Enhancing Database Access Control with XACML Policy

Sonia Jahid, Imranul Hoque, Hamed Okhravi, Carl A. Gunter

seclab.uiuc.edu

PROBLEM AND SOLUTION

PROBLEM:

- Database access control mechanism does not support Attribute-based Access Control (ABAC) by utilizing information in the database itself.

SOLUTION:

- eXtensible Access Control Markup Language (XACML) is widely accepted to describe attribute-based policies.
- Native database access control mechanism such as access control list (ACL) is efficient in enforcing Identity-based Access Control (IBAC).
- Transfer high level XACML policies into low level database ACLs to unify ABAC and IBAC.

MOTIVATION AND CHALLENGES

MOTIVATION:

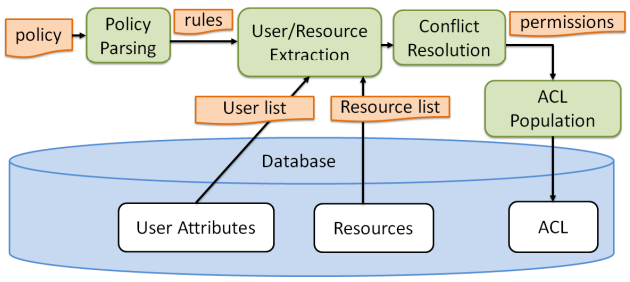
- **EXPRESSIVENESS AND EFFICIENCY:** XACML provides expressiveness while ACL provides efficiency. Utilize the advantages of both the world.
- **MANAGEMENT:** Privilege changes are taken care of by the attributes and policy.
- **SECURITY:** ACLs enforce policy inside the database, the final point of policy enforcement.

CHALLENGES:

- Maintain permission correctness when database changes.

OVERVIEW

Professors or Systems area Assistant professors → Can select from, insert into, and delete from → Graduate Admission related tables



POLICY PROCESSING

- Subject: <rank=Professor> OR <rank=Assistant Professor AND area=Systems>
  Resource: <table_comments=Graduate Admission>
  Action: <select, insert, delete>

EXTRACT USERS AND RESOURCES:

- SELECT username FROM employees.info
  WHERE rank='Professor' OR (rank='Assistant Professor' AND area='Systems');
- SELECT table_name FROM information_schema.tables
  WHERE table_comment='Graduate Admission';

POPULATE ACL

- Users u_0, u_5, and u_7 and tables t_5 and t_8 satisfy the policy.
- GRANT SELECT,INSERT,DELETE ON t_5 TO u_0, u_5, u_7;
  GRANT SELECT,INSERT,DELETE ON t_8 TO u_0, u_5, u_7;

CONFLICT RESOLUTION

CONFLICT (Subject S, Action A, Resource R, Policy P):

- P_1 permits actions A_1 to users S_1 on resources R_1.
  P_2 denies actions A_2 to users S_2 on resources R_2.
- $S1 \cap S2 = S (S \neq nil)$, $R1 \cap R2 = R(R \neq nil)$, $A1 \cap A2 = A(A \neq nil)$.

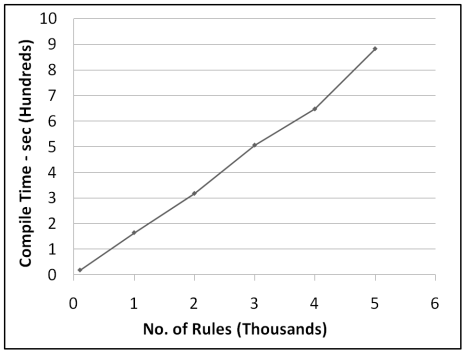
COMBINING ALGORITHMS:

- Permit/Deny-overrides, First-applicable.

EXAMPLE:

- Permissions: <u_0, t_0, select, permit>, <u_0, t_0, insert, permit>, <u_0, t_0, select, deny>
- Combining Algorithm: Deny-overrides.
  - The 1st permission is nullified by the 3rd one.
  - The 3rd permission is revoked from the ACL if it already exists.
  - The 2nd permission is added to the ACL.

EVALUATION



- Implementation of XACML over MySQL database.
- Compilation time includes parsing, user extraction, and ACL update.
- Time is almost linear with number of rules in a policy.

Information Trust Institute
University of Illinois at Urbana-Champaign
www.iti.illinois.edu