Access Control Policy Mining and Management

Scott D. Stoller, Stony Brook University
http://www.cs.stonybrook.edu/~stoller/policy-mining

Mining ABAC Policies

- Attribute-Based Access Control (ABAC) is becoming more common and more important, due to its flexibility and ease of administration, especially for complex, dynamic policies.
- An ABAC policy contains rules that grant permissions based on attributes of users and resources.
  - Example rule (informal): A developer or tester can read and request to work on a non-proprietary task if his/her assigned projects include the project the task is in, and his/her areas of expertise include the task's required areas of expertise.
- Policy mining algorithms can drastically reduce the cost of migrating from legacy access control to ABAC, by partially automating construction of an ABAC policy.
- We developed the first algorithms to mine an ABAC policy from attribute data together with lower-level information about entitlements, which can be:
  - Access Control Lists (ACLs), or
  - Role-Based Access Control (RBAC) policy, or
  - Logs (often maintained for auditing, etc.)
- The goal is to find the highest-quality (e.g., most concise) policy consistent with the given data.
- Mining from logs is harder, because logs provide incomplete information about the policy. Our algorithm attempts to "fill in" appropriate entitlements not reflected in the logs.

Algorithm

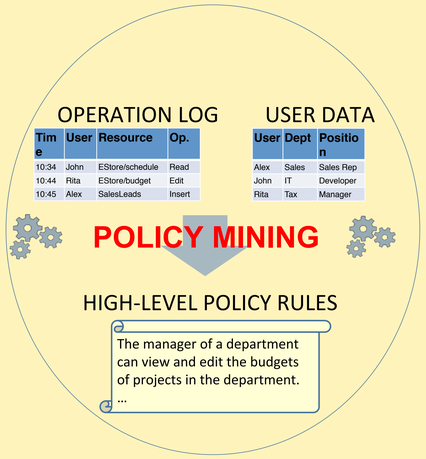
- Iterate over the given entitlements, use selected entitlements as seeds to construct candidate rules, and generalize candidate rules to cover more entitlements. Merge and simplify candidate rules. Select the highest-quality candidate rules for the generated policy.
- The problem is NP-hard. This is a greedy heuristic.
- Algorithm also tries to detect and correct noise in inputs.

Evaluation

- Methodology: Generate ACLs and synthetic logs from ABAC policies. Add noise in some experiments. Apply mining algorithm. Compare original and mined policies.
- Results: Algorithm can mostly or completely reconstruct the original policy, even with up to 12% noise or 20% log incompleteness.

Project Goals

- Access control policies are increasingly large, complex, decentralized, and dynamic.
- Administrators need better tools that help them develop correct policies.
- This project is developing new techniques for policy mining and policy analysis.



Mining Temporal RBAC Policies

- Role-Based Access Control (RBAC) is a widely used access control paradigm today.
- Temporal RBAC extends RBAC to limit the times at which roles are enabled. Repeating time intervals are represented by periodic expressions.
- Several commercial tools support role mining but not temporal role mining.
- We developed first temporal role mining algorithm that
  - produces hierarchical policies, and
  - optimizes multiple metrics, including
    - policy size: # roles + # user-role assignments + …
    - policy interpretability: how well role membership can be characterized in terms of user attributes
    - compound metrics, e.g., size and interpretability
- Algorithm was evaluated and shown effective on datasets based on real-world ACL policies.
- Best Paper Award, DBSec 2016.

Relationship-Based Access Control

- Relationship-Based Access Control (ReBAC) originated in social networks and is spreading to general-purpose computing systems.
- It extends ABAC with relationships between objects (subjects, resources, …) which can be used and chained together in path expressions in access control rules.
  - Example rule (informal): Technicians working for a contractor can view tasks associated with active contracts for customers of the technician's department.
- ReBAC is more flexible than ABAC and well-suited for access control in object-oriented systems.

Mining ReBAC Policies

- We are developing a ReBAC policy mining algorithm for a general-purpose object-oriented ReBAC framework.
- ReBAC mining is harder than ABAC mining, because there are many more legal path expressions than attributes.
- The object-oriented type system, limits on path length, and heuristic quality metrics help reduce the search space.
- Entropy-based attribute selection might also help.
- Evaluation and refinement of the algorithm are driven by two large case studies.

Managing ReBAC Policies

- Several general-purpose ReBAC models were proposed, but none fully addressed policy administration.
- We proposed the first general-purpose ReBAC model with a comprehensive administrative model. It controls changes to all aspects of the policy, including adding and deleting objects, relationships, and access control rules.
- Controlling addition and removal of rules is challenging, because of the tension between providing flexibility for administrators and ensuring global safety of the policy.
- We define a strictness order on rules and specify in the policy the least strict rules that each subject is allowed to add or remove.
- Future work will consider policy language extensions and verification of ReBAC policies.

Interested in meeting the PIs? Attach post-it note below!

National Science Foundation
WHERE DISCOVERIES BEGIN

NSF Secure and Trustworthy Cyberspace Inaugural Principal Investigator Meeting
Nov. 27 -29th 2012
National Harbor, MD

Stony Brook University