The Policy Machine

David Ferraiolo, Serban Gavrila, Steve Quirolgico
National Institute of Standards and Technology
301-975-3046
dferraiolo@nist.gov

Access Control 101

Authentication is the process of identifying an individual user, and his/her processes.

Access control or authorization is the process of controlling which users (and their processes) can perform which operations on which resources, in part based on those identities.

Each resource is protected under one or more access control policies

Each user (and process) possesses capabilities (ability to perform operations on resources) and prohibitions (denied capabilities) with respect to a policy expression

An access control mechanism grants or denies process access requests based on capabilities and prohibitions (enforces the current access state)

The access state can dynamically change as a consequence of successful user and process access requests

The Policy Machine

A logical “machine” comprising of a fixed set of data relations for the expression of any access control policy, and a fixed set of functions for making access control decisions, and enforcing policy based on that expression.

- ⇒ Standardized framework for enterprise specific policy specification and enforcement
- ⇒ Comprehensive protection of resources

Today's Access Control Paradigm

- Enterprises deploy a multitude of access control mechanisms, implemented at both the Operating System (OS) and application levels
- These come in a wide variety of forms, each with:
  - A method for authentication,
  - Access control data (expressing policy)
  - A set of functions for making access decisions and enforcing policy
  - A specific scope of control (over users and data),
  - A specific set of expressible and enforceable policies.

Policies are Complex

- Policy enforcement is instrumental in preventing the unauthorized disclosure of sensitive data, protecting the integrity of vital data, mitigating the likelihood of fraud, protecting privacy of individuals, and is what ultimately enables the sharing of information.
- Policy may dictate, for example that a user in accessing a resource: has a need-to-know, is appropriately cleared, is competent, has not performed a different operation on the same resource, the object was previously accessed by a different user, the user or the user's process is incapable of accessing other enterprise objects, or the user is only capable of accessing an object or any copy of the object in performance of a specific task.

Complexity (Cont.)

- Enforcement must pertain to processes (possibly malicious) that actually access data.
- Objects often need to be protected under multiple policies (e.g., Although a user may have access to a medical record through the role Doctor, other policies may come into play (a medical record may be classified, only accessible to doctors on a ward, or only under the discretionary permission of a primary physician))

Operating System

- Of the many access control models and specifications, today's OS are limited to the enforcement of discretionary access control (DAC), limited types of RBAC, and in very limited numbers, mandatory access control (MAC) policies.
- DAC and RBAC are weak and MAC is heavy handed.
- Other policies, "orphan policies" have no commercially viable OS for enforcement

Application level AC Mechanisms

- Access control mechanisms are commonly implemented within applications
  - Prominent among these applications are DBMS,
  - but access control mechanisms are also in small applications (e.g., enterprise calendars, time and attendance, etc.)
  - Many apps provide services through access control (e.g., email and workflow management)

What are the consequences?

- Need to administer a multitude of access control systems, each with a local scope of control (user, data)
  - Identity and access management is hard, costly, prone to error
- Policy is not comprehensively enforced (e.g., although a file management system may narrowly limit access to a file, chances are that file can be copied to or attached to a message and mailed to anyone in the organization or the world.)
- Information can be “leaked” to unauthorized users
- Copies of sensitive data can’t be tracked or controlled
- Many types of information (e.g., PII data, classified data, medical records, financial information) can’t be appropriately protected, or require specialized operating systems or applications
- …

What is new?

- One generic mechanism for comprehensive enforcement of many policies
- Can protect resources any combination of currently configured policies
- No decision making or enforcement at application level
- A single framework with a single administrative domain and scope of control that extends over a multitude of OSs and applications

How does the PM work

- User logs on to the PM,
- PM logically presents the user with all his/her accessible resources (e.g., files, inbox, work items …)
- User requests access to resources through a process
- PM mediates the access to resources by those processes based on capabilities derived through user and object attributes, and process and user prohibitions (capability denies).
- Machine state can dynamically changes as users and processes access resources
- Note: Policy is created through data configuration alone

PM Data, Relations, and Functions

- Data Sets (e.g., users, attributes, objects)
- Relations
  - Permissions
  - Prohibitions
  - Event Pattern-Response (obligations)
- Functions
  - Authentication
  - Session management
  - Reference mediation

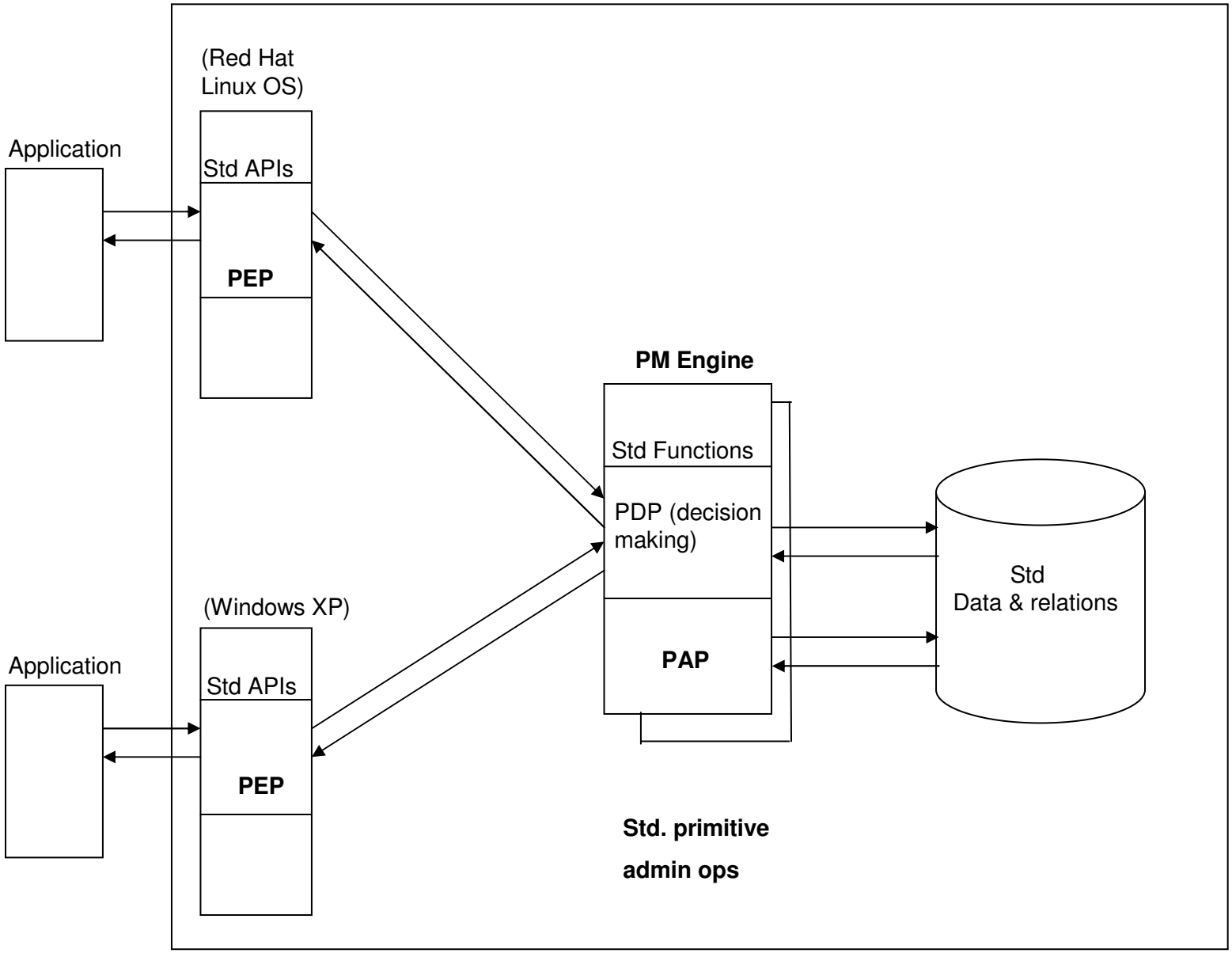
Define the current process access state

Define the overall policy state

Architecture

- The PM standard recognizes policy enforcement points (PEP), Policy Decision Point(s) (PDP), a Policy Administration Point (PAP), a policy database, and two types of enterprise applications - those applications that afford services in the absence of access control (e.g., Word, Power Point) and those applications that afford services through the use of access control (e.g., email, workflow management).

Our Policy Machine Implementation



State of PM Development

- PM Specification completed
  - Data relations, functions and algorithms
- Working Reference implementation
  - PM enforcement functions are implemented in an OS Kernel simulator (PM Sim.)
  - Variety of applications
- Working with INCITS (Std development)
- Moving toward Tech Transfer and Pilot Deployment

Conclusion 1: Benefits to the Vendor

- OS Vendor
  - No need to change system to accommodate the policy de jour,
  - No need to cater to special needs of different user communities
  - No need to make access control decisions, or maintain or manage access control data
- Application developers
  - No need to provide functionality for making access control decisions or policy enforcement (up to 60% of logic)
  - No need to maintain or manage access control data

Conclusion 2: Benefits to the User

- General Purpose Protection Machine (one mechanism fit for many purposes)
- Large library of policies available for immediate configuration
- Naturally provides interoperability and single sign-on
- Operational Assurance
  - Can render many Trojan horse attacks harmless
  - No enforcement or decision making at the application level
- Fine-grained, flexible and comprehensive protection
- Promotes greater sharing of information (through protection)
- Promotes greater sharing of computers (through logical access)
- Can prevent “leakage” of sensitive data to unauthorized principals, (e.g., through email, and storage devices (hard-drives, memory sticks))
- Can track and control copies of information (under the same policies as the original)
- Truly secure application services through access control (email, workflow management)

Questions ?

David Ferraiolo

301-975-3046

dferraiolo@nist.gov