APPLICATION NO. 11,062,004

LICENSING OPPORTUNITY:
A METHOD AND SYSTEM FOR CENTRALIZED ABAC POLICY ADMINISTRATION AND LOCAL POLICY DECISION AND ENFORCEMENT USING ACCESS CONTROL LISTS

DESCRIPTION

Problem

An Access Control List (ACL) is a simple mechanism, dating back to the 1970s and remains in widespread use for the protection of system resources of varying types (e.g., files and directories). Resources are associated with an ACL that stores lists and groups users along with their approved rights (e.g., read, write) for controlling access to those resources.

Benefits:

- Extremely fast
- Easy to determine user access rights to a resource

Drawbacks:

- Difficult to directly update and manage
- Difficult to enforce modern-day access policies
- Difficult to determine and manage the access capabilities of users

Invention

An Access control method where user requests to perform operations on resources are granted or denied based on attributes assigned to users, attributes assigned to resources, and a set of policies that are specified in terms of those attributes.

Benefits:

- Ease in management of access policies
- Enforcement of sophisticated policies
- Ease in determining the access capabilities of users

Drawbacks:

- Less efficient than ACLs in computing access decisions and enforcing policy
- Challenging and costly to integrate into existing systems

BENEFITS

Competitive Advantage

Benefits of ACLs and ABAC without their drawbacks.

When a user enters on duty or when a user's job function, authority, affiliation, or any other characteristic changes, an administrator simply assigns/reassigns the user to appropriate attributes, and corresponding ACLs are automatically created and/or updated.

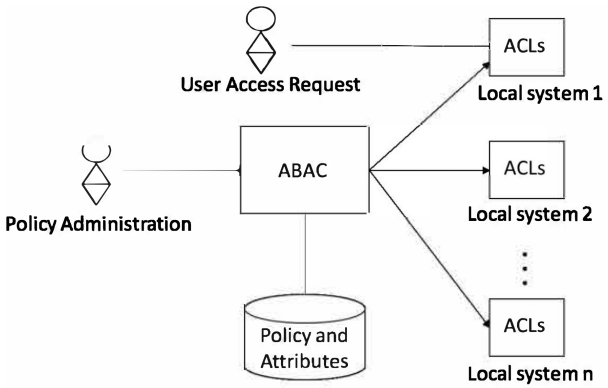


Diagram showing centralized ABAC policy managed and local enforced of the ABAC policy in local systems using native ACL mechanism.

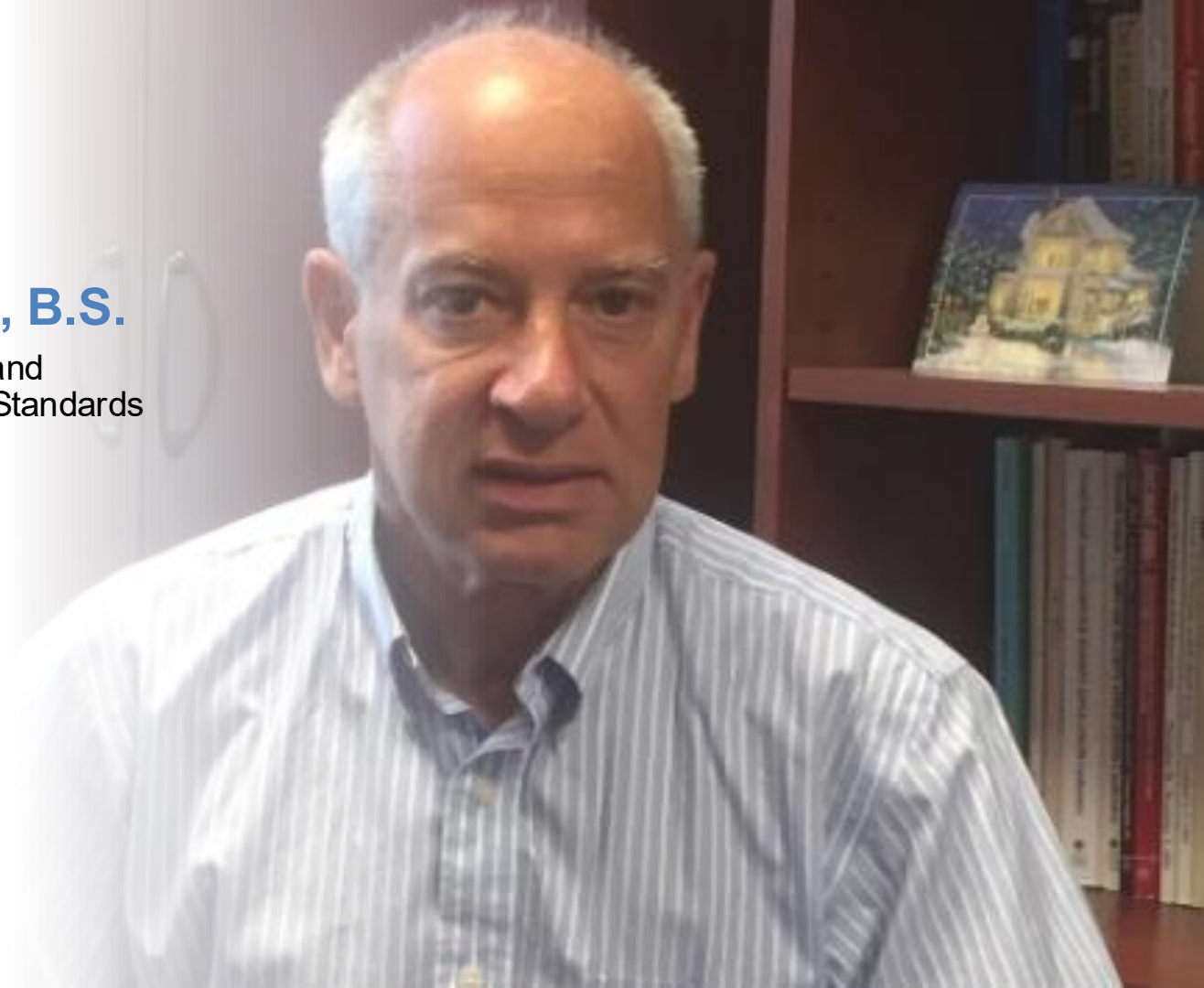
Contact: licensing@nist.gov

NIST TECHNOLOGY PARTNERSHIPS OFFICE

NIST Technology Partnerships Office
National Institute of Standards and Technology
100 Bureau Drive, Gaithersburg, MD 20899-2200

Meet **David Ferraiolo, B.S.**

Group Leader, Secure Systems and Application, National Institute of Standards and Technology



A method and system for centralized ABAC policy administration and local policy decision and enforcement using access control lists

David Ferraiolo and Gopi Katwala
Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technologies

Innovative Technology Showcase #2
March 13, 2023

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY U.S. DEPARTMENT OF COMMERCE

Invention

- Method of automatically creating and updating Access Control Lists (ACLs) based on Attribute Based Access Control (ABAC) policies
- Bridge the two access control approaches
- ABAC policies are centrally managed in an ABAC system and the ABAC policies are locally enforced in systems using their native ACL mechanisms

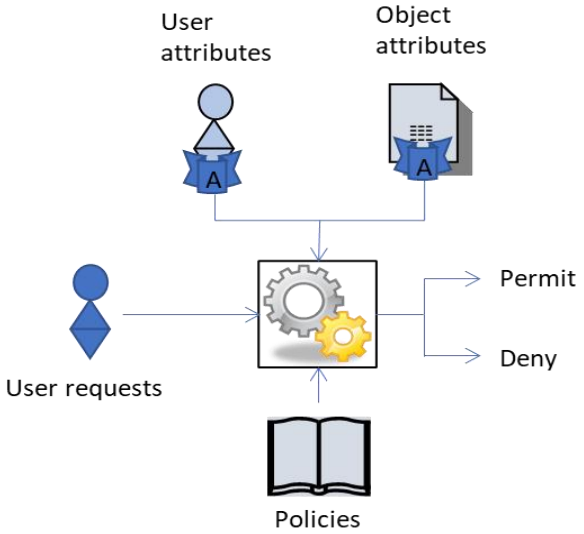
**Policies are expressions of access control rules*

Access Control Lists (ACLs)

- A list of individual users and/or groups of users with their access rights (e.g., read, write) attached to a system resource (e.g., file or directory)
- Date back to the early 1970s and **remain in widespread use today** for controlling access to system resources
- Benefits
  - Extremely fast in computing access decisions
  - Easy to review user access rights to a resource
- Drawbacks:
  - Difficult to directly update and manage
  - Difficult to enforce modern day access policies
  - Difficult to determine and manage access capabilities of users

Attribute-based Access Control (ABAC)

- Latest milestone in an evolution of authorization approaches
  - ACLs → RBAC → ABAC → ABAC/NGAC*
- Access is granted/denied based on user attributes, object attributes, and a set of policies specified in terms of those attributes
- Benefits:
  - Ease in management of access policies
  - Enforcement of sophisticated policies
  - Efficient in reviewing access capabilities of a user and user access rights to a resource
- Drawbacks:
  - Less efficient than ACLs in computing access decisions
  - Challenging and costly to directly integrate into existing systems



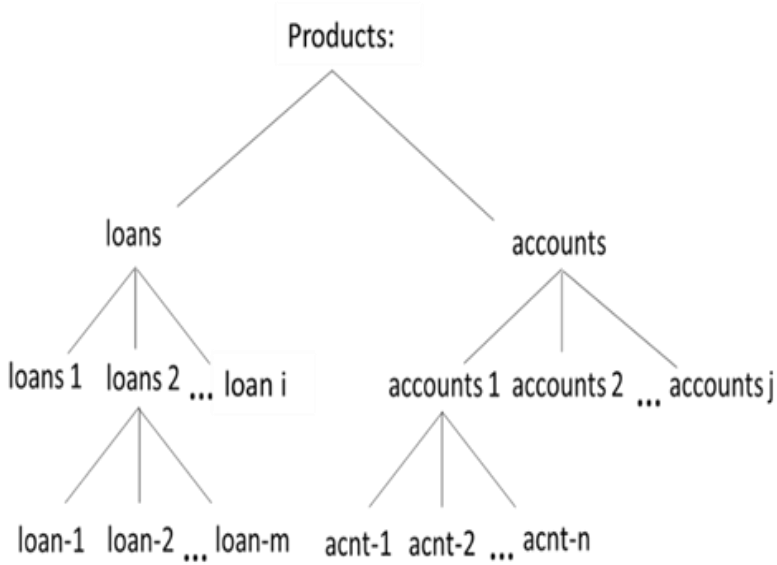
*Next Generation Access Control (*NGAC) is an ANSI/INCITS Standard (565) with open-source implementations*

Method

1) Centrally express an ABAC (NGAC) policy that conforms to the access control rules of the enterprise
2) Introduce representations of local host resources (file, directories) needing protection into the ABAC policy expression as objects or object attributes
3) Maintain a correspondence between ABAC representations of resources and the actual resources
4) Formulate ACLs for representations through policy reviews in accordance with the ABAC policy
5) Create: user accounts (if needed), groups, and ACLs on local resources using the ACL data of their corresponding representations
6) As the ABAC policy configuration changes, update the ACLs on affected representations and automatically update accounts, groups, and corresponding ACLs on local resources
7) Operationally, users attempt to access resources in local host systems, and the ABAC policy is enforced in those systems in terms of their native ACL mechanisms

Example ABAC Bank Policy

- Tellers can read and write accounts in all branches
- Tellers can create and delete accounts in the branches to which they are assigned
- **Loan Officers can read and write loans in all branches**
- **User u2 (a Loan Officer) cannot write to loan-1 (u2's loan)**
- **Loan Officers can only create and delete loans in the branches to which they are assigned**
- An Auditor can read account and loan products in all branches

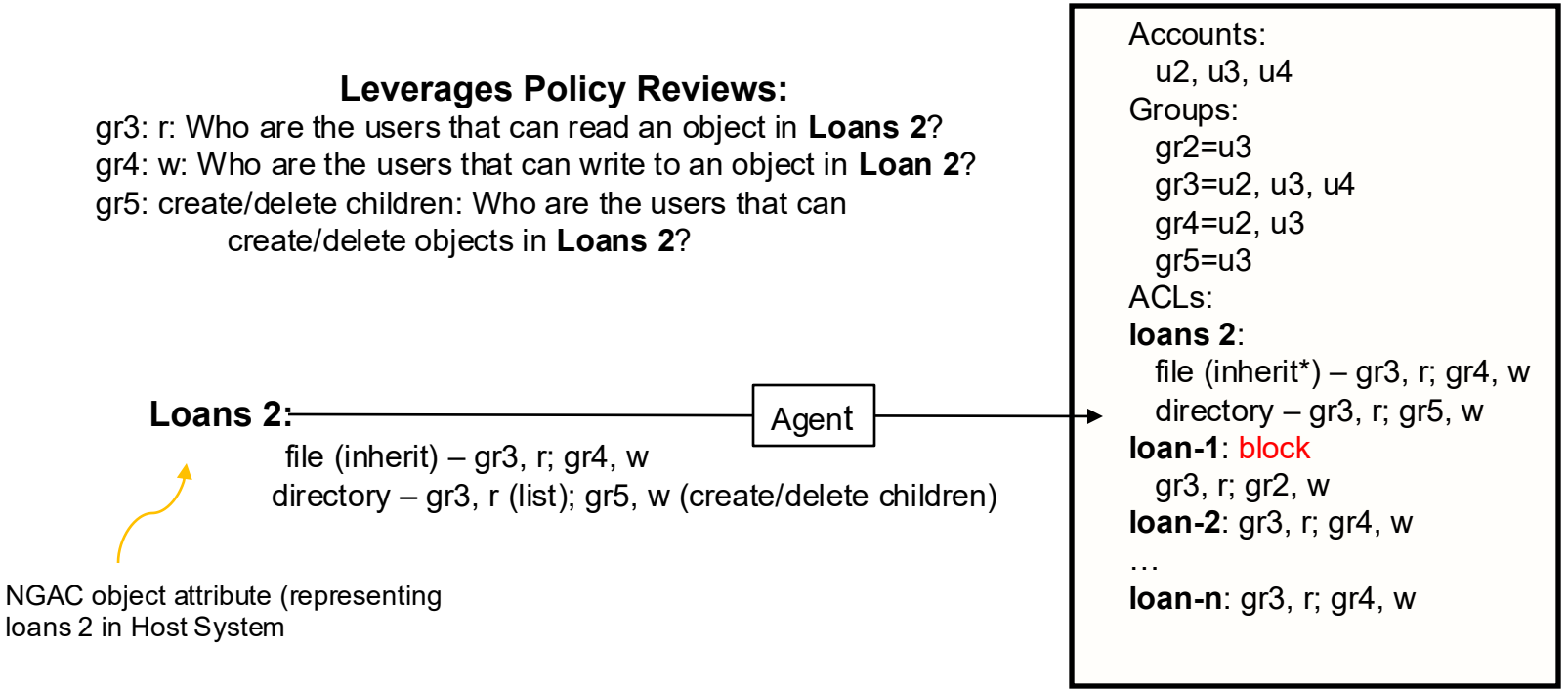


Directory Structure

Under the hood: generate ACLs for host resource loans 2 (loans in branch 2)

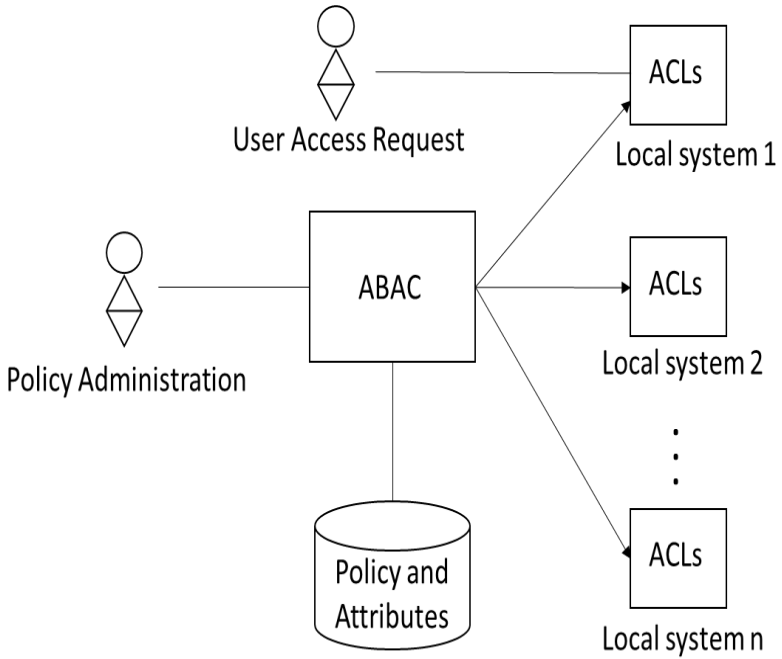
Leverages Policy Reviews:

gr3: r: Who are the users that can read an object in **Loans 2**?
gr4: w: Who are the users that can write to an object in **Loan 2**?
gr5: create/delete children: Who are the users that can create/delete objects in **Loans 2**?



In Operation

- Administrators centrally express/manage ABAC policies and introduce representations of local resources into the expression, automatically creating/updating ACLs in local systems
- Users attempt to access resources in local systems, and ABAC policies are enforced in those systems in terms of native ACL mechanisms



Summary/Conclusion

- Best of both ABAC and ACLs
  - Simpler authorization management than direct management of ACLs
  - Policy support goes far beyond what is feasible through direct management of ACLs.
  - Enforcement of ABAC policies at the speed of ACLs
  - Policy analytics beyond what is feasible through ACLs
  - Enforces ABAC policies in local systems with minimal changes to those systems