Week 10 - Friday

COMP 4290

Last time

- Intrusion detection systems
- Database background
- Database security requirements

Questions?

Project 3

Nfaly Toure Presents

Finish Database Reliability and Integrity

Concurrency

- Most database systems allow more than one user or process to access it at the same time
- Updates must be controlled to avoid **race conditions**
  - Race conditions are sequences of commands that result in different states depending on timing
  - If there is one ticket left to a Bad Bunny concert, it should be impossible for two people to buy it
- Commands that both query (is there a ticket remaining) and update (buy the ticket) should be executed atomically
- Reading data also needs to be protected
  - If a user is writing data, it should be locked so that it can't be read

Constraints

- A **monitor** is the part of the DBMS responsible for structural integrity
- **Range comparisons** check newly entered numerical data for sanity
- **Filters** or **patterns** can be arbitrarily complex to make sure that a zip code or a VIN is correctly formatted
- The job of a DBA is to set these up, as well as the more complex state and transition constraints

State and transition constraints

- A **state constraint** is a characteristic that should be invariant over the database
  - Only one person is labeled president
  - Only one table has a given name
  - If such a constraint is violated, something has gone wrong in the database
- A **transition constraint** must be met before certain changes can be made to the database
  - A vacant position has to be listed before a new employee can be added
  - A student record must exist before that student's ID can be added to a class

Sensitive Data

Sensitive data

- **Sensitive data** are data that should not be made public
  - This is the confidentiality aspect of a database
- It can be sensitive because it is:
  - Inherently sensitive
  - From a sensitive source
  - Declared sensitive
  - Part of a sensitive record or field
  - Sensitive in relation to previously disclosed information

Access decisions

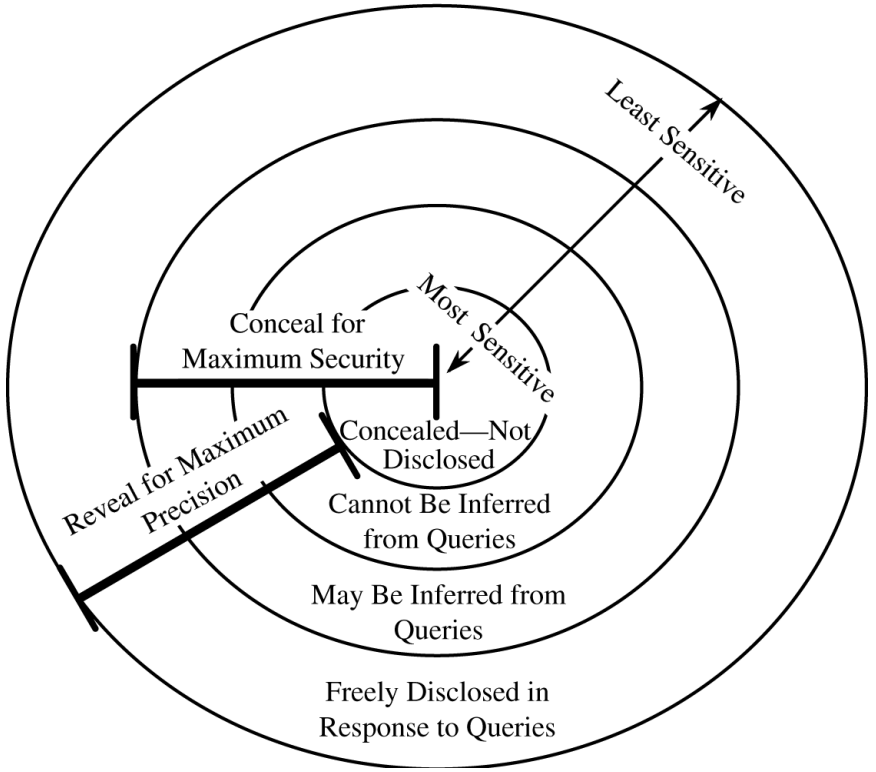
- The DBA has to make decisions about who can access what based on a number of factors
- Availability of data
  - What policies should be used to prevent two users from accessing the same data?
  - What if one user locks the data up, preventing the other from reading?
- Acceptability of access
  - Sensitivity is complex
  - Can I get a list of people whose fines are not zero, if the FINES field is sensitive?
  - As a student, can I get statistical data about an exam?
- Assurance of authenticity
  - Time or location can be used to determine access
  - Accessing some information may make others inaccessible

Types of disclosure

- The most serious disclosure of sensitive data is its exact value
- Bounds can also be disclosed
  - Example: highest salary and lowest salary
  - If the user can manipulate the bounds, he or she can search for specific values
- Negative result
  - Felonies is not zero
  - Visits to the oncology ward is not zero
- Existence
  - Knowing that a field even exists means someone is using it
- Probable value
  - How many people are in Bob's dorm room? 2
  - How many people in Bob's dorm room pirate movies? 1
  - There's a 50% chance that Bob pirates movies

Security versus precision

- The goal of a database is to collect data to analyze and get users' jobs done
- The DBA is trying to keep the data secure
- Users want to access data easily
- **Precision** is knowing the data you need to know
- In a perfect world, we could have perfect precision and perfect security, but we don't live there



Inference

Direct attack

- In a direct attack on sensitive information, a user will try to determine the values of a sensitive field with the right query

Name	Sex	Race	Aid	Fines	Drugs	Dorm
Adams	M	C	5000	45	1	Holmes
Bailey	M	B	0	0	0	Grey
Chin	F	A	3000	20	0	West
Dewitt	M	B	1000	35	3	Grey
Earhart	F	C	2000	95	1	Holmes
Fein	F	C	1000	15	0	West
Groff	M	C	4000	0	3	West
Hill	F	B	5000	10	2	Holmes
Koch	F	C	0	0	1	West
Liu	F	A	0	10	2	Grey
Majors	M	C	2000	0	2	Grey

Direct attack

- **`SELECT NAME FROM STUDENTS WHERE SEX="M" AND DRUGS="1"`**
  - This query might be rejected because it asks for a specific value of sensitive field Drugs
- **`SELECT NAME FROM STUDENTS WHERE (SEX="M" AND DRUGS="1") OR (SEX<>"M" AND SEX<>"F") OR (DORM="AYRES")`**
  - This query might be accepted by some systems because it appears to mask the sensitive information by including other information
  - However, the additional OR clauses add no records

Indirect attack

- To avoid leaking sensitive data, some DBMSs allow statistics to be reported
- Each of the following statistics can be attacked in different ways:
  - Sum
  - Count
  - Mean
  - Median

Sum example

- A single carefully chosen query can leak information
- The sum of financial aid broken down by gender and dorm reveals that no female student in Grey receives financial aid
  - If we know that Liu is a female student in Grey, we know she gets no aid

Count

- Count statistics are often released in addition to sum
  - Together these two allow averages to be computed
  - Alternatively, if count and average are released, sums could be computed
- A query of the count of students reveals that there is 1 male in Holmes and 1 male in Grey
- We can get their names because that is unprotected data
- Using the previous sums of financial aid, we can determine exactly how much they get

Mean

- If you are able to get means for slightly different sets, you can determine the values for the difference of those sets
- Example (with made-up numbers)
  - Average salary for an Otterbein employee: $47,600
  - Average salary for all Otterbein employees except for the president: $47,000
  - Given that there are 500 employees at Otterbein, how much does Dr. Comerford make?

Medians

- We can use the median values of lists to reconstruct the user who has that median value
  - Some extra information might be needed
- If someone knows that Majors is the only male whose drug-use score is 2, they can find the median of the financial aid value for males and the median of the financial aid value for people whose drug-use score is 2
  - If the two values match, that number is Majors's aid, with high probability

Tracker attacks

- A DBMS may refuse to give back a result that has only a single record
  - `COUNT(SELECT * FROM STUDENTS WHERE SEX="F" AND RACE="C" AND DORM="HOLMES")`
- However, we can use the laws of set theory to defeat the DBMS
- $|A \cap B \cap C| = |A| - |A \cap (B \cap C)^c|$
- Thus, we find:
  - `COUNT(SELECT * FROM STUDENTS WHERE SEX="F")`
  - `COUNT(SELECT * FROM STUDENTS WHERE SEX="F" AND (RACE<>"C" AND DORM<>"HOLMES"))`
  - Then, we subtract the two values
- This attack can be extended to a system of linear equations to solve for a particular set of values

Controls for statistic inference attacks

- **Suppression** means that sensitive values are not reported
- **Concealing** means that values close to the sensitive values are returned, but not the values themselves
- This represents the tension between security and precision

Limited response suppression

- The ***n* items over *k* percent rule** means that the data should not be reported if the number of records ***n*** makes up more than ***k*** percent of the total
- This is a standard rule for suppression
- However, if counts are supplied, additional values may need to be hidden so that the hidden values can't be computed
- It's like Sudoku!

Combined results

- Instead of failing to report values, we can combine ranges
  - Though drug use values of 0, 1, 2, and 3 would allow us to recover individual names, we can report the category of 0 – 1 and 2 – 3
- Numerical values such as money can reported in ranges as well
- Finally, values could be rounded (note that this is essentially the same as using ranges)

Randomization

- In **random sample control**, results are given from a random sample of the database, not the whole thing
  - Useless for count, but not bad for mean
  - To prevent users from getting more information from repeated queries, the random sample for a given query should always be the same
- **Random data perturbation** adds small amounts of error to each value
  - Because the errors are positive or negative, means and sums will be only slightly affected

Inference summary

- Suppress obviously sensitive information
  - Easy, but incomplete
- Track what the user knows
  - Expensive in terms of computation and storage requirements
  - Analysis may be difficult
  - Multiple users can conspire together
- Disguise the data
  - Data is hidden
  - Users who are not trying to get sensitive data get slightly wrong answers

Multilevel Databases

Database security isn't simple

- A single element may have different security from other values in the record or other elements in the same attribute
- Sensitive and non-sensitive might not be enough categories to capture all possible confidentiality relationships
- The security of an aggregate (sums, counts, etc.) may be different from the security of its parts
- We want to add different levels of security, similar to Bell-La Padula

Integrity and confidentiality

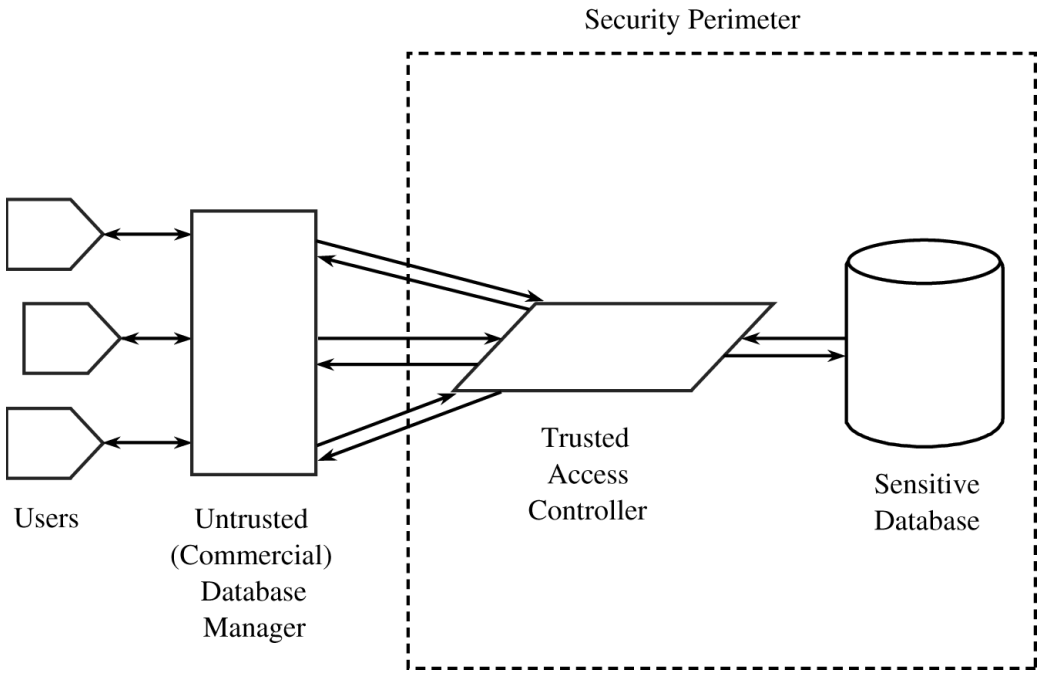
- Integrity is difficult, but we can assign levels of trust
  - It is necessarily not going to be as rigorous as Biba
- Confidentiality
  - Difficult and causes redundancies since top secret information cannot be visible in any way to low clearance users
  - Worse, we don't want to leak any information by preventing a record from being added with a particular primary key (because there is a hidden record that already has that primary key)
  - **Polyinstantiation** means that records with similar or identical primary keys (but different data) can exist at different security levels

Proposals for multilevel databases

- Partitioning
  - Each security level is a separate database
  - Simple, but destroys redundancy and makes access harder
- Encryption
  - A single key for a security level isn't good enough: a break in the key reveals everything
  - A key for every record (or field) gives good security, but the time and space overhead is huge

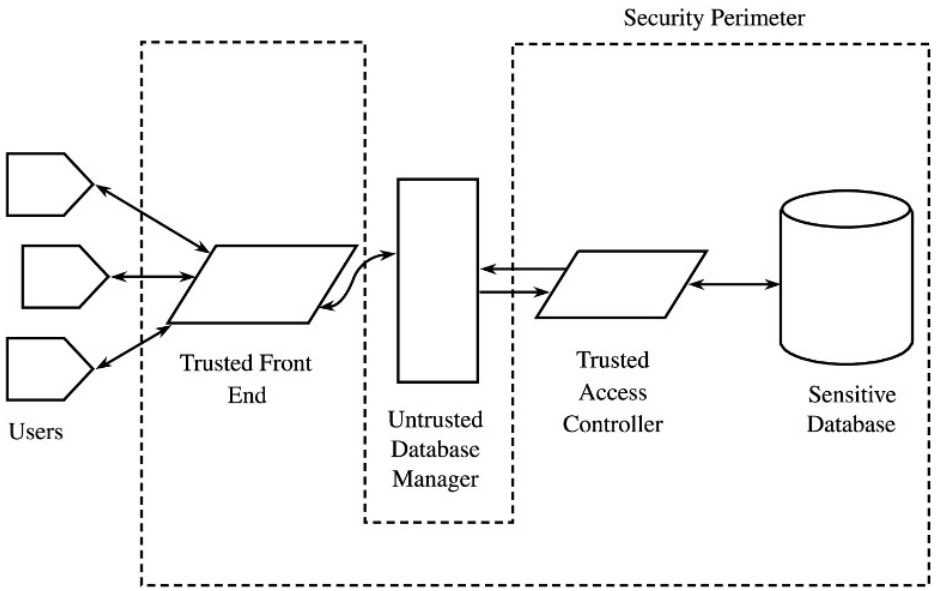
Integrity lock databases

- An integrity lock database has a security label for each item, which is:
  - Unforgeable
  - Unique
  - Concealed (even the sensitivity level is unknown)
- If these labels are preserved on the data storage side, an untrusted front-end can be used as long as it has good authentication
- Storage requirements are heavy



Trusted front-end databases

- A **trusted front-end** is also known as a **guard**
- The idea isn't that different from an integrity lock database with an untrusted front end
- It is trying to leverage DBMS tools that are familiar to most DBAs
  - The front-end can be configured instead of interacting with database internals
- The system can be inefficient because a lot of data can be retrieved and then discarded by the front end if it isn't at the right level



Ticket Out the Door

Upcoming

Next time…

- Data mining
- Cloud computing
- Anu Regmi presents

Reminders

- Work on Project 3
  - User names and passwords and phrases need to be turned in **next Friday in class!**