A Categorized Review on Software Security Testing

Neha Mahendra
Department of Computer Application
Integral University, Lucknow, India

Suhel Ahmad Khan
Department of Computer Application
Integral University, Lucknow, India

ABSTRACT
The main objective of security testing is to check the weaknesses of the implemented security mechanism. It is done for finding the vulnerabilities of a system and to determine whether the system is protected from intruders or not. Security testing can be done prior to production or after the production of the system. But, if the security testing is done after the production, then cost will be more and the huge amount of rework will be required to remove the problems. Also the time between the vulnerability is get known and the malicious attack against it, is becoming less. Therefore it is required to include the security testing in the early phases of software development life cycle. The present paper deals with the review of software security testing approaches and techniques proposed so far. The review is presented in a categorized way and tabulated for the last one and half decade (2000-2015).

Keywords
Security testing, software development life cycle, SDLC phase

1. INTRODUCTION
Security is a way of protecting an application against actions that cause it to stop functioning or being exploited. Actions can be either intentional or unintentional. Intentional actions comprise the planned attacks by hackers that harm the system. Unintentional actions are the errors that get the system in an undesirable state. Security of a system is affected by the software, middleware, hardware, communication networks, client and end users involved. The motive of security testing is to find out the possible threats in the system and determine its potential vulnerabilities. Normally, security testing has the attributes: Confidentiality, Integrity, Resilience, Availability, Authentication, Authorization and Non-repudiation [1]. Security testing is must to deal for avoiding the disturbance to the online means of revenue, website downtime and the loss of customer trust. In today's competitive market everything is available but the product which gives the best security can only beat the market. An effective security testing of a system will greatly affect the industry as well as the academics.

1.1 Security Testing
Security testing must be performed in time before a breach harms the system. Consequently, the time loss and expenditures in recovering from damage is reduced. Therefore, in order to implement security testing properly there is a need of a systematic process. A good security testing should incorporate the proper training for all developers, designing threat models for the overall system, regular code reviews and penetration testing. There are seven main types of security testing as given in open source security testing methodology manual- Vulnerability Scanning, Security Scanning, Risk Assessment, Penetration Testing, Security auditing, Posture assessment and ethical hacking [2].

1.2 Incorporating Security Testing in Software Development Life Cycle (SDLC)
The software development life cycle (SDLC) provides a framework which defines various tasks performed at each and every step of the software development process. It describes how to develop and maintain software. It basically consists of six phases- Define (Requirement analysis), Design, Develop (Coding), Test, Deploy (Implementation), Support (Maintenance) (see Table-1). Security testing must be done as a continuous process with SDLC, especially in earlier phases.

Table-1: Phases of SDLC

Phases Of SDLC	Security Processes
Define	Security analysis for requirements.
Design	Security risk analysis for designing and development of security test plan
Develop	Static / Dynamic Testing and white box testing
Test	Black Box Testing, Vulnerability scanning
Deploy	Vulnerability Scanning, Penetration Testing
Support	Impact analysis

2. LITERATURE REVIEW
The various security testing approaches are proposed so far in which some have been reviewed here. The security testing is considered as a continuous process throughout the SDLC. The Security Development Lifecycle (SDL) is given by the Microsoft [3, 4] for including testing of security in the Software Development Life Cycle. The Secure Software Development Lifecycle (SSDL) given by Wysopal [5] and the security touchpoints for a SDLC given by McGraw [6] are proposed for the same purpose. Software security testing can be upgraded with the help of security attributes, tools, models and most importantly test case used in testing [7].

The review is presented in a categorized manner as follows (depicted in Figure-1):

1. Frameworks
2. Techniques
3. Methodologies
4. Reviews

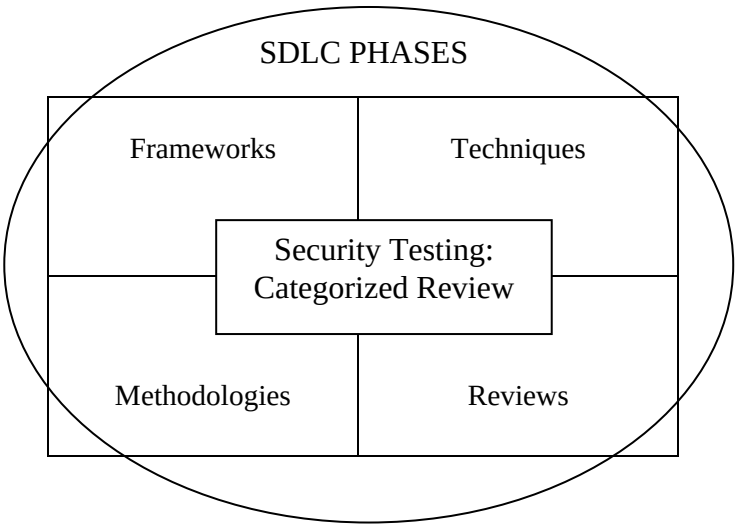

Figure-1: Categorized Review

Different frameworks/methodologies used are applied in different phases of SDLC. Therefore, while reviewing papers the phase of SDLC in which the particular worked is also analyzed and summarized in tables. The year wise tabulation is given in chronological order.

2.1 Frameworks

Framework is a structure-in-support to perform the task. It describes the environment for doing the task. Some of the frameworks are reviewed and presented in Table-2.

Table 2: Frameworks Used in Security Testing

S. N.	Year	Author	Framework	SDLC Phase
1.	2002	K. Jiwnani and M. Zelkowitz [8]	Three Dimensional Classification of vulnerabilities (based on Landwehr's classification)	Deploy Phase
2.	2004	J. A. Wang[9]	Relatively Complete Coverage (RCC) Principle	Develop and Test Phase
3.	2004	Bruce Potter, Gary Mcgraw [6]	Risk Based Approach	All phases
4.	2004	S. Lipner [4]	Security Development Lifecycle (SDL)	All phases
5.	2007	K. Karppinen, R. Savola, M. Rapeli and E. Tikkala [10]	Iterative Security Evaluation Process	Test and Deploy Phase
6.	2008	I. A. Tondel, M. G. Jaatun and J. Jensen [11]	Vulnerability Study	All phases
7.	2010	R. Hassan, M. Eltoweissy, S. Bohner and S. El-Kassas [12]	Formal Analysis And Design For Engineering Security (FADES)	All phases
8.	2012	C. Rudolph and A. Fuchs[13]	Inter-disciplinary approach	All phases
9.	2013	S.A.Khan and R.A.Khan [14]	Phased approach	All phases
10.	2013	Suhel Ahmad Khan, Raees Ahmad Khan [15]	Integrated approach	All phases

In 2002, K. Jiwnani and M. Zelkowitz [8] proposed a security testing strategy based on the three dimensional classification (based on Landwehr's classification) of vulnerabilities with their genesis, location and impact. This classification scheme fixes flaws in the early stages of the development cycle and helps to derive security metrics for testing. They applied the taxonomy on a file of 1200 vulnerabilities found in Windows NT from Harris Corporation Rand 160 in Linux compiled from Red Hat Linux Errata.

In 2004, J. A. Wang [9] suggested a relatively complete coverage (RCC) principle and approach for generating and conducting destructive security test sets and stated that security testing phase should be added to software development process. A Component Based Development (CBD) is profitable from the security perspective. Further he stated that complexity is the main source of errors which lead to security vulnerabilities.

In 2004, Bruce Potter, Gary Mcgraw [6] used a risk-based approach to software security testing. They stated that non functional security testing is important. Security problems can be solved production of the software.

In 2004, S. Lipner [4] proposed the security development lifecycle (SDL) which have many sub-processes distributed across all phases of SDLC. Threat modeling is described as the highest priority component of SDL. Further gives explanation with its implementation across a range of Microsoft software.

In 2007, K. Karppinen, R. Savola, M. Rapeli and E. Tikkala [10] discussed the security evaluation process and found that there is a need of iterative process based on risk, threat and vulnerability. They performed a case study on security testing project of Technical Research Centre of Finland.

In 2008, I. A. Tondel, M. G. Jaatun and J. Jensen [11] presented a review of security testing and proposed a software security testing scheme based on vulnerabilities inside the organization and used the output of one application as the input to the next application to be tested.

In 2010, R. Hassan, M. Eltoweissy, S. Bohner and S. El-Kassas [12] proposed FADES that is formal analysis and design for engineering security as the security engineering approach and using FADES also proposed an automated process to find out the security specifications. They also derived the acceptance test cases from security requirements.

In 2012, C. Rudolph and A. Fuchs [13] suggested that the different tasks of security engineering should be integrated with SDLC. Various approaches to the security engineering and the relation of functionality and security have been discussed. Further, three embedded scenarios are used to identify some core requirements for a security engineering process.

In 2013, S.A.Khan and R.A.Khan [14] proposed a Phased approach for software security testing. They described in detail the seven activities to be performed in phases for security testing. Each phase is described with help of a diagram.

In 2013, Suhel Ahmad Khan, Raees Ahmad Khan [15] proposed a prescriptive framework for security testing consists of seven phases with the objective of identifying defects early. Integration of these phases with SDLC has been shown diagrammatically.

2.2 Techniques

Technique is a specific method applied to do a task on the basis of a special skill. Some of the techniques are reviewed and presented in Table-3.

Table 3: Techniques Used in Security Testing

S.N.	Year	Author	Technique	SDLC Phase
1.	2007	D. Byers and N. Shahmehri [16]	Vulnerability Cause Graphs (VCG)	All phases
2.	2010	Zhanwei Hui [17]	Software Security Testing (SST) Model based on Software Security Defects (SSD).	Define, Design, Develop and Test Phase
3.	2012	S. J. Lincke, T. H. Knautz and M. D. Lowery[18]	Misuse Deployment Diagram (MDD) based on Unified Modeling Language (UML)	Define and Design Phase
4.	2013	T. Kobashi, N. Yoshioka, T. Okubo, H. Kaiya, H. Washizaki and Y. Fukazawa [19]	Extended Security Requirement Pattern (Ex-SRP) And Extended Security Design Pattern (Ex-SDP)	Define and Design Phase

In 2007, D. Byers and N. Shahmehri [16] presented a process consisting vulnerability modeling together with vulnerability cause mitigation and process component definition. These are based on vulnerability cause graphs. This paper explains the criteria that have influenced the process design.

In 2010, Zhanwei Hui [17] presents a software security testing (SST) model based on Software Security Defects (SSD). He performed the defects behavior analysis using SSD, software vulnerabilities, software security threats and accidents.

In 2012, S. J. Lincke, T. H. Knautz and M. D. Lowery [18] used the Misuse Deployment Diagram (MDD) based on UML for system architecture when analyzing security. They also performed a case study on web registration project.

In 2013, T. Kobashi, N. Yoshioka, T. Okubo, H. Kaiya, H. Washizaki and Y. Fukazawa [19] suggested a method using Extended Security Requirement Pattern (Ex-SRP) And Extended Security Design Pattern (Ex-SDP) for security testing. A model testing process is proposed and a case study is performed.

2.3 Methodologies

Methodology is the concept and gives a way of applying methods to accomplish a task. Some of the methodologies are reviewed and presented in Table-4.

Table 4: Methodologies Used in Security Testing

S.N.	Year	Author	Methodology	SDLC Phase
1.	2010	Andrea Avancini , Mariano Ceccato [20]	Taint analysis and genetic algorithms	Test, Deploy and Support Phase
2.	2011	B. Smith [21]	Black Box security tests based on software system requirements specifications	Define and Design Phase
3.	2012	A. Rein, C. Rudolph, J. F. Ruiz and M. Arjona [22]	Security Building Block (SBB) Metamodel	Design Phase
4.	2014	J. Bozic and F. Wotawa [23]	UML state charts	Design, Develop and Test Phase
5.	2014	L. b. Othmane, P. Angin and B. Bhargava [24]	Extension of the Scrum method	Define, Design and Develop Phase

In 2010, Andrea Avancini, Mariano Ceccato [20] proposed a methodology through the investigation of candidate's vulnerable points on basis of the integration of the static analysis with genetic algorithms. They stated that test cases for security testing can be generated on this basis.

In 2011, Smith [21] proposed a way to enhance the security of applications using a methodology based on software system's requirements specification statements that generates a set of black box security tests.

In 2012, A. Rein, C. Rudolph, J. F. Ruiz and M. Arjona [22] described the new Security Engineering Process with the

analysis of requirements and definition of the properties of the security and how the Security Building Block Model fits into this approach. SecFutur Patterns (SFPs) and Security Building Blocks (SBBs) are used for implementation of security.

In 2014, J. Bozic and F. Wotawa [23] made use of UML state charts based on attack patterns. They followed Moore and colleagues rules of attack patterns. From the proposed model test cases can be generated and executed automatically.

In 2014, L. b. Othmane, P. Angin and B. Bhargava [24] proposed the use of security assurance cases that are developed iteratively. The extension of the Scrum method discovered by Takeuchi and Nonakais are used. The three phases of Scrum are pregame, game, and Postgame are discussed in detail.

2.4 Reviews

Review is a survey and performs re-examination of the previous given facts/articles. It is generally presented in a periodical manner. Some of the reviews studied are presented in Table-5.

Table 5: Reviews in Security Testing

S. N.	Year	Author	Review Based On
1.	2007	Sattarova Feruza Y. and Prof.Tao-hoon Kim [1]	Strategies and methods
2.	2010	Gu Tian-yang, Shi Yin-sheng, and Fang You-yuan [25]	Major methods and security testing tools
3.	2011	Smriti Jain, Maya Ingle [26]	Security metrics
4.	2012	Hossian Shahriar, Mohammad Zulkernine [27]	Security Vulnerability Mitigation Techniques

In 2007, Sattarova Feruza Y. and Prof.Tao-hoon Kim [1] review the security testing components and basic principles. Different frameworks for assuring security in components are also discussed in detail. They also reviewed the technologies in IT security.

In 2010, Gu Tian-yang, Shi Yin-sheng, and Fang You-yuan [25] have given the major methods and testing tools for software security testing. They suggested that security testing can be classified into security functional testing and security vulnerability testing. They also proposed the taxonomy of security testing tools.

In 2011, Smriti Jain, Maya Ingle [26] reviews the software metrics in software development process and suggests that there is still the scope of development of metrics for quantitative assessment of security using the reasons for security loop holes in the software identified during SDP.

In 2012, Hossian Shahriar, Mohammad Zulkernine [27] mainly compared the security vulnerability mitigation techniques with static analysis and hybrid analysis. Secure programming, program transformation, and patching are also discussed.

3. CONCLUSION

In this paper, the review of various security testing approaches and techniques have been presented and the findings are tabulated in chronological order. Review reveals that most of the security testing techniques are implemented at various phases of software development life cycle. Advance concepts like UML and Scrums are also used. Some emphasis is given on the early phases of SDLC, but the proper attention to the design phase for security testing implementation is not drawn. In this phase, the various artifacts like Application logic, Interface design, Database design, User interfaces, Data dictionary, Process diagrams, and Screen layout diagrams are available. Therefore, security testing can be profusely performed in the design phase, prior to implementation. There is a need of any viable and perspective framework for security testing process at design phase of SDLC. Hence there is a wide scope for research in this context.

4. REFERENCES

[1] Sattarova Feruza Y. and Prof.Tao-hoon Kim, “IT Security Review: Privacy, Protection, Access Control, Assurance and System Security” International Journal of Multimedia and Ubiquitous Engineering, Vol. 2, No. 2, April, 2007.

[2] Https://Www.Owasp.Org/Index.Php/Testing_Guide_Introduction, last accessed: 23 August 2016.

[3] Michael Howard and Steve Lipner. The Security Development Lifecycle: SDL, a Process for Developing Demonstrably More Secure Software. Microsoft Press, 2006.

[4] S. Lipner, "The trustworthy computing security development lifecycle," Computer Security Applications Conference, 2004. 20th Annual, 2004, pp. 2-13.

[5] Chris Wysopal, Luke Nelson, Elfriede Dustin, Lucas Nelson, and Dino Dai Zovi. The Art of Software Security Testing. Addison-Wesley, 2006.

[6] Bruce Potter, Gary Mcgraw, “Software Security Testing”, IEEE Computer Society, October 2004

[7] Stig F. Mjolsnes, “A Multidisciplinary Introduction to Information Security”, CRC press, 2012.

[8] K. Jiwnani and M. Zelkowitz, "Maintaining software with a security perspective," Software Maintenance, 2002. Proceedings. International Conference on, 2002, pp. 194-203.

[9] J. A. Wang, "Security testing in software engineering courses," Frontiers in Education, 2004. FIE 2004. 34th Annual, 2004, pp. F1C-13-18 Vol. 2.

[10] K. Karppinen, R. Savola, M. Rapeli and E. Tikkala, "Security Objectives within a Security Testing Case Study," Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on, Vienna, 2007, pp. 1060-1065.

[11] I. A. Tondel, M. G. Jaatun and J. Jensen, "Learning from Software Security Testing," Software Testing Verification and Validation Workshop, 2008. ICSTW '08. IEEE International Conference on, Lillehammer, 2008, pp. 286-294.

[12] R. Hassan, M. Eltoweissy, S. Bohner and S. El-Kassas, "Formal analysis and design for engineering security automated derivation of formal software security

specifications from goal-oriented security requirements," in IET Software, vol. 4, no. 2, pp. 149-160, April 2010.

[13] C. Rudolph and A. Fuchs, "Redefining Security Engineering," 2012 5th International Conference on New Technologies, Mobility and Security (NTMS), Istanbul, 2012, pp. 1-6.

[14] S.A.Khan and R.A.Khan, ”software security testing process: phase approach” In: A.Agarwal etal (Eds): IITM 2013. CCIS 276. pp. 2011-2017, 2013 c Springar-verlag Berlin Heidelbarg 2013.

[15] Suhel Ahmad Khan, Raees Ahmad Khan,” ,”software security testing process”, UACEE International Journal of Advances in Computer Science and its Applications – IJCSIA Volume 3 : Issue 2 [ISSN 2250 – 3765], June 2013.

[16] D. Byers and N. Shahmehri, "Design of a Process for Software Security," Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on, Vienna, 2007, pp. 301-309.

[17] Zhanwei Hui, Song Huang, Bin Hu and Yi Yao, "Software security testing based on typical SSD: A case study," 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, 2010, pp. V2-312-V2-316.

[18] S. J. Lincke, T. H. Knautz and M. D. Lowery, "Designing System Security with UML Misuse Deployment Diagrams," Software Security and Reliability Companion (SERE-C), 2012 IEEE Sixth International Conference on, Gaithersburg, MD, 2012, pp. 57-61.

[19] T. Kobashi, N. Yoshioka, T. Okubo, H. Kaiya, H. Washizaki and Y. Fukazawa, "Validating Security Design Patterns Application Using Model Testing," Availability, Reliability and Security (ARES), 2013 Eighth International Conference on, Regensburg, 2013, pp. 62-71.

[20] Andrea Avancini , Mariano Ceccato, “Towards security testing with taint analysis and genetic algorithms” Proceedings of the 2010 ICSE Workshop on Software Engineering for Secure Systems, Pages 65-71, ACM New York, NY, USA ©2010, ISBN: 978-1-60558-965-7 doi>10.1145/1809100.1809110.

[21] B. Smith, "Systematizing security test case planning using functional requirements phrases," 2011 33rd International Conference on Software Engineering (ICSE), Honolulu, HI, 2011, pp. 1136-1137.

[22] A. Rein, C. Rudolph, J. F. Ruiz and M. Arjona, "Introducing Security Building Block Models," BioMedical Computing (BioMedCom), 2012 ASE/IEEE International Conference on, Washington, DC, 2012, pp. 132-139.

[23] J. Bozic and F. Wotawa, "Security Testing Based on Attack Patterns," Software Testing, Verification and Validation Workshops (ICSTW), 2014 IEEE Seventh International Conference on, Cleveland, OH, 2014, pp. 4-11.

[24] L. b. Othmane, P. Angin and B. Bhargava, "Using Assurance Cases to Develop Iteratively Security Features Using Scrum," Availability, Reliability and Security (ARES), 2014 Ninth International Conference on, Fribourg, 2014, pp. 490-497.

[25] Gu Tian-yang, Shi Yin-sheng, and Fang You-yuan , “Research on Software Security Testing”, World Academy of Science, Engineering and Technology 69 2010.

[26] Smriti Jain, Maya Ingle, “A Review of Security Metrics in Software Development Process”, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (6), 2011, 2627-2631, ISSN: 0975-9646.

[27] Hossian Shahriar, Mohammad Zulkernine, “Mitigating program security vulnerabilities: Approaches and challenges”, ACM Computing Surveys (CSUR), Volume 44 Issue 3, June 2012 Article No. 11, ACM New York, NY, USA, ISSN: 0360-0300 EISSN: 1557-7341.