ABSTRACT

Name of Candidate : **Faisal Anwer**

Name of Supervisor : **Dr. Mohammed Nazir**

Name of Co-supervisor : **Prof. Khurram Mustafa**

Name of the Department : **Computer Science**

Topic of Research : **Security Testing of Object Oriented Code**

Keywords: Security Testing, Security issues and concerns, Code level security issues, Security testing techniques, Security testing methods, OOP security issues, OOP security testing, Security testing frameworks.

Now a days software applications are facing serious security threats and are more vulnerable to attack than ever before. Flaws in the code playa an instrumental role in application vulnerability and even a single aw in the code can make millions of application vulnerable. The applications based on object oriented programming are far more affected by security threats as it is one of the most popular paradigms in software industry. Further, major chunk of software are being developed using this paradigm. Security vulnerabilities such as denial of services (DoS), cross site scripting (XSS), SQL injection (SQLi) and Buffer over flow (BoF) are found to be prominent one, often being used to exploit the applications.

Among these vulnerabilities, application DoS is one of the most occurring and quite a significant number of instances have been reported in the vulnerability databases. Application DoS makes the application or part of it unavailable or unusable for the valid users. Improper exception handling (IEH) is found to be one of the prominent reasons of application DoS. Improper use of exception handling may raise several security issues including program crash, program inconsistency, resource leakage and information leakage, which might be misused by the attackers. Program crash directly leads to application DoS, while resource leakage and program inconsistency in the long run may also lead to DoS.

Consequences of IEH in a broader sense, the exception bugs can be very severe. It might affect the whole or part of the application and hence may affect a number of users. For instance, improper exception handling caused worst shutdown of the Facebook in the year 2010 and restricted around 135 million people to log-in. In another similar incident, a crash incident was reported in the year 2014 in SQL Server Management Studio due to unhandled exceptions. A high impact severe bug has been discovered in the year 2015 in SAP application server for JAVA that may possibly lead to application DoS. The significant number of cases of program crashes have been observed or noticed in the popular applications such as Hadoop and Eclipse under different releases.

It is therefore, very much desirable to test the application for IEH before the deployment to make it more robust. Moreover, testing of IEH needs special attention because developers generally miss the handling of exceptions or not trained enough for proper handling. It is the responsibility of the testers to expose the exception bugs. Testing IEH has always been a difficult task and it remains a typical challenge mainly because of two reasons. Firstly, reaching the path

that might raise an exception and secondly, generating the required exceptions. The required exceptions can be generated by the program input parameters or abnormality of external resources including intermittent failures.

Previous works have mainly focused on testing IEH due to program input parameters. Literature survey reveals that very few researchers worked on testing IEH due to abnormality of external resources and yet, with certain limitations. This thesis focuses on testing of IEH to detect program crash, resource leakage and program inconsistency due to abnormality of external resources including intermittent failures. Moreover, it is also revealed that the individual security issues of IEH has been the focused of researchers till now. In this study, an attempt has been made to tackle almost all the security issues of IEH in totality through a single method.

A systematic methodology has been followed in the study to carry out the research. First of all, a systematic review and critical analysis of the pertinent literature on `Security Testing of Object Oriented Code' has been performed, following the guidelines by Kitchenham and Charters. The review includes critical study of security vulnerabilities, along with their prominent reasons. Moreover, the critical comparison on security testing techniques and on security testing methods/tools has also been performed and summarized in tabular form. This has prepared a strong background or theoretical basis on the domain of DoS vulnerability detection due to improper exception handling. Finally two frameworks `Symexc' and `Expatt' have been developed for this purpose and validated on java packages.

The framework `Symexc' detects the improper exception handling bugs such as program crash (application DoS), resource leakage and program inconsistency due to environmental failures. Symexc handles almost all the security issues of IEH through a single method. This is implemented in java and validated on java packages. The result shows that program crash and resource leakage are serious concerns for the applications that interact with external resources.

The second framework `Expatt' is an extension of Symexc with respect to testing of program crash, with the focus on intermittent failures of external resources. Exaptt has also been implemented in java and validated on java packages. The result shows that intermittent failures of external resources lead to several instances of program crash. In addition, during the course of study, a phase embedded security testing (PEST) framework is also proposed to carry out security testing in each phase of secure software development life cycle (SSDLC).

The work is very much significant in current scenario where program crash is a very serious concern. The work provides an insightful in code level security concerns, their related security testing techniques/methods and proposed two working security testing frameworks. This may lead to multifaceted benefits for security practitioners and researchers. In order to detect application DoS well in advance, both the frameworks may be incorporated by the security testers with their existing testing mechanisms. Although, the study focuses on security aspect of the code but the proposed methods can also be very much significant for dealing with the safety concerns of applications. The proposed methods may be modified after making some settings for testing the robustness of medical, smart vehicles, air carriers and other software applications to make them safer to use.

Based on the study, the Symexc and Expatt may be extended for handling program input parameters. In addition to these, researcher also want to explore hybrid search based security testing and parallel symbolic execution, which are recent trends in the area. However, an immediate extension may be the implementation and validation of PEST.