ABSTRACT

In recent years, managing the security over the web has gained its importance. Use of appropriate security handling techniques help to solve controversies and to extract interesting scenarios based on the content of the web page. Many varieties of vulnerabilities prevail and Cross-Site Scripting (XSS) vulnerability is ranked among the top ten risks found over the web which is a mandatory issue that requires a solution. XSS vulnerability injects malicious code in many ways that rise during the browsing session. Analysis should be made over the web page to identify whether the page is vulnerable or not.

A dataset is formulated that contains malicious and benign data. Malicious data are obtained from the XSS archive [source: www.xssed.com] which contains the vulnerable XSS web pages and benign data are the web pages that are obtained through queries from the Google search engine. The major constraint is the number of Lines of Code (LOC) present in the web page.

Classification techniques present in data mining are good enough to manage large amount of data. The technique creates an instance of the supervised learning technique. That is learning from the existing patterns that were spotted to be correct. An excellent classifier is required to find the status of the web page about its vulnerability towards XSS. Many classifiers were analysed in the ‘XSS Detector phase’ and a comparison of classifiers has been performed over the dataset. A better classifier named J48 which suits the dataset is finally chosen for the usage. Generally the efficiency depends on the performance and the accuracy of the classifier.
It is important to recover the web page, if it is found vulnerable. An Application Programming Interface (API) named Enterprise Security Application Programming Interface (ESAPI) [source: www.owasp.org] can be applied to the web page after String Comparison (SC) using hash function. Various methods can be invoked from the API for correction of source code in the ‘XSS Handler phase’. The vulnerable web page will undergo a string comparison procedure and appropriate actions will be handled. The altered web page can be made visible at the client end (the browser) after the prevention process.

In order to apply ESAPI rules effectively, it is necessary to position the area of XSS handling. Path tracking for a web page is required and the process of identifying the HTML elements is done. Optimization of path is the apt solution to resolve the problem. The existing classification algorithms projected in the research are futile for high dimensional datasets and optimization.

To accommodate the process, an evolutionary algorithm namely Particle Swarm Optimization (PSO) (Eberhart & Kennedy 1995) can be applied and the optimum path can be found for XSS prevention. From the literature, it is found that the algorithm is better than Genetic Algorithm (Booker et al. 1989) which models the origin of species. PSO algorithm is able to produce a better detection by identifying the path of the tree. Two thresholds are set with the solution namely 60% (PSO-60) and 80% (PSO-80) and it is found that PSO-80 performs better than PS0-60.
In order to improve the performance of the PSO algorithm, an updated Ant Colony Optimization (ACO) technique named PPACO (Prioritization in Path using ACO) is applied and the performance is compared on par with ACO (Dorigo 1991). The PPACO algorithm is made to merge in the XSS Detector phase to generate a small set of remarkable tracks for building the efficient optimizer.

Five samples from the dataset were considered and algorithms are applied. About 24 attributes are used by the classifier. The samples vary in terms of content and size. Different optimization techniques are applied and the results are analyzed. Evaluation measures like Detection Rate (DR), False Detection Rate (FDR) and F Score (FS) are calculated based on the Confusion Matrix.

The final content obtained after the ‘XSS Handler phase’ that is to be displayed on the browser is tested using black box testing technique and also using XSS and SQL Injection Scanner tool. The tool is capable of identifying promising XSS code available in web pages.

Based on the experiments, it was observed that the generation of paths using PPACO achieves better results in terms of DR, FDR and FS than other algorithms.