

Adage

by Rich Simon, Mary Ellen Zurko, et. al.

The Open Group Research Institute

<http://www.camb.opengroup.org/www/adage/>

presented by James Hoagland

hoagland@cs.ucdavis.edu

Security Lab Seminar, 99-01-20

Computer Security Research Laboratory

Department of Computer Science

University of California, Davis

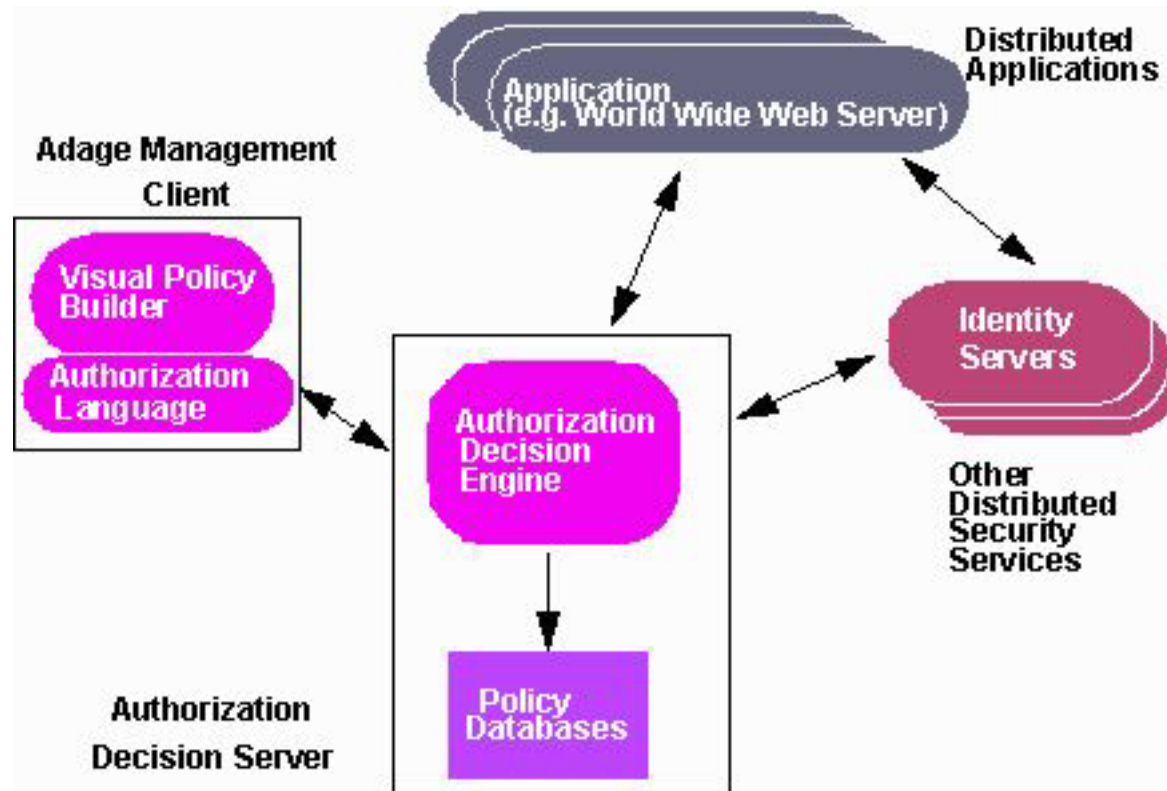
Outline

- Adage overview
- Policy context in Adage
- Policy description in Adage
- Conclusions

Adage Overview

Adage= Authorization for Distributed Authorization and Groups

□ DARPA sponsored project; 11/95 through 7/98



Objectives & Benefits (from web page)

- ❑ Leverage existing work in distributed authentication and communications security
- ❑ Develop new approaches to policy-neutral, distributed authorization
- ❑ Provide high-level, user-friendly authorization language and visual policy builder for ease of use
- ❑ Develop service for rich, flexible groupings and roles for subjects, objects and actions
- ❑ Design an Authorization Trust Model that determines which external information sources are trusted to have input to local authorization decisions
- ❑ Provide consistent mechanisms and policies that can be shared across applications within an enterprise
- ❑ Provide a single Authorization Enforcement Engine for use by all components

Other Goals and Approaches

- ❑ user-centered security
 - emphasize usability
 - so that it will be deployed
 - so that it will be used
 - see the conference paper
- ❑ want easy correspondence between natural language model and its expression in Adage terms
 - so, Adage protection mechanisms are designed to reflect real world security policies and concepts

Outline

- ✓ Adage overview
- Policy context in Adage
- Policy description in Adage
- Conclusions

Policy Implementation and Enforcement (1)

- ❑ a *cell* is the basic administrative domain of Adage
- ❑ cells have a consistent site policy in place
- ❑ other cells are considered non-local

Process:

- ❑ Security Administrator in a cell creates a natural language version of a security policy
- ❑ the SA uses the *Adage Visual Policy Browser* GUI to state the policy to Adage
- ❑ the VPB translates its policies into equivalent *Authorization Language* statements
 - AL is text-based and more program-like
 - advanced users can work at the AL level directly

Policy Implementation and Enforcement (2)

- ❑ AL is sent to the *Authorization Database Server* to create entries in the *User Authorization Database*
 - UAD is a collection of policy descriptions for a cell
- ❑ UAD is translated into the *Engine Authorization Database*
 - semantically equivalent but designed for faster access for authorization decisions
- ❑ the *Authorization Engine* makes the access control decisions based on the EAD
 - invoked by applications to make access control decisions
 - access is denied if no relevant rules are found (considered an error)
 - all rules must succeed for access to be granted

Adage Trust Model

- ❑ several authentication authorities may identifies actors
- ❑ rules describe for what policies/actions/targets authentication authorities may be used, based on trust metrics
- ❑ 3 trust metrics measure trust based on how a system comes to know an authentication authority or principal
- ❑ citizenship metric (base trustworthiness)
 - citizenship model: native, foreign, tourist, alien, suspect, enemy
- ❑ trusted content (what it is trusted to provide)
 - all, ID info, referrals, nothing
- ❑ quality of referral (how much a link is trusted to refer others)
 - unimpeachable, trustworthy, plausible, unreliable, hostile

Outline

- ✓ Adage overview
- ✓ Policy context in Adage
- Policy description in Adage
- Conclusions

Visual Policy Browser

- ❑ GUI for creating and describing Adage authorization rules
- ❑ runs on Windows NT
- ❑ two work areas:
 - browsing space to display/select different parts of the policy
 - workspace for manipulating policy building blocks, each represented by an icon

- ❑ authorization *rules* are made up of 3 building blocks:
 - actors
 - targets
 - regulations
- ❑ a *clause* is a heterogeneous group of these

Actors

- ❑ *actors* are human users and computer entities that can make requests
 - i.e. subjects and principals in security literature
- ❑ *attributes* are associated with each actor
 - label (i.e. compartments)
 - group membership
 - security level, integrity level
 - administrator-defined named label sets (i.e. country of citizenship)
- ❑ *team*: a group of actors
- ❑ *actor template*: a partial description of an actor
 - describes a kind of actor, for creating new ones or for use in rules
- ❑ external authentication authorities introduce authenticated identities of its users
 - any authentication authority may be integrated with Adage

Targets

- ❑ a *target* is an information container or other application abstraction that an actor may request access to
- ❑ targets have attributes (same kinds as actors)
- ❑ *collection*: a group of targets
- ❑ *target template*: a partial description of a target

Applications

- ❑ targets are relative to the application that maintains its namespace
- ❑ applications register with cell upon installation including:
 - application actions available
 - (optionally) actor and target templates, clauses, regulations

Regulations

- ❑ *regulations* are the actions actors are allowed to take on targets
- ❑ always with respect to a given application
- ❑ *manual*: a group of regulations
- ❑ regulations consist of a set of constraints to be satisfied
 - each constraints is for a particular action
 - can state “disallowed”
 - can state constraints between attributes of actors and targets
- ❑ *regulation templates* are used as filters or prototypes

Rules

- ❑ rules contain:
 - at least 1 actor or actor template
 - at least 1 target or target template
 - at least 1 regulation or regulation template
- ❑ for example:
 - “all actors”
 - “all targets”
 - “secrecy level” of the TheActor \geq TheTarget
- ❑ each rule is marked as active or latent
- ❑ the set of active rules is the site policy

Advanced Policy Description Features

Available through VPB

- ❑ NoOverlap constraint between 2 teams
- ❑ AtMost constraint, restricting team maximum membership

Mentioned in earlier technical reports, but not clearly in 7/28/98 reports

- ❑ history-based constraints
 - <actor> HasDone <action> To <target>
 - <actor> NeverDid <action> To <target>
 - <actor> NeverUsed <target>
- ❑ time of validity for regulations
- ❑ multi-person policies (“one from each group”)
- ❑ delegation of identities

Adage Protection Mechanisms

- ❑ the protection mechanisms implemented in the Authorization engine
- ❑ based on access matrix model and role-based access control, enhanced with rules
- ❑ not a 1-1 correspondence to VPB/AL concepts, i.e.
 - VPB actor \Leftrightarrow each authenticated identity for use, as own principal
 - VPB regulation \Leftrightarrow might be used in several rules
 - roles not exposed at UI level
 - groups & namespaces flattened
- ❑ *roles* consist of <a team, a collection, a set of actions> which (by default) means that each <actor, target, action> therein is allowed
- ❑ rules have 4 parts at this level:
 - the scope for each of the principal, action, and target
 - a constraint, the criteria for a rule to pass

Outline

- ✓ Adage overview
- ✓ Policy context in Adage
- ✓ Policy description in Adage
- Conclusions

Some Conclusions

Some apparent successes

- ❑ VPB makes stating policies user-friendly
- ❑ a generic policy engine for applications

Some unanswered questions

- ❑ What is the range of (kinds of) policies that can be stated using Adage? What is its expressability?
- ❑ How can we be sure that the full semantics of policies are preserved across translations?