Chapter 4.8
A Service-Based Approach for RBAC and MAC Security

Charles E. Phillips, Jr.
United States Military Academy, West Point, USA

Stephen A. Demjurian
University of Connecticut, USA

Thuong Doan
University of Connecticut, USA

Keith Bessette
University of Connecticut, USA

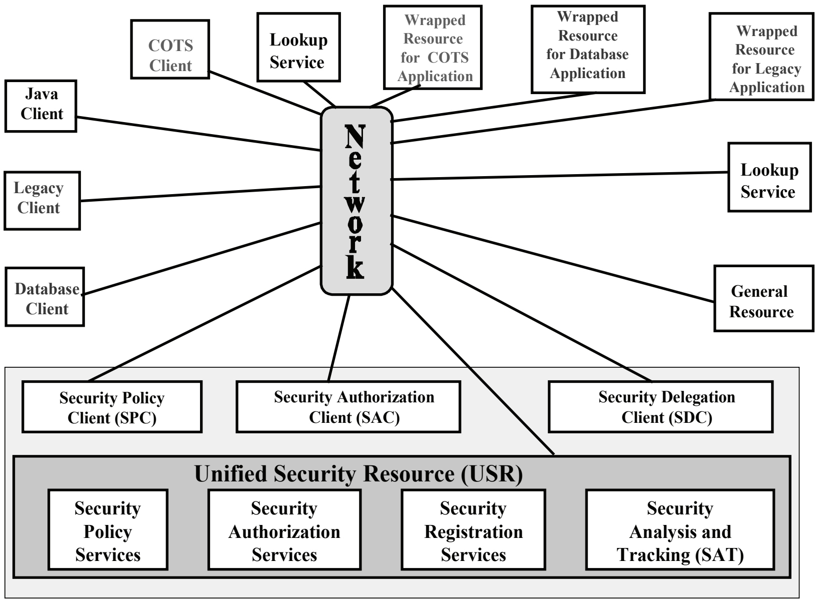
ABSTRACT

Middleware security encompasses a wide range of potential considerations, ranging from the ability to utilize the security capabilities of middleware solutions (for example, CORBA, .NET, J2EE, DCE, and so forth) directly out-of-the-box in support of a distributed application to leveraging the middleware itself (paradigm) to realize complex and intricate security solutions (for example, discretionary access control, role-based access control, mandatory access control, and so forth). The objective in this chapter is to address the latter consideration: examining the attainment of advanced security capabilities using the middleware paradigm, namely, role-based access control (RBAC) and mandatory access control (MAC). The resulting security provides a robust collection of services that is versatile and flexible and easily integrates into a distributed application comprised of interacting legacy, COTS, GOTS, databases, servers, clients, and so forth.

INTRODUCTION

One challenge facing government and corporations today is to architect and prototype solutions that integrate new and existing software artifacts (that is, legacy applications, COTS, GOTS, databases, clients, servers, and so forth), facilitating their interoperation in a network-centric environment via middleware (collections of services), thereby providing the computing infrastructure to

Figure 1. The security framework


COTS Client
Lookup Service
Wrapped Resource for COTS Application
Wrapped Resource for Database Application
Wrapped Resource for Legacy Application
Java Client
Legacy Client
Database Client
Network
Lookup Service
General Resource
Security Policy Client (SPC)
Security Authorization Client (SAC)
Security Delegation Client (SDC)
Unified Security Resource (USR)
Security Policy Services
Security Authorization Services
Security Registration Services
Security Analysis and Tracking (SAT)

support day-to-day operations, as shown in the top portion of Figure 1. In these distributed collections of software artifacts, security must play a fundamental role, considered at early and all stages of the design and development life cycle. Middleware security encompasses a wide range of potential considerations, ranging from utilizing out-of-the-box security services of middleware platforms, that is, DCE (Open Software Foundation, 1994; Rosenberry, Kenney & Fischer, 1992), CORBA (Object Management Group, 2002; Vinoski, 1997; Yang & Duddy, 1996), DCOM/OLE (Microsoft Corporation, 1995), J2EE/EJB (Roman, 1999; Valesky, 1999), Jini (Arnold et al., 1999; Waldo, 1999), and .NET (Riordan, 2002; Sceppa 2002), to custom-built service-based solutions that realize complex and intricate security approaches (for example, discretionary access control, role-based access control, mandatory access control, and so forth).

In such a scenario, one can conceptualize each of the software artifacts in terms of *resources* that provide *services* (methods) for use within the environment, and as such, each artifact publishes an *application programmer interface (API)*. The problem with these APIs is that they contain all of the public methods needed by all users without regard to security. If one user (for example, a physician) needs access to a method (for example, prescribe_medicine) via a patient tool, then that method must be part of the API, and as such, the responsibility would be on the software engineer to ensure that the method is only accessible via the patient tool to users who are physicians and not all users of the patient tool (which may include nurses, administrators, billing, and so forth). Thus, in many applications, the ability to control the visibility of APIs (services) based on user role would be critical to ensure security.

Towards this end in this chapter, we present a service-based approach using middleware that unifies role-based access control (RBAC) and mandatory access control (MAC) into a security model and enforcement framework for a distributed environment comprised of interacting software artifacts (Liebrand et al., 2003; Phillips et al., 2002a; Phillips et al., 2002b; Phillips et al., 2003a; Phillips et al., 2003b). Our approach concentrates on the APIs of software resources, the services, providing the means for them to be