The Security Model

The Legion environment raises a number of security concerns and, just as significantly, invalidates many assumptions made in previous distributed systems. What follows is an outline of a new security model proposed to address these issues.

Object Security

Security in Legion will be centered on the object, making every object responsible for ensuring its own security. Instead of being dictated a set of global access rules, every object class will decide who is permitted to communicate with that object. Rather than having a set of standard rights, every object will permit access on a per-method basis. Thus, a file system could grant rights to member functions that create, read, write, seek, and delete files, while another object for whom these rights are meaningless could define rights based on its own functions.

Every Legion object will contain a number of hooks, whereby additional functionality can be attached. These hooks can be left undefined, in which case no additional execution costs would be invoked, or they can be as complex as the object implementor desires, with corresponding increases in run-time cost. These hooks will be implemented as member functions that are declared in some base class but can be overridden as desired in derived classes. The member functions include hooks for authentication protocols, message encryption, access control, and delegation.

The most important of the member functions, that which is responsible for enforcing access control, is the "MayI" function. This function will be implicitly invoked whenever some other object attempts to invoke one of the object in question's member functions. The MayI function is provided data about the present environment, including the identity of the caller, and decides whether to permit the call to proceed or not. In order to reduce the overhead, MayI can optionally issue licenses that permit later function calls to bypass the MayI function for a limited time or number of invocations.

It is worth noting that since Legion uses inheritance to define security functionality, the creator of a new class can inherit all, some, or none of the desired functionality from an existing class. It is our intent to provide classes implementing a number of basic security schemes as part of a standard Legion distribution in order to facilitate this.

External Policies

While we believe that object-centric security is the correct approach for enforcing security among communicating objects, there are many existing security approaches that must still be supported if Legion is to succeed. Most existing security schemes are implemented as a central authority that authorizes inter-entity communication. It might be easier to port such a system if some sort of central authority were available. In fact, in some schemes such as the government Multi-Level Security (MLS) approach, objects are not trusted to behave and security must be enforced externally.

For these reasons, Legion includes a facility for extra-object security. If a collection of objects requires an external scheme, it can provide a Security Agent (SA) - an object that monitors and must approve of all inter-object communication. Whenever an object attempts to invoke a method of another object, the acting security agent is invoked and has the opportunity to either approve or disallow the missive. Note that:

- This is in addition to any additional protection that the object itself wishes to enforce.
- If no SA is active, no additional calls and accompanying overhead are involved.
- Since the SA traces execution, it can be used to track information flow dynamically.
- There are a number of ways to optimize repeated permitted function calls

The Physical Model

Because of the lack of physical security, we must assume that tampering with Legion programs and data will occur. While at the functional level it is useful to view the entire system as one seamless virtual supercomputer, the security-conscious programmer will have to be able to collect information about the actual hosts on which objects reside, and even dictate restrictions on the object-host mapping. This may permit proprietary service software to be made available without worry that it will leave a host that is trusted not to copy it. It will also permit a security-conscious site to trust Legion to whatever degree it deems appropriate, rather than being forced to take an all-or-nothing stance. Finally, in conjunction with existing firewall software, this mapping will permit Legion firewalls to be constructed by only allowing the execution of certain trusted gateway objects on the firewall machine.

Legion Basis

Legion security functionality can be divided into those basic services that Legion will provide, and those additional services that user objects will have to provide, either by inheriting from an existing base class or by providing themselves. This distinction is much like that between the C++ language and the standard library. While we may provide a number of useful base classes, the core Legion security functionality will be quite small. At present it consists of:

- The empty base object template with "hooks" for other functionality
- Message signature and authentication support
- Integrity of environmental information
- The MayI invocation and optimization code
- The SA invocation and optimization code
- The physical model

Summary

The Legion security model is by no means complete. As implementation proceeds, the model will be refined to handle the difficulties that will arise. Whatever the future holds for Legion, the object-centric model of security has tremendous potential for the large scale distributed systems of the future. At present Legion motivates a number of interesting security questions and provides an environment in which solutions to these problems can be evaluated.

For a much more detailed look at the proposed Legion security model, see:

- William A. Wulf, Chenxi Wang, Darrell Kienzle, "A New Model of Security for Distributed Systems," University of Virginia Technical Report CS-95-34, August 1995.

legion@Virginia.edu
http://www.cs.virginia.edu/~legion/