Authorization and Delegation in CORBA *

Vipin Swarup
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420
swarup@mitre.org

The CORBA Security Specification V2.0 document contains a detailed, voluminous description of authorization and delegation for method invocations in CORBA. A concise, rigorous description of the underlying models will help convey the gist of CORBASec to current and future implementors, administrators, and users.

1 Authorization

In the classical access control model, an access control matrix is a relation that specifies whether a subject s can invoke operation m on an object o. The access control matrix completely describes the security-relevant state of the system (as pertaining to method invocation), which we call the *protection state*. Administrative operations and other operations such as delegation characterize how this state can evolve.

CORBA authorization is based on a refinement of the classical access control model. The protection state is not just a matrix but has more structure. In particular, it is partitioned into four components which we call the *subject attribute state*, *object attribute state*, *access rights state*, and *access decision function*.

CORBASec uses privilege attributes, domains, and access rights to group together subjects, objects, and methods respectively. Privilege attributes are *granted* limited access rights in domains, and they *require* specified access rights in order to invoke methods on objects. This framework is very general and admits many different semantic models; thus different CORBASec implementations may not be interoperable.

*This work is funded by the National Security Agency under Army CECOM Contract DAAB07-97-C-E601. It does not represent the position of the NSA or of the MITRE Corporation.

2 Delegation

A delegation model describes how security attributes in an authorization model can be propagated among entities. We distinguish between two kinds of delegation: privilege attribute delegation and policy enforcement delegation.

Privilege attribute delegation describes how the privilege attribute state of a subject changes when the subject engages in a communication event such as invoking a method on an object or receiving a message from another subject. A subject may lose privileges when it invokes an untrusted operation and may gain privileges when it invokes a privileged operation. For instance, invoking a setuid operation in Unix causes a subject to acquire the privilege attribute "root". Similarly, a subject's privileges may increase if it receives a capability (e.g., e-cash) in a message from another subject.

Policy enforcement delegation describes how the enforcement of an object's access control policy may be delegated to a host. For instance, when an object migrates from one host to another (e.g., as a call-by-value argument of a remote method invocation), the source host may delegate enforcement of the object's access control policy to the destination host.

CORBASec includes a limited form of privilege attribute delegation and does not address policy enforcement delegation.

3 Summary

In my presentation, I will describe the models that underlie CORBASec authorization and delegation. Further, I will elaborate on various issues and ambiguities inherent in CORBASec.