CHAPTER 5

PERFORMANCE EVALUATION OF XSS DETECTOR USING PSO OPTIMIZER

5.1 INTRODUCTION

To overcome the XSS security issues at the client end, it is mandatory for a browser to scan the web page as and when the request is received. The process is time consuming as the content of the web page contains raw facts including text, images and many more that is bundled in the form of a DOM object.

PSO being a computational method aims for an optimal solution which is similar to the concept of a bird identifying its group. The algorithm applies a similar concept of grouping where each particle utilize the search space with reference to the function until the process ends, neither when an optimal solution being obtained nor the number of iterations gets over.

Wang et al. (2007) compared the performance of PSO over the other conventional methods with respect to the automatic document classification process. The advantages of using PSO are its path relinking capability, iterative process and greater exploration of the population. Since PSO is based on the velocity value of the particles, it formulates the local subsets with the best global value.

In this chapter, PSO is adopted in the XSS detector phase. The work acquires essential inspiration from social living in biological organism for constructing an effective detector. A customized fitness function is
designed for identifying the best track from the list of paths. PSO optimizer thus built using PSO achieves high detection rate. The experiments are conducted with the datasets and promising results are obtained.

5.2 PATH TRAVERSAL USING PSO OPTIMIZER

The proposed PSO optimizer is used to effectively track paths from a given web page. This section deals with the formulation of the mathematical model for generating track of paths along with the stepwise procedure for using PSO optimizer.

5.2.1 Problem Statement

The web browser is capable of presenting the content of the web application. The web application is a composition of web pages with numerous elements referred as nodes. The nodes and their connections are to be traced before viewing in the browser.

Tracing the connecting nodes is the goal of the optimizer. The node and its value need to act as a parameter in order to produce optimization with a larger impact. It is important to note that the web page is not always static. The objective of the proposed system is to optimize the path of the web page tree.

5.2.2 PSO Algorithm

PSO is a stochastic algorithm, where each solution is a bird or fish or node or particle in the search space. All the nodes are initialized with a group of particles which have fitness values. The fitness values are evaluated by a fitness function with various velocities and finally determine the best function value and the best location are determined. Finally the particles are
updated with the local best and global best values. The process is repeated until no leaf nodes are found.

A mathematical model $M = (S, f, C)$ of an optimization problem consists of:

A search space $S$ defined over a finite set of nodes. A velocity update function $V_{id}$ for each particle is given by

$$V_{id} = V_{id} + P_1 H_1 \ POB_{id} - POS_{id} + P_2 H_2 (POB^n_{id} - POS_{id})$$  \hspace{1cm} (5.1)$$

The acceleration positive coefficients are $P_1$ and $P_2$. $H_1$ and $H_2$ are independent numbers ranging between 0 to 1. $POB_{id}$ represents the best position of the $i^{th}$ element and $POS_{id}$ represents the position of the $i^{th}$ element. $POB^n_{id}$ represents the best position found by the next element. The particle position is updated by

$$Pos_{id} = Pos_{id} + v_{id}$$  \hspace{1cm} (5.2)$$

The constraint $C$ is the stopping criterion where the entire nodes are covered at least once.

5.2.3 Feature Set

Paths ($T_p$) are identified and the feature set $F_j$ is extracted which satisfy the user support minimum threshold ($\delta$) with a value of 60% and 80%.

$$F_{T_p} = \{ F_j \mid F_j \in S \} \ \forall \ F_j \text{ and } \exists \ supp(F_j) \geq \delta$$ \hspace{1cm} (5.3)$$

5.2.4 PSO Representation

For the requested dynamic web page from the browser, all possible nodes of the tree are generated by assuming the initial population or particles.
In general, the initial population represents a potential feasible solution to the problem. With respect to the constraints defined, obtaining the feasible solution is a challenging task. The feasible solutions are evaluated using the fitness function and the optimum or near optimum solution is obtained by updating the functions of the particles and the velocity respectively.

### 5.2.5 Algorithm Complexity

The value of the cost function determines the computational complexity of the PSO optimizer. The number of iterations required is directly proportional on the three factors namely the cost of the current position of the node, updating of particle position and its velocity. Since the research problem is associated with separate cost function, each node and its dimensions can be optimized independently and it is to be noted that most of the real world problems include separate cost criterions with quick convergence.

### 5.3 FRAMEWORK FOR DETECTING PATHS OF A WEB TREE USING PSO OPTIMIZER

Figure 5.1 outlines the proposed methodology of the research problem which has an XSS Detector phase and an XSS Handler phase. The XSS Detector phase tries to discern the path of the given web page using the PSO algorithm with related constraints. The extracted feature set is then passed as input to the other phase. The XSS Handler phase provides sanitization for the paths using escaping rules. The sanitized web page obtained, is then presented by the browser for usage at the client-end.
Figure 5.1 Workflow of the PSO Optimizer
5.3.1 Dataset Pre-Processing

The webpage obtained as input is pre-processed using WEKA 3.7 as mentioned in section 3.3.2. The resultant of the pre-processing will classify whether the web page is XSSed or NonXSSed.

If it is a NonXSSed page, the web page is presented to the browser, else it is provided to the PSO optimizer with parser operations of the tree. The optimizer identifies the path and the chosen paths are associated with the Handler process. Finally, the content is presented in the browser for the user view.

5.3.2 XSS Detector

The PSO optimizer handles the web page that is given as input and the nodes are identified using the parser. The velocity values are initiated for every node and iterations are performed until all the nodes are involved in updating at least once or the criterion is satisfied. Tracing the path is a hectic process if the LOC of the input page is high. The optimizer formulates the path thereby undergoing a sequence of steps. The process of this phase is briefed in Table 5.1.

<table>
<thead>
<tr>
<th>Table 5.1 Path Traversal using PSO Optimizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSOOptimizer(Input, Output)</td>
</tr>
<tr>
<td>// Input: Uniform Resource Locator (URL)</td>
</tr>
<tr>
<td>// Output: Significant Paths Selected</td>
</tr>
<tr>
<td>// ic –iteration count; mi- maximum number of iteration; ( f_b ) –best fitness</td>
</tr>
<tr>
<td>(1) {</td>
</tr>
</tbody>
</table>
(2) Initialize $POS_{id}$ and $V_{id}$ based on pre-processing

(3) Calculate the fitness for $POS_{id}$

(4) Assign $POB_{id}$ with $POS_{id}$

(5) Assign $POB^n_{id}$ with $POS_{jd}$ where $j$ is the index of best next element

(6) Assign $ic$ with zero

(7) Loop until $ic$ is less than $mi$

(8) Find $POB^n_{id}$ if $f(POB^n_{id}) < f(POS_{jd})$ for every element

(9) Assign $POB_{id}$ with $POS_{id}$ when $POS_{id} < f(POB_{id})$

(10) Assign $f_b$ with $f$ $POB_{id}$ and $POS_b$ with $POB_{id}$

(11) Continue the loop

(12) Update $V_{id}$

(13) Update $POS_{id}$

(14) Calculate the fitness of $POS_{id}$

(15) Increment $ic$ by 1

(16) Return $POS_b$

(17) }

5.3.3 Evaluation of the Traversal Path

Identifying the most valid paths in a web page from the obtained tracks is the important task to be performed at this stage. It requires tremendous analysis of the web page and its content respectively which is
very difficult within the user browser in a shorter period of time. So, a threshold is set to filter the longest path of the page tree. A possibility of omitting the important track is also possible due to the usage of threshold value. Moreover the threshold value is set to avoid the overload of the browser.

5.3.4 XSS Handler

This phase performs the process of applying the guidelines specified by the OWASP to the tracks which convince the threshold value. The threshold is compared to the ranked fitness values obtained from the detector phase. Then reformulation of the web page takes place for the usage of the client.

5.4 EXPERIMENTS AND RESULTS

5.4.1 Dataset

Any webpage is an eligible data for the conduct of the experiment. For comparison, datasets mentioned in Section 1.4 are considered. The featured datasets belong to the XSSed class and are thereby considered for involving in the first phase of the optimizer.

5.4.2 Experimental Setup

The class label is obtained after pre-processing as in Section 3.3.2 over the requested web page. If found XSSed, the handler needs to apply rules to the page content in an effective manner.

The following parameters namely $POS_{id}$ and velocity of the element are set by the parser by analysing the considered web page. In this research work, the optimizing algorithm should not consider the insignificant
features and utilise only the relevant ones without information loss. Parsing causes the process of generating the nodes and the significant paths are identified by PSO as explained in section 5.3.2. Two threshold values are set with values 60% (PSO-60) and 80% (PSO-80) respectively. The path retrieved by means of the PSO with specific threshold can be considered to be the significant path.

After the procedure is over, the handler phase takes over control and sanitation takes place. Furthermore the page is reconstructed and presented to the browser. The performance of the optimizer is tested under the two threshold values using the XSS & SQLi tool.

5.4.3 Results and Discussion

The result of the experimentation done with PSO-60 and PSO-80 optimizer for the test domain is shown in Table 5.2. It is observed that PSO-80 could yield more significant paths than PSO-60 since more number of paths is prevented through the handler process.

Table 5.2 Comparison of Evaluation Measures (%) for the Various Test Domains Using PSO

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Detection Rate (DR)</th>
<th>False Discovery Rate (FDR)</th>
<th>F Score (FS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Domain</td>
<td>PSO-60</td>
<td>PSO-80</td>
<td>PSO-60</td>
</tr>
<tr>
<td>Events</td>
<td>46.91</td>
<td>49.38</td>
<td>15.56</td>
</tr>
<tr>
<td>classifieds</td>
<td>43.90</td>
<td>46.34</td>
<td>12.9</td>
</tr>
<tr>
<td>Roomba</td>
<td>28.10</td>
<td>30.72</td>
<td>12.24</td>
</tr>
<tr>
<td>Personal Blog</td>
<td>29.76</td>
<td>32.14</td>
<td>10.71</td>
</tr>
<tr>
<td>Jgossip</td>
<td>31.73</td>
<td>32.69</td>
<td>13.91</td>
</tr>
</tbody>
</table>

PSO-80 holds better for all the considered test domains in terms of DR. While analysing the FDR score, the values of PSO-60 is found low and
thereby PSO-60 scores to be the best algorithm. All the five test domain for FS votes for PSO-80 with an increased difference of about 1% to 3% from PSO-60. Hence the optimizer PSO-80 suits better than PSO-60 for auto sanitization at the client end.

5.5 SUMMARY

In this chapter, PSO was used to develop the path tracking operation of a web page that is valuable for identifying the XSS. It presents a sequential procedure to identify the possible path and to utilize it. The method points out the track and ensures a high rate of XSS detection as in Table 5.2. The content is further cleansed for effective display in the browser. Observations also proved that the significant path using PSO-80 optimizer presents a better performance when compared to PSO-60.