Chapter I

Integrating Security and Software Engineering: An Introduction

H. Mouratidis, University of East London, UK

P. Giorgini, University of Trento, Italy

ABSTRACT

This chapter serves as an introduction to this book. It introduces software engineering, security engineering, and secure software engineering, providing definitions and explanation of terms necessary for readers to understand the subsequent chapters. Characteristics of each of the above areas are presented followed by an overview of the current advances in these areas. Finally, the 10 approaches described in the remaining chapters of the book are briefly introduced.

INTRODUCTION

Software systems become more and more critical in every domain of the human society. Transportation, telecommunications, entertainment, health care, military, education, and so on; the list is almost endless. These systems are used not only by major corporations and governments but also by individual users. Such wide use of information systems has resulted in these systems containing a large amount of critical information, which inevitably need to remain secure. Therefore, although it is important to ensure that software systems are developed according to the user
needs (functional requirements), it is equally important to ensure that these systems are secure.

However, traditionally, security is considered after the definition of the system, meaning that security mechanisms are fitted into pre-existing designs. Usually, in practise, a fit-all solution is assumed where security mechanisms, such as authentication, are inserted into the system with very little consideration of the implications of inserting such mechanisms into the existing system’s design. As a result, security may conflict with the system’s requirements and this can lead to problems, which most of the times translate into security vulnerabilities (Anderson, 2001; Stallings, 1999).

One of the reasons for this situation is the fact that traditionally the two associated research areas of software engineering and security engineering work independently. On one hand, software engineering techniques and methodologies do not consider security as an important issue, although they have integrated concepts such as reliability and performance, and they usually fail to provide precise enough semantics to support the analysis and design of security requirements and properties (Crook, Ince, & Nuseibeh, 2003; Mouratidis, 2004a). On the other hand, security engineering research has mainly produced formal and theoretical methods, which are difficult to understand by non security experts and which, apart from security, they only consider limited aspects of the system.

From the viewpoint of the traditional security paradigm, integrating security and software engineering would result in a situation where security is considered as part of the development process, leading to the development of more secure software systems. We call this area of research secure software engineering, and we consider it a branch of research concerned with the development of secure software systems, which integrates security and software engineering.

In the rest of the chapter, the research areas of software and security engineering are introduced and a discussion emphasising the characteristics of the secure software engineering research area is presented. Then the current state of the art on software and security engineering are presented, emphasising the latest approaches to secure software engineering. The chapter concludes by introducing the approaches presented in the rest of the book.

SOFTWARE ENGINEERING

Trying to explicitly and accurately define something as wide and dynamic as software engineering is a very difficult task. Therefore, there is a tendency from researchers and practitioners to develop personal definitions (Pressman, 2005). As a result of this, various different definitions regarding software engineering appear on texts (see for example Macro & Buxton, 1990; Pressman, 2005; Sommerville, 2004; Vliet, 1993). These definitions often use different words and different ideas to describe software engineering and range from very simple ones, such as software