Chapter XXI
Access Control Frameworks for a Distributed System

Rajeev R. Raje
Indiana University-Purdue University Indianapolis, USA

Alex Crespi
Indiana University-Purdue University Indianapolis, USA

Omkar J. Tilak
Indiana University-Purdue University Indianapolis, USA

Andrew M. Olson
Indiana University-Purdue University Indianapolis, USA

Carol C. Burt
2AB Inc., Helena, AL, USA

ABSTRACT

Component-based software development offers a promising technique for creating distributed systems. It does require a framework for specifying component properties, analyzing the behaviors of a system before composition, and validating them during operation. This chapter focuses on access control properties of a distributed system. It provides a framework that addresses the following issues: (a) specifying access control properties for individual components, (b) identifying components with required access control properties, and (c) formulating compositional models for predicting the access control properties of a composed system from those of its individual components.

INTRODUCTION

Although component-based software development is a valuable approach for creating a complex distributed system, it requires a framework for specifying component properties in order to analyze the system’s behavior before its assembly and, then, validate it during operation. Both functional and quality of service (QoS) features of components require specification. One common QoS characteristic is security, whose importance cannot be understated in many sensitive application domains, such as medical or military applications. A framework that supports composing and
predicting a distributed system’s security characteristics from the properties of its individual components would aid in the creation of more secure systems for such sensitive domains. Access control is an important type of security. Thus, any framework to specify and predict the security properties of a composed system from the properties of individual components should provide a means to model and predict the system’s access control properties. This chapter describes one such framework based on the principles of UniFrame (UniFrame Project, 2006)—an on-going research effort that aims to automate the creation of Distributed Computing Systems (DCS) from geographically scattered, heterogeneous software components.

The specific objectives of this chapter are to provide a means of:

- Expressing the access control characteristics of individual software components within the UniFrame paradigm.
- Identifying individual software components on a network that meet system access control requirements.
- Predicting the access control behavior of an integrated system based on the properties of its individual components.

**BACKGROUND**

There have been many attempts made at modeling access control in computing systems. This section describes a few prominent efforts.

**Access Matrix**

The basic model for access control is the access control matrix, which consists of a two dimensional matrix relating subjects to objects. Each cell in the matrix contains the access privileges of one subject for accessing one object (Saunders, Hitchens, & Varadharajan, 2001). This matrix can be extremely large for large systems and may be sparsely populated, leading to great inefficiencies in implementation (Sandhu & Samarati, 1994). Therefore, most systems implement access control models that can be mapped back to the concept of the access control matrix while avoiding these inefficiencies.

**Access Control Lists (ACL)**

ACL are a means of implementing efficiently access control matrices. Each secured object has an ACL that consists of data from a column of the access control matrix. Only entries for subjects allowed to use the object are present in the ACL, thereby eliminating the access control matrix’s inefficiencies. Replacing the ACL for an object is easy, but determining all of the privileges for a single subject is difficult in such a system (Sandhu & Samarati, 1994). For instance, if all of the access privileges of a subject must be revoked, then all ACLs must be examined. This may still be more efficient than examining the whole matrix.

**Discretionary Access Control (DAC)**

In DAC, an owner of an object determines if a given subject may be allowed access to that object or not. Thus, a DAC policy consists of a subject’s identity, pertinent object identities, and a series of rules decided by the object’s owner. This policy specification determines whether or not a particular subject may perform a specific operation on an object. Closed DAC systems only allow access when the policy specifically allows the access, whereas an open DAC system only disallows access when the policy denies access. One weakness of DAC is that once a user gains access to information, potentially nothing prevents it from sharing the information with an unauthorized user (Sandhu & Samarati, 1994). In addition, with DAC systems, it can be difficult to represent and maintain complex access control policies for a large organization.

**Mandatory Access Control (MAC)**

MAC assigns each object a security level (e.g., Top Secret, Secret, Confidential, and Unclassified), and each subject a clearance level (i.e., Top Secret, Secret, Confidential, and Unclassified) reflecting the subject’s trustworthiness. A subject may access an object only if the subject’s clearance level is sufficiently high to permit access to the object’s security level. A subject may read objects of equal or lower security level than the subject’s clearance level, whereas the subject may write only to objects that are of equal or higher security level than the subject’s clearance level. This structure ensures that information from an object with a higher security level does not get written to a document with a lower security level, thus preventing a breach in security (Sandhu & Samarati, 1994).