Issues in Performing Access Control for Java RMI at the Enclave Boundary

Gregg Tally
Gary Lamperillo
Network Associates

May 7, 1999

Many organizations are developing distributed object applications that lack application level access control. In addition, the organizations find that the applications would be useful to clients that are outside the enclave boundaries. While it is possible to implement access control mechanisms in the client and server applications, it is often preferable to implement access control at the enclave boundary, such as a firewall. Adding security at the enclave boundary minimizes the impact on existing application code and centralizes the administration of access control policy.

Under funding from DARPA, NAI Labs has recently developed and demonstrated a prototype implementation of enclave boundary access control for Java RMI. The Multi-Protocol Object Gateway (MPOG) performs fine-grained access control at the enclave boundary for both CORBA and Java RMI applications. The MPOG is based on our prior work with the ORB Gateway, which has been presented at previous DOCsec workshops. In the original ORB Gateway, the access control mechanism uses an implementation of Object Oriented Domain and Type Enforcement (OO-DTE) which provides role-based access control for distributed object applications. This mechanism has also been applied to Java RMI access control in the MPOG.

An enclave boundary access controller must make decisions based on the wire-protocol messages passed between the client and server. We observed that IIOP is a largely self-contained protocol and that it was relatively easy to obtain the information required to make an access decision (target object, interface name, operation name, and principal's privilege attributes). JRMP is much less self-describing than IIOP and relies upon the registry to provide meaning to the JRMP identifiers.

The registry is central to Java RMI. Servers register objects with the registry at creation. Clients access the registry to find objects. If clients and servers reside on opposite sides of a firewall, the registry must be located on a host that is accessible to both the client and server. The registry, however, is intended to be local to the server, which would normally make it inaccessible to clients on the outside of a firewall.

Java Security and CORBA Security take much different approaches to access control. CORBA Security is oriented towards determining if the requesting principal has the required privileges to invoke an operation on an object. Java Security does not authenticate principals and does not use a principal's privilege attributes in access control decisions. Instead, Java Security makes access decisions based upon the reliability of the code source that is requesting access to a resource (object).

The prototype MPOG implementation resolves these and other issues with fine-grained access control at the enclave boundary. There are also some problems remaining to be solved, and we will provide information on these issues as well.