3. OVERVIEW OF THE STRATEGIES TO DETECT AND MITIGATE WEB APPLICATION VULNERABILITIES

Web applications are versatile. They are used as both desktop applications or as smartphone applications. Over the last couple of years web applications have gone through huge transformations. Web applications, which were used in desktop environment, have been adapted to work on smartphone environment. But, the security scenarios in both the cases are different. The nature of attacks on these applications is also different. To deal with this difference, strategies specific to desktop applications and smartphone applications are used.

The attacks that are predominant in the desktop applications are the SQL injection attack followed by the Cross site scripting attack. Literature reveals that Android based Smartphones are most vulnerable to personal information leakage and data theft. This commonly occurs through three factors, namely the permissions, the updates and the WebView. Figure 3.1 shows the overview of the strategies employed to curb these attacks.

3.1 CONTRIBUTION OF THE THESIS

In today’s scenario, Web applications are essential part in our day to day life. All the activities that were once done person to person are now computerized and provided as a web based software application to the clients. These web applications reside on the server and these servers provide service to the clients when requested. The information provided by the client is received by the server and stored for later use. Request and Information thus received may be benign or malicious. To segregate a benign request from a malicious request, a security feature has to be employed. Many of the web applications are developed by amateurs, hence they are most vulnerable to the input validation attacks, involving the SQL Injection attack and the Cross – Site scripting attack.
Figure 3.1: Overview of Strategies to Detect and Mitigate Vulnerability
Most common SQL injection attacks are usage of tautology, union queries and piggybacking. Taking these into consideration a simple yet effective system was proposed to safeguard these applications. The input data cleansing algorithm detects and mitigates these types of vulnerabilities with great accuracy and a negligible overhead.

Attacks on Smartphone’s Web applications are mainly related to personal information theft. The contribution of this thesis to detect and mitigate this attack is three fold.

a) The application when downloaded from the play store may be benign, but once installed, it may receive updates from its server and the updates may have malicious code that extracts information from the user illegally. When an application is from a third party, there is no means of checking the updates. A novel algorithm has been proposed to provide security check during updates.

b) The ‘permission’ appealed by the application is considered as the major security feature provided by the Android operating system. However, these security features could be vanquished by a malicious application that demands excess permissions. To check on this type of attack, permission enforcer algorithm has been proposed and implemented. The custom permissions that are defined by the developer have also been verified to ensure that they do not induce an attack. This is an exclusive feature of this algorithm.

c) Apart from the permission and updates, the other most important loophole in the smartphone environment which is susceptible to attacks is the WebView. Five commonly prevalent attacks have been considered and an extensive study has been made to illustrate the nature of these attacks. Moreover, a critical attack named Supplementary Event -Listener Injection attack has been explored in this work. To detect all of the above attacks, a model has been designed and implemented. This model works on Inference rules.