Chapter 15
An Integrated Secure Software Engineering Approach for Functional, Collaborative, and Information Concerns

J. A. Pavlich-Mariscal
Pontificia Universidad Javeriana, Colombia

S. Berhe
University of Connecticut, USA

A. De la Rosa Algarín
University of Connecticut, USA

S. Demurjian
University of Connecticut, USA

ABSTRACT

This chapter explores a secure software engineering approach that spans functional (object-oriented), collaborative (sharing), and information (Web modeling and exchange) concerns in support of role-based (RBAC), discretionary (DAC), and mandatory (MAC) access control. By extending UML with security diagrams for RBAC, DAC, and MAC, we are able to design an application with all of its concerns, and not defer security to a later time in the design process that could have significant impact and require potentially wide-ranging changes to a nearly completed design. Through its early inclusion in the software design process, security concerns can be part of the application design process, providing separate abstractions for security via new UML diagrams. From these new UML diagrams, it is then possible to generate security policies and enforcement code for RBAC, DAC, and MAC, which separates security from the application. This modeling and generation allows security changes to have less of an impact on an application. The end result is a secure software engineering approach within a UML context that is capable of modeling an application's functional, collaborative, and information concerns. This is explored in this chapter.

DOI: 10.4018/978-1-4666-6026-7.ch015

1 INTRODUCTION

The software development process has had significant improvements of the past forty plus years, from the introduction of the *waterfall model* (Winston, 1970) to the *iterative model* (Larman and Basili, 2002) in the late 70s to the *spiral model* (Boehm, 1986) in the mid-1980s to the *unified process model* (Scott, 2001) to the *agile development lifecycle* (Craig, 2003) in the early 21st century. Despite this progress, there remain many challenges when one attempts to design and develop large-scale applications, where there are a myriad of concerns such as user interfaces, server functionality, database support, logging and historical tracking, and secure information modeling, access, and enforcement. Rather than separation, there is often an entanglement of these different concerns, e.g., in an object-oriented application, code to read/write the database can be spread across multiple classes even if the database is abstracted via Hibernate. Also consider that security can be realized across the entire application, with security checks and enforcement at the GUI level, the server level, the database level, the network communications level, etc. All of these different concerns end up being tangled with one another, and spread out across the application's varied components. As a result, the traceability of security in terms of an application's functional, collaborative, and information concerns cannot be easily isolated; in such a situation, changes to the security policy often requires code-level alternations which are not acceptable in practice. The intent of this chapter is to elevate security to a primary and early priority in the software development process to provide a secure engineering approach that encompasses functional, collaborative, and information concerns.

To place this into a proper perspective, Figure 1 conceptualizes a secure software engineering approach for functional, collaborative, and information concerns via UML to visually model access control security. Over the past five years, our focus has been on extending UML with new diagrams that supports secure software engineering for role-based access control (RBAC) (Ferraiolo, et al., 2001), discretionary access control (DAC) (DoD, 1985), and mandatory access control (MAC) (Bell & LaPadula, 1976). In this chapter, we bring together our work for secure software engineering in three areas. First, from a functional perspective that focuses on object-oriented design, we have developed a framework of composable security features that preserves separation of security concerns from models to code through the extension of UML with new diagrams for RBAC, DAC, and MAC with the automatic generation of enforcement code in AspectJ that allowed the security definitions to be separated (untangled) from the code (Pavlich-Mariscal, 2005, 2010a, 2010b). Second, from a collaboration perspective, we have developed a framework for secure, obligated, coordinated, and dynamic collaboration that extends the RBAC to allow for the definition and enforcement of security for collaborative RBAC applicable to situations such as medical care where physicians from different specialties need to collaborate with one another to treat a patient in an effective and timely manner (Berhe, 2009, 2010, 2012); this work also defines new UML diagrams based on our functional work (Pavlich-Mariscal, 2005, 2010a, 2010b). Third, from an information perspective, we have defined and developed an XML security framework (De la Rosa Algarín, 2012, 2013a, 2013b) with new XML oriented UML diagrams that integrates RBAC, DAC, and MAC to further extend (Pavlich-Mariscal, 2005, 2010a, 2010b) by allowing the definition of security policies for XML for sharing and exchange of information in a secure manner via XACML. Our combined work promotes security as an integral part of a secure software engineering approach, while tracking software quality assurance in terms of the consistency of the security and non-security requirements.