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

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
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>
<thead>
<tr>
<th>S. N.</th>
<th>Year</th>
<th>Author</th>
<th>Framework</th>
<th>SDLC Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>2004</td>
<td>Bruce Potter, Gary Mcgraw [6]</td>
<td>Risk Based Approach</td>
<td>All phases</td>
</tr>
</tbody>
</table>

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. Eltoweissiy, 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**

<table>
<thead>
<tr>
<th>S N</th>
<th>Year</th>
<th>Author</th>
<th>Technique</th>
<th>SDLC Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2007</td>
<td>D. Byers and N. Shahmehri [16]</td>
<td>Vulnerability Cause Graphs (VCG)</td>
<td>All phases</td>
</tr>
</tbody>
</table>

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. Kobushi, 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**

<table>
<thead>
<tr>
<th>S N</th>
<th>Year</th>
<th>Author</th>
<th>Methodology</th>
<th>SDLC Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>2011</td>
<td>B. Smith [21]</td>
<td>Black Box security tests based on software system requirements specifications</td>
<td>Define and Design Phase</td>
</tr>
</tbody>
</table>

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
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>
<thead>
<tr>
<th>S. N.</th>
<th>Year</th>
<th>Author</th>
<th>Review Based On</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.</td>
<td>2011</td>
<td>Smriti Jain, Maya Ingle [26]</td>
<td>Security metrics</td>
</tr>
</tbody>
</table>

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


