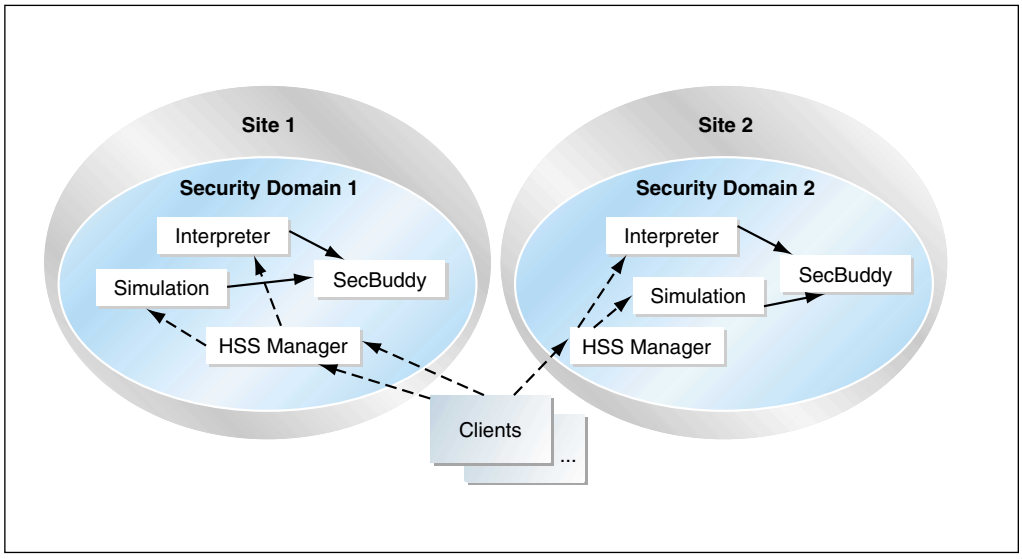
Software To Secure Distributed Propulsion Simulations

CORBASec brings role-based security to CORBA-object-wrapped simulations.

John H. Glenn Research Center, Cleveland, Ohio

Distributed-object computing systems are presented with many security threats, including network eavesdropping, message tampering, and communications middleware masquerading. NASA Glenn Research Center, and its industry partners, has taken an active role in mitigating the security threats associated with developing and operating their proprietary aerospace propulsion simulations. In particular, they are developing a collaborative Common Object Request Broker Architecture (CORBA) Security (CORBASec) test bed to secure their distributed aerospace propulsion simulations. Glenn has been working with its aerospace propulsion industry partners to deploy the Numerical Propulsion System Simulation (NPSS) object-based technology. NPSS is a program focused on reducing the cost and time in developing aerospace propulsion engines.



The **Functional Blocks** and the relationships among them depicted in this diagram represent the CORBASec multiple security domain, multiple ORB interceptor services, and application invoked architecture.

NPSS has been developed by Glenn and sponsored by the NASA Ames Research Center. Glenn is an active domain member of the Object Management Group (OMG) — an open membership, not-for-profit consortium that produces and manages computer industry specifications (i.e., CORBA) for interoperable enterprise applications. When NPSS is deployed, it will assemble a distributed-aerospace propulsion-simulation scenario from proprietary analytical CORBA servers and execute them with security afforded by the CORBASec implementation.

The NPSS CORBASec test bed utilizes the Portable Object Adaptor (POA) architecture from the VisiBroker 4.x Object Request Broker (ORB) [Borland Software Corp., Scotts Valley, CA], and the Orbix 2000 ORB [IONA Technologies, Dublin, Ireland, with U.S. headquarters in Waltham, MA]. Quadrasis [Software Solutions Division of Hitachi Computer Products (America), Inc., Waltham, MA] integrated both VisiBroker 4.x and Orbix 2000 ORB architectures with their security service [Hitachi Security Service (HSS)]. The NPSS required a security service that was compatible with both ORBs. Glenn and two United States aeropropulsion-industry companies are the initial partners contributing to the NPSS CORBASec test bed. The test bed uses Security SecurID [RSA Security Inc., Bedford, MA] two-factor, token-based authentication together with HSS digital-certificate-based authentication to validate the various NPSS users.

The CORBASec test bed was integrated across firewalls. The process of getting CORBASec to communicate across firewalls was a large accomplishment. Unlike processing Hypertext Transfer Protocol (HTTP) messaging, firewalls do not have designated ports for CORBA Internet Inter-ORB Protocol (IIOP) traffic. NPSS also verified the CORBASec and firewall design by testing with multiple vendors' firewalls. The OMG is now working on a Firewall Traversal specification that promises to provide a standard solution to the CORBASec firewall-integration problem. It will take some time before the CORBA and CORBASec vendors implement this standard solution. NPSS chose to move ahead with a workable solution.

The CORBASec architecture is a flexible ORB security architecture that supports both security-unaware and security-aware application development. CORBASec security-unaware security features are primarily configuration-based, requiring very little programming and its security services are implicitly invoked at the ORB interceptor layer. The CORBASec security-unaware architecture spares the application developer from having to write large amounts of security code as the ORB interceptor layer has been configured to handle CORBASec message traffic automatically. The CORBASec security-aware architecture is for applications that need fine-grain security. Security-aware applications enforce fine-grain or application-specific security policies via the CORBASec application programming interface (API) explicitly invoked security services. Unlike security infrastructure-based APIs such as the java.security package, the Java Cryptography Extension (JCE), and the Java Authentication and Authorization Service (JAAS), COR-

BASec [like Enterprise Java Beans (EJB)] supports container-based security, in which a rich array of security services enforce security transparently, allowing the developer to concentrate on building the application rather than the supporting infrastructure. The paradigm shift away from a security API results in security software that is controlled and managed at the ORB interceptor layer and is less prone to programming error.

Within the computer-security discipline there is much talk about role-based security. In the distributed-computer-security world, CORBASec, unlike other distributed role-based security models (e.g., EJB security), defines security domains to allow partitioning of enterprise systems that need to secure large numbers of resources. The NPSS team needed a design suitable for the enterprise systems and therefore, we chose to use the CORBASec approach. The choice of a design that supports multiple security domains has enabled the NPSS team to develop a highly scalable architecture that allows room for growth.

The CORBASec test bed is designed to provide peer (client and server) role-based authorized security at the CORBA object (interface, method, and variable) levels. For the purposes of discussion, this article focuses on the functionality of the peer client: The CORBASec client authorization architecture (role-based design) allows each client access to a simulation object's functionality based on a run-time comparison of the clients' granted roles and credentials against the required rights of the object. The test bed uses the HSS administration tool to configure three NPSS client user roles: developer, general user, and restricted user. Initially, each client role-based user is checked for the proper security domain access. Developers are granted full access to the private and public simulation object variables and methods as well as access to methods unique to programmers. General users are granted full access to all private and public simulation object variables and methods (except developer-only methods). Restricted users are granted only public access. If necessary, various other role-based configurations can be developed with the CORBASec test bed and its HSS administration tool.

In the figure, the two ovals labeled Site 1 and Site 2 represent separate aerospace propulsion company network sites. The two ovals labeled Security Domain 1 and Security Domain 2 contain CORBA servers shown as boxes labeled HSS Manager, Interpreter, Simulation, and SecBuddy. The dashed arrows depict the flows of public information among these servers. The solid arrows depict the flows of delegated private information among these servers. The boxes outside the ovals in the figure represent CORBA clients. The HSS Manager at each site authenticates clients for access to specific security domains.

The test bed is expected to demonstrate NPSS CORBASec-specific policy functionality, confirm adequate performance, and validate the required Internet configuration in a distributed collaborative aerospace propulsion environment.

This work was done by Tammy M. Blaser of **Glenn Research Center**. *Further information is contained in a TSP (see page 1).*

Inquiries concerning rights for the commercial use of this invention should be addressed to NASA Glenn Research Center, Commercial Technology Office, Attn: Steve Fedor, Mail Stop 4-8, 21000 Brookpark Road, Cleveland Ohio 44135. Refer to LEW-17214.