B-649 Project Proposal: Automated detection of Logic Vulnerabilities in Web Applications

Varun Patil

28th September 2014

Introduction.

Web applications are the most common way to make services available on the internet. However, with the increase in the number and the complexity of the web applications, the vulnerabilities associated with the web applications have also increased. Logic vulnerabilities result from the insufficient validation of the of the business process within the web application. The web application vulnerabilities can be classified into two broad categories

1)Vulnerabilities that have common characteristics across different applications

2)Vulnerabilities that are application specific

Vulnerabilities like cross-site scripting (XSS) and SQL injection belong to the first category. It is essential to have the knowledge of the logic of the web application to identify and categorize the second type of vulnerabilities.

The traditional approaches to identify security problems in the web application focuses on the input validation flaws. The vulnerabilities that are specific to the web application are often out of scope of the existing tools. These vulnerabilities have to be discovered manually.

Importance

The programmers that develop the web application often are under time and financial constraints. As a result, they often end up ignoring the application specific vulnerabilities but take sufficient precaution against generic vulnerabilities like SQL injection and Cross Site Scripting. Consequently, these vulnerabilities mount up and project ends up being an easier target for the attacker.

Since the web application for health centers or banks hosts sensitive user data. It is very essential to safeguard against these vulnerabilities and prevent the attacks that expose these vulnerabilities. Hence, it is essential to have a system in place that automatically detects the logic vulnerabilities in a web application.

The logic vulnerabilities are specified to the intended functionality of a web application. Hence it is difficult to propose a general specification that identifies and mitigates the flaws in logic of a web application.

Objective

The objective of the research is to come up with a platform independent and generalized technique that identifies the logic vulnerabilities. Such a generalized technique can be utilized for a varied range of user applications like banking, e-commerce and health applications.

Approach

One of the approaches to identifying the logic vulnerabilities in the web application is to analyze the dynamic execution of the program invariants and model checking to identify the specification violations. In the first step involves automatically obtaining the specification that reflect the business logic of the specific application. In the second step we understand the inferred variations and identify the violations with respect to the web application's code. The success requires the combination of this two application.

Another option is to implement the black box technique where we can study the way in which the user interacts with an application. Based on the extracted behavioral patterns. target test cases can be generated with respect to some common attack scenarios. The key to extract the behavioral patterns is to analyze the network traces during normal operation.

Other techniques to successfully predict the user behavior while using a web application also need to be analyzed and compared in the light of our proposed technique in order to judge the success of our application.

Plan

The plan is to analyze the various techniques that ensure automated detection of logic vulnerabilities in the web application. The next goal would be to write a document citing each of the techniques with their advantages and drawbacks. This will help us infer which technique is both effective and platform-independent. An optimal solution would be to combine the advantages of more than one technique to achieve greater security.