ABSTRACT

The Internet, one of the wonders of modern science, has impacted the lifestyle of people, regardless of age, education and standard of living. The relation between customers and business has assumed newer dimensions and approached new horizons because of Internet based applications, also called the Web Applications. However, there are also problems that tag along, and mainly the problem of security issues related to Web Applications that needs to be addressed first. Attacks on the Web application can basically be on applications used in desktop devices or handheld devices. In desktop based Web applications, in accordance with the client input to a Web application, data are either stored or processed or retrieved. If this client input could be infected, then the whole Web application system would be compromised. This compromise is caused by the Structured Query Language (SQL) Injection attack and the Cross Site Scripting (XSS) Attack, which are called the Input – validation attacks. Studies have proved that insufficient validation on the client input is the major reason for these attacks. To detect and mitigate these attacks a novel server side approach using data cleansing algorithm has been proposed and implemented in this work. This system uses a reverse proxy that interprets the input from the user, extracts the user query, and sanitizes it before sending the processed input to the server. To test the system, four applications from the test bed www.gotocode.com have been used. The inputs to these applications were the SQL injection commands from the cheat sheet developed by Halfond et al., for their work AMNESIA. The accuracy of the system accounted to 100% with a minimal trade-off of 3.49% increase in response time.

In recent years, the Web Applications have migrated from the client server architecture to the Smartphone systems. The Web Applications have been tailored for use in Smartphones. The security issues in the Smartphone scenario could be caused through two ports, the permissions and the WebView. To identify a system under attack and to dynamically identify the application causing the attack, an algorithm called ‘Amendment Capture...
Algorithm’ has been designed and implemented. The Smartphone may incur the attack after installation, that is, during the update. When there is an update or a behavior change, protective system is triggered. The accuracy of the training and testing system is 94% and that of the monitoring system is 93.5%.

In Smartphones, an application can interact with another application if it is provided with the permission. But there are possibilities to bypass this rule and connect to certain applications. This vulnerability paves the way for personal information theft attacks. To pro-actively identify the vulnerabilities caused by the permissions, an algorithm called ‘Permission Enforcer Algorithm’ has been proposed and implemented. Permission can be of two types, namely the system defined and the user defined. To implement and test the proposed system, seven different categories of applications have been considered. For each category, the permissions are segregated into three lists, namely white list: the list of permitted permissions, the red list: user defined permissions, and the black list: forbidden permissions. Whenever an application is first downloaded, the system reverse engineers the application and extracts the permissions. Depending on the permissions, the application is tagged as legal, illegal or suspicious one. If the application is suspicious, further monitoring of the system is carried out and a verdict on the application is presented to the user. The system, when presented with three applications with different permission scenarios, is able to correctly categorize them as benign and malicious.

WebView is another vulnerable part of the Smartphone that could launch an attack on the systems. WebViews are web-based interface that can be included as part of the application to retrieve and display web contents from remote servers. The WebView component enables the creation of hybrid web-based applications possessing more powerful functionalities than the traditional desktop web-browsers. However, these excess features provided by the WebView can open up portals for launching malicious attacks onto the Smartphones through them. To address the vulnerabilities caused by the WebView, inference rules have been formulated and implemented. The proposed system performs automated static analysis
on the WebView embedded applications and generates a report in a comprehensive format containing the results obtained from the processing of WebView. This analysis is based on the inference rules formulated for six major attacks that could be caused by WebView Vulnerability. A possibility of a new attack, named “Supplementary Event injection attack” has also been addressed. The proposed system has been evaluated to have an accuracy of 85%.

These algorithms have been benchmarked with standard test beds and a comparative study with the existing work has also been presented. Experimental results have shown significant improvement in enhancing the security of the systems when exposed to Web applications.