# Software Security Testing Process: Phased Approach

Suhel Ahmad Khan and Raees Ahmad Khan

Department of Information Technology
BabasahebBhimraoAmbedkar University (A Central University),
Lucknow-226025
ahmadsuhel28@gmail.com, khanraees@yahoo.com

**Abstract.** Early identification of defects and prevention of defects migration are key goals of the software security testing process. Early integration of security testing activities into the development lifecycle leads to secure software development. The prescribed key activities of security testing are closely interconnected with security development life cycle to deliver secure software. Software test process elaborates various testing activities and describes which activity is to be carried out when. Given the need and significance of phased approach of security testing, this paper proposes different testing activities to be carried out while integrating it within the security development life cycle.

**Keywords:** Software Security Testing, Security Test Life Cycle, Security Test Cases.

## 1 Introduction

Security testing is part of the analysis of security properties in the development systems. These security properties are verified relative to the functional specification, guidance documentation, and high level design of the system [1]. Software Security testing is a continuously growing field. Advance techniques and technologies are rapidly being developed in order to ensure continued security. The intruders or hackers targets on application level which can be easily lacerate security. The hackers exploit its vulnerabilities to obtain sensitive information to take control of the system. The possible disruption to ongoing behavior can be control through security testing.

Security testing is regarded as an important means to improve security of software. Security testing with a structured approach throughout the entire development life cycle gives a good understanding of the software quality and protects from known security risks[2]. If security testing is not considered during development, the application can contain dangerous vulnerabilities and sufficient vulnerability propagation will occur throughout the development phases making enormous risks to the organization. The security development life cycle process positioning as a separate process that has a strict interconnection with security testing activities to perform secure development actions.

A. Agrawal et al. (Eds.): IITM 2013, CCIS 276, pp. 211–217, 2013.

## 2 Involved Activities in Perspective Framework of Software Security Testing

The motive of software security testing is that the software behavior would be fully operative under hostile conditions. The ultimate objectives of security testing are to validate the robustness and to prevent security vulnerabilities from ever entering the software. While addressing security testing, one of the major issues to consider is to discuss how the system under design deals with possible attacks at early design stage[3, 4]. An effective security testing should test the entire phases, rather than just implementation. A test process is needed to ensure that the designed system can protect asset from attacks with the help of mitigation. An appropriate and accurate security test activity implemented during development may make the software more profitable [5]. In addition to this, an effective and prescriptive process of security testing specifying very clear prioritized activities may be advantageous in different perspectives as follows:

A. Security Test Strategy & Security Test Plan
   - Preparation of high-level security test plan
   - Designing strategy document for various types of security testing
   - Analyzing various security testing approaches available
   - Selection of security test tools, test effort estimation and scheduling for optimal security testing
   - Planning of resources for complete security testing

B. Design Security Test Cases
   - Include statement of purpose, what is being tested
   - Elaborate methods, how it will be tested
   - Include step environment data
   - Write the basis security test cases and security test procedures
   - Consider actual functional and security flow

C. Execute Security Test Cases
   - Prioritize execution of security test cases, identify essential features that must be tested
   - Identify the risk or consequences of not testing some features
   - Run the security test cases, document test results and log defects for failed cases
   - Map defects to security test cases, rested the defect fixes
   - Design security test execution records containing overall results

D. Capture Security Test Results
   - Prepare security test log
   - Document security test incidents reports
   - Identify the events requiring further analysis
   - Prepare security test summary report

E. Captures Security Test Metrics
   - Identify base metrics for direct measures
   - Design calculated metrics for indirect measures
   - Metric calculation for software security testing

F. Qualitative Assessment
- Prepare a set of parameters
- Document qualitative reporting
- Make interpretation
- Give suggestive measures

G. Security Test Closure Report
- Prepare a checklist
- List out activities that must be performed after the project is closed
- Document traceability matrix
- Do defect trade analysis

## 3 Security Test Life Cycle Process Integration within the Development Process

In order to gain insight into the quality of software, an integrated approach of security testing is necessary to spot the vulnerabilities in each phase of development life cycle and to mitigate the same then and there to avoid its propagation. An integration of security testing within the development process will reduce the cost of damages and risks associated [6]. Security testing strategy for software product should be developed for each phase. Security testing aims at finding security vulnerabilities prior to making them available to end users. One of the fundamental objectives of security testing is to identify whether the security features of the software implementation are consistent with design.

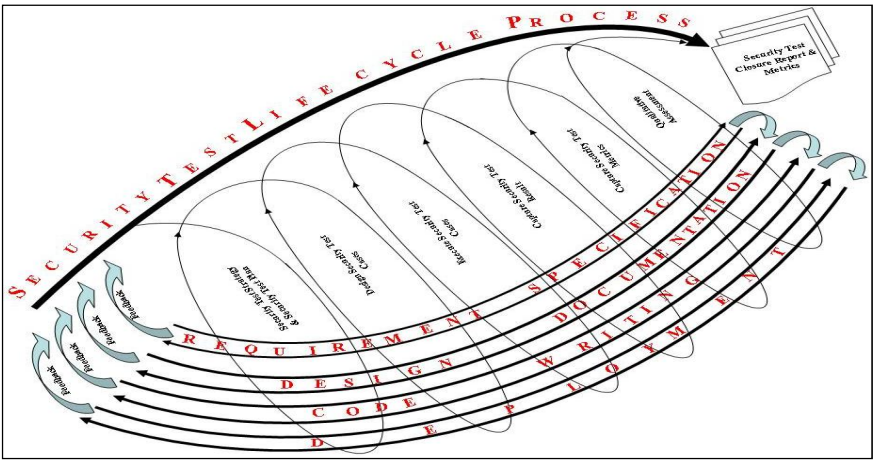


**Fig. 1.** (a): Security Test Life Cycle Process Integration within the Development Process

Figure 1(a) shows security test life cycle process stating as a separate process that has a very tight interconnection with development activities. Each phase is briefly described in the following section.

### 3.1 Security Test Strategy and Plan

An appropriate security test plan should be prepared enabling the mapping of security tests to secure requirements and defining the entry and exit criteria for each phase of testing. High level test plan should identify the items to be tested, the features to be tested, the type of testing to be performed, the personnel responsible for testing, the resources and schedule required to complete testing, and risks associated with the plan. A prescriptive step in preparing security test strategy and plan is depicted in figure 2.

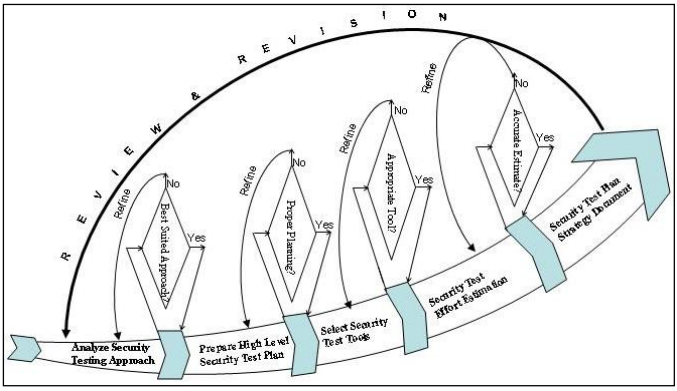


**Fig. 2.** Process of Security Test Strategy & Security Test Plan

### 3.2 Design Security Test Cases

This phase should perform creation, verification and rework of security test cases and test scripts. Test cases should be performed based on the four scenario including positive scenario, negative scenario, boundary conditions and real world scenario. During test case creation, correct and pre-requisites for conducting the security test cases should be configured. A prescriptive step in preparing security test design is depicted in figure 3.

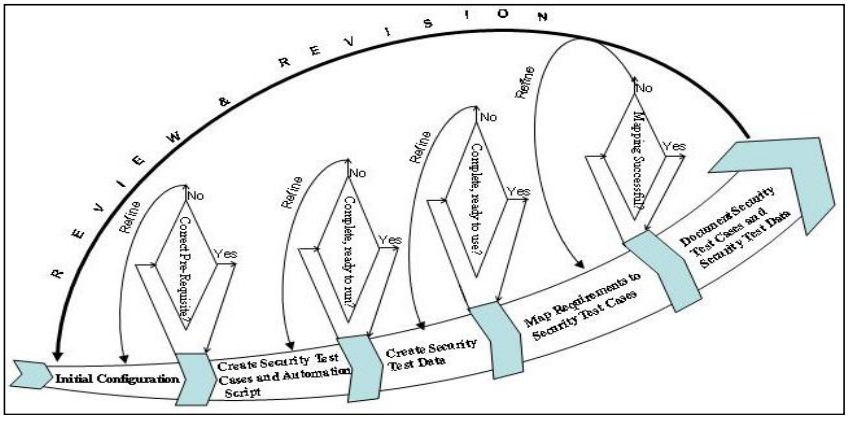


**Fig. 3.** Process of Designing Security Test Cases

### 3.3 Execute Security Test Cases

In this phase, security test scripts should be executed in a logical sequence with specific input data. The results should be monitored and output should be recorded in test sheets. Out of the larger cluster of test cases, their priorities of execution should be decided based on some rationale, non-arbitrary criteria. Test results should be documented and log should be prepared with defects for failed cases. Accordingly, security test plan and test cases should be reviewed and revised. A prescriptive step in executing security test cases is shown in figure 4.

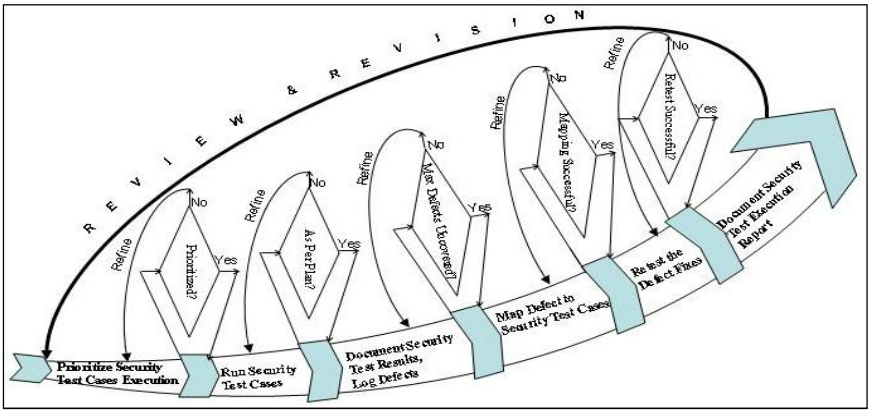


**Fig. 4.** Process of Executing Security Test Cases

### 3.4 Capture Security Test Result

After successful security test case execution, the identified bugs should be fixed and retested to declare the result as pass or fail. Test reports should be documented properly. A test log should be prepared to present chronological notes of relevant

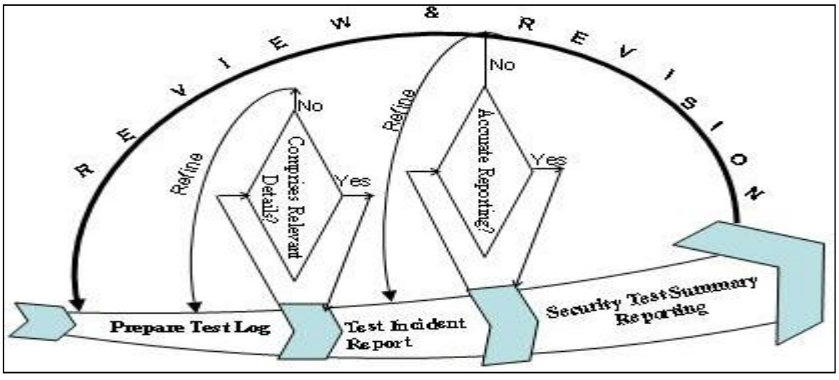


**Fig. 5.** Process of Capturing Security Test Result

details about the execution of tests. A test summary report should be prepared at last summarizing the results of test activities associated with one or more test design specification and provide evaluation based on such result. A prescriptive step in executing security test cases is shown in figure 5.

### 3.5 Capture Security Test Metrics

Security test metric should be identified which may aid in precise estimation of project effort, address the interest of metric group, software managers who are interested in estimating software security test effort and improve both development and security testing processes. Basic security test metrics should be used to provide project status reports to the test lead and project managers. Many of the basic metrics are simple counts that most test analyst already tracks in one form or other. Calculated security test metric converts the basic metric data into more useful information. A prescriptive step in capturing test metrics is depicted in figure 6.

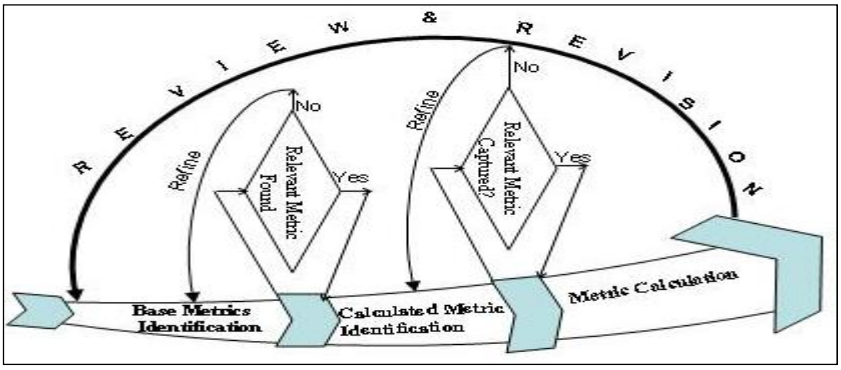


**Fig. 6.** Process of Capturing Security Test Metrics

### 3.6 Qualitative Assessment

Product quality measurement parameter is set forth in this phase in order to assess the quality of the product. A commonly accepted set of parameters is identified for which the assessment is to be made. On the basis of the quantitative results, a qualitative reporting of quality of the work products is made to the end user. Based on the qualitative assessment, interpretations are made and suggestive measures are proposed. A prescriptive step in qualitative assessment is depicted in figure 7.

### 3.7 Security Test Closure Report

Once the security test meets the exit criteria based on the time, test coverage, cost, software, critical business objectives, quality, the activities including capturing key outputs, lesson learned, results, logs, documents related to the project are archived

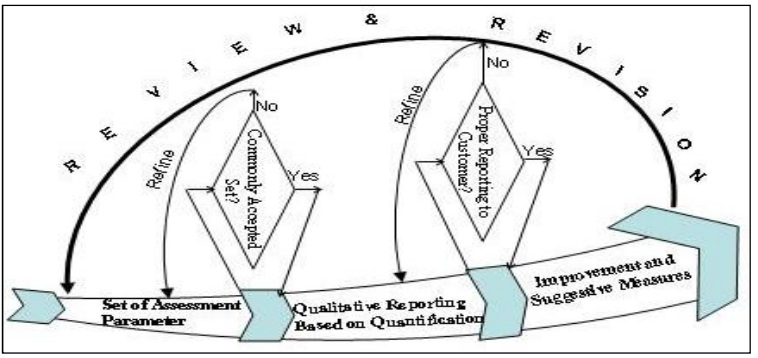


**Fig. 7.** Process of Qualitative Assessment

and used as a reference for future project. Test closure document should contain a checklist of all of the items that must be met in order to close a test project as well as a list of activities that must be performed after the project is closed.

## 4 Conclusion

It has become essential to integrate phased security testing process within the development life cycle with the intent of finding errors at each phase well in advance in order to reduce developmental cost, delivery time and rework efforts. The prescribed activities of software security testing process are interconnected with development activities. These security activities are verified through development process. The proposed work may help testers to better understand and execute test in an efficient and effective manner.

## References

1. Abrams, M.D.: FAA System Security Testing and Evaluation. MITRE Technical Report (May 2003)
2. Turpe, S.: Security Testing: Turning Practice into Theory. In: IEEE International Conference on Software Testing, Verification and Validation Workshop (ICSTW 2008). IEEE Computer Society (2008)
3. He, K., Feng, Z., Li, X.: An Attack Scenario Based Approach for Software Security Testing at Design Stage. In: 2008 International Symposium on Computer Science and Computational Technology, pp. 782–787. IEEE Computer Society (2008)
4. Gu, T.-Y., Shi, Y.-S., Fang, Y.-U.: Research on Software Security Testing. World Academy of Science, Engineering and Technology, 647–651 (2010)
5. Software Security Testing, Software Assurance Pocket Guide Series: Development, Volume III, Version 1.0 (May 21, 2012)
6. Potter, B., McGraw, G.: Software Security Testing. IEEE Security & Privacy, 32–36 (2004)