A Static Malicious Javascript Detection Using SVM

WANG Wei-Hong, LV Yin-Jun, CHEN Hui-Bing, FANG Zhao-Lin
Zhejiang University of Technology
HangZhou, China
e-mail: 492166641@qq.com

WANG Wei-Hong, LV Yin-Jun, CHEN Hui-Bing, FANG Zhao-Lin
Zhejiang University of Technology
HangZhou, China
e-mail:492166641@qq.com

***Abstract*—Malicious script,such as JavaScript, is one of the primary threats of the network security. JavaScript is not only a browser scripting language that allows developers to create sophisticated client-side interfaces for web applications, but also used to carry out attacks taht used to steal users' credentials and lure users into providing sensitive information to unauthorized parties. We propose a static malicious JavaScript detection techniques based on SVM(Support Vector Machine). Our approach combines static detection with machine learning technique, to analyze and extract malicious script features,and use the machine learning technology,SVM, to classify the scripts.This technique has the characteristics of high detection rate,low false positive rate and the detection of unknown attacks. Applied to experiments on the prepared data set, we achieved excellent detection performance.**

Keywords-Keywords; SVM; static detection; malicious script detection

I. INTRODUCTION

With the rapid development of network information technology, information security issues gains more and more attentions. The malicious script is one of the primay security threats of computer networks. By constructing a special web page, which contains Trojans, viruses, worms, or aggressive programs, malicious script propagate to the user's computer when the user access to these pages.

Based on the execution state of malicious script, the current detection methods of malicious script can be divided into the static analysis and dynamic analysis method:

Without executing the script, the static analysis method uses the static characteristic, the structure of the scripts to identify malicious scripts, take [1] as example, it counts malicious signatures, then weights the different statistical methods with Judgment matrix method, and at last uses the weighted geometric mean method to obtain the results. This method not only requires some obvious features, but also weak at finding unknown attacks.

Dynamic analysis method, which runs malicious scripts in the controlled environment, detects malicious scripts by observing the execution states, processes. In [2][3], they monitor system ports, network connections, the registry, system configuration files , to detect abnormal procedures. The method has to run malicious code, which increases the risk of the system, and the efficiency is also a problem.

Malicious script is the special code hidden in the scripting language, such as js files. Thanks to its standardized script format, grammar, we tend to get enough static characteristics information of the file, to distinguish the malicious script and the benign script[4]. This article uses machine learning techniques to analyze the feature of the script, proposes a static detection method based on SVM.

II. MALICIOUS SCRIPT FEATURE EXTRACTION

JavaScript[5] is a lightweight, object-based and event-driven scripting language. JavaScript based on HTML could develop interactive Web pages, making web users achieve real-time, dynamic interaction [6]. However, JavaScript is also an attractive choice for attackers to implement their assaults and distribute them over the Internet., such as cross-site scripting attacks, SQL injection attacks and passive download attack.

According to a survey to 90 sites in the China Education and Research Network (CERNET) in 2008, nearly one-third of the sites was attacked. And 39% of the attacks is caused by the malicious JavaScript [6]. Its characteristics make JavaScript easy to become a carrier of malicious programs.

JavaScript has two characteristics: First, JavaScript, a description language as a file, can be executed directly through the browser; Second, Without protection, JavaScript written in the HTML can be seen and copy by anyone directly.

Therefore, these characteristics have made JavaScript the one of attackers' favorite tools. To solve this problem, sand-boxing mechanism is provided to prevent malicious JavaScript from compromising the security of client's environment[8]. And it allows the code to perform a restricted set of operations only. What's more, the sand-boxing mechanism not only brings the problem of efficiency, but also constraints the execution of JavaScript in client. In this paper, we turn to machine learning classification techniques to solve this problem.

To achieve this goal, features are analyzed and extracted at first. According to [9], we can extract 17 malicious JavaScript features. And 10 features more are added based on the analysis of the data. The part of 27 features are explained as follows:

In most benign cases, the number of some special functions is limited while there are a relatively large number of these functions in malicious script, such as the eval function, escape function,DOM-modifying function. The exploits usually call several of DOM functions in order to instantiate vul-nerable components and/or create elements in the page for the pur-pose of loading external scripts and exploit pages.And the escape function could be called to

Published by Atlantis Press, Paris, France.

code malicious string.The abnormal use of special keyword, tag,string are also considered.

Unfortunately, obfuscation techniques, which was intended to protect the source code, is taken by the attackers to circumvent these feature extraction. In order to reduce the impact of the obfuscation, we also do a certain degree of strength analysis [10]. Some features such as the scripts' whitespace percentage, the maximum entropy of the strings, the entropy of the script, are measured. Table.I shows one of the results :

TABLE I. 27 FEATURES OF DATASET

1	the number of eval() function	15	the number of DOM modification functions
2	the number of the setTimeout() functions	16	the script's whitespace percentage
3	the ratio between keywords and words	17	the average length of the strings used in the script
4	the number of built-in functions used for deobfuscation	18	the average script line length
5	the entropy of the strings declared in the script	19	the number of strings containing "iframe"
6	the entropy of the script as a whole	20	the number of suspicious tag strings
7	the number of long strings(>40)	21	the length of the script in characters
8	the maximum entropy of all the script's strings	22	the number of unescape and escape
9	the probability of the script to contain shellcode	23	the number of classid
10	the maximum length of the script's strings	24	the number of parseInt and fromcharcode
11	the number of string direct assignments	25	the ratio between document.writeln and line
12	the number of string modification functions	26	the number of chars in hex
13	the number of event attachments	27	the number of CreateObject,ActiveXObject
14	the number of suspicious strings		

III. MALICIOUS SCRIPT DETECTION BASED ON SVM

The machine learning technology,SVM, which could help summarize the knowledge of identifying known malicious JavaScript, carry out a similarity search to find unknown malicious JavaScript, with a high detection rate and low false alarm rate [11].

A. SVM

SVM (Support Vector Machine), originated in statistical learning theory by Vapnik et al in 1995, was focused on pattern classification problems [12]. It is a statistical learning algorithm that classifies the samples using a subset of training samples called support vectors.In simple terms, SVM, which creates a feature space with the attributes in the training dataset, is to search a decision boundary or an optimal hyperplane to separates the feature space with the maximum interval,as shown in Fig.1.

There are two types of SVM. The linear SVM which separates the data points with a linear boundary and the non-linear SVM which separates the data points with a nonlinear boundary.

In the case of linearly separable problems, it is easy to find the plane in the feature space that separate two types of samples. Therefore, our optimal plane is the one that has maximum geometry interval. As the following formulas shows:

$$\min \frac{1}{2} \| \omega \|^2$$

$$\text{s.t.}, y_i(\omega^T \cdot x_i + b) \geq 1, i = 1, \ldots, n$$

Obviously, it's a convex quadratic programming problems. To solve this problem, firstly, the Lagrange function should be brought in to turn it to its dual problem,.The slack variable and penalty function are proposed to deal with linearly inseparable problem caused by noise. Then the objective function convert to:

$$\min \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i$$

$$\text{s.t.}, y_i(w^T x_i + b) \geq 1 - \xi_i, i = 1, \ldots\ldots, n$$

$$\xi_i \geq 0, i = 1, \ldots\ldots, n$$

Linear SVM performs well on datasets that can be easily separated by a hyper-plane into two parts. But sometimes datasets are complex and are difficult to classify using a linear kernel. Non-linear SVM classifiers can be used for such complex datasets.

In the non-linear case, it maps the data into a high dimensional space, where an optimal separating hyperplane would be found. With appropriate mapping function, most of the non-linear problem can be transformed into the linear problem in high-dimensional space. However, the high-dimensional mapping also brings the curse of dimensionality, and it is a disaster to calculate separating hyperplane in the feature space. The inner product can be realized in the feature space with kernel function satisfies Mercer, which is a trick to this problem:

$$\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$s.t., \alpha_i \geq 0, i = 1, \ldots, n$$

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

Common kernel functions are polynomial kernel, Gaussian kernel, Sigmoid kernel function. Gaussian kernel is a universal nuclear function, by selecting the appropriate parameters, it can achieve a high correct rate. Gaussian kernel:

$$k(x_i, x_j) = \exp(-\gamma \cdot \| x_i - x_j \|^2), \gamma > 0$$

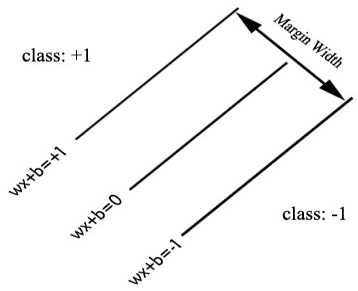

Figure 1. Optimal hyperplane

B. *The malicious script analysis framework based on SVM*

As mentioned before, the script analysis can be divided into static analysis and dynamic analysis. Here, we propose an SVM-based static analysis method, combined with machine learning classification techniques, to distinguish malicious scripts and benign script. Its script training flowchart and script test flow chart are shown in Fig.2.

a) Dataset preparation: collect enough malicious JavaScript and benign JavaScript from the site.

b) Data cleaning: cleaning the sample data, such as the removal of the Notes, excess carriage return and line feed, which increases the processing speed and accuracy.

c) Feature extraction: extract 27 features based on the analysis above.

d) Pre-treatment: data normalization processing, scaled to [0,1]. This process reduces the training error while the data characteristic value is too large, or too small. Second, the efficiency could also be improved.

e) Parameter tuning: the WEKA is the platform to train models. With a grid of binary classification SVM traverse GridSearch algorithm and ten-fold cross-validation,it selects the best SVM model parameters.

f) Model training: training best SVM model to obtain the optimal parameters.

g) The data prediction: using the best model to predict the classification of the test set.

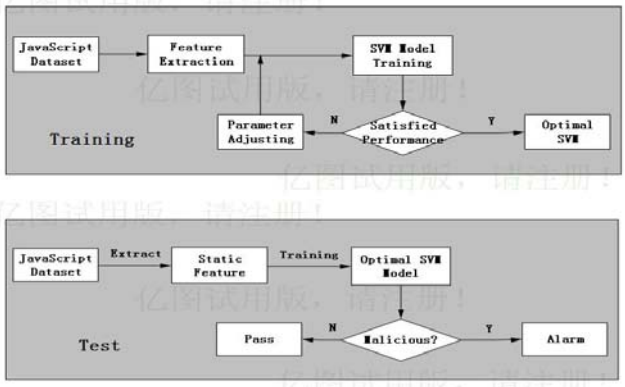

Figure 2. The flowchart of malicious JavaScript

IV. EXPERIMENTAL ANALYSIS AND IMPLEMENTATION

A. *Experimental Analysis*

The experimental data is composed of 1000 malicious JavaScript collected from VX Heavens [13] and 1000 benign ones from reputable sites. The dataset is divided into three, one third as the training set and two thirds as the test set.

According to the analysis previously,we extract 27 features of the dataset, scale on the extracted features, and converts it into WEKA file format.

The Table.II above shows that , SVM obtains more than 90% both on accuracy and recall, and the accuracy on the training set even raised to 93.8% . SVM shows a better accuracy even in the case of less training samples.

In this paper, we choose the RBF kernel to get the best classification model. Two parameters would be adjusted, the penalty factor C and kernel function parameter γ.

C is used to weigh the "Find largest interval hyperplane" and "make sure minimum deviation of the data points", C set large value easily causes overlearning, and reduceing the generalization performance. When set small value, it results in less learning, which all the sample are classified into the strong class. γ stands for the nuclear radius, directly impacts the classification performance of SVM. With too large value, it will end in zero generalization ability, while with too small value, the classify ability of new samples close to zero,even it has a high accuracy on the training set[14].

The optimization algorithm, GridSearch on WEKA, is used in this paper to search the optimal parameters.Set accurately rate as criterion, 1 as Step of C, γ steps as a base unit, and obtain the experimental results of Table.III. when C = 27, γ= 4, the training set accuracy up to 96.59%. And get the best optimization model parameter training.

As shown in Table.IV, SVM gains higher accuracy on the training set and a test set than ADTree and NaveBayes. NaveBayes even don't obtains 90%, while SVM has an accuracy of 94.38% on the test set. It is clear that the SVM is better at handling binary class.

These experimental results shows that, the static detection method based on SVM we proposed, is excellent both on the accuracy and detection efficiency.

TABLE II. THE WEKA FILE OF NORMALIZED EIGENVECTORS

	TP Rate	FP Rate	Precision	Recall	F-Measure	Roc Area
malicious	0.912	0.038	0.958	0.912	0.934	0.937
benign	0.962	0.088	0.919	0.962	0.940	0.937
average	0.937	0.064	0.938	0.937	0.937	0.937

TABLE III. PARAMETER OPTIMIZATION WITH GRIDSEARCH

C	γ	Optimal parameter	Accuracy of training set	Accuracy of test set
1~128	$2^{-10} \sim 2^{6}$	C=27 γ=4	96.59%	94.38%
1~128	$3^{-10} \sim 3^{6}$	C=30 γ=1	95.48%	95.46%
1~128	$5^{-10} \sim 5^{6}$	C=8 γ=5	96.48%	93.38%

TABLE IV. THE COMPARISON OF THE ACCURACY OF TRAINING SET AND TEST SET OF SVM, ADTREE, AND NAVEBAYES ALGORITHM

Learning algorithm	Accuracy of training set	Accuracy of test set
ADTree	*94.94%*	*91.68%*
NaiveBayes	*86.36%*	*84.31%*
SVM	*96.59%*	*94.38%*

B. System implementation

The implementation of prototype system for the detection of malicious JavaScript is introduced in this section. The system can directly detect a JavaScript script, or deal with a URL to detect the JavaScript the page contains.

The module of feature extraction and SVM detection is developed by C, while the script extraction is by PHP. As shown in Figure 3.

1) Script extraction module:

This module is developed for the user as interface, provides services of script detection . Users can either choose to upload a JavaScript script, or submit a URL address. This module will analyze the page, and extract the JavaScript and then package to feature extraction module for further analysis.

2) Feature extraction module:

Firstly,this module would accept the JavaScript from the last one, do data cleaning and remove extra blank lines, comments and so on; Then extract 27 feature previously mentioned; At last, the data is scaled to [0,1] to improve computational efficiency, and converted into the standard form of the next detection module.

3) SVM detection module

The model used here,SVM, is trained with optimal parameters. It accepts a standard data from feature extraction module. Detected by SVM, the results are then delivered to display in the script extraction module.

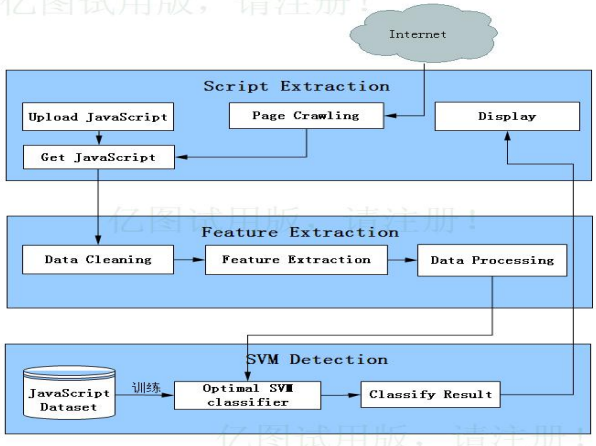

Figure 3. The implementation of prototype system

V. CONCLUSIONS

This paper proposed a SVM-based malicious JavaScript detection method, which,based on fully analysis of scripting language, extracts the static information of the script, and improves the detection efficiency and safety of the system, without parsing and compiling the script; The SVM has a good reputation in the practical application of machine learning, and helps detect unknown attacks. The experimental results show that this method has a high accuracy and low false alarm , and could detect unknown attacks.

ACKNOWLEDGMENT

This research was supported by grant R1090569 and LY12F02039 from the Natural Science Foundation of Zhejiang Province.

REFERENCES

[1] Hao Zhang, Ran Tao, Zhiyong Li, Hua DU , The Detection Methods of Malicious Script . Ordnance Academic Journal, 2008.

[2] Ming Zhu, Qian Xu, Chunming Liu. The Analysis And Detection Of Trojan, Computer Engineering and Applications, 2003.

[3] Oystein Hallaraker, Giovanni Vigna, Detecting Malicious JavaScript Code in Mozilla.IEEE, 2005.

[4] Min Dai, Ya-Lou Huang, Wei Wang, Trojan detection Model Based On Static File Information. Computer Engineering, 2006,3 (6): 176 - 179.

[5] D. Flanagan. JavaScript: The Definitive Guide, 4th Edition.December 2001.

[6] Yinhe Zhang, Wenxin Liang, Xinlei Li, Self-study manual of JavaScript . Tsinghua University Press,2008-10.

[7] Bin Liang, Jianjun Huang, Fang Liu, Dawei Wang, Daxiang Dong, Zhaohui Liang. Malicious Web Pages Detection Based on Abnormal Visibility Recognition. IEEE, 2009.

[8] V. Anupam and AJ Mayer. "Secure Web Scripting". IEEE Internet Computing, 1998, 2 (6) :46-55.

[9] Likarish P., Jung E., Jo I. Obfuscated malicious JavaScript detection using classification techniques. IEEE :47-54.

[10] Byung-Ik Kim, Chae-Tae Im, Hyun-Chul Jung. Suspicious Malicious Web Site Detection with Strength Analysis of a JavaScript Obfuscation.International Journal of Advanced Science.2011.

[11] Xiaokang Zhang. Malicious code detection technology based on data mining and machine learning research [D]. Master's degree thesis of USTC .2010.

[12] Vapnik VN The nature of statistical learning theory [M]. Springer Verlag, 2000.

[13] VX Heavens. Http://vx.netlux.org/ [EB / OL]. 2006-09-28.

[14] Xiaofei Yan, Hongwei Ge, Sheng Yan, RBF kernel SVM and Its Application, Computer Engineering and Design, 2006.