NEXT GENERATION SOFTWARE SECURITY THROUGH TESTING STAGE OF SDLC

Vidyabhushan A. Upadhye and Shashank D. Joshi

1-2Department of Computer COE, Bharati Vidyapeeth University, Pune, India.
E-mails: bhushan_u2001@yahoo.com, sdj@live.in

ABSTRACT

Keeping in view the daily increase in software security threats, developing secure software has become a necessity and challenge. Early detection of vulnerabilities in software while developing it and countering them in the software development cycle will save time, money and energy spent on removing them after software release. We are developing continuous process for systematic and improvement of Software security throughout the various software lifecycle stages, that is suitable for industrial adoption, and focuses on preventing vulnerabilities in all phases of software development cycle. To aim of this paper we propose methodology for software security through Testing Stage of software development life cycle.

Keywords: Software Development Life Cycle, Vulnerabilities.

1. INTRODUCTION

Software Testing is the process of executing a program or system with the target of finding errors. It involves any activity aimed at evaluating a characteristic, capability of a program or system and influential that it meets its required results. Software bugs will almost always exist in any software module with sensible size not because programmers are careless or reckless, but because the complexity of software is generally intractable and humans have only limited ability to manage complexity. It is also true that for any intricate systems, design defects can never be completely ruled out.

Software quality, reliability and security are tightly coupled. Flaws in software can be exploited by intruders to open security holes. With the development of the Internet, software security problems are becoming even more severe and painful.

Many critical software applications and services need integrated security measures against malicious attacks. The purpose of security testing of these systems include identifying and removing software flaws that may potentially lead to security violations, and validating the effectiveness of security measures. Simulated security attacks can be performed to find vulnerabilities.

A vulnerability scanner is a program that performs the analytical phase of a vulnerability analysis, also known as vulnerability assessment. Vulnerability analysis defines, identifies, and classifies the security holes and their vulnerabilities in a computer, server, network, or communications channel, infrastructure.

In addition, vulnerability analysis can forecast the effectiveness of proposed countermeasures, and evaluate how well they work after they are put into use. A vulnerability scanner relies on a database that contains all the information required to check a system for security holes in services and ports, anomalies in packet construction, and potential paths to exploitable programs or scripts. Then the scanner tries to exploit each vulnerability that is discovered. This process is sometimes called ethical hacking.

2. RELATED WORK

The methodology aims to minimize vulnerabilities in software under development. To this aim, each time, output of each phase of SDLC is fed to the security checklist where it is verified whether the output fulfills prerequisites for security of the phase. If yes, the phase is declared as secure. If not, the output is properly analyzed for the detection of the vulnerabilities [1].

We are developing a process for systematic and continuous improvement of software security throughout the software life cycle that is suitable for industrial adoption, and focuses on preventing vulnerabilities in all phases of software development [2].

An ideal vulnerability scanner has capabilities such as: Preservation of an up-to-date database of vulnerabilities. Detection of authentic vulnerabilities without an excessive number of false positives. Ability to perform trend analyses and provide clear reports of the results. Recommendations for countermeasures to eliminate discovered vulnerabilities. If security holes are detected by a vulnerability scanner, a vulnerability disclosure may be required. The person or organization that discovers the vulnerability, or a responsible industry body such as the Computer Emergency Readiness Team (CERT), may make the disclosure, sometimes after
alerting the vendor and allowing them a certain amount of time to remedy or moderate the problem.

Fig 1 Shows UI for Vulnerabilities Detection in software application before process starts.

After scanning predefined Vulnerabilities the tool presents report summery as shown in Fig.2. Stress testing in IT industry (hardware as well as software sectors) means testing of software for its effectiveness in giving consistent or satisfactory performance under extreme and unfavorable conditions such as heavy network traffic, heavy processes load, under or over clocking of underlying hardware, working under maximum requests for resource utilization of the peripheral or in the system etc.

In other words, stress testing helps find out the level of robustness and consistent or satisfactory performance even when the limits for normal operation for the system (software/hardware) is crossed. Most important use of stress testing is found in testing of software and hardware that are supposed to be operating in critical or real time situation. Such as a website will always be online and the server hosting the website must be able to handle the traffic in all possible ways (even if the traffic increases manifold), a mission critical software or hardware that works in real time scenario etc.

Stress testing in connection with websites or certain software is considered to be an effective process of determining the limit, at which the system/software/hardware/website shows robustness, is always available to perform its task, effectively manages the load than the normal scenario and even shows effective error management under extreme conditions.

**Importance of Stress Testing**

Stress testing is considered to be important because of following reasons:

1. Almost 92% of the software/systems are developed with an assumption that they will be operating under normal scenario. And even if it is considered that the limit of normal operating conditions will be crossed, it is not considerably as high as it really could be.

2. The cost or effect of a very important considerable software, system and website failure under extreme conditions in real time can be huge (or may be catastrophic for the organization or entity owning the software/system).

3. It is always better to be prepared for extreme conditions rather than letting the system/software/web services crash, when the limit of normal, proper operation is crossed.

4. Testing carried out by the developer of the system/software/website may not be sufficient to help reveal conditions which will lead to crash of the system/software when it is actually submitted to the operating environment.

5. It’s not always possible to reveal possible problems or bugs in a system/software, unless it is subjected to such type of testing.
3. COMPARISON OF RESULTS

![Comparison Graph]

4. CONCLUSION

This paper focuses on enhancement of software through testing stage of SDLC. Testing is moderately expensive. Vulnerabilities Detection Tool is very good technique to cut down cost and time. Testing efficiency and effectiveness is the criteria for coverage-based testing techniques. Present tool well tested on Banking Domain Future work is to suit the tool for all domains.

REFERENCES


