**Problem**

Data Leakage: undesired disclosure of the organization’s sensitive information, be it intentional or accidental.

Data leakage can cause extensive monetary loss, damage to reputation and legal problems.

**Goal**

Detect anomalies indicative of data leakage by monitoring (SQL) database usage/activities.

**Approach**

1. Monitor users’ (SQL) activities for the time necessary to capture normal behavior.
2. Learn normal behavior profiles based on users’ activities.
3. Detect anomalous activities and the root cause of the anomaly.
4. Quantify the anomaly severity based on data sensitivity (annotated data model).
5. Use feedback (e.g., false alarms) to update the user profiles.

**What’s New**

- Profiling considers the result set and the context in addition to the SQL query.
- Root cause of an anomaly is explicitly indicated (e.g., the table accessed, or the amount of data retrieved).
- Actual data retrieved is used to rate severity of anomalies.
- A feedback loop is introduced to reduce false alarms (profiles are updated when false alarms arise).

**The Framework**

1. Activity Monitoring
   - SQL query
   - Users
   - SQL database
   - Result set

2. Learning Phase
   - Algorithm (e.g., histograms, k-means)
   - User profiles

3. Detection Phase
   - Detection Model
     - No matching profile
     - User activity
     - Anomaly Alert (with root cause)

4. Quantification Phase
   - Anomaly Quantification
     - Annotated data model

5. Feedback Loop
   - Admin decision
   - Take Action

**Development and Validation**

Development and validation of the framework with extensive tests on real data from different application domains:

- Healthcare
- Financial sector
- Service management

**Acknowledgment**

This research has been partially supported by the COMMIT ThECS Project.

Contacts: Elisa Costante (e.costante@tue.nl), Sokratis Vavilis (s.vavilis@tue.nl)