IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-12, Dec 2018 E-ISSN: 2347-2693

Security Testing on Web Based Application

A. Punitha[1*], D. Sukanya Bai[2], K. Lavanya[3]

[1,2,3]Department of Computer Science, CMR Engineering College, JNTU University, Hyderabad, India

[*]*Corresponding Author: punitha26@gmail.com, Tel.: 9014787455 / 8897619309/9492017594*

Available online at: www.ijcseonline.org

Accepted: 19/Dec/2018, Published: 31/Dec/2018

Abstract- Prime objective of security testing [1] is to find out how vulnerable a system may be and to determine whether its data and resources are protected from potential intruders. Online transactions have increased rapidly of late making security testing as one of the most critical areas of testing for such web applications. Now a days, privacy and security play an important role. Software applications need to focus on data and its operations security which requires urgent attention, but it's ignored. Security resembles the essential feature in software. In fact, our main intention is to focus on the web based testing which is one of the types of application testing with its importance, implementation and its methodologies that has been defined from developer's point of view which is extremely helpful to the developers**.** Software Testing is an important process in SDLC [2] which provides assurance for quality to both software developer's (Company) and users as well. Just like testing the performance of an application, it is also important to perform security testing before the app is open to real users. Security testing is performed to detect vulnerabilities in an application, while ensuring that the data is protected and that the application works as required .Web testing is a software testing practice to test the websites or web applications for potential bugs. It's a complete testing of web-based applications [3] before making live. A web-based system needs to be checked completely from end-to-end before it goes live for end users [4]. By performing web site testing, an organization can make sure that the web-based system is functioning properly and can be accepted by real-time users. The UI design and functionality are the captains of website testing.

Keywords— Security Testing, Software Tesing, Web based Testing, Web-based Applications

I. INTRODUCTION

The basic principle of software testing is similar to security testing, to ensure system security. Security testing resembles a systems state where it can secure itself from unwanted actions and does not allow other entities/intruders to vanish system's integrity .The unwanted action may comprise systems unauthorized access to suspiciously alter file(s). So, Security testing is an act of making system defendable from attacks.

Security Testing Aspects: Security testing assures that the following aspects of data and information are maintained at any cost [5]:

1. Authentication
2. Authorization
3. Availability
4. Confidentiality
5. Integrity
6. Non-repudiation

Security protects applications against external malware and other unanticipated threats that may result in malfunction or exploitation of the application. These unanticipated threats could be either deliberate or unplanned. Studies suggest that security should, in fact, be made a business priority, as businesses of the day are running the show predominantly through digital platforms, organizations, therefore, need to be able to invest in security, in order to guarantee products and services of utmost quality.

II. RELATED WORK

Software Testing is included at end to find software operational response. Testing describes implemented software functionalities in practical way. Testing verifies the developed system with expected and unexpected inputs and observes its output thoroughly. Observations decide system correctness. Following approaches are very common to test software [6]:

1. Black Box Testing approach
2. White Box Testing approach
3. Gray Box Testing approach
4. Risk Based Testing approach

© 2018, IJCSE All Rights Reserved

1. ***Black box Testing Approach***: It is a very simple and efficient technique to test system as it just observes behaviour of system under test. It analyzes programs behaviour with various input combinations to look for any abnormal behaviour or wrong output. It does not perform code inspection checking. It executes program with valid and invalid inputs. The system is looked for its response and in case, an abnormal behaviour is observed, it must be corrected. It is also called Functional testing. To check security, tester injects malicious input, resultant system behaves abnormally and developers fix it [7].

2. ***White box Testing Approach:*** White box testing approach is important software testing which actually looks into code to find systems flaws. It is also known as structural testing. It requires tester to understand design and implementation knowledge of source code. It is very efficient in locating programming errors. Some testers use static analyzers and pattern matching to test programming errors in code, but it can provide false positive results.

3. ***Grey-box testing Approach:*** Grey-box testing (or gray-box testing) is defined as combination of black box and white box testing, and increases testing coverage of software testing . It allows testers to test software with basic information about it. The basic information required for Grey Box testing includes knowledge of internal data structures and algorithms, used for designing test cases. The test cases are executed at exposed interfaces. Grey box testing is best suited for testing integration of two modules. The interface is checked or tested for modules connectivity and data flow mechanism between them. It requires that tester must have knowledge about applications operation and functionality.

4. ***Risk based Testing Approach:*** Risk based testing technique refers risk associated with software system under test. in, author defines risk as "any threat to the achievement of one or more of the cardinal aims of the project". Another definition of risk state that "a risk is a problem that has yet to occur, and a problem is a risk that has already materialized .the risk associated with software may cause great loss to organization, why risk based testing is considered of great importance. risk based security testing also considers attackers intentions and his abilities to perform attack. Developer's identify risk associated with an attack and try to minimize it. So, it provides very good methodology to improve software quality. It also helps management personnel to make necessary decisions regarding software release in market.

III. METHODOLOGY

There are seven main types of security testing as per Open Source Security Testing methodology manual. They are explained as follows [8]:

Figure 1: Types of Security Testing

- **Vulnerability Scanning**: This is done through automated software to scan a system against known vulnerability signatures.
- **Security Scanning:** It involves identifying network and system weaknesses, and later provides solutions for reducing these risks. This scanning can be performed for both Manual and Automated scanning.
- **Penetration testing**: This kind of testing simulates an attack from a malicious hacker. This testing involves analysis of a particular system to check for potential vulnerabilities to an external hacking attempt.
- **Risk Assessment:** This testing involves analysis of security risks observed in the organization. Risks are classified as Low, Medium and High. This testing recommends controls and measures to reduce the risk.
- **Security Auditing:** This is an internal inspection of Applications and Operating systems for security flaws. Audit can also be done via line by line inspection of code
- **Ethical hacking:** This is a hacking organization Software system. Unlike malicious hackers, who steal for their own gains, the intent is to expose security flaws in the system.
- **Posture Assessment:** This combines Security scanning, Ethical Hacking and Risk Assessments to show an overall security posture of an organization.

© 2018, IJCSE All Rights Reserved

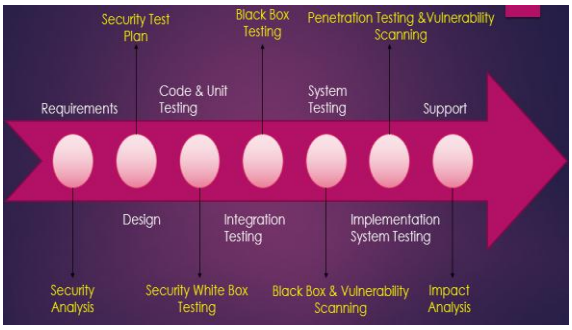

Figure 2: Security Testing Process in SDLC

IV. RESULTS AND DISCUSSION

Why Securiy Testing is Needed: According to Cisco's 2017 Annual Cyber security Report, over 33% of the organizations all over the globe had to deal with a cyber-breach in 2016. This resulted in a severe loss of users, business opportunities and overall revenue by a whopping 20%. The report surveyed nearly 3,000 chief security officers (CSOs) and security operations leaders from 13 countries .

Security testing tools [9] are many in number, each with the ability to focus on a certain element of the intricate interconnectedness of a software system. Security testing helps to avoid [10]:

- Loss of customer trust
- Inconsistent website performance
- Additional costs required to repair website after an attack
- Other legal implications that arise due to lax security measures

3 TIPS FOR WEB APPLICATION SECURITY TESTING [11]:

1**) If a system is business-critical, it should be tested often**: Any system that stores customer data—including credit card numbers, personally identifiable information (PII), or any other sensitive information—should be tested for security vulnerabilities; in fact, it's often a requirement of many government- or industry-mandated COMPLIANCE guidelines. Keep this in mind when looking at the potential scope of web application security testing [11] in your organization.

2) **The earlier security is tested in software's design lifecycle, the better:** You do not want to leave security testing as a last step in software development inevitably, vulnerabilities will be found and this can throw a big wrench into the development and maintenance processes. Bring security into the process early in the development lifecycle, preferably with the full involvement of your development operation team, to streamline response, minimize risk, and minimize any costs or time spent on remediation [12].

3) Keep development teams on track by prioritizing remediation and bug fixes: The output of web application security testing will often be a list of items that development will need to address at some point. Security calls them vulnerabilities, but development calls them bugs. The key is to not simply drop a list of these issues into a Developers team's lap; instead, be sure to prioritize the vulnerabilities and fully integrate with the bug tracking system in place, in order to maximize time to remediation.

SECURITY TESTING TOOLS:

These are just a few of the security testing tools available for web applications [13].

Tools	Description	Requirement
BeEF	BeEF (Browser Exploitation Framework) is a tool which focuses on the web browser – this means it takes advantage of the fact that an open web-browser is the crack into a target system and designs its attacks to go on from this point onwards.	Linux, Apple Mac OS X and Microsoft Windows
Brakeman	Brakeman is an open source vulnerability scanner which is designed for Ruby on Rails applications. It statically analyzes Rails application code to find security issues at any stage of development.	Rails 3
Netsparker	It is the automated web application security scanner .the comparison contains a good wealth of information and for those who have time, it is worth to dive into and analyse al of the results.	Java,Perl,.NET,PHP,Phython language applications
Gendarme	Gendarme is an extensible rule-based tool to find problems in .NET applications and libraries. Gendarme inspects programs and libraries that contain code in ECMA CIL format (Mono and .NET) and looks for common problems with the code, problems that compilers do not typically check or have not historically checked.	.NET (Mono or MS runtime)

© 2018, IJCSE All Rights Reserved

NET SPARKER (TESTING TOOL): Netsparker Desktop is an easy-to-use yet powerful web application security scanner that allows you to scan websites, web applications, and web services automatically and identify vulnerabilities and security flaws in them. Netsparker Desktop can scan all types of web applications, irrelevant of the platform they run on or the technology they are built with [14].

FEATURES:

A. Ease of Use
B. Advanced Scanning
C. Proof-Based Scanning
D. Web Services Scanning
E. Built-in Tools
F. Flexibility
G. Productivity
H. SDLC Integration
I. Vulnerability Details
J. Reporting
K. Manual Testing

COMPARISION WITH OTHER TOOLS:

Netsparker detected 1,112 vulnerabilities and is only second to IBM App Scan, which detected 1,147 vulnerabilities. Next in line is NTO Spider with 958 vulnerabilities, then HP Web inspect with 917 vulnerabilities followed by Acunetix, which detected 819 vulnerabilities. Burp Suite, Syhunt and N-stalker follow with 791, 716 and 484 identified vulnerabilities respectively [15].

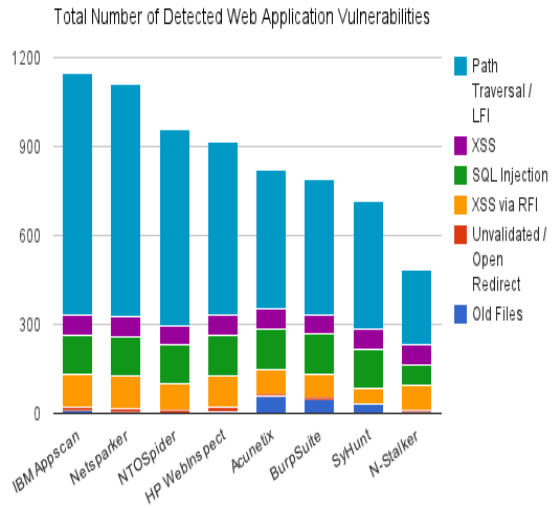

Figure 3: Comparing Netsparker with other Tools

V. CONCLUSION AND FUTURE SCOPE

As more and more web technologies have moved a long way to create web application. Security testing of the Web based application [16] of software plays an important role. Here in this paper we discussed while these approaches are tested successful on various case studies, many problem remains, related to mainly security issue. Strive to use some systematic process that starts from identifying and scoping the entire application, followed by planning multiple tests to follow the process of testing, analyzing and reporting on the security level and/or posture of a Web application. It is used by Web developers and security administrators to test and gauge the security strength of a Web application using manual and automated security testing techniques. We considered security testing with respect to various security testing techniques and Web testing tools. The main conclusion is that all testing are fully dependent on implementation technologies and future testing techniques have to adapt heterogeneous and dynamic nature of security testing to use best automation tool [17] to deliver quality software to the customer.

ACKNOWLEDGMENT

This work was done as part of a research work in the year 2018 in CMR Engineering College, Hyderabad.

REFERENCES

[1] Gu Tian-yang, Shi Yin-sheng, and Fang You-Yuan, "Research on Software Security Testing", World Academy of Science, Engineering and Technology, Vol. 70, p 647-651, September 2010.
[2] Qianxiang WANG, Lining QUAN, Fuchen YING, ―Online Testing of Web-Based Applications‖, 0730-3157/04 2004 IEEE.
[3] K.K. Aggarwal, Yogesh Singh, "Software Engineering", (3rd ed.), Copyright © New Age International Publishers, 01-Jan-2005.
[4] Yu Qi, David Kung and Eric Wong, ―Leveraging User-Session Data to Support Web Application Testing‖, IEEE Transactions on Software Engineering, vol. 31, no. 3, March 2005.
[5] Gencer Erdogan, Ketil Stolen, "Risk-driven Security Testing versus Test-driven Security Risk Analysis", First Doctoral Symposium on Engineering Secure Software and Systems.
[6] Mohd. Ehmer Khan& Farmeena Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques", International Journal of Advanced Computer Science and Applications, (IJACSA) Vol. 3, No.6, 2012.
[7] Jovanovich, Irena, "Software Testing Methods and Techniques".
[8] Ould, M. A. (1999). Managing software quality and business risk. Chichester: John Wiley & Sons.
[9] B. Potter, G. McGraw, "Software Security Testing," IEEE Security & Privacy, v2, n5, 81-85, Sept.-Oct. 2004.
[10] S. Barnum, G. McGraw, "Knowledge for Software Security", IEEE Security & Privacy, v3, n2, 74- 78, March-April 2005.
[11] Thompson, H.H., "Why security testing is hard", IEEE Security & Privacy, v 1, n 4, 83-6, July-Aug. 2003.
[12] DeMarco, T. and T. Lister (2003). Waltzing with Bears: Managing Risk on Software Projects. New York: Dorset.

© 2018, IJCSE All Rights Reserved

[13] Bruce Potter & Gary McGraw, "Software Security Testing", IEEE Security & Privacy, 2004, pp. 32-36.
[14] Suhel Ahmad Khan, Raees Ahmad Khan "Software Security Testing Process", Proc. of the Intl. Conf. on Recent Trends In Computing and Communication Engineering-- RTCCE 2013, p39-42.
[15] J. Viega and G. McGraw, Building Secure Software: How to Avoid Security Problems the Right Way, Addison Wesley, 2001.
[16] Arora A., Sinha M. ―Web Application Testing: A Review on Techniques, Tools and State of Art‖, International Journal of Scientific & Engineering Research, Volume 3, Issue 2, February-2012 ISSN 2229-5518.
[17] G. McGraw, "Testing for Security During Development: : Why We Should Scrap Penetrate-and-Patch," IEEE Aerospace and Electronic Systems, Vol. 13, no. 4, 1998, pp 13-15.

AUTHORS PROFILE

A.Punitha pursued Master of Technology in Computer Science & Engineering from NRI Institute of Technology, Hyderabad,Telangana State. Completed her M.B.A System in Databse Madras University.I am currently working as Assistant Professor in CSE Department at CMR Engineering College, Kandlakoya (V), Medchal (D), Hyderabad, Telangana, India-501401. I have 3 Years of experience in Education field.One year in training field on Software Testing .My areas of interest include Software Engineering & Software Testing.

D. Sukanya Bai pursued Master of Technology in Computer Science & Engineering from the University of JNTUCEA, Anantapur in year 2016. She is currently working as Assistant Professor in Department of CSE, CMR Engineering College, Kandlakoya (V), Medchal (D) HYDERABAD since 2017. She has published a research papers in reputed international journal and conference including (IJERCSE) it's also available online. Her main interests focuses on Network Security and Privacy. She has 2 years of teaching experience and 1 year of Research Experience.

K.Lavanya pursued Master of Technology in Computer Science & Engineering from the University of JNTUCEA, Anantapur in year 2016,currently working as Assistant Professor in CSE Department at CMR Engineering College, Kandlakoya (V), Medchal (D), Hyderabad, Telangana, India-501401. She has 2 Years of experience in Education field.. Areas of interest include Data warehousing & Data Mining, BigData Analytics.

© 2018, IJCSE All Rights Reserved