Provenance-Based Access Control (PBAC)

Prof. Ravi Sandhu
Executive Director and Endowed Chair

April 15, 2016

ravi.sandhu@utsa.edu
www.profsandhu.com
Ultimate Unified Model

Attributes

Security
Access Control
Trust
Risk

Relationships

Provenance
What is Provenance?

Art definition of provenance

- Essential in judging authenticity and evaluating worth.

Data provenance in computing systems

- Is different from log data.
- Contains linkage of information pieces.
- Is utilized in different computing areas.
Provenance Data

- Information of operations/transactions performed against data objects and versions
  - Actions that were performed against data
  - Acting Users/Subjects who performed actions on data
  - Data Objects used for actions
  - Data Objects generated from actions
  - Additional Contextual Information of the above entities

- Directed Acyclic Graph (DAG)
- Causality dependencies between entities (acting users / subjects, action processes and data objects)

- Dependency graph can be traversed for the discovery of Origin, usage, versioning info, etc.
Provenance-aware Systems

- Capturing provenance data
- Storing provenance data
- Querying provenance data
- Using provenance data
- Securing provenance data

Provenance Data Model
Access Control
Access control in Provenance-aware Systems

- **Provenance Access Control (PAC)**
  - Controlling access to provenance data which could be more sensitive than the underlying data
  - Needs access control models/mechanisms (e.g., RBAC)
  - (Meaningful) control granularity? Right level of abstraction?

- **Provenance-based Access Control (PBAC)**
  - Using provenance data to control access to the underlying data
  - Provenance-based policy specification
Access control in Provenance-aware Systems

PBAC
Extended PBAC
Base PBAC

Access control

PAC
Sanitization/Filtering/Redaction/…
Role-based PAC
Prov-based PAC

Common Foundations:
Base Provenance Data, DName (named abstraction) and matching DPath (Dependency Path Pattern)
Open Provenance Model Example

- **Cake**
  - **100g Sugar**
  - **100g Flour**
  - **100g Butter**
  - **Two Eggs**
  - **John**

Relations:
- **wasDerivedFrom**: Cake
- **wasGeneratedBy**: Bake
- **used**: Bake
- **wasControlledBy**: John
Provenance Data Model

Subject (agent) \( \rightarrow \) Action (process) \( \rightarrow \) Attribute \( \rightarrow \) Object1 (artifact) \( \rightarrow \) Object2 (artifact)

- \( c \): wasControledby
- \( u \): used
- \( g \): wasGeneratedBy
- \( t \): hasAttributeOf

Attrb. edge \( \rightarrow \) Dep. edge
Direct vs. Indirect Dependencies

**Direct dependencies**
- Used (u), wasGeneratedBy (g), wasControlledBy (c)
- Captured from transactions as base provenance data

**Indirect dependencies**
- System-computable dependencies
  - using pre-defined dependency names and matching dependency path patterns
- User-declared dependencies
  - using pre-defined dependency names
Object Dependency List (DL₀)

- A set of pairs of
  - abstracted dependency names (DNAME) and
  - regular expression-based object dependency path patterns (DPATH)

- Examples
  - `<wasSubmittedVof, gSubmit.uInput>`
  - `<wasAuthoredBy, wasSubmittedVof?.wasReplacedVof *gUpload.c>`
A Sample Base Provenance Data
Sample Object Dependency List ($DL_o$)

- $< \text{wasReplacedVof}, \text{greplace}.u_{\text{input}} >$
- $< \text{wasSubmittedVof}, \text{gsubmitt}.u_{\text{input}} >$
- $< \text{wasReviewedOof}, \text{greview}.u_{\text{input}} >$
- $< \text{wasReviewedOby}, \text{greview}.c >$
- $< \text{wasGradedOof}, \text{grade}.u_{\text{input}} >$
- $< \text{wasAuthoredBy}, \text{wasSubmittedVof}?\cdot\text{wasReplacedVof} \ast .\text{gupload}.c >$
- $< \text{wasReviewedBy}, \text{wasReviewedOof}^-1.\text{wasReviewedOby} >$
A Sample Base Provenance Data

1. wasReviewedOby
2. wasSubmittedVof
3. wasReviewedOof
4. wasGradedOof

DL: < wasReviewedOby, greview, au3 >
< wasSubmittedVof, gsubmit, ginput >
< wasReviewedOof, greview, au3 >
< wasGradedOof, ggrade, ginput >
< wasReplacedVof, greplace, ginput >
wasAuthoredBy

$\text{DL}_O: \langle \text{wasAuthoredBy, wasSubmittedVof, wasReplacedVof} \rangle$

$\ast. g_{\text{upload}}. c >$
A Sample Base Provenance Data

DL₀: < wasReviewedBy, wasReviewedOof⁻¹, wasReviewedOby >
1. Anyone can upload a homework.
2. A user can replace a homework if she uploaded it (origin-based control) and the homework is not submitted yet.
3. A user can submit a homework if she uploaded it and the homework is not submitted already. (workflow control)
4. A user can review a homework if she is not the author of the homework (DSOD), the user did not review the homework earlier, and the homework is submitted already but not graded yet.
5. A user can grade a homework if the homework is reviewed but not graded yet.
PBAC\textsubscript{b} Model Components

- Acting Users (AU)
- Request \((au,a,o)\)
- Actions (A)
- Objects (O)
- User Authorization
- Action Validation
- Access Evaluation
- Dependency List (DLIST)
- Policies (P)
- Provenance Data (PD)

Access
Decision Activity
Utilized by
PBAC\textsubscript{B}: A Base Model

- System-captured Base Provenance Data only
  - Using only 3 direct dependencies (u, g, c)
  - No user-declared provenance data

- Object dependency only

- Policy is readily available
  - No policy retrieval required
A Family of PBAC Models

Combined Models

PBAC_U

PBAC_A

PBAC_B

PBAC_PR
PBAC Model Components

- Acting Users (AU)
  - Request (au, a, O)
- Actions (A)
  - Action on O
- Objects (O)

User Authorization

Access Evaluation

Dependency Lists (DL)
- Object DL (DLc)
- Acting User DL (DLAU)

Provenance Data (PD)
- Policies (P)
- Base Prov. Data (PDb)
- User-Declared Prov. Data (PDU)

P Retrieval Policies (PRP)

Access
Decision Activity
Utilized by

Policy Retrieval