



Information System Security Operation Configuration Synthesis and Policy Enforcement

Simplify The Processes Of Access Policy Construction And Implementation

Motivation

The ability to securely share information system resources has become critical to the tightly integrated systems and process of business, military and other coalitions. Secure sharing requires that organizations be able to exercise fine-grained, policy-governed control over access to shared resources. Unfortunately, configuring mechanisms like firewalls or web servers to enforce access control policies is currently a manual, ad hoc, and error-prone process. Incorrect or inconsistent configurations can lead to access surprises and significant vulnerabilities. The configuration problem is even more severe in coalition environments with multiple systems and networks, cross-organizational resource sharing, a broad user base, and limited inter-organizational trust.

To enable collaborative resource sharing, we have developed proof-of-concept tools to support the accurate configuration of enforcement mechanisms for complex and dynamic distributed systems. These tools:

- Help the administrator precisely express and refine the intended coalition access policy at an “enterprise level” appropriate to coalition sharing.
- Automatically generate accurate, coordinated configurations for heterogeneous access policy enforcers.
- Warn the administrator when the existing policy enforcers are unable to enforce the desired access control policy.
- Deploy updated configurations to policy enforcers either automatically (to continue to enforce an existing policy amid system

changes) or at the administrator’s command.

Improving Scalability of Security Administration

The Configuration Synthesis for Policy Enforcement or SPiCE project is the third in a sequence of projects in which we have addressed the problem of scalable security administration for complex, collaborative distributed systems. Under the Secure Virtual Enclaves (SVE) project, we developed an access control system to support coalition sharing, while preserving organizational autonomy over local resources. Under the Security Policy Automation, Modeling and Bridging (AMBer) project, we introduced the first collection of coalition-based access control models (CBAC) and developed a framework for specifying the semantic interoperation of diverse authorization systems. The SPiCE project builds on these foundations by providing a language (called Cape) for specifying CBAC policies, a tool to assist the administrator in writing and refining those policies, a tool to translate Cape policies into configurations for access policy enforcers, and a deployment system to distribute updated configurations to the enforcers.

Synthesizing Configurations using the SPiCE System

The figure on the next page shows components of the proof-of-concept SPiCE access policy translation system. The SPiCE compiler is the heart of the system, taking coalition-level access control policies, written in the Cape language, and producing configurations for the access policy enforcers shown at the bottom of the figure: a modified Java runtime environment and a web servlet (both implementing a form of

This work sponsored by DARPA through SPAWAR, Contract Number N66001-02-C-6021, with McAfee Research, which is now the Security Research Division of SPARTA.

role-based access control) and Telcordia's Smart Firewalls system (which controls packet-filtering firewalls). The compiler requires three inputs: a CBAC domain specification, a Cape policy and a virtual system model.

The CBAC domain specification, stored in the database shown on the left side of the figure, contains information about coalition entities, such as coalitions, missions, organizations, resources, and roles. The administrator uses the policy editor to create policies that authorize organizations or roles to access system resources. The editor ensures that the process of policy refinement—narrowing authorizations from organizations, down through sub-organizations, and finally to individual roles—is performed correctly by the administrator. An individual cannot be authorized to access resources unless the organization that employs her is authorized.

The network oracle creates a model of the distributed system to be protected, identifying the enforcers (firewalls, web servlets, Java runtime, etc.) that protect each resource. In a production system, the network oracle would provide an interface between the SPiCE system and a network management system, such as

HP's OpenView or CA's Unicenter. From the domain specification, Cape policy, and system model, the SPiCE compiler attempts to generate configurations for the policy enforcers in the system. We have proven that the translation algorithm used by the compiler is both safe and complete: Configurations generated by the compiler accurately implement the Cape policy and if the compiler fails to generate a configuration, then we know that there is no configuration of the system's enforcers that can implement that Cape policy. This translation approach is extensible to enforcers beyond the three targeted by our proof-of-concept implementation.

Finally, the distributor component communicates configuration updates to the policy enforcers throughout the system. The operations of the policy editor, compiler and distributor are managed by the administrator via the SPiCE controller, shown at the right of the figure.

The SPiCE policy translation tools with their formal underpinnings, provide a strong foundation to ensure scalable, accurate, automated and extensible access policy administration.

