PolicyReplay:
Misconfiguration-Response Queries for Data Breach Reporting

Daniel Fabbri*, Kristen LeFevre*, Qiang Zhu+
*Electrical Engineering and Computer Science, University of Michigan
+Computer and Information Science, University of Michigan - Dearborn

Motivation:

- Databases use access control policies to restrict access
- Unfortunately, these access policies are complex
- As a result, the policies are sometimes misconfigured
- Users may read or modify data inappropriately

- Recent legislation increases requirements to report breaches
  - American Recovery and Reinvestment Act of 2009 increases HIPAA reporting rules for hospitals

Preliminaries:

- The database is a transaction-time backlog DB
- The database stores a log of all operations executed on the DB

Problem:

Find operations that were affected by the misconfiguration

Misconfiguration-Response (MR) Queries

Returns all queries that were substantively affected by the misconfiguration

Naïve Solution:

1. Go back in time to when the misconfiguration started
2. Apply the correct access control policies
3. Replay all operations on a **copied** database
4. Compare query results under the old and new policies
5. If the results differ, then the query is **suspicious**

Weaknesses:

- Replays all operations
- Copies the entire database

PolicyReplay Optimizations

1. **Pruning**: Reduce the number of operations re-executed

Static Pruning (Queries Only):

- Delta Expressions: Change in visibility of a row due to the policy misconfiguration
- Analyze the query's selection condition
- Determine, without re-executing, that the query is not suspicious

Delta Tables (With Updates):

- For each table, stores the row-wise difference between the old policy and the new policy.

Pruning Condition for an Operation:

1. Statically prunable
2. Does not use rows from the delta tables

Putting It All Together

1. Create empty delta tables
2. Replay all operations (no need to copy the DB)
3. If the operation is prunable, skip.
4. Otherwise, re-create the DB from the old DB and the delta tables.
5. Re-execute the operation
   a) Determine if a query is suspicious
   b) Update the delta tables

2. **Re-Execution**: Reduce re-execution costs

Re-Execution:

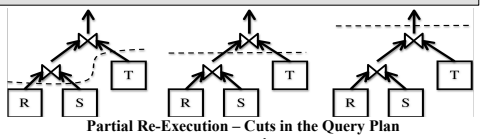
- When an operation cannot be pruned, the operation must be re-executed

Partial Re-Execution:

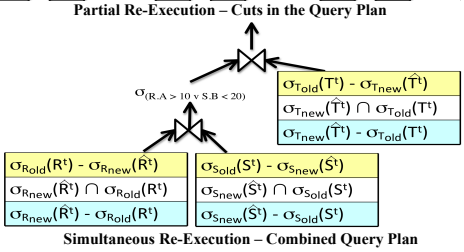
- Determine mid-execution that an operation is unsuspicious

Simultaneous Re-Execution:

- Execute the query on the old policy and new policy simultaneously (shared computation)

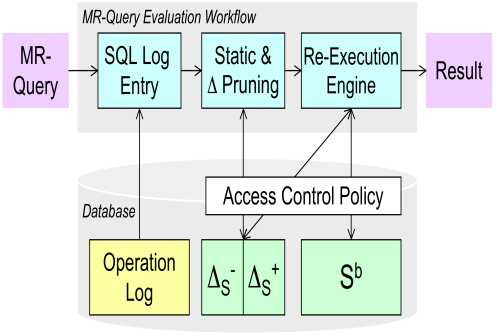


Partial Re-Execution – Cuts in the Query Plan



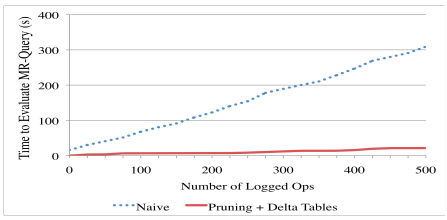
Simultaneous Re-Execution – Combined Query Plan

MR-Query Components

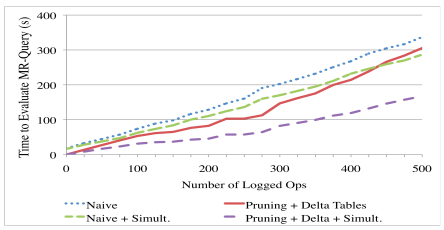


Experimental Evaluation

Vary the following parameters: Size of the policy misconfiguration, operation selectivity, select to update ratio, size of the DB, the number of predicate attributes.



1% Policy Misconfiguration,
1% Selectivity, 250 K Rows, Predicate Attributes=1



10% Policy Misconfiguration,
1% Selectivity, 250 K Rows, Predicate Attributes=1

Results

1. Optimizations can reduce reporting time by up to an order of magnitude.
2. As the misconfiguration gets larger, the cost of re-executing with the optimizations increases.
3. There exists a trade-off point when pruning plus delta tables is not efficient; rather, the naïve approach is better.
4. Simultaneous re-execution improves the performance of the naïve and optimized methods.