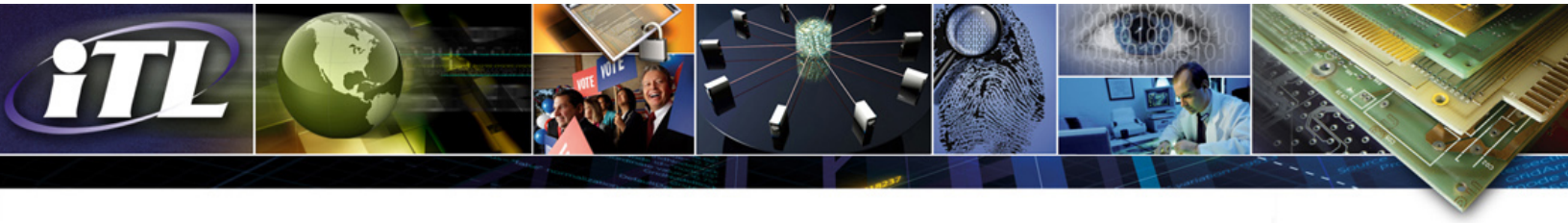


ITL BULLETIN FOR AUGUST 2014

POLICY MACHINE: TOWARDS A GENERAL-PURPOSE, ENTERPRISE-WIDE OPERATING ENVIRONMENT

David Ferraiolo, Larry Feldman, and Greg Witte, Editors
Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
U.S. Department of Commerce

Overview

The Policy Machine is an access control framework designed to implement the security-critical portion of the program logic of arbitrary data services, and to enforce tailored access control policies over data services, solely through the configuration of its access control data. Previous publications [4, 5] have described the Policy Machine's capabilities in expressing and enforcing a wide variety of access control policies. The focus of this bulletin is to highlight the benefits of the Policy Machine's integration of access control and data services.

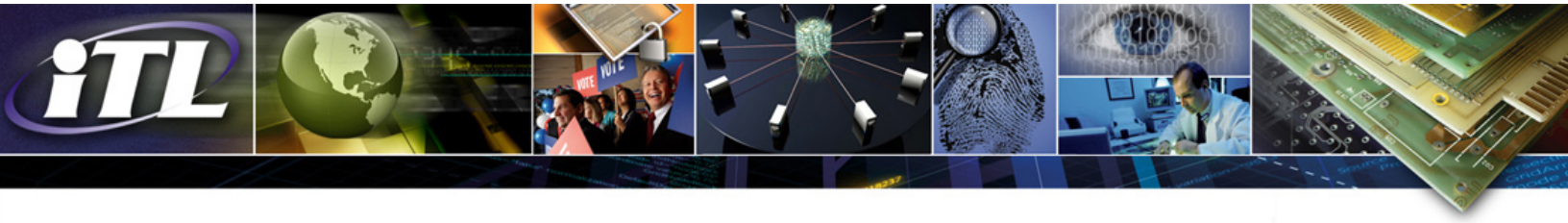
Background

Policy Machine (PM) has evolved from a concept to a formal specification [1], to a reference implementation and open source distribution, and has served as a basis for an American National Standards Institute, International Committee for Information Technology Standards (ANSI/INCITS) standardization effort under the title of *Next Generation Access Control* [2, 3].

Controlled delivery of data services is a primary objective of enterprise computing. In addition to the ubiquitous electronic mail and file management, data services commonly deployed in an enterprise include services for time and attendance reporting, payroll processing, health benefits management, and workflow management. All data services provide users with a subset of security-relevant computational capabilities to read, write, manage, and share data. Control over the execution of data service capabilities is achieved through authentication and access control mechanisms, typically made available through an operating environment.

While access control currently plays an important role in securing data services, by building access control and data services from the same underlying elements, access control can serve a more substantial role in computing. The PM was designed with this unification in mind, resulting in a multiuser, enterprise-wide operating environment.

To appreciate the PM's benefits to computing, it is important to recognize the methods by which data services are delivered today. Each data service runs in an operating environment, which can be of many



types (e.g., operating systems, database systems, middleware, many applications), each implementing its own routines to enable the execution of data service specific operations (e.g., read, send, approve, select) on their respective data types (e.g., files, messages, work items, records). To impose control, each operating environment typically implements its own method for identifying and authenticating its users. In addition to authentication, many operating environments implement their own method of access control to selectively limit a user's ability to perform operations on their objects (e.g., resources, and access control data).

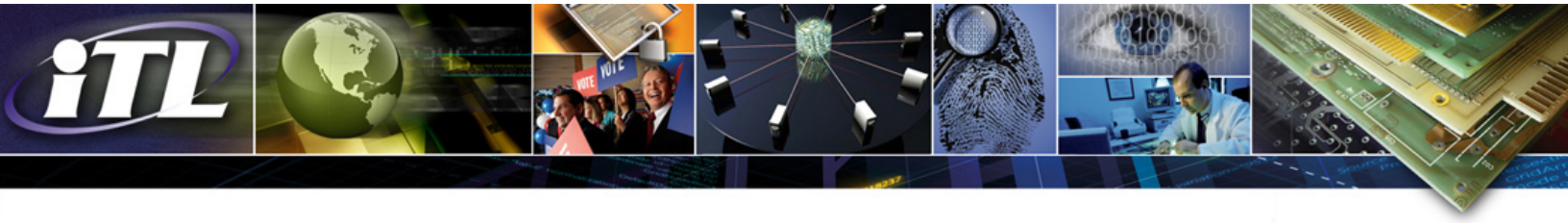
The heterogeneity among operating environments introduces a number of data interoperability, security management, policy enforcement, and usability challenges. Operating environments do not necessarily recognize each other's operation and object types. For instance, an operating system sees a database management system as just one giant file. Administrators must contend with a multitude of security domains when managing access policy, each with a local scope of control (user, data). Ordinary users and administrators alike must authenticate to and establish sessions within different operating environments in order to exercise legitimate data service capabilities. Even if properly coordinated across operating environments, access control policies are not always globally enforced. An email application may, for example, distribute files to users regardless of an operating system's protection settings on those files. Moreover, special types of controls that are required over sensitive data can be especially difficult to implement in a piecemeal fashion across different operating environments.

Policy Machine Approach

To address these challenges, the PM offers a multiuser, enterprise-wide operating environment. The approach taken is to provide a generic access control mechanism that can implement computational capabilities of arbitrary data services, and displace access controls of different operating environments, with a single administrative domain and scope of control. Under this approach, enterprise computing goes from the need for multiple operating environments, each delivering different data services and access controls, to a single enterprise-wide operating environment with a single access control delivering all data services.

Like most other access control mechanisms, the PM comprises: (1) access control data that expresses access control policies and delivers capabilities of data services to perform operations on objects; (2) a set of administrative operations for configuring access control data; and (3) a set of functions for enforcing policy on requests to execute operations on objects and for computing access decisions to accommodate or reject those requests based on the current state of the access control data.

The PM is distinguished from other access control mechanisms by: (1) the data elements and relations that define its access control data; (2) the type of operations that are recognized; and (3) the functions that it implements. These factors are driven by a redefinition of access control and data services to integrate common and underlying elements, relations, and functions.



Access control provides the underpinnings for this integration. Data service capabilities are delivered to users through access control requests and policy is enforced over those requests, but only with respect to the operation and object types of the operating environment in which the access control is implemented. The key question is whether a single access control scheme can be generalized enough to support the operation and object types of arbitrary data services. To accommodate arbitrary data service object and operation types, the PM takes a data-centric approach. That is, the PM does not control access to data services, but to data service data types (e.g., documents, messages, work items), which are treated simply as objects that can be read and written. Data service operations (e.g., read, send, submit, approve, schedule) are implemented as combinations of read/write operations on data service data and administrative operations on access control data that alter the access state for which users can read/write data. In addition, the PM uniformly organizes the access control data and data service data using containers, which are instrumental in the distribution of data service capabilities to users and the expression and enforcement of policies. PM provides a systematic approach to the creation of administrative roles and delegation of administrative and read/write capabilities, beginning with a super user and ending with users with policy administrative and data service capabilities.

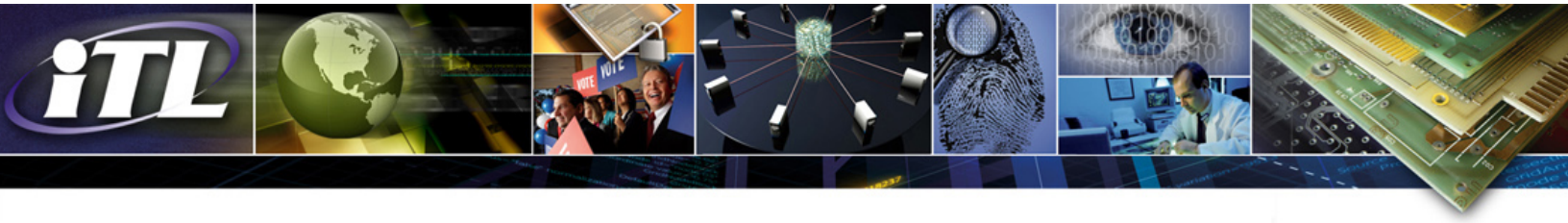
Some aspects of data service functionality are not handled by the PM. For example, operations such as spell checking, character manipulation, and user presentation pertain to specific methods for writing and reading data and must be implemented in data service logic outside of the PM's purview. Being able to ignore these methods makes the PM data type agnostic and the data of data services naturally interoperate.

Conclusion

The theme of advanced access control efforts is separation of policy from mechanism. We believe that the PM provides the next level of evolution through integration of access control and data services. PM was designed with this goal in mind, resulting in a general-purpose operating environment.

PM is a generic operating environment in the sense that through the reuse of the same access request interface, set of operational routines, access control data elements and relations, and functional components, arbitrary data services can be delivered to users, and arbitrary, mission-tailored access control policies can be expressed and enforced over executions of data service capabilities.

The practical benefits are many. Rather than a user having to authenticate to multiple operating environments to exercise legitimate data services capabilities, a user can access all of their data, regardless of type, in a manner consistent with policy, under a single authenticated session. Rather than administrators having to contend with a multitude of operating environment-specific security domains when managing access policy, the PM provides a single administrative domain and enables a systematic approach to the delegation of administrative capabilities, resulting in users with data service capabilities and policy management responsibilities. The data of data services interoperate and policy is comprehensively enforced over data services. For instance, a protected file attached to an email, or its



content pasted into the body of the message, could only be read by recipients in a manner consistent with the protection settings of the file. Finally, because the PM displaces access control features that are often implemented in application code to an underlying access control framework, those features can be made less susceptible to bypass and less vulnerable to attack.

PM is more than conceptual. Through its reference implementation, its features and capabilities have been shown to be viable. The current implementation is available from GitHub as an open source distribution to allow widespread experimentation and transfer. Example data services are provided with the distribution, and include email, file management, records management, work flow, cut/copy and paste, and several representative office applications.

References/Additional Resources

[1] David Ferraiolo, Serban Gavrila, and Wayne Jansen, NISTIR 7987, *Policy Machine: Features, Architecture, and Specification*, National Institute of Standards and Technology, Gaithersburg, Maryland, 109 pp., May 2014.

[2] *Information technology - Next Generation Access Control - Functional Architecture (NGAC-FA)*, INCITS 499-2013, American National Standard for Information Technology, American National Standards Institute, March 2013.

[3] Working DRAFT *Information technology - Next Generation Access Control – Generic Operations and Data Structures (NGAC-GOADS)*, INCITS 499-2013, American National Standard for Information Technology, American National Standards Institute, April 2014.

[4] David Ferraiolo, Serban Gavrila, Vincent Hu, and Rick Kuhn, *Composing and combining policies under the policy machine*, Symposium on Access Control Models and Technologies (SACMAT), Stockholm, Sweden, pp. 11-20, June 2005.

[5] David Ferraiolo, Vijayalakshmi Atluria, and Serban Gavrila, *The Policy Machine: A novel architecture and framework for access control policy specification and enforcement*, Journal of Systems Architecture, Volume 57, Issue 4, pp. 412-424, April 2011.

[6] Policy Machine Project Page

ITL Bulletin Publisher: Elizabeth B. Lennon
Information Technology Laboratory
National Institute of Standards and Technology
elizabeth.lennon@nist.gov