



Information System Security Operation Security for Object-Oriented Distributed Systems

Standards that Support Scalability and Management

Overview

One of the challenging aspects of security in Object-Oriented Distributed Systems is managing large-scale deployments while retaining fine-grained access control. Scaling problems develop when there are large numbers of users, large numbers of objects, complex interfaces, or all of the above. Although there are security standards for popular object-oriented architectures, the standards have largely ignored issues of scalability and management.

Under Defense Advanced Research Projects Agency (DARPA) funding, McAfee[®] Research, now the Security Research Division of SPARTA, developed two key security technologies for object-oriented distributed systems, Object Oriented Domain and Type Enforcement (OO-DTE), and the Multi-Protocol Object Gateway (MPOG).

- OO-DTE is a security plug-in for CORBA (Common Object Request Broker Architecture from the Object Management Group) that provides access control for distributed objects.
- MPOG is a security gateway (i.e., firewall proxy) for CORBA and Java RMI traffic.

Object Oriented Domain and Type Enforcement

OO-DTE provides selective access to specific objects and methods by individual users according to their assigned roles. For example, an individual acting as a Planner might be allowed to invoke the “changePlanObjective” method belonging to a crisis action plan object, while a Logistics Clerk is allowed to invoke the “assignSuppliesToMission” method.

The prototype OO-DTE plug-in is designed to be inserted into an Object Request Broker (ORB), the foundation of the CORBA infrastructure. An ORB is typically implemented as a set of libraries linked into each client and server application process. As shown in the adjacent figure, when a client invokes a method a remote object, the client-side ORB transmits an invocation request message to the ORB in the corresponding server. Both the client-side and server-side ORBs perform authorization checks before forwarding the request. The client-side checks that the server is authorized to perform the request and the server-side checks that the client is authorized to make the request. If both checks succeed, the request is passed to the server application; if either check fails, the request is rejected and an exception is transmitted to the client. In many cases, the presence of OO-DTE can be transparent to the application, requiring only initialization changes. The prototype OO-DTE plug-in has been integrated with Inprise’s Visibroker ORB and The ACE ORB (TAO).

Controlling access in large, object-oriented systems can be difficult because there may be many thousands of objects, classes, and methods that must be protected. Using conventional access control lists compounds this problem because each list can contain multiple entries whose combined effect is based on ordering and precedence rules. The result is a proliferation of complex access control information and a loss of understandability.

OO-DTE addresses these problems by providing a compilable high-level policy language called DTEL++ for specifying the desired access control configuration. DTEL++ closely resembles CORBA’s interface definition language (IDL) so

application architects can use the same identifiers and terminology in DTEL++ that they used to define the client-to-server interfaces in IDL. DTEL++ provides a variety of “wild-card” techniques so access control attributes can propagate by default through the inheritance hierarchy of the lexical name space. As a result, a small number of DTEL++ statements can easily specify the access control configuration for a large application system.

OO-DTE improves scalability by separating the authentication and authorization steps. The Secure Socket Layer (SSL) protocol provides authentication and transport security. A Role Authorization Database (RAD) provides authorization information by mapping each user identity onto a set of authorized roles. After establishing an SSL session, the OO-DTE plug-in sends the user’s desired role before the first request is sent to the server. The OO-DTE plug-in validates the user’s requested role against the RAD using the identity information provided by SSL. The plug-in uses that role for subsequent access checks on the connection.

OO-DTE has a policy distribution mechanism that allows rapid role authorization, revocation, and access control policy changes to be pushed to all OO-DTE hosts from a central management point.

The policy mechanism scales well by using a hierarchy of multiple policy distribution servers.

Multi-protocol Object Gateway (MPOG)

As shown in the figure below, the MPOG prototype acts like an application firewall proxy protecting CORBA and Java RMI servers on a local area network while providing selective access to them by remote clients. The MPOG uses OO-DTE to limit the objects and methods that such clients can access. Although the MPOG interoperates with OO-DTE clients and servers using SSL, it has been designed to support multiple authentication technologies for non-OO-DTE hosts. MPOG provides highly configurable facilities for weighting and combining multiple security attributes into composite authorization ratings. MPOG also supports dynamic role authorization and policy updates using OO-DTE’s policy distribution facilities.

Object-Oriented Domain and Type Enforcement

