Indian Journal of Science and Technology, *Vol 11(16), DOI: 10.17485/ijst/2018/v11i16/102356,* April *2018*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A Framework for XSS Attack Prevention in Web Browser using Interceptor Approach

Nayeem khan[1] and Abdullah Saleh Alqhatani[2]

[1]Faculty of Computer Science and Information Technology, Albaha University, Albaha, Saudi Arabia; 15010049@siswa.unimas.my

[2]Deanship of Common First Year, Department of Self-development Skills, King Saud University, Riyadh, Saudi Arabia; shamer_84@hotmail.com

Abstract

Objectives: Cross site scripting attacks are performed through malicious JavaScript's with the intention to attack client side. This paper proposes an efficient approach for detection of previous unknown malicious JavaScript attacks using machine learning techniques with high detection accuracy. **Methods/Statistic Analysis:** Despite the plethora of prevention and detection techniques, detection of malicious code such as XSS at the client side during execution by the browser is still a threatening and time-consuming process which degrades the browsing performance due to increased configuration overheads. The proposed approach can efficiently detect such attacks, which are in the form of malicious scripts before they get executed on the browser by employing an interceptor for all the HTTP traffic coming from the server to the client using machine learning classifiers for novel XSS attacks. **Findings:** It is expected that proposed framework once implemented will be able to achieve high detection accuracy with low false positives and fewer performance overheads. **Improvement**: This study provides a strong base for the detection of malware in real-time and experiments will be conducted based on this framework.

Keywords: Attack, Interceptor, Prevention, XSS

1. Introduction

With the rapid expansion of the Internet and rich features of the web application has led to many security flaws in a web application. Some flaws are due to poor programming practices while some are intentionally scripted by the attackers behind the scene. Attackers are constantly working on techniques to get sensitive data through web applications. Applications which are vulnerable to malicious users can break the security and protection mechanism of the system by gaining access to personal information or taking control over system resources. The purpose of the attack is to get access to personal information and system resource, which may cause damage to assets of individuals and organizations and are performed by using the executable code, scripts, active content and other software[1]. Any individual or organization which has its existence over the web has some exposure of being attacked. Depending upon various factors the level of risk varies. Among the reported vulnerabilities, Open Web Application Security Project (OWASP) has ranked cross-site scripting 2nd the most dangerous vulnerability among top ten vulnerabilities. The first attack of XSS was reported in early 90's. Currently, XSS holds a share of 43% among all the reported vulnerabilities. The target of XSS attack is the client side whereas SQL injections target server with the intention to modify the SQL statement to achieve the privileges on the system[2]. XSS attack is vulnerability at the application layer of network hierarchy, which occurs by injecting malicious scripts to break security mechanism. About 70% attacks are reported to occur at Application Layer. Web browsers are the most susceptible application

**Author for correspondence*

layer software for attacks. The purpose of the web browser is to get the requested web resource from the server and displayed in browser's windows. The format of the supplied resources is not restricted to HTML but can also be PDF, image, etc. Attackers run malicious JavaScript in a web browser to target users. Malicious and obfuscated URL's also serve as a carrier for XSS attacks.

Several techniques for detecting of XSS attacks, either at client side or server side, which is commonly distinguished between static and dynamic analysis has been proposed. Static analysis involves reviewing, testing and examining the source code or bytecode of an application without executing the application to find the faults. Static analysis allows analyzing the data flow, checking the syntax and verifying if the states of application are finite[3]. Signature-based methods are used primarily to implement static analysis[4] but rely on the identification of unique strings in binary code[5]. Static analysis has an advantage that it provides a rapid classification to detect a malicious file without executing it[6]. In the static analysis, the interaction of multiple functions causes unpredicted errors, which are visible only when the application is in running state. This is considered the main drawback of this approach. In the dynamic analysis also known as behavioral analysis, the process of testing, evaluating and collection of information from the operating system in real time takes place during program execution takes place. The objective is to detect any malicious code or malware when a program is in running state rather than by evaluating the code offline.

Security solution providers who are considered as the main party in bringing down attacks use technology, which is primarily based on two complementary approaches, signature-based detection, and heuristic-based detection methods. However, their classification is purely based on the type of features, which are employed in the detection of malicious code. Signature based method relies on the identification of unique string patterns in binary code. Signature based detection approach is unable to survive when a new type of attack occurs. Vendors first need to catch an instance of a new threat in order to create a new signature and respectively update their clients. Since the gap of time between catch of instance of a new attack and creation of new signature is long, it could lead to millions of devices vulnerable. Therefore, this approach of tackling attacks is considered as counterproductive when a new threat arrives. Heuristic-based detection which proves to be very much helpful in detecting primarily unknown attacks and defending them by updating the definition file of the detection system. This method analyzes the characteristics and behavior of a suspected file by using rules determined by experts of the field to decide the malicious or benign behavior of a code or file in order to detect the attack[7]. Heuristic detection is an effective way of detecting unknown attacks in real time but the downside of implementing this approach is that it can take some time in scanning and analyzing code and can increase false positives. Since both the approaches have some limitations. Recently, Machine learning approaches have been employed to use the idea of heuristic based methods for detection of unknown malicious code. To classify new malicious code classifiers is brought into play to learn the pattern in binary code file to perform classification.

Despite a number of techniques for mitigating XSS have been proposed either at client side or server side, XSS still remains a threat to users. Thus an efficient approach to mitigate XSS is demanded. Researchers took the services of machine learning to find the accurate detection approach for detection of previously unknown arracks[8], were the first to explore the possibility of using machine learning for detection malicious web page. Their work limited up to the detection of a malicious web page based on URL by performing lexical analysis. This study does not check the code of the web page for any malicious code. Authors in study[9] used regression analysis by using about 18 selected features for detection of a phishing website. An accuracy of 97.3% was achieved by using a small data set of about 2500 URLs. Authors conducted a study[10], try to detect phishing URLs by performing a comparative analysis of phishing and now- phishing URLS. Their study does not use any classifier.[10] carried a study for drive-by exploit of URLS by using machine learning classifiers pre-filters. The limitation of this approach it is time-consuming as it employs a heavyweight classifier for classification. Authors[11] used a very small dataset for classification of fake medical websites. This study is restricted only up to the detection of the fake medical website and cannot detect other types of fake or malicious websites. Researchers[12] conducted a study for detection of malicious code based on the behavior of malicious code using API sequence calls. Experimental analysis shows that this approach is effective only in detection of previously known malware variants. Authors[13] propose a non-machine learning based approach for securing web application and web users from injection attacks, claims accuracy above 90%.

Motivated by the above stated problem, we are proposing a framework for detection of such attacks, which are in the form of malicious scripts and malicious URL's before they get executed by the browser by employing an interceptor for all the traffic coming from the client side from server using advanced filtering and machine learning classifiers to thwart XSS attacks in real time. It is expected that the proposed approach will have a high detection rate of XSS attacks with low false positives and no performance overheads. The proposed work is the continuation of our previous work[14].

2. Technical Background

XSS is a vulnerability that allows attackers to inject malicious code into the web page to be executed at victim's browser. If the malicious code gets successfully executed in the victim's web browser, then attacker takes the control of the victim's resource and sensitive data. An XSS attack is the composition of an attack vector to penetrate into the system with a payload to perform the effective attack. Figure 1 depicts the principle of XSS attack. Three types of XSS are: Reflected, Stored and DOM based XSS.

2.1 Stored XSS (Persistent XSS)

Stored XSS occurs when malicious javascript is stored on the target server in database, guest book's message forum's etc. Figure 1, Shows Persistent XSS scenario. The malicious scripts get executed when the user visits the malicious site thereby passing the privileges of the user to the attacker who then takes unauthorized actions without user permission.

2.2 Reflected XSS (Non-persistent XSS)

In reflected XSS, the attacker injects the malicious code into the server. Figure 2 depicts non-persistent XSS scenario. The injected code is reflected back to the attacker in the form of error message or search result, which may include some or part of inputs provided to the server as a request. Then reflected XSS attacks are sent to target victim through email or links embedded on the web pages to steal the confidential or take control over the victim's computer.

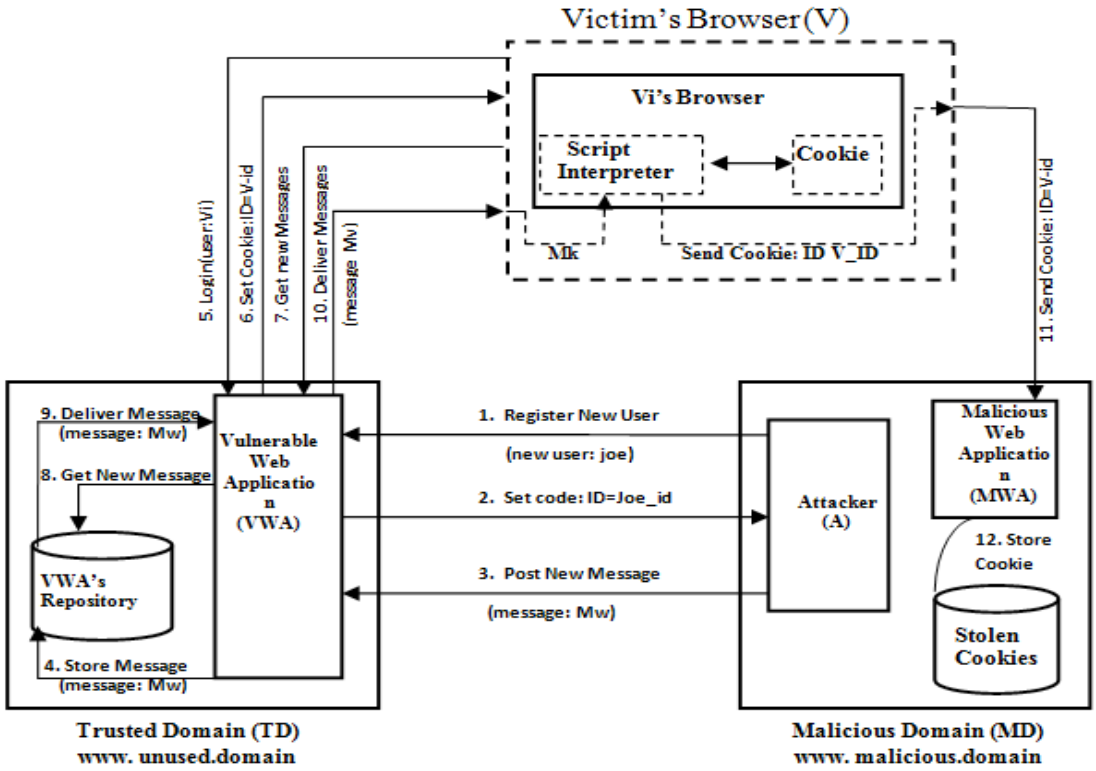

Figure 1. Persistent XSS scenario[15].

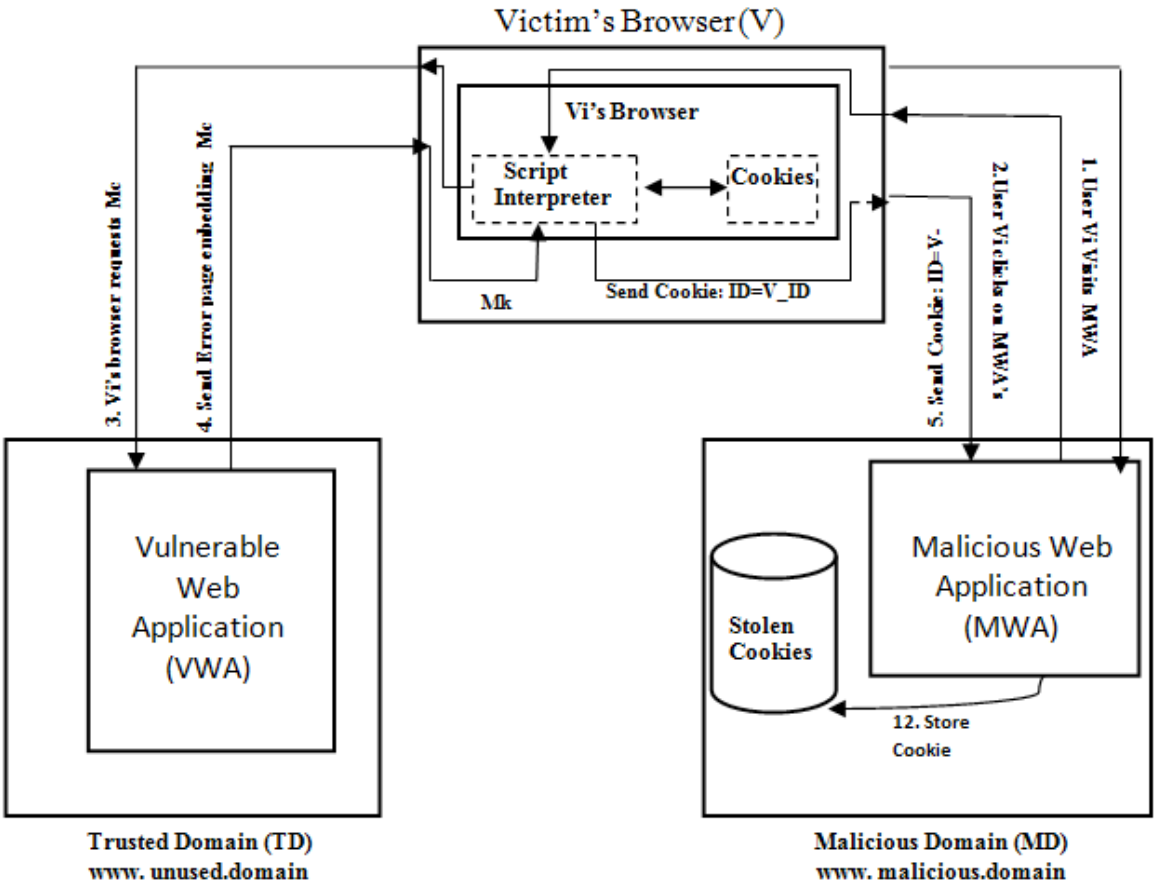

Figure 2. Non-persistent XSS scenario[15].

2.3 DOM based XSS

In DOM based XSS the entire tainted data flow from source to sink takes place in the web browser. The source of XSS can be any HTML element or the web page's URL, while any method with the sensitive call which can cause malicious code to be executed is the sink.

3. Genesis of URL based Attacks

Malicious URL's are created with an intention to attack users by downloading malicious code or malware into the target machine, which can be contained in a spam or phishing mail. Phishing with malicious URL is considered as one of the most common methods of initiating a direct attack. Malicious URL's can also be found inside the source code of malicious web pages. Malicious URL's work by finding vulnerability in application to perform the attack. The malicious code gets executed when a user clicks on a URL without having any knowledge of any attack. Once clicked by the innocent user the attacking party analyzes the type of operation system, browser and other plug-inns for vulnerability to carry out the attack which could lead to buffer overflow attack, format string attack, dangling pointer attack, integer flow attack, etc. In order to bypass data input validation filter's and access restrictions for malicious code and malicious URLs. Attackers use obfuscation of URL's to initiate an attack. Obfuscation is a mechanism to hide details by using encoding schemes such as Hexadecimal, Decimal, Octal, Unicode, Base64, are widely used in attacks[15,16]. Figure 3 illustrates the obfuscated URL with malicious code and has a size larger than URL without encoding.

Multiple domains in a single URL are also used to redirect the presented URL to malicious URL[17], as shown:

www.trustedsite.com/redirect.php?url=http://www.malicioussite.com

```
1  http://www.trustedsite.com/search.html?type=<"<<sCrIpT>alert
2  (document.cookie)</sCrIpT><"<<sCrIpT>alert(document.cookie)
3  </sCrIpT>
4  http://www.trustedsite.com/search.html?type=%3C%22%3C%3C
5  %73%43%72%49%70%54%3E%61%6C%65%72%74%28%64
6  %6F%63%75%6D%65%6E%74%2E%63%6F%6F%6B%69%
7  65%29%3C%2F%73%43%72%49%70%54%3E%3C%22%3C
8  %3C%73%43%72%49%70%54%3E%61%6C%65%72%74%28
9  %64%6F %63%75%6D%65%6E%74%2E%63%6F%6F%6B%
10 69%65%29%3C%2F%73%43%72 %49%70%54%3E
```

Figure 3. Attack using obfuscated malicious code.

Some malicious URL's are more likely to be advertisement-related. Such type of URL's contain certain keywords like 'ad', `advert', `popup', `banner', `sponsor', `iframe', `googlead', `adsys', and `adser. An example of ad URL is shown here:

http://www.mycashkit.com/?from=googleads&click_id=ad1&gclid=CjwKEAjw1riwBRD61db6xtWTvTESJACoQ04QLHAtV1qDPK4WL2zHS2dDXvPLWUB3j1pvQJkvHcZWxoCRDfw_wcB

4. Malicious Code Detection using Machine Learning

Malicious code which is being put into the wild over the web every day thus, making it strenuous for existing approaches to preventing attacks. The static analysis which primarily uses signature-based methods is completely ineffective against unknown malicious code. Recent approaches advocate the use of Machine Learning for detection of unknown malicious code. Attackers very often reuse code in creating a new attack. The property of inheritance exists when the code is being used, thus creating a weakness and important clue that can be investigated for designing an appropriate solution. In order to gain advantages from inherent affinity and code patterns has turned researchers towards Machine Learning. Machine learning is concerned with teaching mechanism to recognize concepts by detecting the sign of patterns in a group of objects[18]. Machine learning has a natural competence for detection of new unknown attacks on the property of swift learning and speedy identification of patterns in a code. Machine learning algorithms do not take raw data directly, but preprocessing is necessary like feature extraction. For categorizing malicious code Machine Learning uses classification and clustering, which splits data into groups. The classification has a set of predefined classes and wants to know which class a new object belongs to[19]. As the training data exits in the classification, it is referred a supervised learning. However, clustering attempts to group a set of objects and tries to find the relationship between the objects that exist. Since no training data exits in clustering, it is referred as unsupervised learning. The task of detection of previously unknown malicious scripts has extensively been studied and significant progress has been made using some Machine Learning approaches such as Support Vector Machines (SVM), Naïve Bayes, Decision Tress, Nearest Neighbor (K-NN), Neural Networks, Random Forests, Genetic Algorithm, latent Semantic Analysis and Rocchio's Algorithm etc. The process of applying supervised ML to a real-world problem is depicted in Figure 4 which is self-explanatory in nature.

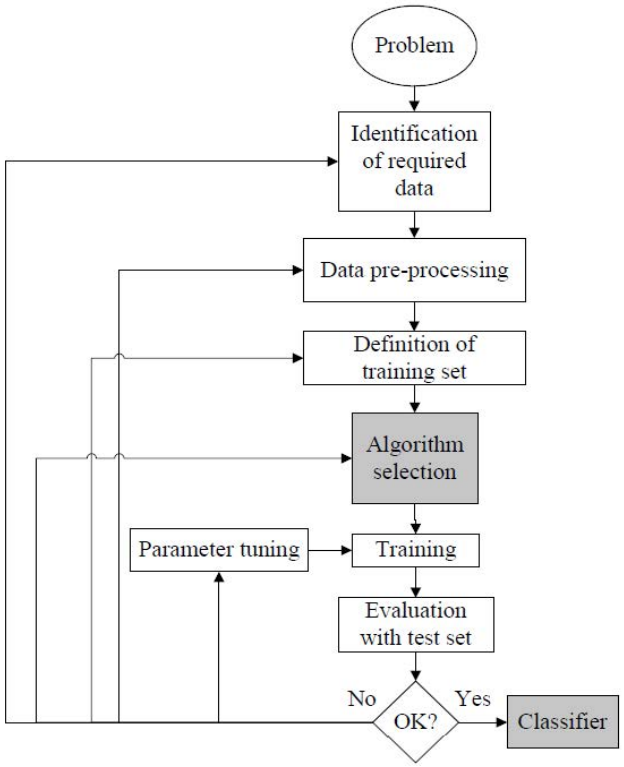

Figure 4. The process of Machine learning based classification.

5. Impact of XSS

An XSS attack occurs due to flaws in server-side applications, and the reasons behind these flaws are due to not properly sanitized HTML characters of an input from the user. The browser is considered as the most directly affected application from XSS attack on client side[20].

The browser interprets and displays HTML Pages. Java's scripts, AJAX and other content hosted on the web server. The content hosted on the web server may be malicious with the intention to target users. It can attack the confidentiality and integrity of browser. Some of the common ways by which attackers target users through a browser are Cookie and Session stealing, browser hijacking, sniffing the browser history, tracking user behavior on the web, buffer overflow, format string attack, dangling pointer attack, integer flow attack, drive-by download and a variety of other ways through which sensitive information maintained by the browser is stolen or access to resource is denied. The end result of such attacks leads to information leak about the cookie and session and disclosure of end user documentation. Other consequences of such as type of attacks on organization may force that organization to issue a press statement about the attack occurred, which may lead to a financial loss by affecting the stock price and lessens the customer confidence. URL's are also used by attackers to target end users by encoding the URL and hiding the parameters. Malwares are used, which act as a spy on web browsers to get the current activities of the user and traffic statistics which leads to credentialed misinformation. Some malicious scripts when interrupted by browsers change the appearance and behavior of web page. Malicious scripts are engineered in such a way that they remain silent about victim's machine but only its impact is visible. To avoid detection of these types of scripts, they are sent to victim's victim's browser for execution in small batches.

6. Architecture of Proposed Approach

The proposed approach consists of an interceptor between browser and server for detection of malicious code. As In this approach, all the traffic between server and client is exchanged through the interceptor to check for possible attacks in the source code to be executed by the browser as shown in Figure 5 there are no direct communication channels between browser and server. Normally, when a client intends to visit a website by typing the URL into the address bar. The request is sent to the web server for lookup and if found response is generated, and the cookie is set up in the browser. In this approach, the response from the server is passed via an interceptor to find a malicious code.

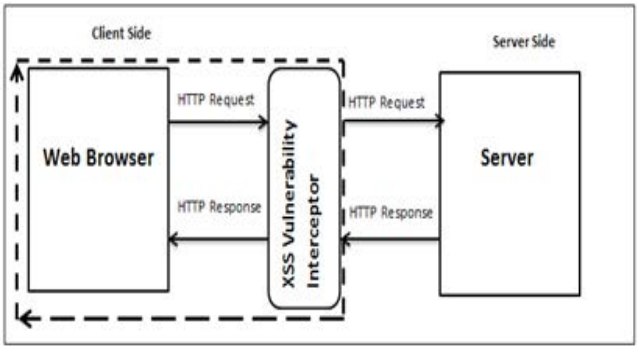

Figure 5. System figure of XSS vulnerability interceptor.

7. Method for Building the Detection Model

The process for building an XSS detection model using Machine Learning approach is achieved by following many steps as depicted in Figure 6.

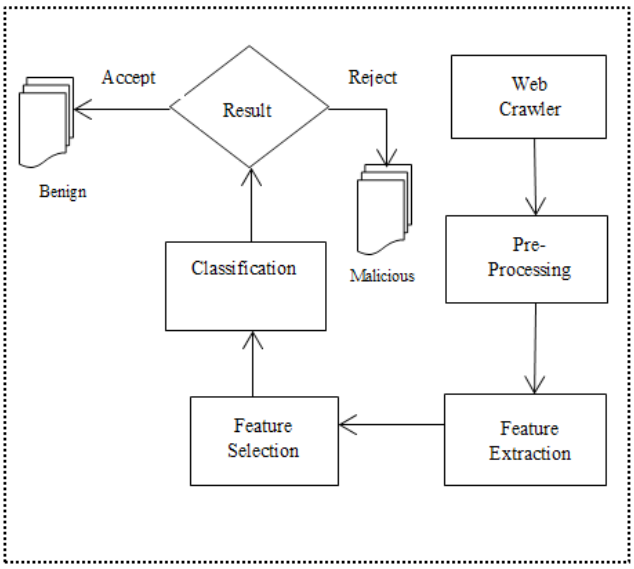

Figure 6. XSS detection model.

7.1 Data Collection

For the preparation of dataset for proposed approached will be collected by a web crawler. Two types of data will be required in this process that is malicious and benign java scripts and URLs. The malicious data will be collected from leading repositories such as XSSed, VX heavens, Phishtank and leading commercial security solution provider F-Secure. Benign data will be collected from top 500 Alexa's websites using the crawler.

7.2 Pre-processing

Since machine learning, algorithms learn from data. It is very important and critical to input required data to solve the problem. Data processing is a data mining technique that involves transforming data into a readable form, removing noise, filling of missing values and resolving other consistencies in data to make it ready for next stage.

7.3 Feature Extraction

Feature extraction is the pivotal step in malware detection. It deals with extracting features from the collected code and generates a feature vector from it. The process of transformation of a large collection of vague inputs into a set of features is referred as feature extraction. This process is required when there a large number to input data to an algorithm which leads to redundancy. If feature extraction is not done in order, it may introduce computational overheads and will have a bad impact on results. The method which is employed in feature extraction has a direct impact on the system efficiency, robustness, and accuracy. Some of the feature extraction methods are byte n- gram, Opcode, Executables, etc. The proposed approach will use a novel set of features while extraction.

7.4 Feature Selection

In order to enhance learning efficiency, increasing predictive accuracy and reducing complexity feature selection plays an important role. The main objective of feature selection in machine learning is to remove irrelevant attributes or no predictive information that may be present in the feature set. Machine learning algorithms do not perform well when there are a lot of features. Selection of a right and important features is necessary for better results. This step is implemented before any machine learning algorithm is used. Advantage of using this step is to remove the problem of overfitting, improves the predictive model with high performance.

7.5 Classifier Selection

Researchers in the field of machine learning have proposed many algorithms for classification in the past. Some of the popular know algorithms are SVM, Naïve Bayes, Decision Tree, K-Nearest Neighbour (K-NN), Random Forests etc. The selection of algorithm depends on the size of the training set and also on the basis of accuracy, training time, linearity, the number of parameters and number of features. If the training set is small in size, then high variance Low bias classifiers are used and if the training set is large for the low variance, high bias classifiers are used. Machine learning algorithms are divided into three categories based on their learning style. They are supervised, unsupervised and semi-supervised algorithms. The proposed approach will use a novel combination of 5 supervised and 2 unsupervised machine learning, and their results will suggest which classifier to be used in the interceptor for the real-time detection malicious code.

7.5.1 Supervised Learning

In supervised learning, the input data is called training data with known labels and is used only when labeled data is available. A model is achieved through a training process where it is required to make predictions and is corrected when the predictions are wrong. The process is continued till the desired level of accuracy is achieved in the training data. Figure 7 shows the process of supervised Learning. Supervised learning is further divided into classification and regression. In classification, labels are discrete while as in regression labels are continuous.

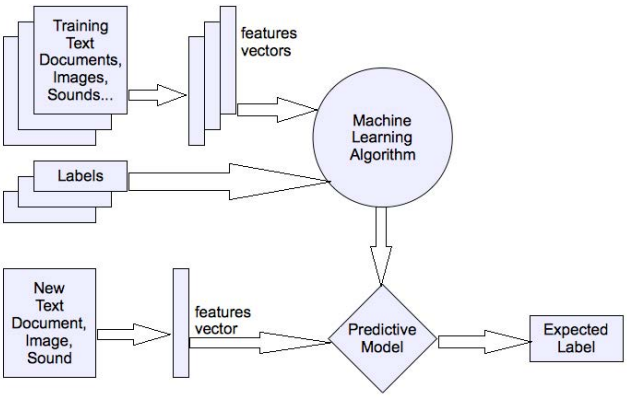

Figure 7. The process of supervised Machine Learning.

7.5.1.1 Naïve Bayes Classifier

Naïve Bayes classifier technique is based on Bayes theorem with independent assumption between predictors. Naïve Bayes model is easy to build when the dimensionality of the dataset is very large. In the context of learning the process, the classifier tokenizes training data to some tokens x_i $(i = 1 \ldots n)$ and counts the number of occurrences of x_i in each class. Based on this process the likelihood of each class is computed with some test data and classifies that test data to the class which has the highest likelihood.

Despite its simplicity, Naïve Bayes classifier is widely used as it surpasses more sophisticated classification methods. Bayesian classifier is based on Bayes theorem, which says.

$$p(c_j|d = \frac{p(d|c_j)p(c_j)}{p(d)} \quad (1)$$

$p(c_j \mid d)$ = probability of instance d being in class c_j
$p(d \mid c_j)$ = probability of generating instance d given class c_j
$p(c_j)$ = probability of occurrence of class c_j
$p(d)$ = probability of instance d occurring

In our study, there are only two classes malicious and benign for classification the range of *j* is from 1 to 2 only. Naïve Bayes has shown that it can classify data across various domains accurately[21].

7.5.1.2 Support Vector Machines

Support Vector Machines (SVM) was developed by[22] is considered as of the most effective models for binary classification of high dimensional data. SVM is devised for linear separation but can also be extended for nonlinear separation. SVM tries to find a linear hyper plane separation that will classy the example of distinctive classes and maximizes the distance between hyper planes and class examples from distinct class. SVM uses a kernel function to map the data into high dimensional space and separates the data on the mapped dimension.

7.5.1.3 K-nearest Neighbour

The K-Nearest Neighbour Algorithm (KNN) is the simplest machine learning algorithm[23]. To determine the category of the test data, K-NN performs a test to check the degree of similarity between documents and k training data to store a certain amount of classified data. Since k-NN classifies instances, in our research, it will be malicious and benign code instances nearest to the training space. The classification of unknown instances is performed by measuring the distance between the training instance and unknown instance. Sine instances are classified based on the majority vote of neighbor; the most common neighbor is measured by a distance function. If k=1 then the instance is assigned to the class of its nearest neighbor. In n-dimensional space distance between two points x and y is achieved by using any distance function:

Euclidean Distance Function

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \quad (2)$$

Manhattan Distance Function

$$\sum_{i=1}^{k}|x_i - y_i| \quad (3)$$

Minkowski Distance Function

$$\left(\sum_{i=1}^{k}\left(|x_i - y_i|\right)^q\right)^{1/q} \quad (4)$$

7.5.1.4 Artificial Neural Networks

Artificial Neural Networks (ANN) is a supervised, powerful and robust classification technique, which is used to approximate real, discrete and vector valued functions[24]. ANN is inspired by a biological immune system. The human nervous system is based on the composition of a large number of interconnected neurons working together to produce feel and reaction. In ANN, artificial neurons are interconnected using a mathematical model to construct a specific application such as spam detection. ANN in our case is the process of separating a code into different classes by finding features between malicious and benign code. In document classification and pattern recognition, many ANN approaches have been used, such as single layered perceptron which has only one input layer and one output. Multi-layered perceptron consists of an input layer with many hidden layers and an output layer and is commonly used for the classification process. The advantage of using ANN is that it works efficiently with high dimensional features and documents with noisy and contradictory data.

7.5.1.5 Decision Trees

Decision tree learners are a non-parametric supervised method used for classification and regression. In a decision tree, classifier is represented as a tree whose internal nodes represent the condition of the variable and final nodes or leaves are the final decision of the algorithm. In the process of classification, a well-formed decision tree can efficiently classify a document by running a query from the root not until it reaches a certain node. The main

advantage of using decision tree it is simple and easy to understand and interpret for naïve users. The risk associated with decision tree is overfitting which occurs when a tree is fully grown, and it may lose some generalization capabilities. Some common reasons of over-fitting are the presence of noise, lack of representation instance and multiple comparison procedures. Overfitting can be avoided by several approaches such as pre- pruning and post-pruning. Figure 8 shows a simple example of the decision tree.

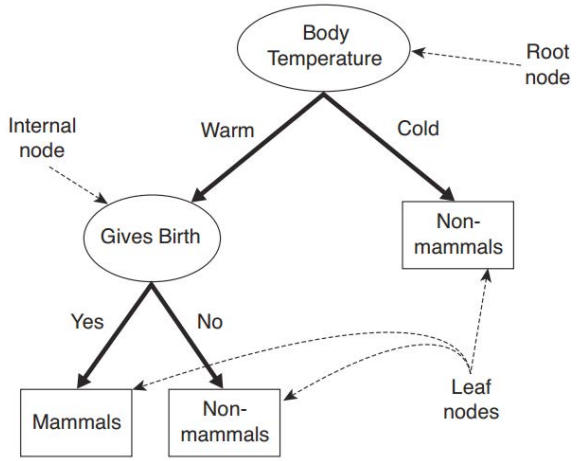

Figure 8. A decision tree for Mammal Classification.

7.5.2 Unsupervised Learning

In unsupervised learning, input data is not labeled like in supervised learning. A model is prepared by analyzing similarities between the objects. Unsupervised learning is the process of discovering the labels from the data itself. Unsupervised learning comprises of tasks such as dimensionality reduction, clustering, and density estimation. Figure 9 illustrates the process of supervised learning.

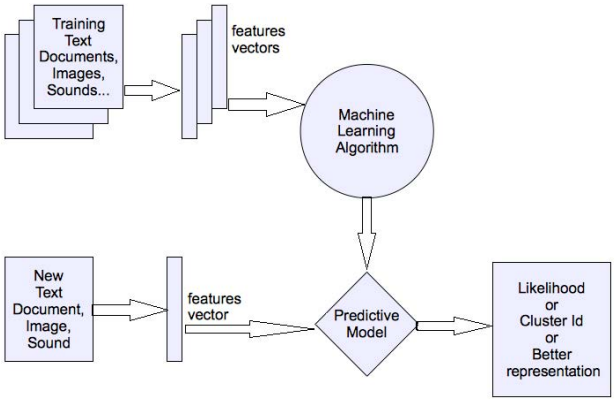

Figure 9. The process of unsupervised Machine Learning.

7.5.2.1 Affinity Propagation

Affinity propagation is a new clustering algorithm which works based on the concept of message passing between data points of samples until convergence and simultaneously considers all the data points as exemplars[25]. Affinity propagation has been used in solving many clustering problems. Based simplicity, general applicability, and performance, we are hopeful that affinity propagation will provide high detection rate in our study.

7.5.2.2 K-Means

K-Means is an unsupervised clustering algorithm that solves well-known clustering problems. Most of its variants involve an iteration scheme that operates over a fixed number of clusters[26]. The purpose of K-Means clustering is to partition n observations into k- clusters with each observation belong to a certain cluster with the nearest mean. Initially, k number is chosen as a centroid data point at the center of a cluster. Each centroid is an existing data point in a given input data set, which is picked at random so that all the centroid is unique and are a distance away from each other until there is no remaining point. Now the need is to re-calculate k- new centroid as a common center of the cluster obtained previously. After obtaining new k-centroid a new binding is established between same data points and centroid, so a loop is generated from this, which changes in the locations of k- centroid can be observed step by step until no change is required. This algorithm aims at minimizing an *objective function*, in this case, a square error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{x} \left\| x_i^{(j)} - c_j \right\|^2 \quad (5)$$

where, $\left\| x_i^{(j)} - c_j \right\|$ a chosen distance measure between a data point the cluster center is c_j is an indicator of the distance of the n data points from t e r respective cluster centre.

8. Expected Results

It is highly expected that the proposed approach for detection of XSS in real-time will have a high rate of accuracy with very low false positives. For the first time, a broad range of supervised and unsupervised classifiers will be used with enormous and diversified malicious

and benign data sets for detecting previously unknown XSS attacks. Results will suggest which classifier to use in which kind of environment, and subsequently we will be able to decide which classifier to use for our real-time detection.

9. Evaluation

Performance evaluation acts as a multi-purpose tool which is used to measure the actual values of the system against expected values. Our main goal of using evaluation is to study and analyze malicious code detection correctness of our proposed approach against the previously used approaches by studying theirs. Although the process of evaluating keeps on going throughout, especially by selecting a most-recent malicious data set to be used in classification. In order to have high results from the proposed approach, we are highly concerned about the accuracy which is defined by Eq. 1.

$$Accuracy = \frac{No\ of\ classified\ Benign\ scripts}{Total\ Benign\ Samples} \times 100 \quad 1$$

A false-positive scenario occurs when the attack detection approach mistakenly treats a normal code as a malicious code. The other situation in which implemented approach is unable to detect malicious code despite its illegal behavior such as situations leads to be false negative. Detection rate is measured by using confusion matrix or error matrix for the assessment of false positives and false negatives. False positive and false-negative detection rate is calculated by Eq. 2

$$FNR = \frac{FP}{N} = \frac{FP}{FP + TN} \quad (2)$$

And the false-negative rate is calculated by Eq. 3

$$FNR = \frac{FN}{N} = \frac{FN}{FN + TP} \quad (3)$$

where,

FPR = False Positive Rate
FNR = False Negative Rate
FN = False Negative
TN = True Negative
TP = True Positive

True Negative shows a number of negative samples correctly identify, False Negative implies a number of malicious samples identified as negative, False Positive indicates the number of negative samples identified as malicious and true positive shows a number of malicious samples correctly identified. The performance of the proposed detection approach will be the rate at which the malicious scripts are processed. The performance will be calculated by latency time is taken in presence and absence of interceptor to display a page and another by calculating the system resource consumption in both scenarios. Achieving real-time detection is impossible where the detection system is poor.

10. Some Characteristics of the Interceptor

10.1 Spatial (Location)

Many approaches for prevention and detection of XSS attacks have been proposed. These techniques have been either implemented on client side location or server-side location. Client side solutions are usually implemented on web browsers. As the malicious scripts are being sent from the server to target client, strict separation between contents produced by malicious sites needs to be enforced to avoid loss of confidentiality. The primary focus for implementing mitigation techniques at the client side is to parse the scripts coming from the server towards client and perform validation. The advantage of using the client-side location for implementing mitigation technique is to reduce of overheads on the server as the sanitation and validation are done at client side location. Client side solutions have capabilities of detection and protection against all types of XSS attacks. Using client side is useful, but it has got some limitations too. As the mitigation on the client side is done by scripting, it may provide a way for attackers to attack by using different attack vectors. Several server solutions which have been implemented at server-side location do exist but are restricted in mitigation of only one form of XSS attack. Server-side mitigation techniques have been globally discarded due to exploitation overhead. Few approaches which have been implemented at both clients and server-side locations called as hybrid does exist. Those techniques which are implemented at hybrid locations, server side is responsible for content inspection and safety rule creation while

as the client side responds by implementing the rules on contents. The advantage of hybrid approaches is that there are very fewer performance overheads. Hybrid detection type is a two- fold scheme containing vulnerability detection phase with the aim to detect, exploit and attack detection phase with the aim to prevent the attack. The proposed approached will be a hybrid which will use both client and server-side locations, but the interceptor will remain at the client side.

10.2 Temporal

Detection and prevention of XSS at the right time are important before the loss of confidentiality and integrity of data. Many techniques which could predict the attack prior it occurs to exist, but their mitigation level is very limited. Post attack techniques are only responsible for the detection of XSS worms. Recently, XSS worm was found in MySpace, Yahoo, and twitter. XSS worms have the capability of self-propagating. The proposed approach will be able to detect and prevent XSS attacks in real time.

10.3 Performance

The performance of the proposed XSS detection approach is the rate at which the malicious scripts are processed. The performance will be calculated by latency time is taken in presence and absence of interceptor to display a page and another by calculating the system resource consumption in both scenarios. Achieving real-time detection is impossible where the detection system is poor.

10.4 Detection Technique

Our proposed protection technique uses static analysis approach in detection rather than dynamic analysis. Static analysis approach evaluates and examines the code without executing the application. The static analysis examines the path and variable of a program which is important in revealing errors. The advantage of static analysis is that it provides full code coverage for analysis, and it is not a compiler dependent. However, drawbacks are memory leaks and concurrent errors. On the other hand, dynamic analysis is the testing and evaluation of a program in runtime mode. The disadvantage of dynamic analysis is that it is complex to work with. Dynamic analysis is performed on only executed paths and does not give any guarantee about non-traversed paths during runtime.

11. Summary

Due to the importance of the Internet and wide usage of the web browser as a medium to access the Internet, vulnerability presented to a web browser is a serious issue. In this paper, we have discussed a specific type of web vulnerability known as Cross Site Scripting (XSS), the technical background of XSS, the different type of XSS, and its impact on the browser. We have also outlined the requirement and the characteristic of the proposed vulnerability prevention model using the interceptor approach. Finally, we have outlined the scope of work in order to achieve the proposed model.

12. References

1. Lee S. New malware analysis method on digital forensics. Indian Journal of Science and Technology. 2015 Aug 5; 8(17):1–6.
2. Beulah S, Dhanaseelan FR. Survey on security issues and existing solutions in cloud storage. Indian Journal of Science and Technology. 2016 Apr 14; 9(13):1–8. crossref
3. Pérez PM, Filipiak J, Sierra JM. LAPSE+ static analysis security software: Vulnerabilities detection in java EE applications. Future Information Technology Springer Berlin Heidelberg; 2011. p. 148–56. PMid:21742481
4. Griffin K, Schneider S, Hu X, Chiueh TC. Automatic generation of string signatures for malware detection. International Workshop on Recent Advances in Intrusion Detection. Springer Berlin Heidelberg; 2009 Sep 23. p. 101–20. crossref
5. Moser A, Kruegel C, Kirda E. Limits of static analysis for malware detection. Computer Security Applications Conference. 2007. ACSAC 2007. Twenty-Third Annual IEEE; 2007 Dec 10. p. 421–30.
6. Potashnik D, Fledel Y, Moskovitch R, Elovici Y. Monitoring, analysis, and filtering system for purifying network traffic of known and unknown malicious content. Security and Communication Networks. 2011 Aug 1; 4(8):947–65. crossref
7. Jacob G, Debar H, Filiol E. Behavioral detection of malware: from a survey towards an established taxonomy. Journal in Computer Virology. 2008 Aug 1; 4(3):251–66. crossref
8. Kan MY, Thi HO. Fast webpage classification using URL features. Proceedings of the 14th ACM International Conference on Information and Knowledge Management; 2005 Oct 31. p. 325–6. crossref
9. Garera S, Provos N, Chew M, Rubin AD. A framework for detection and measurement of phishing attacks.

Proceedings of the 2007 ACM Workshop on Recurring Malcode. ACM; 2007 Nov 2. p. 1–8.
10. McGrath DK, Gupta M. Behind phishing: An examination of phisher modi operandi. LEET. 2008 Apr 15; 8:1–4.
11. Abbasi A, Zahedi F, Kaza S. Detecting fake medical websites using recursive trust labeling. ACM Transactions on Information Systems (TOIS). 2012 Nov 1; 30(4):22. crossref
12. Alazab M. Profiling and classifying the behavior of malicious codes. Journal of Systems and Software. 2015 Feb 28; 100:91–102. crossref
13. Saravanan A, Ahmed MI, Bama SS. Policy approval engine - a framework for securing web applications and web user. Indian Journal of Science and Technology. 2016 Jan 19; 9(4):1–7. crossref
14. Khan N, Abdullah J, Khan AS. Towards vulnerability prevention model for web browser using interceptor approach. 2015 9th International Conference on IT in Asia (CITA), IEEE; 2015 Aug 4. p. 1–5. crossref
15. Baranwal AK. Approaches to detect SQL injection and XSS in web applications. Term Survey paper-EECE 571b, University of British Columbia; 2012 Apr.
16. Likarish P, Jung E, Jo I. Obfuscated malicious javascript detection using classification techniques. MALWARE; 2009 Oct 13. p. 47-54. crossref
17. Nunan AE, Souto E, dos Santos EM, Feitosa E. Automatic classification of cross-site scripting in web pages using document-based and URL-based features. 2012 IEEE Symposium on Computers and Communications (ISCC). IEEE; 2012 Jul 1. p. 000702–000707.
18. LeDoux C, Lakhotia A. Malware and machine learning. Intelligent Methods for Cyber Warfare Springer International Publishing; 2015. p. 1–42. crossref
19. Błaszczyński J, Greco S, Matarazzo B, Słowiński R, Szelag M. jMAF-Dominance-based rough set data analysis framework. Rough Sets and Intelligent Systems-Professor Zdzisław Pawlak in Memoriam, Springer Berlin Heidelberg; 2013. p. 185–209.
20. Pelizzi R, Sekar R. Protection, usability and improvements in reflected XSS filters. ASIACCS; 2012 May 2. p. 5.
21. Domingos P, Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning. 1997 Nov 1; 29(2–3):103–30. crossref
22. Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory ACM; 1992 Jul 1. p. 144–52. crossref
23. Fix E, Hodges Jr JL. Discriminatory analysis-nonparametric discrimination: consistency properties. California University Berkeley; 1951 Feb.
24. De Castro LN, Timmis J. Artificial immune systems: a new computational intelligence approach. Springer Science & Business Media; 2002 Sep 23.
25. Frey BJ, Dueck D. Clustering by passing messages between data points. Science. 2007 Feb 16; 315(5814):972–6.
26. MacQueen J. Some methods for classification and analysis of multivariate observations. Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. 1967 Jun 21; 1(14):281–97. PMCid:PMC1748987