Outline

• Security basics
• Access control in Databases
• SQL injection attacks
• Private Data Analysis
Security properties

• Confidentiality
  • A guarantee that data has not been disclosed to an unauthorized party.

• Authenticity
  • A guarantee that data has not been modified from its original state by an unauthorized party.

• Availability
  • A guarantee that data is available when needed.
Confidentiality or Authenticity?

- Which security properties matter for:
  - Student grades for this course stored in database.
  - Resume or CV posted on webpage.
  - Medical records stored in database.
Privacy

Informational Privacy

The ability to determine for ourselves when, how, and to what extent information about us is communicated to others.

- Westin

(Requires aspects of both data confidentiality & authenticity)
Access control

• Regulates direct access to resources
  • Subjects (i.e. registered users)
  • Objects (files, directories, tables)
  • Privileges (read, write, insert, delete, etc.)

• **Discretionary** access control
  • Users can grant access at their discretion.

• **Mandatory** access control
  • All subjects and objects classified by an authority and global rules determine privileges.
SQL Security

- Core security features present in nearly all database systems:
  - User authentication
  - Discretionary access control:
    - Subjects (database users)
    - Privileges (select, insert, delete, update)
    - Objects (tables, columns, views)
  - In SQL: GRANT / REVOKE

System R authorization model [Griffith and Wade’76], [Fagin’78]
Discretionary AC in SQL

GRANT privileges ON object TO users

[WITH GRANT OPTIONS]

privileges = SELECT | INSERT(column-name) | UPDATE(column-name) | DELETE | REFERENCES(column-name)

object = table | view
Examples

GRANT INSERT, DELETE ON Customers TO Yuppy
WITH GRANT OPTIONS

Queries allowed to Yuppy:

- INSERT INTO Customers(cid, name, address)
  VALUES(32940, ‘Joe Blow’, ‘Seattle’)
- DELETE Customers WHERE
  LastPurchaseDate < 1995

Queries denied to Yuppy:

- SELECT Customer.address
  FROM Customer
  WHERE name = ‘Joe Blow’
Examples

GRANT SELECT ON Customers TO Michael

Now Michael can SELECT, but not INSERT or DELETE
Examples

GRANT SELECT ON Customers TO Michael
WITH GRANT OPTIONS

Michael can say this:
GRANT SELECT ON Customers TO Yuppi

Now Yuppi can SELECT on Customers
Examples

GRANT UPDATE (price) ON Product TO Leah

Leah can update, but only Product.price, but not Product.name
Examples

Customer(cid, name, address, balance)
Orders(oid, cid, amount)     cid= foreign key

Bill has INSERT/UPDATE rights to Orders.
BUT HE CAN’’T INSERT ! (why ?)

GRANT REFERENCES (cid) ON 
Customer TO Bill

Now Bill can INSERT tuples into Orders
CREATE VIEW PublicCustomers
SELECT Name, Address
FROM Customers;
GRANT SELECT ON PublicCustomers TO Fred

David owns

Fred is not allowed to see this

David says

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Huston</td>
<td>450.99</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
<td>-240</td>
</tr>
<tr>
<td>Joan</td>
<td>Seattle</td>
<td>333.25</td>
</tr>
<tr>
<td>Ann</td>
<td>Portland</td>
<td>-520</td>
</tr>
</tbody>
</table>

CREATE VIEW PublicCustomers
SELECT Name, Address
FROM Customers;
GRANT SELECT ON PublicCustomers TO Fred

Friday, May 14, 2010
CREATE VIEW BadCreditCustomers
    SELECT *
    FROM Customers
    WHERE Balance < 0;
GRANT SELECT ON BadCreditCustomers TO John

David owns

Customers:

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Huston</td>
<td>450.99</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
<td>-240</td>
</tr>
<tr>
<td>Joan</td>
<td>Seattle</td>
<td>333.25</td>
</tr>
<tr>
<td>Ann</td>
<td>Portland</td>
<td>-520</td>
</tr>
</tbody>
</table>

John is allowed to see only <0 balances

Friday, May 14, 2010
Views and Security

- Each customer should see only her/his record.

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
<th>Balance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary</td>
<td>Huston</td>
<td>450.99</td>
</tr>
<tr>
<td>Sue</td>
<td>Seattle</td>
<td>-240</td>
</tr>
<tr>
<td>Joan</td>
<td>Seattle</td>
<td>333.25</td>
</tr>
<tr>
<td>Ann</td>
<td>Portland</td>
<td>-520</td>
</tr>
</tbody>
</table>

CREATE VIEW CustomerMary
SELECT * FROM Customers
WHERE name = 'Mary'
GRANT SELECT ON CustomerMary TO Mary

CREATE VIEW CustomerSue
SELECT * FROM Customers
WHERE name = 'Sue'
GRANT SELECT ON CustomerSue TO Sue

Doesn’t scale.

Need row-level access control!

David says
Revocation

[Syntax]

REVOKE [GRANT OPTION FOR] privileges
ON object FROM users { RESTRICT | CASCADE }

Administrator says:

REVOKE SELECT ON Customers FROM David CASCADE

John loses SELECT privileges on BadCreditCustomers
SQL Injection

- Popular attack on databases accessed through web interfaces.
- Attacker is able to insert SQL statements into a query by manipulating application input data.
- Ranked as a top 10 security vulnerability
  - SANS Institute
  - Open Web Application Security Project (OWASP)

<table>
<thead>
<tr>
<th></th>
<th>Top Vulnerabilities in Web Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>Unvalidated Input</td>
</tr>
<tr>
<td></td>
<td>Information from web requests is not validated before being used by a web application. Attackers can use these flaws to attack backend components through a web application.</td>
</tr>
<tr>
<td>A2</td>
<td>Broken Access Control</td>
</tr>
<tr>
<td></td>
<td>Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access other users' accounts, view sensitive files, or use unauthorized functions.</td>
</tr>
</tbody>
</table>
## SQL injection: example

### Treatment table

<table>
<thead>
<tr>
<th>patient</th>
<th>doctor</th>
<th>date</th>
<th>diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>fred</td>
<td>Dr. Lee</td>
<td>9/1/2005</td>
<td>cancer</td>
</tr>
<tr>
<td>mary</td>
<td>Dr. Lee</td>
<td>5/2/2004</td>
<td>flu</td>
</tr>
<tr>
<td>fred</td>
<td>Dr. Ash</td>
<td>1/18/2005</td>
<td>diabetes</td>
</tr>
<tr>
<td>joe</td>
<td>Dr. Boul</td>
<td>6/4/2005</td>
<td>flu</td>
</tr>
</tbody>
</table>
SQL Injection

Your health insurance company has a web site for claims:

First login:
User: fred
Password: ********

Then search:
Search claims: Dr. Lee

SELECT…FROM…WHERE doctor='Dr. Lee' and patient='fred'

[Chris Anley, Advanced SQL Injection In SQL]
SQL Injection

Now try this:

Search claims: `Dr. Lee' OR patientID = 'mary'; --

.....WHERE doctor='Dr. Lee' OR patientID='mary'; --' and patientID='fred'

Even better:

Search claims: `Dr. Lee' OR 1 = 1; --
SQL Injection

• Those attacks threaten confidentiality
• There are also authenticity attacks
  • these often require knowledge of schema
  • can be discovered using error messages!

Extreme case:

Search claims: Dr. Lee’; DROP TABLE Patients; --
Solutions

• Input validation
  • check content, length, format
  • generally a pain, hard to check, may accidentally reject good input

• Stored procedures with parameters

• Deeper answer:
  • Move policy implementation from apps to DB
Database humor

Hi, this is your son's school. We're having some computer trouble.

Oh, dear - did he break something? In a way-

Did you really name your son 'Robert'); DROP TABLE Students;--?

Oh, yes. Little Bobby Tables, we call him.

Well, we've lost this year's student records. I hope you're happy.

And I hope you've learned to sanitize your database inputs.
Private Data Analysis

• Goal: derive useful information from collection of data without violating privacy of individual participants

• Oldest form: statistical database
  • A database which permits users to retrieve aggregate statistics (e.g. count, average) only for subsets of entities.
Statistical databases

SELECT count(*)
FROM Patients
WHERE age=42
    and sex='M'
    and diagnostic='schizophrenia'

SELECT name
FROM Patient
WHERE age=42
    and sex='M'
    and diagnostic='schizophrenia'
Solution approaches

Query restriction

(Send) queries

Exact responses / denials

Data perturbation

Perturbed SDB

Client

Output perturbation

Perturbed responses

(Part) queries

Client
Mass. Group Insurance Commission (GIC) is responsible for purchasing health insurance for state employees.

GIC collects data, and publishes it: "Anonymized" data publishing.

<table>
<thead>
<tr>
<th>name</th>
<th>ssn</th>
<th>gender</th>
<th>dob</th>
<th>zip</th>
<th>diagnosis</th>
</tr>
</thead>
<tbody>
<tr>
<td>joe</td>
<td>123-56-7634</td>
<td>male</td>
<td>1/4/64</td>
<td>1045</td>
<td>cancer</td>
</tr>
<tr>
<td>mary</td>
<td>113-36-4252</td>
<td>female</td>
<td>3/24/45</td>
<td>1312</td>
<td>flu</td>
</tr>
<tr>
<td>bob</td>
<td>124-46-574</td>
<td>male</td>
<td>5/4/55</td>
<td>1452</td>
<td>HIV</td>
</tr>
</tbody>
</table>

- **Identifier**: name, ssn
- **Sensitive Attr.**: gender, dob, zip, diagnosis

Friday, May 14, 2010
Sweeney paid $20 and bought the public voter registration list for Cambridge Mass.:

<table>
<thead>
<tr>
<th>name</th>
<th>party</th>
<th>gender</th>
<th>dob</th>
<th>zip</th>
</tr>
</thead>
<tbody>
<tr>
<td>joe</td>
<td>Dem</td>
<td>male</td>
<td>1/4/64</td>
<td>1045</td>
</tr>
<tr>
<td>mary</td>
<td>Rep</td>
<td>female</td>
<td>3/24/45</td>
<td>1312</td>
</tr>
<tr>
<td>bob</td>
<td>Dem</td>
<td>male</td>
<td>5/4/55</td>
<td>1452</td>
</tr>
</tbody>
</table>
Re-identification

INSURED(zip, dob, sex, diagnosis, procedure, ...)
VOTER(name, party, ..., zip, dob, sex)

- William Weld (former governor) lives in Cambridge, hence is in VOTER
- 6 people in INSURED share his dob
- only 3 of them were male (same gender)
- Weld was the only one in that zip
- Sweeney learned Weld’s medical records!
87% of the US population (216 million out of 248 million) are likely to be uniquely identified based on:

zipcode, gender, date-of-birth
K-Anonymity

- Intuition: privacy is gained by hiding individuals in groups of sufficient size

- Alter data so that:
  - At least k individuals share each pseudo-identifier occurring in the database.
  - Attribute suppression and generalization

Data perturbation

SDB \(\rightarrow\) Perturbed SDB \(\rightarrow\) Client

queries \(\leftarrow\) Pert. responses
K-anonymity example

<table>
<thead>
<tr>
<th>Non-Sensitive</th>
<th>Sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zip Code</td>
<td>Age</td>
</tr>
<tr>
<td>1</td>
<td>13053</td>
</tr>
<tr>
<td>2</td>
<td>13068</td>
</tr>
<tr>
<td>3</td>
<td>13068</td>
</tr>
<tr>
<td>4</td>
<td>13053</td>
</tr>
<tr>
<td>5</td>
<td>14853</td>
</tr>
<tr>
<td>6</td>
<td>14853</td>
</tr>
<tr>
<td>7</td>
<td>14850</td>
</tr>
<tr>
<td>8</td>
<td>14850</td>
</tr>
<tr>
<td>9</td>
<td>13053</td>
</tr>
<tr>
<td>10</td>
<td>13053</td>
</tr>
<tr>
<td>11</td>
<td>13068</td>
</tr>
<tr>
<td>12</td>
<td>13068</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-Sensitive</th>
<th>Sensitive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zip Code</td>
<td>Age</td>
</tr>
<tr>
<td>1</td>
<td>130**</td>
</tr>
<tr>
<td>2</td>
<td>130**</td>
</tr>
<tr>
<td>3</td>
<td>130**</td>
</tr>
<tr>
<td>4</td>
<td>130**</td>
</tr>
<tr>
<td>5</td>
<td>1485*</td>
</tr>
<tr>
<td>6</td>
<td>1485*</td>
</tr>
<tr>
<td>7</td>
<td>1485*</td>
</tr>
<tr>
<td>8</td>
<td>1485*</td>
</tr>
<tr>
<td>9</td>
<td>130**</td>
</tr>
<tr>
<td>10</td>
<td>130**</td>
</tr>
<tr>
<td>11</td>
<td>130**</td>
</tr>
<tr>
<td>12</td>
<td>130**</td>
</tr>
</tbody>
</table>

Original data → 4-anonymous data

From Machanavajjhala, Gehrke, Kifer. ICDE 06
Analysis

- Higher k -- more privacy
- Fewer suppressions & generalizations -- more accuracy

<table>
<thead>
<tr>
<th></th>
<th>Zip Code</th>
<th>Age</th>
<th>Nationality</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>130**</td>
<td>&lt; 30</td>
<td>*</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>2</td>
<td>130**</td>
<td>&lt; 30</td>
<td>*</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>3</td>
<td>130**</td>
<td>&lt; 30</td>
<td>*</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>4</td>
<td>130**</td>
<td>&lt; 30</td>
<td>*</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>5</td>
<td>1485*</td>
<td>≥ 40</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>6</td>
<td>1485*</td>
<td>≥ 40</td>
<td>*</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>7</td>
<td>1485*</td>
<td>≥ 40</td>
<td>*</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>8</td>
<td>1485*</td>
<td>≥ 40</td>
<td>*</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>9</td>
<td>130**</td>
<td>3*</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>10</td>
<td>130**</td>
<td>3*</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>11</td>
<td>130**</td>
<td>3*</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>12</td>
<td>130**</td>
<td>3*</td>
<td>*</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

From Machanavajjhala, Gehrke, Kifer. ICDE 06
Attacks on anonymized data

- Homogeneity attack
  - Alice knows Bob is 31, living in zip 13053

- Background knowledge attack
  - Alice has japanese friend who is 21 and living in 13068.
Supplemental knowledge

• In controlling inference, must model the adversary’s supplemental knowledge:
  • all information about the database the adversary can acquire from sources other than the system itself.
Differential Privacy

Intuition

• When a database contains personal information about an individual, privacy has been guaranteed for that individual if:

• the statistics released are unchanged whether or not the individual is in the database.
The differential guarantee

Two databases are neighbors if they differ by at most one tuple.
A randomized algorithm $A$ provides **ε-differential privacy** if:

for all neighboring databases $D$ and $D'$, and
for any set of outputs $S$:

$$
Pr[A(D) \in S] \leq e^{\epsilon} Pr[A(D') \in S]
$$

Epsilon is a privacy parameter

Epsilon is usually small: e.g. if $\epsilon = 0.1$ then $e^\epsilon \approx 1.10$

$\downarrow$ epsilon = $\uparrow$ stronger privacy
Calibrating noise

- The following algorithm for answering $Q$ is $\varepsilon$-differentially private:

$$Q(D) + \text{Laplace} \left( \frac{\Delta D}{\varepsilon} \right)$$
Examples of query sensitivity

The sensitivity of a query $Q$ is
$$\Delta Q = \max_{D,D'} | Q(D) - Q(D') |$$
where $D$, $D'$ are any two neighboring databases

- Count of students receiving grade ’90’
- Maximum grade in the class
- Median grade in the class