EVALUATING SCENARIOS BASED ON SECURITY REQUIREMENTS

[1]Khaled Musa, [2]Akram Abdel Qader

Alzaytoonah University of Jordan - Faculty of Science and Information Technology
Amman, Jordan
[1]Dr.khalid@zuj.edu.jo , [2]Akrama@zuj.edu.jo

Abstract

Software requirements in its different functional and non-functional methods are the derived force in producing an error free software systems. One of the main non-functional requirements is security, which focuses on producing secure systems from any intruders. Security and security testing is the mechanism to achieve secure systems. Security testing techniques reveals critical system threats. Security threats are made based on leaks on security scenarios. The purpose this paper is to extract and evaluate security scenarios out of rich story scenarios (RSS) based on security requirements.

Keywords: Software Requirements; Security Testing; Security Scenarios

1. INTRODUCTION

Software requirements are the acceptable benchmark characteristics that aim to developing good systems. Software requirements identify what developers must deliver to software users. System requirements are specified in form of use cases using natural language [1]. Security testing is the mechanism that is implemented on software systems (such as authorizations and users authentications) to verify the expected software behaviors.

Achieving security requirements to test clues to insecure behaviors and then explore potential vulnerabilities, and most of these vulnerabilities arise from unexpected interactions between different system components [1]. Security testing involves testing for vulnerabilities that arise from unexpected interactions between components [1].

To insure security testing, security test scenarios are created. Security test scenarios depict a set of acceptable system behaviors that show how these behaviors are shared among system components. This paper consists of four sec-

tions: Section 2, previous literature. Section 3, discuses the collection of software system requirements. Section 4, focuses on the generation of general rich story security scenarios based on software system requirements. Section 5, addresses the classification of software system security scenarios requirement. Section 6, discusses the extraction and evaluation to specific security scenarios. Section 7, is the conclusion of this paper.

2. PREVIOUS LITERATURE

There are several studies that are made on testing scenarios and security testing. Scenario based specifications such as Message Sequence Chart (MSC) and its Labeled Transition System Analyzer (LTSA) which checks for system behavior [4]. Implied scenario detection for security testing reveals unexpected interactions between components [1]. Out of a formal specification language, a Use Case scenario is declared by extended UML 2.0 sequence diagrams to derive a test model to assist test designers with test-specific information for later execution [2]. Model-based testing (MBT) and test case selection techniques using triggers, guards, and genetic algorithm-based selection are used to detect system real faults [3].

3. COLLECTION OF SECURITY SYSTEM REQUIREMENTS

Functional and non-functional software requirements describe what the system can do and how it should work at the system level. System functional specifications include the description of system processes that is to say the interaction between the user and the system as well as between subsystems [2] .Collecting system requirements are based on the investigation to system processes.

Descriptive software system requirements are the documented results of the intensive software analysis phase as they are collected through various techniques such as interviews, surveys, and sampling methods.

Software security requirement focuses on producing secure systems to illuminate intruders' access control to the system. Security testing is motivated by addressing undocumented assumptions and areas of particular complexity to determine how a program can be broken [5]

Software security testing is heavily addressed during the software analysis phase to verify that it behaves as expected [6] using a set of security test scenario cases that are extracted from security test cases [1].

Security testing scenarios should identify vulnerabilities that arise from unexpected interactions between system components [1], and identifies the behaviors and interactions of system components [12].

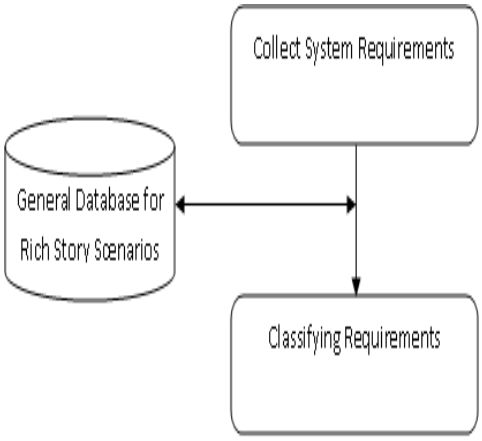
4. GENERAL SECURITY RICH STORY SCENARIOS (RSS)

This section deals with a creation of a new mechanism called Rich Story Scenario (RSS) database. Before exploring the RSS database, lets some information on security scenarios and scenario testing.

Security scenarios are derived from software system requirements where each scenario is a partial behavior between components, and these collections of scenarios are to exhibit more behaviors described by specified test case. Test cases are used to fulfill requirements [2], and such test cases illustrate test scenarios [2]. The generated security requirement should be covered by a minimal one test case, which in turn security scenarios are built manually based on the test cases of requirement specifications [1].

Test scenarios have been employed to identify components interactions [12]. Each scenario is a partial behavior between components having local views of the program execution [12]. The scenarios are to provide more expressive scenario language that allows for infinite number of system behaviors [13].

Requirement security specifications are fulfilled by the implemented security testing scenarios. Our proposed methodology is that to create a general database called Rich Story Scenarios (RSS) as a container for all test case scenarios (security testing scenarios for this matter). From requirements, security test scenarios are extracted and placed in the RSS database. In the RSS, security test cases are addressed through scenarios where each scenario addresses security elements of a system is classified based on certain category such as authorization or authentications figure 1, Generating RSS database.



`

Figure 1: Generate RSS database

5. SYSTEM SCENARIO REQUIREMENT CLASSIFICATION

Model-based testing (MBT) has several strategies for reducing the number of automatically generated test cases and test case scenarios [3]. MBT in its abstract test suite can eliminate any redundant test cases of the contained test cases based on, for example, code coverage criterion.

Another test case selection methodology is the context of regression testing, where test case selection is mostly done on regression testing where the goal is to find a subset of original test suite that assist the execution of fault revealing test cases [10][11]

In the proposed method, security test case scenarios are collected in a general scenarios database RSS; each security scenario is classified to specific category to allow several scenarios to fulfill more needed requirements. The categorized scenarios are built for security factors such as authentication, access control, or other factors. Each category should fulfill the focus of system administrator or user.

6. EXTRACTION SECURITY SCENARIOS

Building security scenarios for system tests to make sure that system operate based on system requirements. Extraction the scenarios that are needed to fulfill certain security category.

Scenario-based specification such as Message Sequence Charts (MSCs) diagram to specify system behavior loops between scenarios, where each scenario is categorized by a set of states and events [1].

Other test case selection, for scenarios to be evaluated, is done based on similarities between test paths using triggers and guards [3]. Fault measurements are done based on similarity-based test case selection techniques.

The extraction of security scenarios, in this proposed system, is to allow testers confirm whether security scenarios pertain or address the desired testing cases. If security scenarios do not fulfill requirements, more scenarios should be extracted from the RSS database. If security scenarios are justified to the system testers, each scenario will be evaluated based on security categories. For example, scenarios should address authorization and other scenarios should address authentications, and more evaluation based on categories, figure 2.

Following scenarios extractions, evaluation is done at this level to determine that security testing scenarios fulfill addressed categories such as authentication, access control, authorization, etc.

Test case scenarios can be evaluated based on MSC to provide more expressive scenario language that allows an infinite number of system behaviors [13]. The MSC and its hMSC graph illustrate the continuation and loops between scenarios.

Evaluation will assist testers in their testing through categorizing requirements. Scenario evaluation can be done based on requirement specification categories to determine whether categories such as authorization, authentication, denial of service, information corruption, etc are addressed, figure 2.

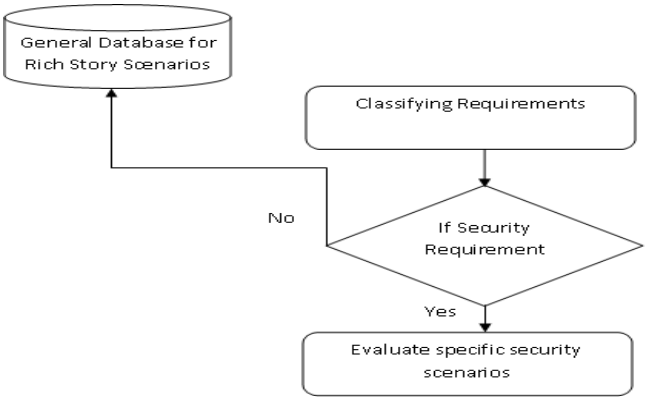


Figure 2: Extracting and Evaluating Scenarios

7. CONCLUSION

Software requirements non-functional requirements is security, which focuses on producing secure systems from any intruders. Security and security testing is the mechanism to achieve secure systems and follow testing techniques to reveal critical system threats. Security threats are made based on leaks on security scenarios. Requirements are collected and classified in an RSS database, which later is used to prompt testers with all possible scenarios testing so they start administer good testing. The security test scenarios that are built in the RSS database are classified to allow thorough type of testing. The applied scenarios are extracted and evaluated to determine the category that the examination that will take place, and that to determine that security requirement is fulfilled.

REFERENCES

[1] Al-Azzani, S., Bahsoon, R. Using Implied Scenarios In Security Testing. SESS'10, May 2 2010, Cape Town, South Africa. Copyright ACM 2010.

[2] Loffler, R., Meyer, M., Gottschalk, M. Formal Scenario-based Requirements Specification and Test Case Generation in Healthcare Applications. SEHC'10, May 3-4, 2010, cape Town, South Africa. Copyright ACM 2010.

[3] Hemmati, H., Briand, L., Arcuri, A., Ali, S. An Enhanced Test Case Selection Approach for Model-Based Testing: An Industrial Case Study. FSE-18, November 7-11, 2010, Santa Fe, New Mexico, USA. Copyright ACM 2010.

[4] Uchitel, S., Kramer, J., Magee, J. Detecting Implied Message Sequence Chart Specifications. ESEC/FSE 2001, Vienna, Austria. Copyright ACM 2001.

[5] M. Howard and D. C. LeBlanc. Writing Secure Code. Microsoft Press, Redond, WA, 2002.

[6] G. Wimmel and J. J¨urjens. Specifiction-based test generation for security-critical systems using muttions. In International Conference on Formal Engineering Methods (ICFEM), 2002.

[7] Object management group, unified modeling language (uml), version 2.2.

[8] R. Alur, K. Etessami, and M. Yannakakis. Inference of message sequence charts. In ICSE '00: Proceedings of the 22nd international conference on Software engineering, USA, 2000. ACM.

[9] S. Uchitel, J. Kramer, and J. Magee. Detecting implied scenarios in message sequence chart specifications. SIGSOFT Softw. Eng. Notes, 26, 2001.

[10] Binder, R. V., *Testing Object-Oriented Systems: Models, Patterns, and Tools*, Addison-Wesley Professional, 1999.

[11] Mathur, A. P., *Foundations of Software Testing*, Addison- Wesley Professional, 2008.

[12] M. A. Babar and I. Gorton. Comparson of scenario-based software architecture evaluation methods. Asia Pacific Software Engineering Conference, 2004.

[13] F. C. de Sousa, N. Mendon, S. Uchitel, and J. Kramer. Detecting implied scenarios from execution traces. Reverse Engineering, Working Conference on, 2007.