Chapter 97
Innovative Strategies for Secure Software Development

Punam Bedi
University of Delhi, India

Vandana Gandotra
University of Delhi, India

Archana Singhal
University of Delhi, India

ABSTRACT

This chapter discusses adoption of some proactive strategies in threat management for security of software systems. Security requirements play an important role for secure software systems which arise due to threats to the assets from malicious users. It is therefore imperative to develop realistic and meaningful security requirements. A hybrid technique has been presented in this chapter evolved by overlapping the strengths of misuse cases and attack trees for elicitation of flawless security requirements. This chapter also discusses an innovative technique using fuzzy logic as a proactive step to break the jinx of brittleness of present day security measures based on binary principle. In this mechanism, partially secure state evolved between safe state and failed state using fuzzy logic provides an alert signal to take appropriate additional preventive measures to save the system from entering into the failed state to the extent possible.

INTRODUCTION

The term "secure software" describes the state of software, where no unauthorized access, no manipulation, and no attacks on the software itself are successful. It is concerned with confidentiality (information disclosed only to authorized users), integrity (information modification only by users who have the right to do so, and only in authorized ways) and availability (use of the system cannot be maliciously denied to authorized users) (Brunil et al., 2009). Unfortunately, it is not possible to achieve such software security with the traditional methods which are post-development activities and are not supported by appropriate methodologies to manage the high complexity of the securing process. Software researchers and practitioners have therefore come up with

DOI: 10.4018/978-1-4666-4301-7.ch097

the view that we should focus on security from early phases of development life cycle as it is much cheaper to prevent than to repair to justify investment (Stoneburner et al., 2004; Davis et al., 2004). A number of security experts have therefore enhanced existing software development life cycle by incorporating various security techniques in all phases for developing secure software systems. Secure software engineering has thus been evolved as a new approach wherein security features are 'in-built' rather than 'on-bolt'.

We have extended this area of research and presented some proactive measures in this chapter to be adopted at the design level itself as a part of secure software engineering. This helps in reducing design-level vulnerabilities which are a major source of security risks in software and has accounted for around 50% of the security flaws (Hoglund & McGraw, 2004). Exploitation of these vulnerabilities result in security threats which are potential attacks, i.e. misuses and anomalies that violate the security goals of system's intended functions. Managing these threats suggests what, where and how security features for threat mitigation should be applied in secure software engineering. This chapter presents some of the proactive strategies in threat management like hybrid technique for elicitation of security requirements and use of fuzzy logic to help avert failed state in security of software systems to meet the challenges of changing security scenario of modern times.

Security requirements form the very basis for developing secure software systems. These requirements identify the vulnerable points in the system that an attacker can exploit to carry out threats. It is therefore necessary to determine realistic and meaningful security requirements. Any flaw in these requirements will mean faulty development of the security system making it vulnerable to threats. A new technique named as Hybrid Technique has therefore been presented in this chapter for elicitation of effective security requirements. This technique has been evolved by overlapping the strengths of misuse cases and attack trees as they provide complementary information which becomes much more comprehensive for the designers for enhancing the security of software systems. This proactive step in threat management is instrumental in enhancing in-built security to meet present day threat perceptions (Gandotra et al., 2009; Gandotra et al., 2011).

Traditional security mechanisms these days are based on binary principle making the system to be in either of the two states i. e. safe state or failed state. We have no means or mechanism to protect our system from complete failure which is a bad omen from security point of view. This can result in catastrophic consequences in terms of leakage of confidential data so vital to the users. A proactive security measure using fuzzy logic has therefore been discussed in this chapter to avert the failed state of the software system to the extent possible. This mechanism helps evolve an intermediate stage i.e. Partially Secure State (Yellow Zone) between the Safe State (Green Zone) and the Failed State (Red Zone). Here, the Yellow Zone provides an 'Alert Signal' in case the security of system starts moving towards Red Zone. This prevents the system from being compromised as the additional preventive measures become operative immediately on sensing this alert signal. This mechanism helps in designing more secure software systems thus eliminating the possibility of catastrophic failure of traditional security measures based on binary principle (Gandotra et al., 2010; Gandotra et al., 2011).

Thus, integration of the above proactive steps in threat management with traditional security development process has the potential to positively influence the level of security inherent in the developed software products.

BACKGROUND

This section discusses some important definitions relating to the subject discussed in this chapter. Works done by leading researchers in this area