A Methodology to Develop Secure Systems Using Patterns

E. B. Fernandez, Florida Atlantic University, USA
M. M. Larrondo-Petrie, Florida Atlantic University, USA
T. Sorgente, Florida Atlantic University, USA
M. Vanhilst, Florida Atlantic University, USA

ABSTRACT
We are developing a methodology to build secure software for complex applications and its related support. This methodology considers the whole software lifecycle, uses security patterns, and is applied at all the architectural levels of the system. A main idea is that security principles should be applied at every stage and that each stage can be tested for compliance with security principles. Patterns help apply security principles. This chapter presents the current status of our work.

INTRODUCTION
A good percentage of the software deployed in industrial/commercial applications is of poor quality, it is unnecessarily complex, and contains numerous flaws that can be exploited by attackers. Every day the press reports of attacks to Web sites or databases around the world, resulting in millions of dollars in direct or indi-
rect losses and the number of vulnerabilities and incidents keep increasing (CERT, 2005). Until recently, the only software vendors’ response to security problems was to provide patches to fix the latest vulnerability found or to blame the users for their lack of caution. However, patches are clearly not a solution: it is hard for system administrators to keep up with the latest patches and the patch itself may open new possibilities for attack. There are two basic approaches to improve application security: (1) examine final production code and look for possible problems (e.g., buffer overflow conditions) (Howard & LeBlanc, 2003) or (2) plan for security from the beginning. We believe that the solution lies in developing secure software from the beginning, applying security principles along the whole life cycle. Part of the problem is that developers are not, in general, acquainted with security methods. We see the use of patterns as a fundamental way, even for developers with little experience, to implicitly apply security principles. We are developing a methodology to build secure software based on patterns (Fernandez, 2004).

Our intended target is the construction of complex applications. These include medical systems, financial applications, legal applications, operating systems, and others. These applications are typically implemented in systems having additional non-functional requirements such as reliability or fault tolerance. Often they are composed of a variety of units, some built ad hoc and some bought or outsourced. In these systems, the security of the application software itself cannot be separated from the security of the rest of the system. Another common aspect of these systems is that they frequently must follow regulatory standards, for example, HIPAA (HIPAA), Gramm-Leach-Bliley (Gramm-Leach-Bliley Act, 1999), or Sarbanes/Oxley (Sarbanes-Oxley, 2002). These systems may include several databases and usually have Internet access as well as distributed and wireless access. Data is typically accessed using a Web application server (WAS) that integrates Web and database applications and has a global enterprise model, usually implemented using components such as J2EE or .NET. These applications are of fundamental value to enterprises and their security is extremely important. A systematic approach is required to build these applications so they can reach the appropriate level of security. We focus on these applications because they define worst-case scenarios where to apply our methodology.

Patterns provide solutions to recurrent problems and there are several catalogs of patterns. We see the use of security patterns as a useful way to incorporate security principles in the design process even by people having little experience with security practices. We have produced many security patterns. (Delessy-Gassant, Fernandez, Rajput, & Larrondo-Petrie, 2004; Fernandez & Pan, 2001; Schumacher, Fernandez, Hybertson, Buschmann, & Sommerlad, 2005). For building conceptual models, we developed a type of pattern called semantic analysis pattern (SAP), which implements a set of basic use cases (Fernandez & Yuan, 2000). We can combine SAPs and security patterns in a natural way to create authorized SAPs, which can be converted into models for secure designs where security constraints are defined.
Addressing Risks in Global Software Development and Outsourcing: A Reflection of Practice


[www.igi-global.com/article/addressing-risks-in-global-software-development-and-outsourcing/205631?camid=4v1a](www.igi-global.com/article/addressing-risks-in-global-software-development-and-outsourcing/205631?camid=4v1a)