavaScript code successfully with a precision of 0.979 and a recall of 0.978

Another well-known method which used by [8] in detecting the malicious web is through the blacklist-based method. Blacklisting is a technique where it was compiled through crowdsourcing technique, and comes from various sources of repository or database. A blacklist is a repository which contains a various list of known malicious URLs. This method particularly used by filtering systems against web-based malware infection and also widely used in search engines and browsers toolbars. However there were there are several limitation, the URL list need to be updated regularly as the web environment is changing at rapidly and unable to detect unknown or zero-day delay malicious web URLs.

2.2. Critical Analysis of Malicious Features Selection

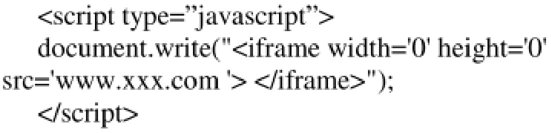
Malicious features is one of the biggest threats in web crime activities. In this paper, different features of malicious has been studied and proposed to help researcher to decide between malicious and benign web pages. In particular, the selection of features is important to prove better performance of the machine learning classifier as well as the accuracy of the result. We focus on content based features which are obtained upon downloading or opening the website.

HTML and JavaScript features are based detail content about the whole web pages. The most common characteristic of malicious JavaScript features are not human-readable, and the character frequency of JavaScript codes being used in the content [9], [10]. While, [11] have used 77 features, among which 45 are new features in their dataset preparation.

This paper [12] implement the vector space representation in extracting semantic features which normally use during classification and clustering. Benign webpages were collected from Alexa (2013), while malicious web pages collected mostly from Phishtank (2013). Furthermore, researcher also used lexical features of URLs, visual features which are based on images and web pages that contain many links to collect information.

We have conducted a thoroughly studies on the other researchers work in order to propose and come out with a set of malicious JavaScript features that are more representative and structured which can be refer by researcher and security practice in future. Besides, this proposed features also could help in detecting malicious web pages as well as any web threats especially zero-day attacks.

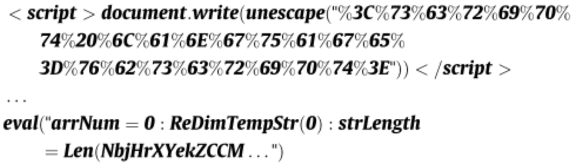
With the dynamic behavior in nature of JavaScript, it executes certain programming behaviors at runtime, using different technique provided by both interpreter and browser. Figure 2 below present an example JavaScript code which attempts to execute the malicious pages where, script elements was stored and pointed to external document at runtime [13].



```
    <script type="javascript">
    document.write("<iframe width='0' height='0'
src='www.xxx.com '> </iframe>");
    </script>
```

Figure 2: Example of malicious JS code (1)

While figure 3 show the example built in JavaScript function of eval( ) and unescape( ) where hacker uses these two function to decode and encode string and write the output on the webpage [14]. This JavaScript function is the most used by attackers in inserting malicious code and activity in the web pages.



```
< script > document.write(unescape("%3C%73%63%72%69%70%
    74%20%6C%61%6E%67%75%61%67%65%
    3D%76%62%73%63%72%69%70%74%3E")) < /script >
...
eval("arrNum = 0 : ReDimTempStr(0) : strLength
     = Len(NbjHrXYekZCCM ...")
```

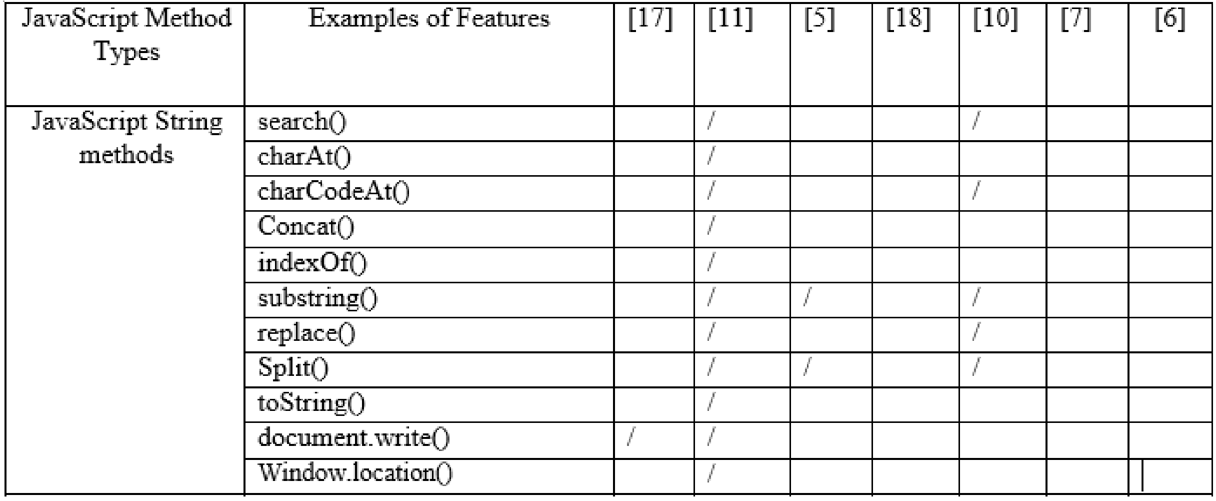
Figure 3: Example of malicious JavaScript located in functions: eval() and unescape()

Another research done by [15], developed a technique based on JavaScript timing attacks for stealing information from the victim machine and from the sites the victim visits during the attack. While [7] proposed a method for detecting malicious JavaScript code based on five features which are: execution time, external referenced domains and calls to JavaScript functions (avgExecTime, maxExecTime, funcCalls, totalUrl, extUrl).

A malicious URL is more likely to be developed with a malicious URL [5]. As discussed in [16], the malware behavior is more resource-consuming than a trusted application. So we have emerge with an analysis within web page and web content only in detecting malicious web pages. This will help researcher in shorten the analysis time and process in detecting the malicious or benign web pages.

Table 1 below summarize the proposed features which have been done and implement by various researcher in detecting malicious web page content. We collected several features which commonly been used by most of the researcher and also few features only can be implemented also in defining the web page whether benign or malicious. Most of the proposed features are from content based features which are primarily from the HTML content and JavaScript method and function used in the web pages.

Table 1: JavaScript string methods type features (1)

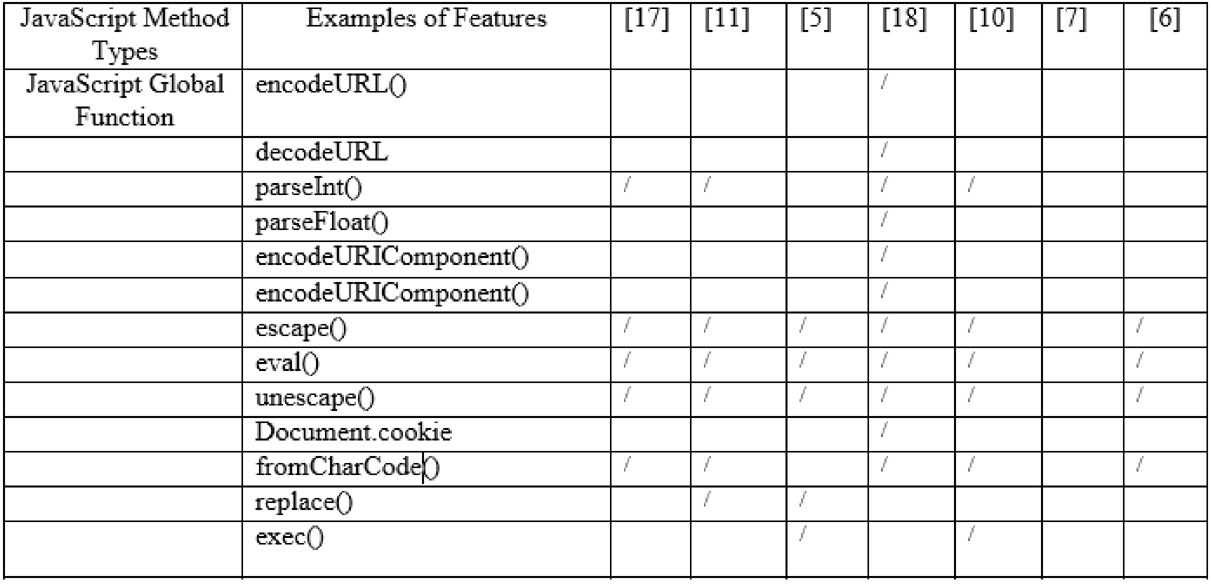


JavaScript Method Types	Examples of Features	[17]	[11]	[5]	[18]	[10]	[7]	[6]
JavaScript String methods	search()		/			/		
	charAt()		/					
	charCodeAt()		/			/		
	Concat()		/					
	indexOf()		/					
	substring()		/	/		/		
	replace()		/			/		
	Split()		/	/		/		
	toString()		/					
	document.write()	/	/					
	Window.location()		/					

Based on Table 1, we categorized several features based on JavaScript strings and method. These feature was classified based on several factor and consideration. The most common consideration where the features representation was denote by most of the attackers in their malicious web pages found in dataset.

Most of this strings and method category are quite vulnerable and frequently used by an attackers to spread the malicious script. Based on thoroughly analysis from other researchers [11, 10] suggest that this features should be considered when detecting a malicious web pages.

Table 2: JavaScript string methods type features (2)



JavaScript Method Types	Examples of Features	[17]	[11]	[5]	[18]	[10]	[7]	[6]
JavaScript Global Function	encodeURL()				/			
	decodeURL				/			
	parseInt()	/	/		/	/		
	parseFloat()				/			
	encodeURIComponent()				/			
	encodeURIComponent()				/			
	escape()	/	/	/	/	/		/
	eval()	/	/	/	/	/		/
	unescape()	/	/	/	/	/		/
	Document.cookie				/			
	fromCharCode()	/	/		/	/		/
	replace()		/	/				
	exec()			/		/		

From the table 2 above, W3Schools have categorized another most popular JavaScript features are from Global Function category. This features are the most common list being used, which are vulnerable and exploited by attackers for injecting this functions with malicious script.

Table 3: JavaScript string methods type features (3)

JavaScript Method Types	Examples of Features	[17]	[11]	[5]	[18]	[10]	[7]	[6]
JS DOM HTML	setInterval()				/	/		/
	setTimeout()	/	/		/	/		/
HTML tags	<iframe>	/	/	/				/
	Number of hidden elements			/				/
	Unequal HTML tags			/				/
HTML Global Event	onload		/			/		/
	onerror		/			/		/
	onunload		/			/		/
	onmouseover		/					/
	onbeforeunload		/			/		/
	addEventListener(), attachEvent(), dispatchEvent()							/

While from the table 3 above we have summarize based on content based HTML types of features. This features was achieved upon downloading the web pages. Most of the researcher [6, 10, 11] proved that this features are important to be considered in detection the malicious web pages.

As mention by Canali et al, [3] classify that this features was very helpful in detection of an early threats of malicious script in web pages. Most of the content based features are derived from HTML and the JavaScript.

Another features have been categorized was other features which are selected by researcher based on pattern or based on their own heuristics. Table 4 below depicted and example of features used.

Table 4: Other features (4)

JavaScript Method Types	Examples of Features	[17]	[11]	[5]	[18]	[10]	[7]	[6]
Others Features	No of Characters in JavaScript		/	/				/
	The ratio between keywords and words	/	/	/				/
	No of blank spaces	/	/	/				/
	Average length of Words	/	/	/				/
	No Hex Values	/	/					/
	String sequence in document	/	/					/
	Length the script string	/	/					/
	No of plugins and ActiveX controls	/	/	/				/
	No of classid	/	/					
	No of suspicious tag string	/		/				/
	avgExecTime,						/	
	maxExecTime						/	
	funcCalls						/	
	totalUrl,						/	
	extUrl						/	

The selection of these features in consideration with content-based feature, which are related to JavaScript and HTML elements. This features primarily contribute to how the patterns of the malicious script that inserted by the attackers during the malicious execution in the web pages.

3. Conclusion

Malicious website has evolved nowadays with various and sophisticated techniques by attackers. Various techniques with difference malicious JavaScript characteristic features was proposed by other researcher in detecting malicious website.

However, some of the features proposed only focus on certain detection threats, such as, malicious URL, obfuscation issues and malicious JavaScript in web pages only. Besides, the total number of the chosen features in each researcher also differ based on the how the researcher declare the features itself. For instance, the tagging and naming of the features sometime repeated or convey the same meaning with the already declare features. The biggest issues in defining the features is detecting the obfuscated malicious JavaScript as attacker used various obfuscation techniques.

Therefore, we conducted a thorough study on the malicious code (JavaScript, HTML & other related code) on web pages and propose the most preferred and more representative which being used by others researchers. By using various features hopefully able to detect the malicious activity or threat in website accurately and precisely with minimum duration of detection.

4. Acknowledgement

The authors would like to thank the supervisor and co-supervisor for their helpful discussions and suggestions. This work was supported by Postgraduate Research Grant Scheme (PGRS), University Malaysia Pahang,PGRS 190388.

References

[1] R. B. Hauke Gierow, “Malware in 2018: The danger is on the web | G DATA Blog,” 2018. [Online]. Available: https://www.gdatasoftware.com/blog/2018/09/31037-malware-figures-first-half-2018-danger-web. [Accessed: 08-Nov-2018].

[2] N. Bielova, “Survey on JavaScript security policies and their enforcement mechanisms in a web browser,” J. Log. Algebr. Program., vol. 82, no. 8, pp. 243–262, 2013.

[3] M. Akiyama, T. Yagi, and M. Itoh, “Searching structural neighborhood of malicious URLs to improve blacklisting,” Proc. - 11th IEEE/IPSJ Int. Symp. Appl. Internet, SAINT 2011, pp. 1–10, 2011.

[4] M. Cova, C. Kruegel, and G. Vigna, “Detection and analysis of drive-by-download attacks and malicious JavaScript code,” Proc. 19th Int. Conf. World wide web - WWW ’10, p. 281, 2010.

[5] R. Wang, Y. Zhu, J. Tan, and B. Zhou, “Detection of malicious web pages based on hybrid analysis,” J. Inf. Secur. Appl., vol. 35, pp. 68–74, 2017.

[6] D. Canali, M. Cova, G. Vigna, and C. Kruegel, “Prophiler : A Fast Filter for the Large-Scale Detection of Malicious Web Pages Categories and Subject Descriptors,” Proc. Int. World Wide Web Conf., pp. 197–206, 2011.

[7] G. Canfora, F. Mercaldo, and C. A. Visaggio, “Malicious JavaScript Detection by Features Extraction,” e-Informatica Softw. Eng. J., vol. Vol. 8, no. nr 1, pp. 65–78, 2014.

[8] A. Y. Daeef, R. B. Ahmad, Y. Yacob, and N. Y. Phing, “Wide scope and fast websites phishing detection using URLs lexical features,” 2016 3rd Int. Conf. Electron. Des. ICED 2016, pp. 410–415, 2017.

[9] P. Likarish, E. Jung, and I. Jo, “Obfuscated malicious javascript detection using classification techniques,” 2009 4th Int. Conf. Malicious Unwanted Software, MALWARE 2009, pp. 47–54, 2009.

[10] G. Canfora and C. A. Visaggio, “A set of features to detect web security threats,” J. Comput. Virol. Hacking Tech., vol. 12, no. 4, pp. 243–261, 2016.

[11] D. R. Patil and J. B. Patil, “Detection of Malicious JavaScript Code in Web Pages,” Indian J. Sci. Technol., vol. 10, no. 19, pp. 1–12, 2017.

[12] H. B. `Kazemian and S. Ahmed, “Comparisons of machine learning techniques for detecting malicious webpages,” Expert Syst. Appl., vol. 42, no. 3, pp. 1166–1177, 2015.

[13] G. Lu and S. Debray, “Automatic simplification of obfuscated JavaScript code: A semantics-based approach,” in Proceedings of the 2012 IEEE 6th International Conference on Software Security and Reliability, SERE 2012, 2012, pp. 31–40.

[14] Y. T. Hou, Y. Chang, T. Chen, C. S. Laih, and C. M. Chen, “Malicious web content detection by machine learning,” Expert Syst. Appl., vol. 37, no. 1, pp. 55–60, 2010.

[15] Dennis Fisher, “JavaScript and Timing Attacks Used to Steal Browser Data | Threatpost | The first stop for security news,” 2013. [Online]. Available: https://threatpost.com/javascript-and-timing-attacks-used-to-steal-browser-data/101559/. [Accessed: 10-Dec-2018].[Online]

[16] M. F. Zolkipli, “An Approach for Malware Behavior Identification and Classification,” 2011 3rd Int. Conf. Comput. Res. Dev., vol. 1, pp. 191–194, 2010.

[17] W.-H. Wang, Y.-J. Lv, H.-B. Chen, and Z.-L. Fang, “A Static Malicious Javascript Detection Using SVM,” Proc. 2nd Int. Conf. Comput. Sci. Electron. Eng. (ICCSEE 2013), no. Iccsee, pp. 214–217, 2013.

[18] Z. S. 3 Javad Hajian Nezhad 1, Majid Vafaei Jahan 3,∗, Mohammad-H. Tayarani-N 2, “Analyzing New Features of Infected Web Content in Detection of Malicious Web Pages,” ISC Int’l J. Inf. Secur., vol. 9, no. 2, pp. 63–83, 2017.

[19] N. Provos, D. Mcnamee, P. Mavrommatis, K. Wang, and N. Modadugu, “The Ghost In The Browser Analysis of Web-based Malware,” *Proc. first Conf. First Work. Hot Top. Underst. Botnets*, vol. 462, p. 4, 2007.