Critical Review on Software Testing: Security Perspective

Mohd Waris Khan[1(✉)], Dhirendra Pandey[1], and Suhel Ahmad Khan[2]

[1] Department of Information Technology, BBAU, Lucknow, India
Wariskhan070@gmail.com, prof.dhiren@gmail.com
[2] Department of Computer Application, Integral University, Lucknow, India
ahmadsuhel28@gmail.com

Abstract. Software plays a crucial role in day to day life; hence its security and reliability cannot be neglected. Creating a secure software system is not just to secure sensitive and confidential information but it needed to establish a system which could stand true on the benchmark set for being a secure software system and further derive a roadmap to construct impregnable and efficient software. In order to fulfill this criterion, security testing is vital for the development of a secure software system as it pursue all the aspects of SDLC. Security should form an integral part of a SDLC, hence to maximize and maintain the defenses of a software system and to keep its development cost in limits, Security Testing Profile (STP) provides a reliable platform for testing software. STP is an uncharted territory and more progress can be made in this area, which may help in developing robust software systems.

Keywords: Security testing · Security factors · Security Testing Profile (STP)

1 Introduction

The purpose of security testing is to find out whether the system meets its specified security objectives, or security requirements. The various phases of security testing during the software lifecycle, starts from requirements elicitation and analysis, design phase, implementation and verification & validation. Figure 1 describes a technical view for security test plan specification. The test plan specifications of software security testing can be classified into four parts, which are as follows: Technical (C1), Strategic (C2), Environmental (C3) and Operational (C4). After classification of these aspects of security, the role of STP in this research paper has been observed and discussed. Furthermore, the importance of STP is defined and its future scope is explained. The test results obtained by these four parameters will provide the background for creating STP. Security Testing Profile will be a sum total of all test results based on the above four parameters.

In this study, we focus on security testing in software development, from a tester's perspective. The main objective is to provide an insight in the use of security testing from the starting point in the SDLC for the sole purpose of creation of secure and robust software. For that, in this paper, we discuss about the security testing profile which will help in the evaluation of software in terms of security. This introduces the

A. Unal et al. (Eds.): SmartCom 2016, CCIS 628, pp. 714–723, 2016.
DOI: 10.1007/978-981-10-3433-6_86

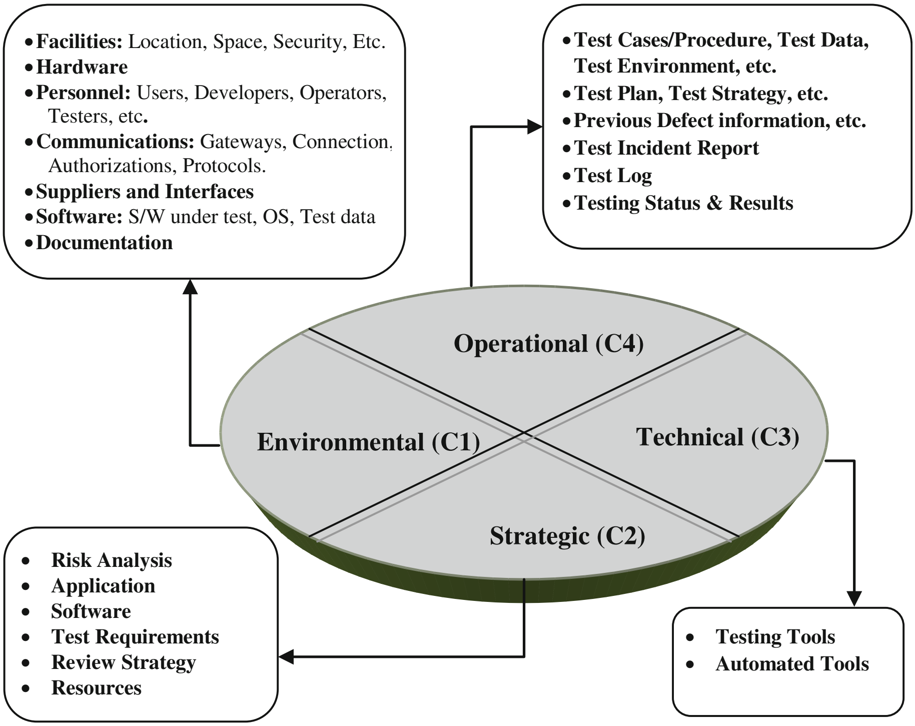

• Facilities: Location, Space, Security, Etc.
• Hardware
• Personnel: Users, Developers, Operators, Testers, etc.
• Communications: Gateways, Connection, Authorizations, Protocols.
• Suppliers and Interfaces
• Software: S/W under test, OS, Test data
• Documentation
• Test Cases/Procedure, Test Data, Test Environment, etc.
• Test Plan, Test Strategy, etc.
• Previous Defect information, etc.
• Test Incident Report
• Test Log
• Testing Status & Results
Operational (C4)
Environmental (C1)
Technical (C3)
Strategic (C2)
• Risk Analysis
• Application
• Software
• Test Requirements
• Review Strategy
• Resources
• Testing Tools
• Automated Tools

Fig. 1. Security test plan specification

basic terminology and concepts regarding security testing profile. We will also examine where and how the security testing profile is useful in software development process.

Additionally, in this article we also discuss how important is testing profiling from security point of view. Which brings us to the uses of security testing profile like it is vital for finding loopholes, for zeroing in on vulnerabilities, for identifying design insecurities, for identifying implementation and dependency in securities and for organization wide software security. For the purpose of reliability testing as well as security testing, profile based testing is a resourceful method. The development of operational profile and security intrusion profile of software are both resource consuming, this is why separate testing for both is required to validate the dependability of the software.

This paper has been divided into 7 sections, starting with an introduction in Sect. 1. Section 2, describes the background and related work presented by researchers. Section 3, defines the Classification of test plan specification. In Sect. 4, detailed description of Software Security Testing Profile is discussed. Section 5, describes the STP with respect to design perspective. Importance of Security Testing Profile is defined in Sect. 6 and conclusion of the paper is discussed in last section i.e., Sect. 7.

2 Background and Related Work

In 2015, the author S. Krishnaveni et al., paper entitled on "Analysis of Software Security Testing Technique in Cloud Computing" was presented, in which issues and challenges to cloud security testing were addressed [1]. Cloud computing is a new field and since there is no such clear methodology to follow in order to complete the cloud security testing, it makes cloud computing vulnerable to attacks and the bulk of information and confidential data is at risk until and proper methodology is developed. This paper focus on the C1 (environmental) and C2 (strategic) aspects as it targets both the environment where the application is set to use and the strategy related to the mitigation of problems and counter mechanism to face threats.

In 2014, the Author R. Kumar et al., have written the paper on "Software Security Testing: A Pertinent Framework", in this paper a mathematical formula is proposed for preparing test cases during any security development life cycle [2]. The paper presents a life cycle for software security and also enlightens the major methods, tools and technique of software security. It signifies the criteria of C4 (operational) specification.

In 2013, Suhel A. Khan et al., presented a paper entitled "Software Security Testing Process: Phased Approach" illustrated the importance of early identification of defects and its mitigation through the software security testing process [3]. It also emphasized on the early incorporation of security testing activities and prescribed the key activities of security testing that are interconnected with security development lifecycle. The paper signifies the various activities that can be classified under C3 and C4 specifications of security test plan.

In the year 2012, German authors Ina Schieferdecker et al., have written the paper on "Model-Based Security Testing" which intend to validate software requirements associated with security properties like confidentiality, integrity, authentication, authorization, availability and non-repudiation [4]. Model based security testing (MBST) is a somewhat new field and focus mainly on the C2 and C4 test plan specification. The paper signifies the requirement of an initial survey on Model based security testing. It is prepared by analyzing the related work and confers the model that can be used for model security testing by outlining the two main approaches namely – Risk-based security and Model-based fuzzing.

In 2011, a paper titled "An Overview of Penetration Testing" published in which the author Aileen G. Bacudio et al., have discussed the benefits, strategies and the methodologies of conducting penetration testing [5]. The methodology of penetration testing consists of three stages: test preparation, test and test analysis. The test stage deals with information gathering, vulnerability analysis and vulnerability exploit. Security is a major concern regarding information systems. This paper explains the need and ways to enhance the defense of software against getting penetrated by any malicious software or internet virus. Focusing on the C1 (Environmental) and C2 (strategic) specifications, the paper describes that because of growing connectivity of computers through the internet the massive rise in the sharing of information, the uncontrolled growth in the size and complexity of systems.

3 Classification of Test Plan Specification

These research papers give us a detail account on the progress, methods and use of various tools and related information regarding current trends in the field of software security testing. After reviewing these research papers, it can be concluded that a lot of research is being performed to enhance the quality of work and to achieve state-of-the-art software security and reliability parameters by the use of newer strategies and testing software systems in various test environments. The test plan specifications most targeted in the above research papers are C2 and C1 respectively as it is imperative to integrate security within the basic software design from the beginning of SDLC. Development of new testing tools i.e., C1 specification is another important issue that has been duly addressed in some papers, as also the development of operational quality of a software security and dependability has been given impetus classified under C4 test specifications (Table 1).

Table 1. Classification of test plan specification

Environmental specification (C1)	Test environment is a stable area for independent systems and integration testing by the test team [9] Test environment includes the physical characteristics of the facilities like hardware, communications and system software, mode of usage/interface and other software or supplies [10] Documentation the physical components required for test execution is also a part of environmental specification [16]
Strategic specifications (C2)	It is the heart of the test plan, contains a descriptive guide of testing procedures to be performed and explain issues that have major impact on the success of the test [11] It describes the overall approach and techniques involved in each phase during test It describes phase wise testing approach Outline the steps involved in overall testing process [12] Identification of all software features, combination of software features to be tested and the associated design specifications [9, 13] It includes test requirements like completeness, accuracy, stability etc. Includes review strategies - walkthroughs, inspection, desk checks etc. Completion criteria for overall test plan will be set in advance Preparation of a checklist at each phase to know whether the testing phase is complete or not [14]
Technical specifications (C3)	Selection of appropriate tools and automation [6, 8, 11]
Operational specifications (C4)	Inputs: Test case/procedures, Test data, Test environment [6, 7] Test Plan, Test Strategy Previous defect information etc. [10, 12] Outputs: Test Incident report, test log, testing status and results [13, 15]

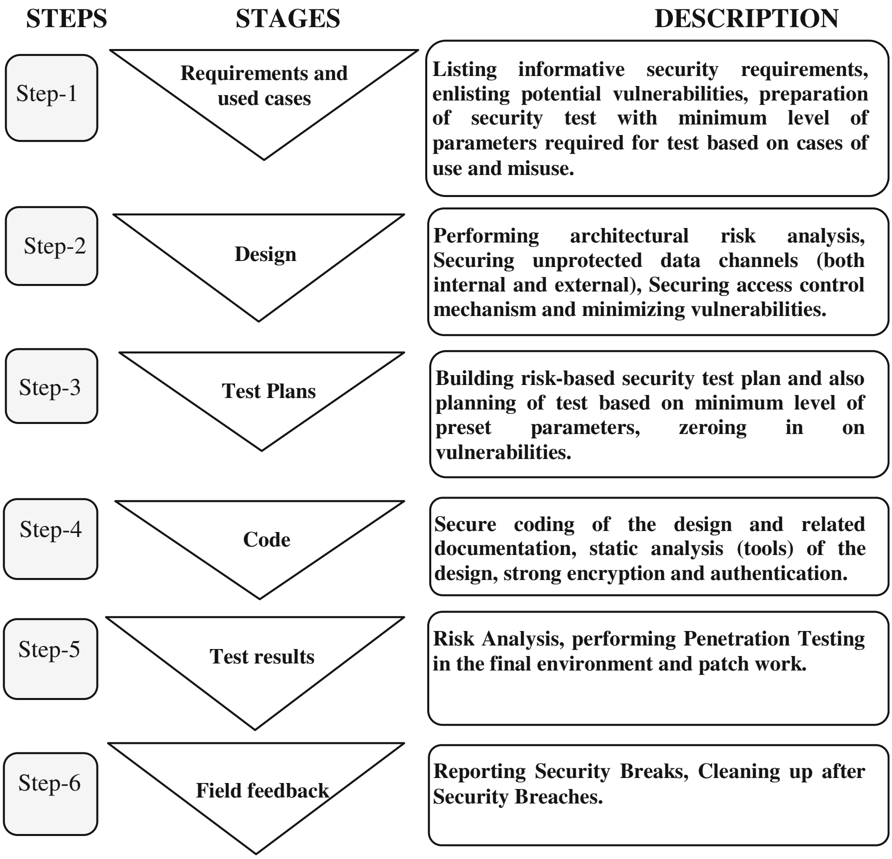
By reviewing these papers, we get a better understanding of various conceivable vulnerabilities and potential security threats in a software system, and also we could study the various methodologies used by the researchers to encounter, endure, contain and fix those problems. The first and foremost research problem is to gather and understand security requirements and constraints. It has become a matter of utmost necessity to integrate security profile testing in the lifecycle of software. The aim is to embed security profile testing into the entire software development life cycle as to achieve the objective of 'defense in depth' concept which focus on integrating and implementing security testing profile to provide a security at different layers. We have presented the importance of the security testing profile wherein the purpose of identification, documentation and setting optimal parameters for a security test of a software system could be realized. A good documentation of security test helps to reduce the system vulnerabilities, security threats and testing vulnerabilities, as also it will help in patching up the security loopholes in a software system.

4 Software Security Testing Profile

Security testing profile is an approach of integrating security requirements, risk-analysis, design, implementation and testing into development life cycle stages or in other words, the structured process of identifying and documenting all optimal security testing is called security testing profile. Profiling of security testing and its use to design a software with minimum loopholes and design faults is guided by the operational profile of the program under testing to ensure that the most used operations receives the most testing. This way by a continuous or repetitive testing most of the nags, latent security breach areas, vulnerabilities to a certain environment and threats by malicious bugs can be detected and mitigated.

Security testing profile in an internet environment can help providing protection to an enormous amount of important and confidential information shared on the internet. It also provides an organization wide data protection and accurate identification of causes of the security leakage. Number of test in a given environment also affects the reliability of the software components. In order to assess the performance of software keeping the factor of reliability in mind, testing profile needs to be evaluated accurately.

The performance of a test during the process of executing a program with an intention of finding design errors in a given environment precisely will define the success of such test. The risk analysis on the basis of these tests will help in plugging the loopholes and developing a framework to fix the program errors and design faults. It goes without saying that you cannot build a secure application without performing sharp security testing on it. It has been defined that a program is correct if it needs its specification for all valid inputs. During testing phase, test cases are selected according to operational profile/testing profile and applied to the software under test. The various stages of the security testing life cycle are shown in the following Fig. 2.



STEPS	STAGES	DESCRIPTION
Step-1	Requirements and used cases	Listing informative security requirements, enlisting potential vulnerabilities, preparation of security test with minimum level of parameters required for test based on cases of use and misuse.
Step-2	Design	Performing architectural risk analysis, Securing unprotected data channels (both internal and external), Securing access control mechanism and minimizing vulnerabilities.
Step-3	Test Plans	Building risk-based security test plan and also planning of test based on minimum level of preset parameters, zeroing in on vulnerabilities.
Step-4	Code	Secure coding of the design and related documentation, static analysis (tools) of the design, strong encryption and authentication.
Step-5	Test results	Risk Analysis, performing Penetration Testing in the final environment and patch work.
Step-6	Field feedback	Reporting Security Breaks, Cleaning up after Security Breaches.

Fig. 2. Profiling of testing in security development life cycle

5 Software Testing Profile: A Design Security Perspective

Software security is to engineer software in such a way that the required application functions uninterrupted and is able to adequately neutralize all the security threats during malicious attacks. In general practice, security is left out in early stages of the software development life cycle (SDLC), rather a software engineering approach must have adequate security criteria from the beginning of SDLC. Inadequate security practice in the software development process can lead to the creation of insecure software. Integrating the security testing profile in the early stages of design and development, working on the behavioral aspect of the software with respect to security in case of malicious attacks and minimizing the potent vulnerabilities in the design phase will ultimately help us in realizing the goal of achieving reliable and secure software. The steps of secure testing profile specification are explained in the Fig. 3 which is as follows:

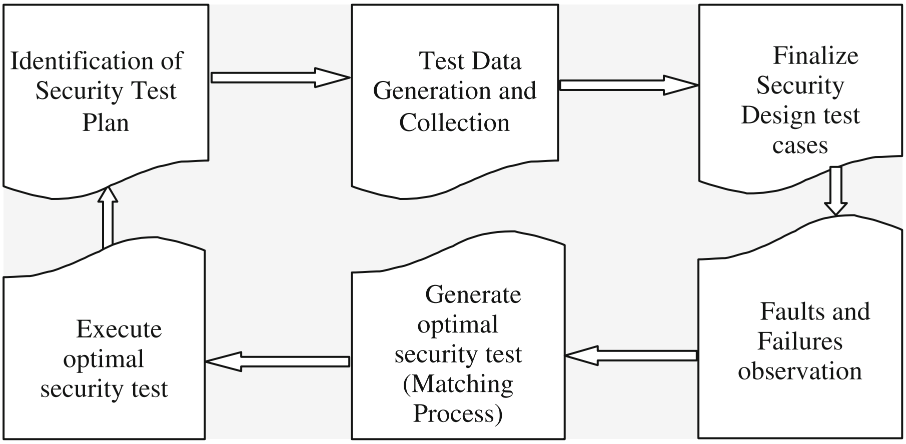

Identification of Security Test Plan
Test Data Generation and Collection
Finalize Security Design test cases
Faults and Failures observation
Generate optimal security test (Matching Process)
Execute optimal security test

Fig. 3. Steps of secure testing profile specification

5.1 Identification of Security Test Plan

In this process problem areas should be identified and program targeted for its testing and mitigation will be prepared. The test plan should also be prepared keeping in mind the original objectives and prepare a risk based test plan should be prepared and implemented.

5.2 Test Data Generation and Collection

It is a crucial part of software testing which describes the process of creating a set of data for testing the reliability and adequacy of the new software application. The main aim of data collection is to reduce the time wasted by manually search for or creating test data. It also eliminated the risk of data breach and improves the quality of testing by early detection of defects and design faults in the original software application. This will also identify the vulnerabilities in the original software design, reduce the infrastructure cost, create missing data and generate large volumes of realistic data.

5.3 Finalize Security Design Test Cases

In this stage a comprehensive security test strategy for software security test cases is developed, focused on the security components of threats, exposures, risks to the assets and controls.

5.4 Fault and Failures Observation

The goal to observe faults and failures is to minimize the efforts of software vendors as well as security researchers so as to efficiently discover and evaluate security vulnerability. The benefit of this process is that it helps in localization of faults that lead to the categorization of individual software failures and distribution of different types of software faults.

5.5 Generate Optimal Security Test (Matching Process)

Software testing has been critical part of the entire software development lifecycle where success of software projects is heavily dependent. Therefore, there has been a widespread effort to improve SDLC practices especially during testing. In this process cases can be prioritized by considering various domains for which software is being developed and conduct optimal security test and validation of the final software that should match with the user's requirements and applicability. The actual goal of this task is to generate optimal security test plan which combine all sorts of security testing and validation processes in single compact form and execute them to produce a final improved, more user requirement specific and secure software system.

5.6 Execute Optimal Security Test

In this phase execution of all the prepared and approved test cases using recommended tools and techniques in the final stage is implemented. The final execution consists of the regression test for software security fixes, execution of new software security test and documentation of all software security defects of overall result.

6 Importance of Security Testing Profile (STP)

In order to estimate correct reliability of software, security testing profile needs to be evaluated accurately. A number of tests affect the reliability of software components with respect to security viewpoint such as accurate identification of test density over lifetime which is essential to estimate correct reliability of the software system. STP may provide the platform that how to select the minimum set of test case adequately, effective for revealing faults in a program. Security testing profile provides a platform for finding loopholes present in the system/software. It may also identify faults in the design phase which helps us zeroing in on vulnerabilities. By identifying dependency insecurities and failures, we could create a better action plan to deal with the problems regarding the implementation insecurities of the software in a given environment. Information security is a one of the major concerns as these days there is huge amount of transfer of information over the internet every second, most of which is unsecured or less secured which creates big proximity areas from where confidential and important information could be stolen. A small bug in software can create a breach in the system which could then be exploited by hackers to siphon off confidential information.

Most of the organizations take their own security systems and countermeasures to secure their important information and process confidential information using software systems via internet. But, it is seen that even after that organization wide security of information is hacked into and stolen. This leads to our original viewpoint that how much security testing profile is essential for developing any software from the beginning of its life cycle. Security testing profile should be embedded from the very start of the preparations and designing, development and releasing a software system so as to minimize the risks and/or eradicate any insecurities or potential vulnerabilities of software system. Figure 4 shows the integration of STP into each stages of SDLC.

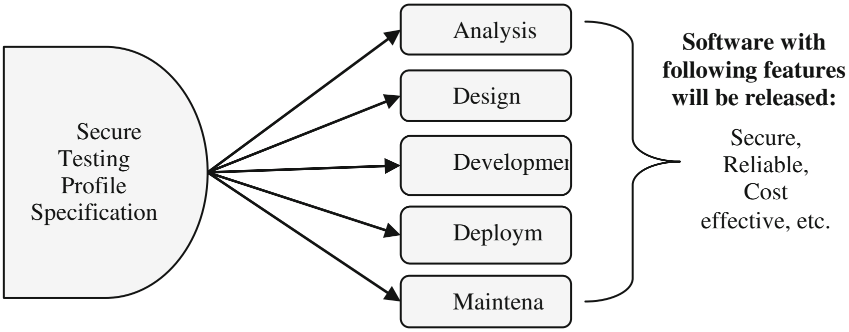

Secure Testing Profile Specification
Analysis
Design
Developmen
Deploym
Maintena
Software with following features will be released:
Secure, Reliable, Cost effective, etc.

Fig. 4. Secure testing profile specification on SDLC

7 Conclusion

This paper outlines the basic problems and constraints that affect the security and reliability of software system. The purpose of this paper is to suggest a new way to prepare the security testing profile and to integrate it in the software development life cycle. Further the paper discusses research problems focused on security testing, understanding the causes of vulnerabilities which will enable us to target our testing on the root cause of those problems. The security requirements that developers have to follow should carefully be documented and future tests of software system shall be based on the data acquired from these documentations. Future work will concentrate on developing a framework for integrating security requirements for STP, its documentation and its implementation into the life cycle stages to achieve maximum security standards at all the layers of software development. A continuous preeminence on the improvement of the process and defect-reduction avoids the degeneration or process statement and it will further the process of improvement in productivity, reduced defect leakage and greater reliability and timeliness of a software system.

References

1. Krishnaveni, S., Prabakaran, Sivamohan, S.: Analysis of software security testing technique in cloud computing. Int. J. Mod. Trends Eng. Res. (IJMTER) **2**(1), 417–424 (2015)
2. Kumar, R., Khan, S.A., Khan, R.A.: Software security testing: a pertinent framework. J. Global Res. Comput. Sci. (JGRCS) **5**(3), 23–27 (2014)

3. Khan, S.A., Khan, R.A.: Software security testing process: phased approach. In: Agrawal, A., Tripathi, R.C., Do, E.Y.-L., Tiwari, M.D. (eds.) IITM 2013. CCIS, vol. 276, pp. 211–217. Springer, Heidelberg (2013). doi:10.1007/978-3-642-37463-0_19
4. Schieferdecker, I., Grossmann, J., Schneider, M.: Model-based security testing. In: Workshop on Model-Based Testing, EPTCS, vol. 80, pp. 1–12 (2012)
5. Bacudio, A.G., Yuan, X., Chu, B.-T.B., Jones, M.: An overview of penetration testing. Int. J. Netw. Secur. Appl. (IJNSA) **3**, 19–38 (2011)
6. Khan, S.A., Khan, R.A.: Software security testing process. In: Proceeding of the International Conference on Recent Trends in Computing and Communication Engineering (RTCCE), pp. 38–42 (2013)
7. Cao, P., Dong, Z., Liu, K., Cai, K.-Y.: Robust dynamic selection of tested modules in software testing for maximizing delivered reliability. (cs.SE), pp. 1–16 (2013)
8. Tian-yang, G., Yin-sheng, S., You-yuan, F.: Research on software security testing. Int. J. Comput. Electr. Autom. Control Inf. Eng. **4**, 1446–1450 (2010)
9. Mao, C.: Experiences in security testing for web-based applications. School of Software, Jiangxi University of Finance and Economics, 330013 China, ICIS 2009, 24–26 November, Seoul, Korea (2009)
10. Zhang, D., Nie, C., Xu, B.: A Markov decision approach to optimize testing profile in software testing. In: 9th International Conference for Young Scientists. IEEE, pp. 1205–1209 (2008)
11. Türpe, S.: Security testing: turning practice into theory. In: First ICST Workshop on Security Testing. IEEE Computer Society, pp. 295–302 (2008)
12. Ma, C., Zhao, J., Gu, G., Ma, X.: Research on software dependability testing profile in internet environment. IEEE Computer Society, pp. 206–209 (2008)
13. Flechais, I., Mascolo, C., Sasse, M.A.: Integrating security and usability into the requirement and design process. Int. J. Electron. Secur. Digit. Forensics **1**(1), 12–26 (2007)
14. Potter, B., McGraw, G.: Software security testing. IEEE Computer Society, IEEE Security & Privacy, pp. 32–36 (2004)
15. Gilliam, D.P., Wolfe, T.L., Sherif, J.S., Bishop, M.: Software security checklist for the software life cycle. In: Twelfth IEEE International Workshops, Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE, pp. 243–248 (2003)
16. Wenliang, D., Mathur, A.P.: Testing for software vulnerability using environment perturbation. Qual. Reliab. Eng. Int. **18**(3), 261–272 (2002). Special Issue on Computer Security