CS 458/658 - Computer Security and Privacy **Fall 2018**

Lecture 19 - 21: November 13 - 20, 2018

Lecturer: Ian Goldberg *Notes By: Harsh Mistry*

19.1 Database Security and Privacy

19.1.1 Security Requirements

- Physical Database Integrity
  - Protect against database corruption
  - Allow only authorized individuals to perform updates
  - Recover from physical problems by performing backups or using transactions
- Logical Database Integrity
- Element Integrity
  - Ensure correctness/accuracy of database elements
  - Access control to limit who can update element
  - Element checks to validate correctness
  - Change log or shadow field to undo erroneous changes
  - Error detection codes to protect against OS or hard disk problems
  - Two-phase update
    - Phase 1 : Gather information required for changes
    - Phase 2 : Make changes permanent
    - Repeat each step if problem arises
  - Concurrency Control - Operations need to be atomic
- Referential Integrity
  - There should be no dangling foreign keys
- Auditability
  - keep audit log of all database accesses
  - Access control can be difficult, audit log allows to retroactively identify users
- User authentication
- Availability

19.1.2 Data disclosure and inference

19.1.2.1 Types of data disclosure

- Exact data
- Bounds
- Negative result
- Existence
- Probable value

19.1.2.2 Security vs. precision

- Security : Forbid any queries that access sensitive data, even if (aggregated) result is no longer sensitive
- Precision : Aggregated result should reveal as much non-sensitive data as possible

19.1.2.3 Data inference

- When sensitive data is derived from sensitive data
- Controls for statistical inference attacks
  - Apply control to query or to data items
  - Suppress sensitive data from result
  - Conceal answer by providing a value close to the actual value
  - n-item k percent rule : For the set of records that were included in the result, if there is a subset of n records that is responsible for over k percent of the result, omit the n records from result
  - Combined results : report set or range of possible values
  - Random sample : Compute result on random sample of database
  - Random data perturbation : Add or subtract small random error to/from each value before computing result
  - Query analysis : Maintain history of user's queries and observe possible inferences

19.1.2.4 Data aggregation

- Related to data inference
- Building sensitive results from less sensitive inputs
- Aggregation can take place outside of a DBMS, which makes it difficult to control

19.1.3 Multilevel Security (MLS) Databases

- Support classification/compartmentalization of information according to its confidentiality
- At element level if necessary
- In an MLS database, each object has a sensitivity classification and maybe a set of compartments

19.1.3.1 *-Property

- Implementing the *-property (no read up, no write down) in an MLS database is difficult
- DBMS must have read and write access at all levels to answer user queries, perform back-ups, optimize database,.

19.1.3.2 Confidentiality

- Depending on a user's clearance, he/she might get different answers for a query
- Existence of a record itself could be confidential
- Keeping existence hidden can lead to having multiple records with the same primary key, but different sensitivity (polyinstantiation)

19.1.3.3 Partitioning

- Have separate database for each classification level
- Might lead to data stored redundantly in multiple databases
- Doesn't address the problem of a high-level user needing access to low-level data combined with high-level data

19.1.3.4 Encryption

- Separate data by encrypting it with a key unique to its classification level
- Must be careful to use encryption scheme in the right way
- Processing of a query becomes expensive, many records might have to be decrypted

19.1.3.5 Integrity lock

- Provides both integrity and access control
- Each data item consist of, the actual item, an integrity level, and a cryptographic signature
- Signature protects against attacks on the above fields, such as attacks trying to modify the sensitivity label, and attacks trying to move/copy the item in the database
- This scheme does not protect against replay attacks
- Any (untrusted) database can be used to store data items and their integrity locks
- (Trusted) procedure handles access control and manages integrity locks
- Have to encrypt items and locks if there are other ways to get access to data in database

19.1.4 Designs of secure databases

- Trusted front end
  - Front end authenticates a user and forwards user query to old-style DBMS
  - Front end gets result from DBMS and removes data items that user is not allowed to see
  - Inefficient if DBMS returns lots of items and most of them are being dropped by front end
- Commutative filters
  - Front end re-writes user query according to a user's classification
  - Benefits from DBMS' superior query processing capabilities and discards forbidden data items early on
  - Front end might still have to do some post processing
- Distributed/federated databases
  - Based on partitioning
  - Front end forwards user query only to databases that user can access based on classification
  - Front end might have to combine the results from multiple databases
- Views
  - Many DBMS support views
  - A view is logical database that represents a subset of some other database
- Truman vs. non-Truman semantics
  - Truman semantics: the DBMS pretends that the data the user is allowed to access is all the data there is
  - Non-Truman semantics: the DBMS can reject queries that ask for data the user is not allowed to access

19.1.5 Data Mining and Data release

- Multilevel databases weren't a commercial success
- data miners actively gather additional data from third parties
- Data mining tries to automatically find interesting patterns in data using a plethora of technologies
- Data mining can be useful for security purposes
- Security Problems
  - Confidentiality - Derivation of sensitive information
  - Integrity - Mistakes in data
  - Availability - Incompatibility of different databases