Chapter 7.7
Software Security Engineering:
Towards Unifying Software Engineering and Security Engineering

Mohammad Zulkernine
Queens University, Canada

Sheikh I. Ahamed
Marquette University, USA

ABSTRACT

The rapid development and expansion of network based applications have changed the computing world in the last decade. However, this overwhelming success has an Achilles' heel: almost every software controlled system faces threats from potential adversaries both from internal and external users of the highly connected computing systems. These software systems must be engineered with reliable protection mechanisms, while still delivering the expected value of the software to their customers within the budgeted time and cost. The principal obstacle in achieving the above two different but interdependent objectives is that current software engineering processes do not provide enough support for the software developers to achieve security goals. In this chapter, we reemphasize the principal objectives of both software engineering and security engineering, and strive to identify the major steps of a software security engineering process that will be useful for building secure software systems. Both software engineering and security engineering are ever evolving disciplines, and software security engineering is still in its infancy. This chapter proposes a unification of the process models of software engineering and security engineering in order to improve the steps of the software life cycle that would better address the underlying objectives of both engineering processes. This unification will facilitate the incorporation of the advancement of the features of one engineering process into the other. The chapter also provides

a brief overview and survey of the current state of the art of software engineering and security engineering with respect to computer systems.

INTRODUCTION

With the proliferation of connectivity of computer systems in the applications where the quality of service depends on data confidentiality, data integrity, and protection against denial of service attack, the need for secure networks is evident. In these applications, the consequences of a security breach may range from extensive financial losses to dangers to human life. Due to heavy dependence of computer network based applications on various software and software controlled systems, software security has become an essential issue. Almost every software controlled system faces potential threats from system users, both insiders and outsiders. It is well accepted that "the root of most security problems is software that fails in unexpected ways when under attack" (McGraw, 2002). Therefore, software systems must be engineered with reliable protection mechanisms against potential attacks while still providing the expected quality of service to their customers within the budgeted time and cost. Software should be designed with the objective not only of implementing the quality functionalities required for their users but also of combating potential and unexpected threats. The principal obstacle in achieving the above two different but interdependent objectives is that current software engineering processes do not provide enough support for the software developers to achieve security goals.

Some of the principal software engineering objectives are usability, performance, timely completion, reliability, and flexibility in software applications (Finkelstein & Kramer, 2000; Pressman, 2001; IEEE, 1999). On the other hand, some of the major objectives of security engineering are customized access control and authentication based on the privilege levels of users, traceability and detection, accountability, non-repudiation, privacy, confidentiality, and integrity (Pfleeger & Pfleeger, 2003; Viega & McGraw, 2001). Having stated that, software security engineering objectives are to design a software system that meets both security objectives and application objectives. However, software security engineering is still considered a difficult task due to inherent difficulties associated with the addressing of the security issues in the core development and maintenance of software systems. Both software engineering and security engineering are ever evolving disciplines and software security engineering is still in its infancy. A precise and well-accepted understanding of software security engineering does not yet exist (ISSEA, 2003; Wolf, 2004).

The principal objectives of software security engineering need to be reinvestigated, and a methodology is required that can be employed for building secure software systems. Software security engineering is a practice to address software security issues in a systematic manner. We believe that the security issues must be addressed in all the stages of software system development and maintenance life cycle such as requirements specifications, design, testing, operation, and maintenance to provide secure software systems. This chapter identifies some possible ways to adapt readily the current practices of security engineering into a software engineering process model. In other words, this chapter suggests a unification of the process models of software engineering and security engineering. The unification is desirable between the two processes by removing the unnecessary differences in their corresponding steps that obscure the fundamental principles of the development and maintenance of secure software systems. This unification will help system designers to employ the techniques and tools of software engineering and security