Introduction to Information Security

Chapter 5
Database Security

Riccardo Spolaor, Ph.D rspolaor@sdu.edu.cn

Shandong University, School of Computer Science and Technology

Database Security

Reasons database security has not evolved together with databases are:

- Dramatic imbalance between the complexity of modern DBMS (database management systems) and the security technique used to protect these critical systems
- The increasing reliance on cloud technology to host part or all of the corporate database
- Most enterprise environments consist of a heterogeneous platforms (DB, OS, enterprise) creating an additional complexity hurdle for security personnel
- The typical organization lacks full-time database security personnel
- Effective database security requires a strategy based on a full understanding of the security vulnerabilities of SQL
- Databases have a sophisticated interaction protocol SQL (Structured Query Language) which is complex

Databases

- Structured collection of data stored for use by one or more applications
- Contains the relationships between data items and groups of data items
- Can sometimes contain sensitive data that needs to be secured

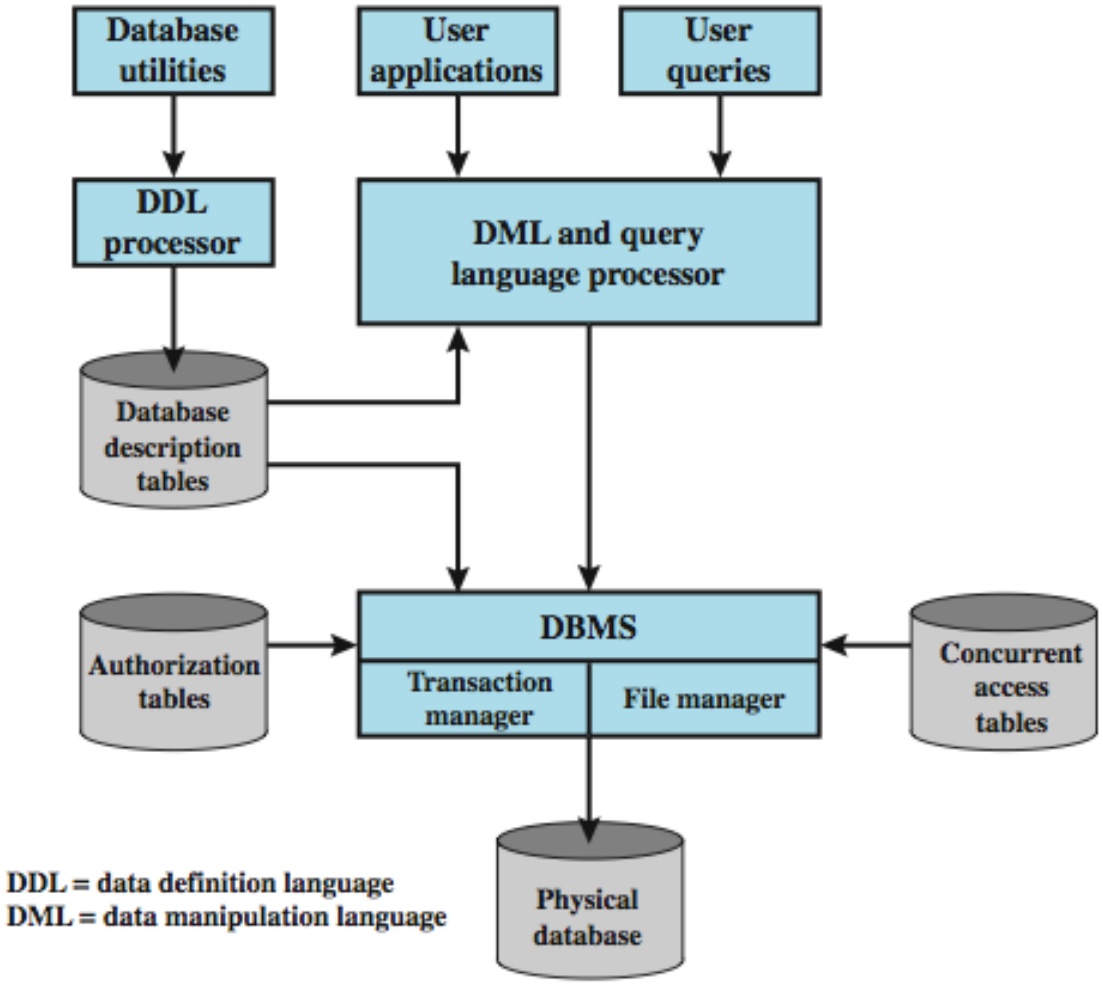
Query language

- Provides a uniform interface to the database for users and applications

Database management system (DBMS)

- Suite of programs for constructing and maintaining the database
- Offers ad hoc query facilities to multiple users and applications

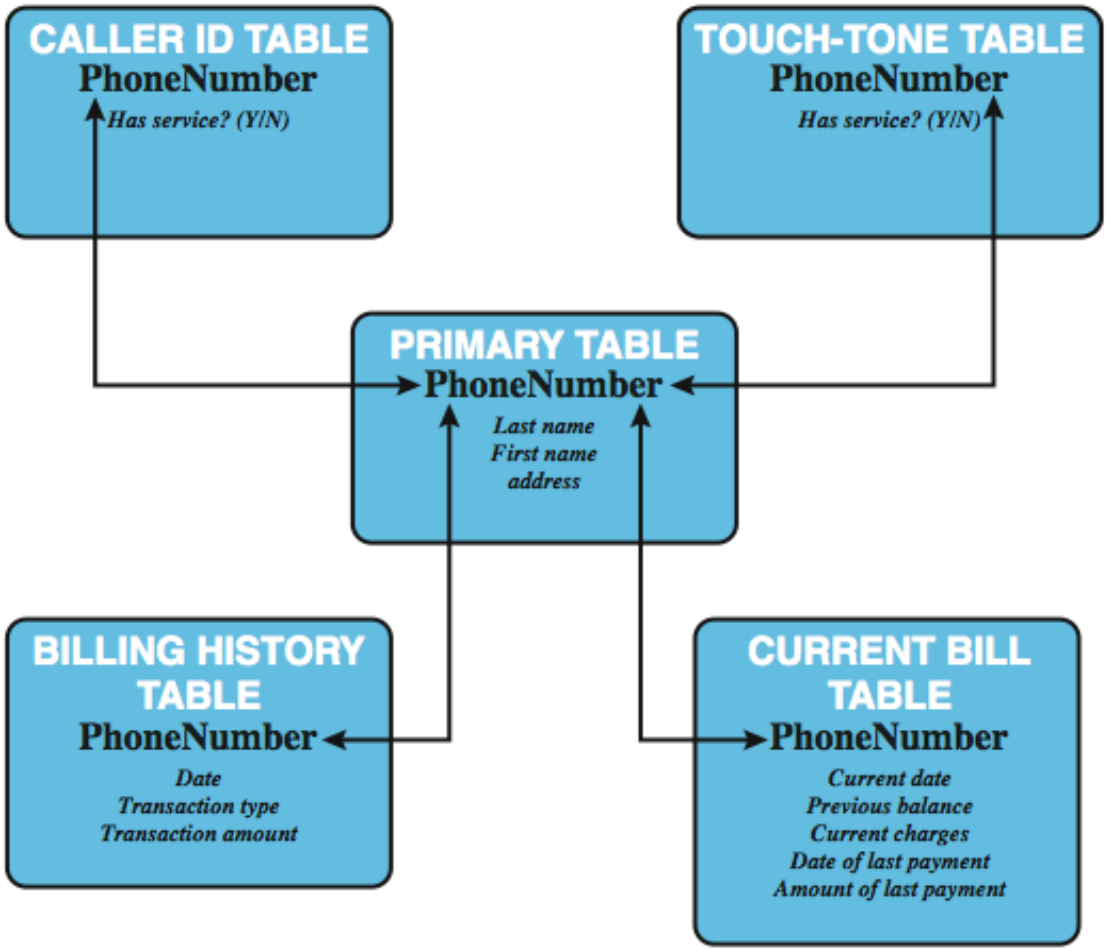
Database Security



Relational Databases

- constructed from tables of data
  - each column holds a particular type of data
  - each row contains a specific value these
  - ideally has one column where all values are unique, forming an identifier/key for that row
- have multiple tables linked by identifiers
- use a query language to access data items meeting specified criteria

Relational Database Example



Relational Database Elements

Basic Terminology for Relational Databases

Formal Name	Common Name	Also Known As
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

- ➢ primary key
  - Uniquely identifies a row
  - Consists of one or more column names
- ➢ foreign key
  - links one table to attributes in another
- ➢ view / virtual table
  - Result of a query that returns selected rows and columns from one or more tables
  - Views are often used for security purposes

Relational Database Elements

Department Table

Did	Dname	Dacctno
4	human resources	528221
8	education	202035
9	accounts	709257
13	public relations	755827
15	services	223945

primary key

Employee Table

Ename	Did	SalaryCode	Eid	Ephone
Robin	15	23	2345	6127092485
Neil	13	12	5088	6127092246
Jasmine	4	26	7712	6127099348
Cody	15	22	9664	6127093148
Holly	8	23	3054	6127092729
Robin	8	24	2976	6127091945
Smith	9	21	4490	6127099380

foreign key — primary key

(a) Two tables in a relational database

Dname	Ename	Eid	Ephone
human resources	Jasmine	7712	6127099348
education	Holly	3054	6127092729
education	Robin	2976	6127091945
accounts	Smith	4490	6127099380
public relations	Neil	5088	6127092246
services	Robin	2345	6127092485
services	Cody	9664	6127093148

(b) A view derived from the database

Structured Query Language

- Structure Query Language (SQL)
  - originally developed by IBM in the mid-1970s
  - standardized language to define, manipulate, and query data in a relational database
  - several similar versions of ANSI/ISO standard

SQL statements can be used to:

- Create tables
- Insert and delete data in tables
- Create views
- Retrieve data with query

Structured Query Language

```
CREATE TABLE department (
        Did INTEGER PRIMARY KEY,
        Dname CHAR (30),
        Dacctno CHAR (6) )

CREATE TABLE employee (
        Ename CHAR (30),
        Did INTEGER,
        SalaryCode INTEGER,
        Eid INTEGER PRIMARY KEY,
        Ephone CHAR (10),
        FOREIGN KEY (Did) REFERENCES department (Did) )
```

```
CREATE VIEW newtable (Dname, Ename, Eid, Ephone)
AS SELECT D.Dname E.Ename, E.Eid, E.Ephone
FROM Department D Employee E
WHERE E.Did = D.Did
```

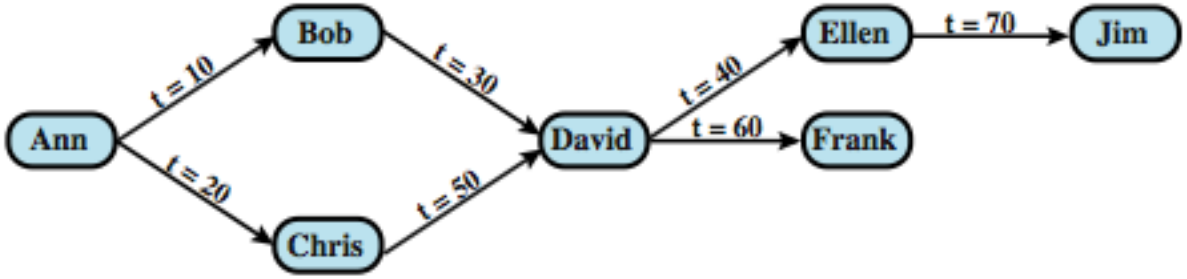
Database Access Control

- DBMS provide access control for database
- assume have authenticated user
- DBMS provides specific access rights to portions of the database
  - e.g. create, insert, delete, update, read, write
  - to entire database, tables, selected rows or columns
  - possibly dependent on contents of a table entry
- can support a range of policies:
  - centralized administration
  - ownership-based administration
  - decentralized administration

SQL Access Controls

- two commands:
  - `GRANT { privileges | role } [ON table] TO { user | role | PUBLIC } [IDENTIFIED BY password] [WITH GRANT OPTION]`
    - e.g. GRANT SELECT ON ANY TABLE TO ricflair
  - `REVOKE { privileges | role } [ON table] FROM { user | role | PUBLIC }`
    - e.g. REVOKE SELECT ON ANY TABLE FROM ricflair
- typical access rights are:
  - `SELECT, INSERT, UPDATE, DELETE, REFERENCES`

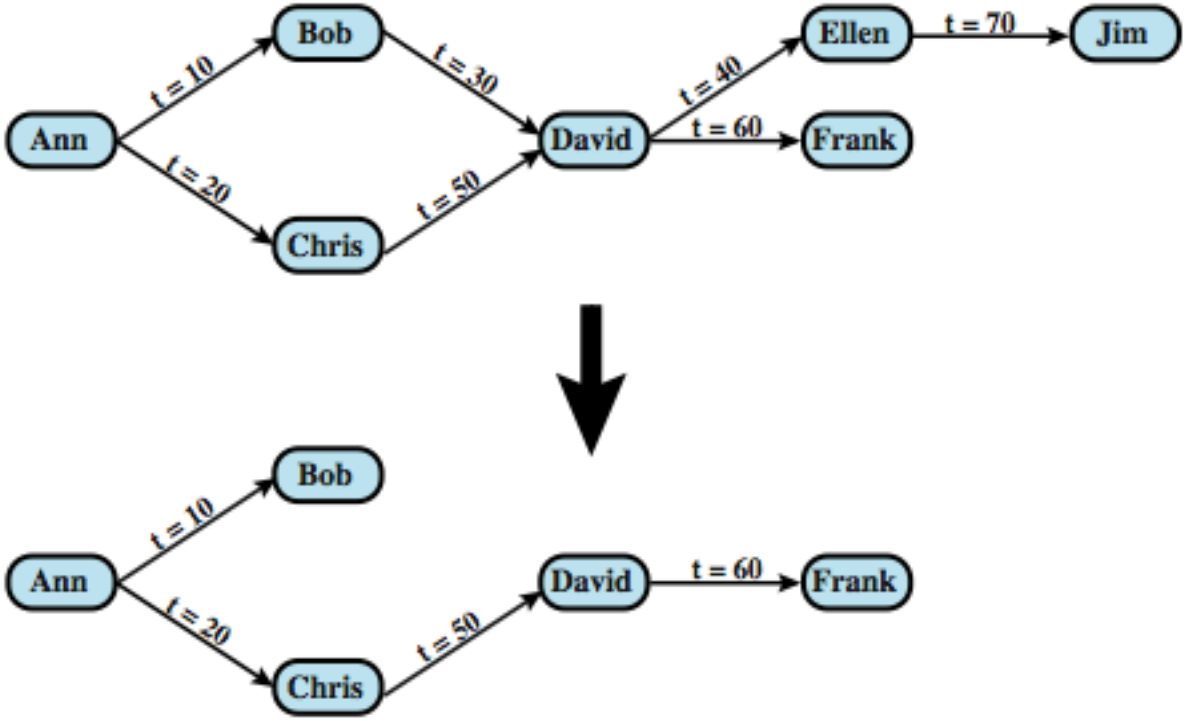
Cascading Authorizations



What if...

Bob revokes privilege from David?

Cascading Authorizations



SQL Injection Attacks (SQLi)

- One of the most prevalent and dangerous network-based security threats
- Designed to exploit the nature of Web application pages
- Sends malicious SQL commands to the database server
- Most common attack goal is bulk extraction of data
- Depending on the environment SQL injection can also be exploited to:
  - Modify or delete data
  - Execute arbitrary operating system commands
  - Launch denial-of-service (DoS) attacks

S (SQLi)

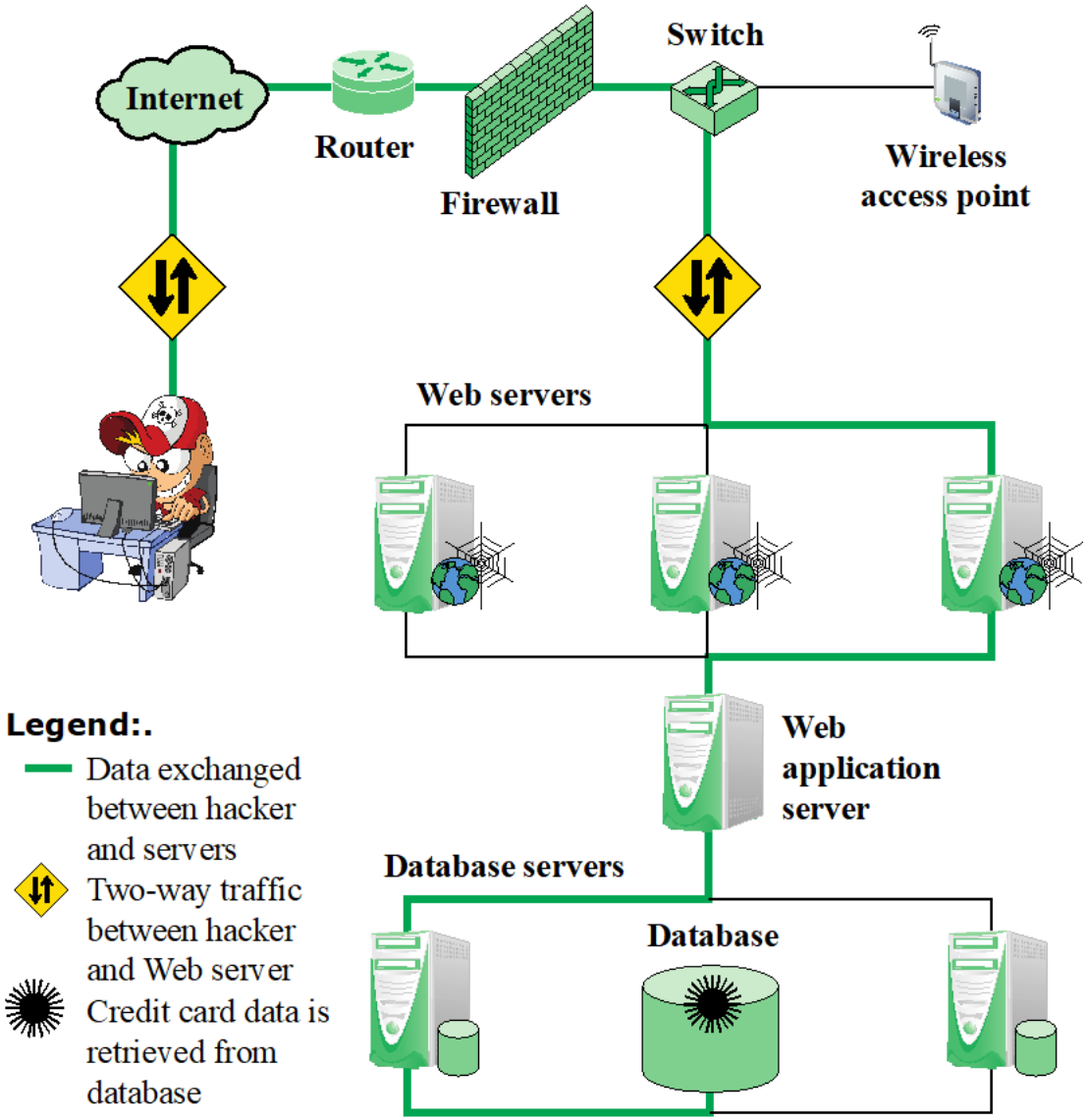


Figure 5.5 Typical SQL Injection Attack

SQL Injection Attacks (SQLi)

The SQLi attack typically works by prematurely terminating a text string and appending a new command

Because the inserted command may have additional strings appended to it before it is executed the attacker terminates the injected string with a comment mark “- -”

Subsequent text is ignored at execution time

SQL Injection Attacks (SQLi)

User input

- Attackers inject SQL commands by providing suitable crafted user input

Server variables

- Forge values in HTTP and network headers

Second-order injection

- Rely on data already present in the system or database to trigger an SQL injection attack
- The input that modifies the query does not come from the user, but from within the system itself

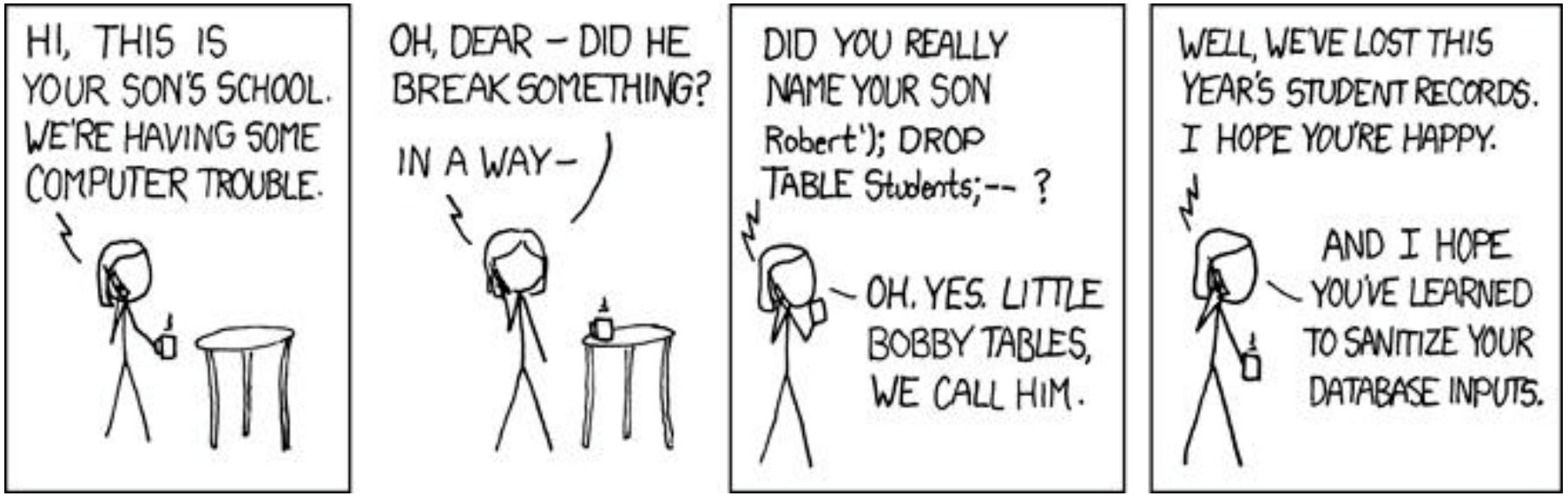
Cookies

- Alter cookies so the application server builds an SQL query based on the cookie's content

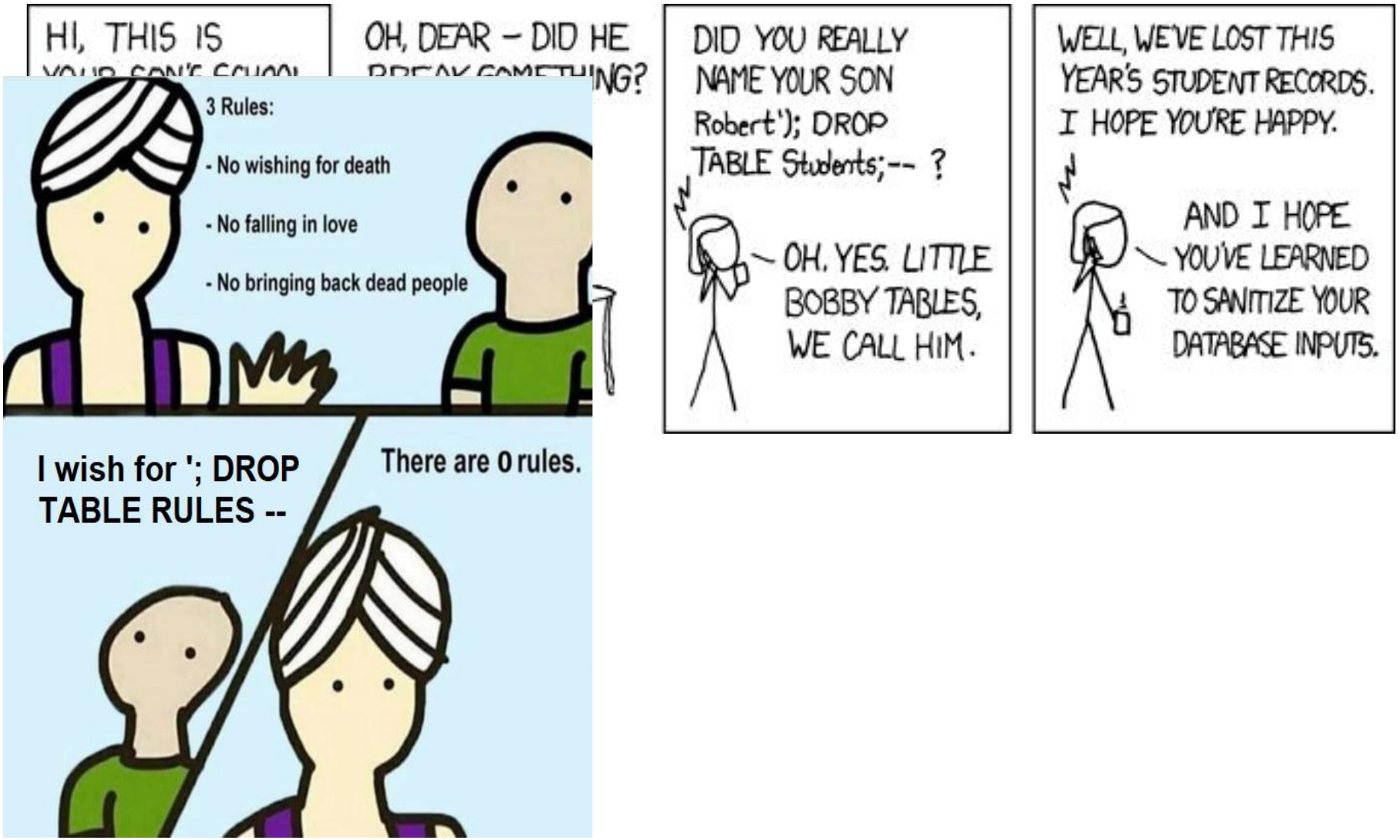
Physical user input

- Applying user input that constructs an attack outside the realm of web requests

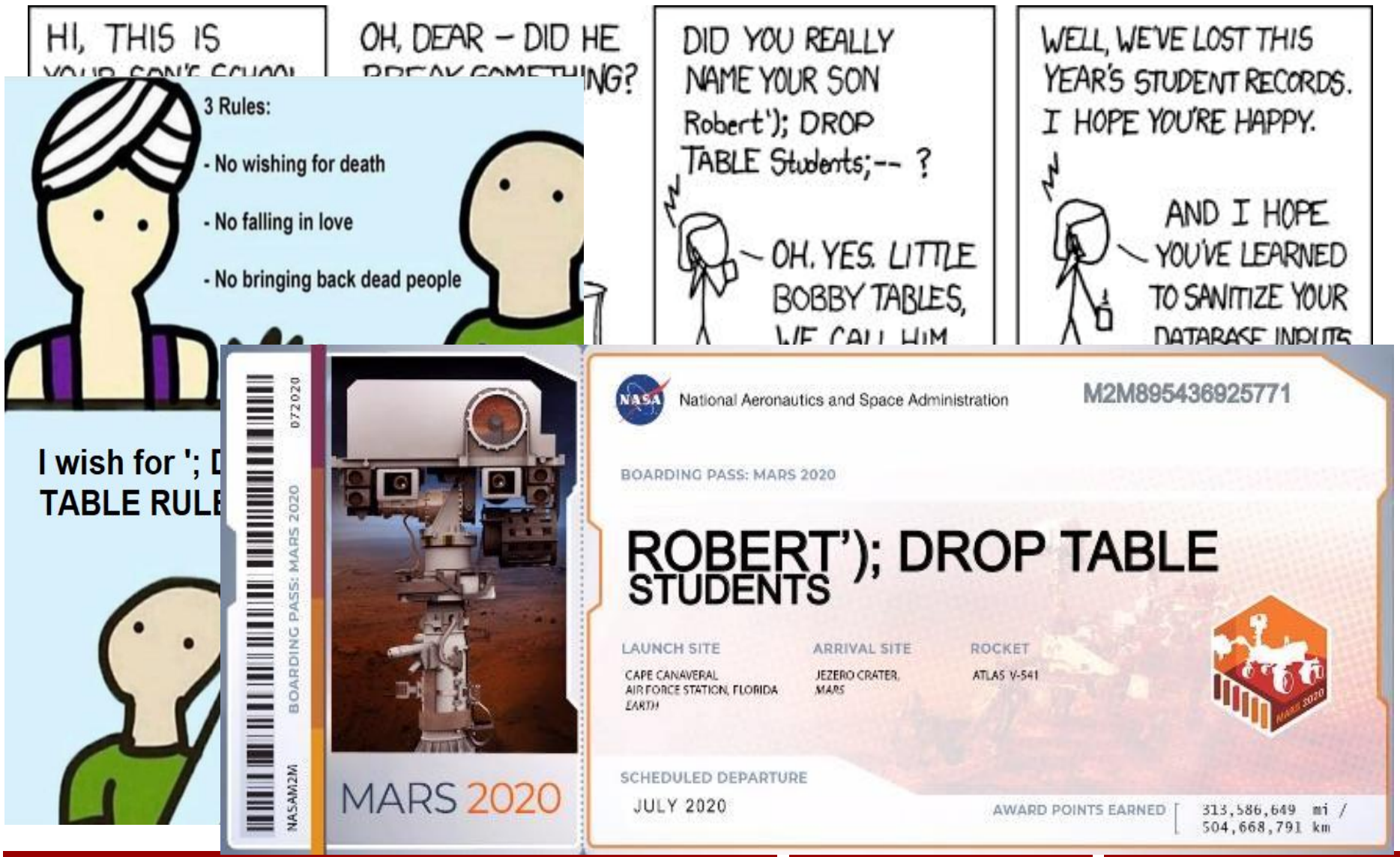
SQL Injection Attacks (SQLi)



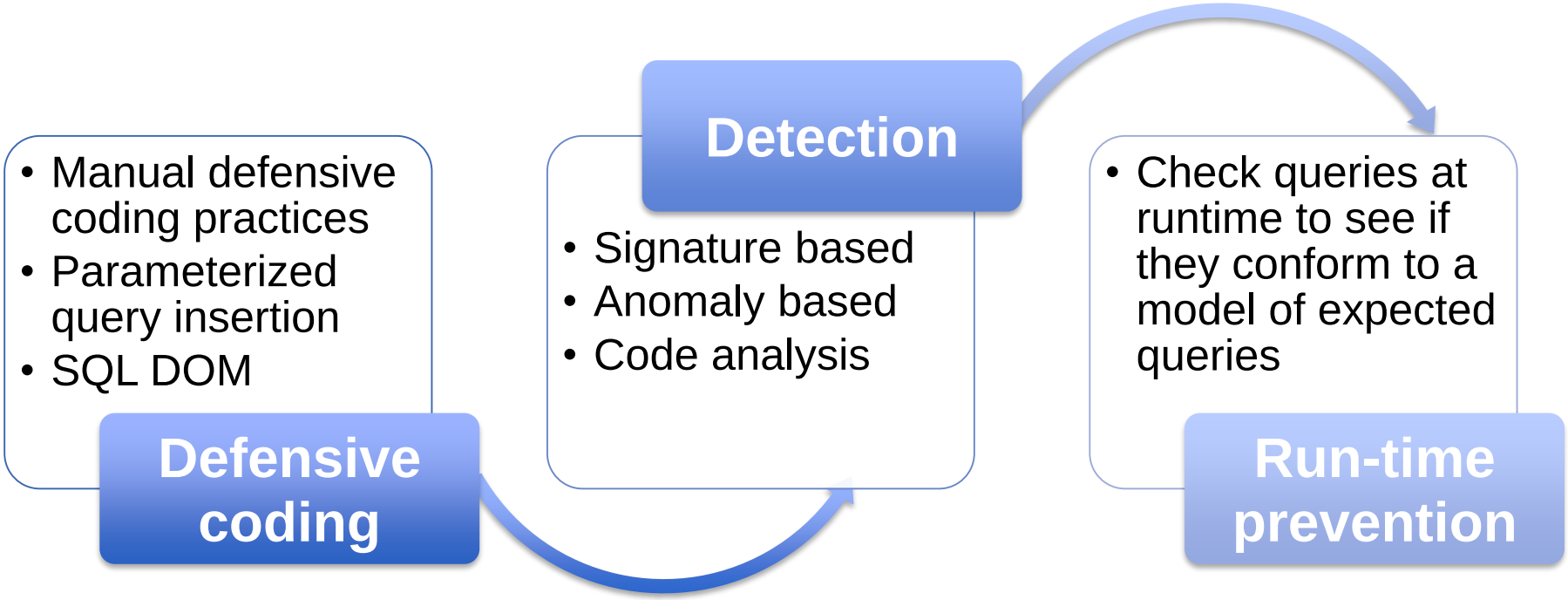
SQL Injection Attacks (SQLi)



SQL Injection Attacks (SQLi)



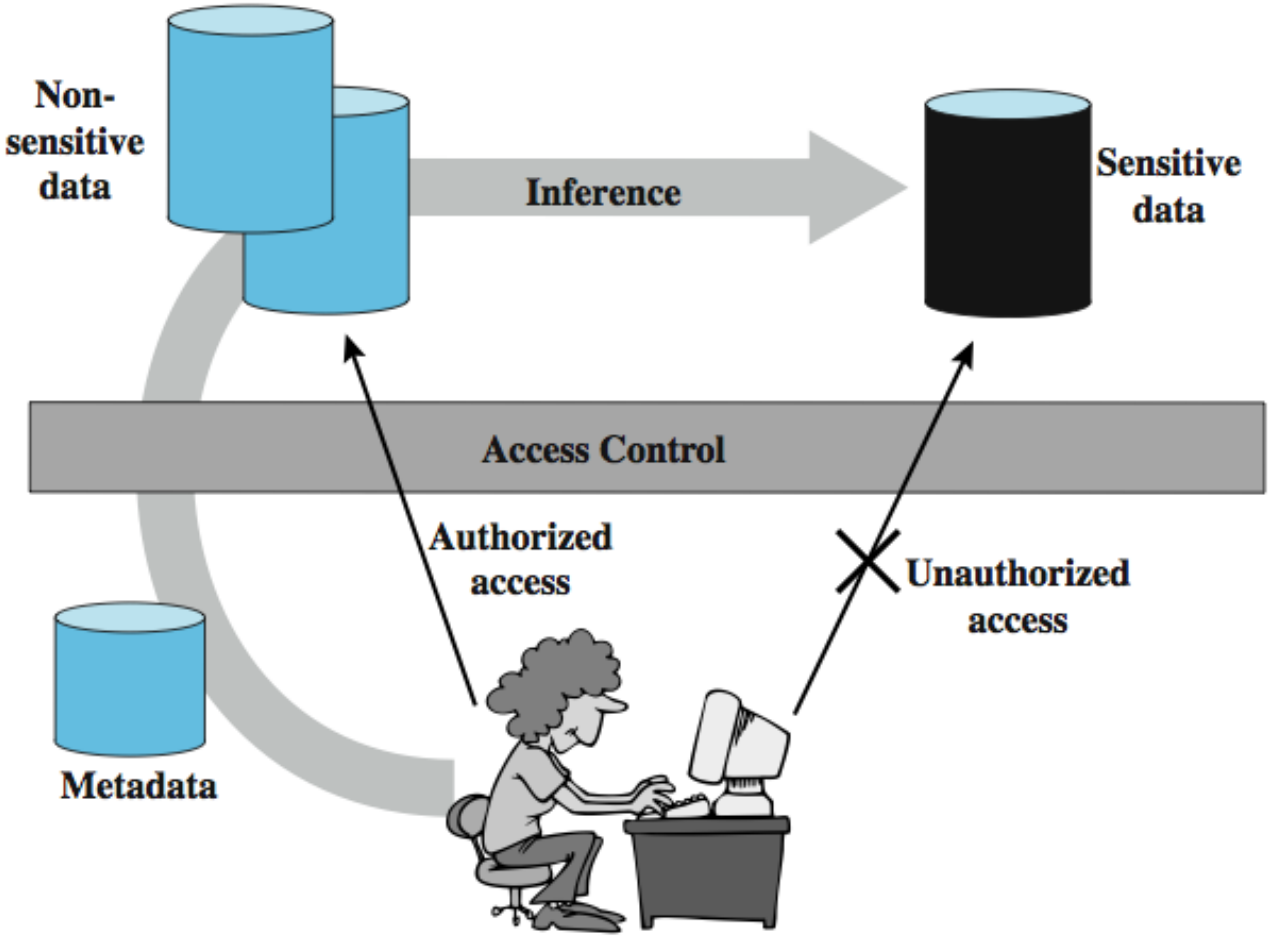
SQLi Countermeasures



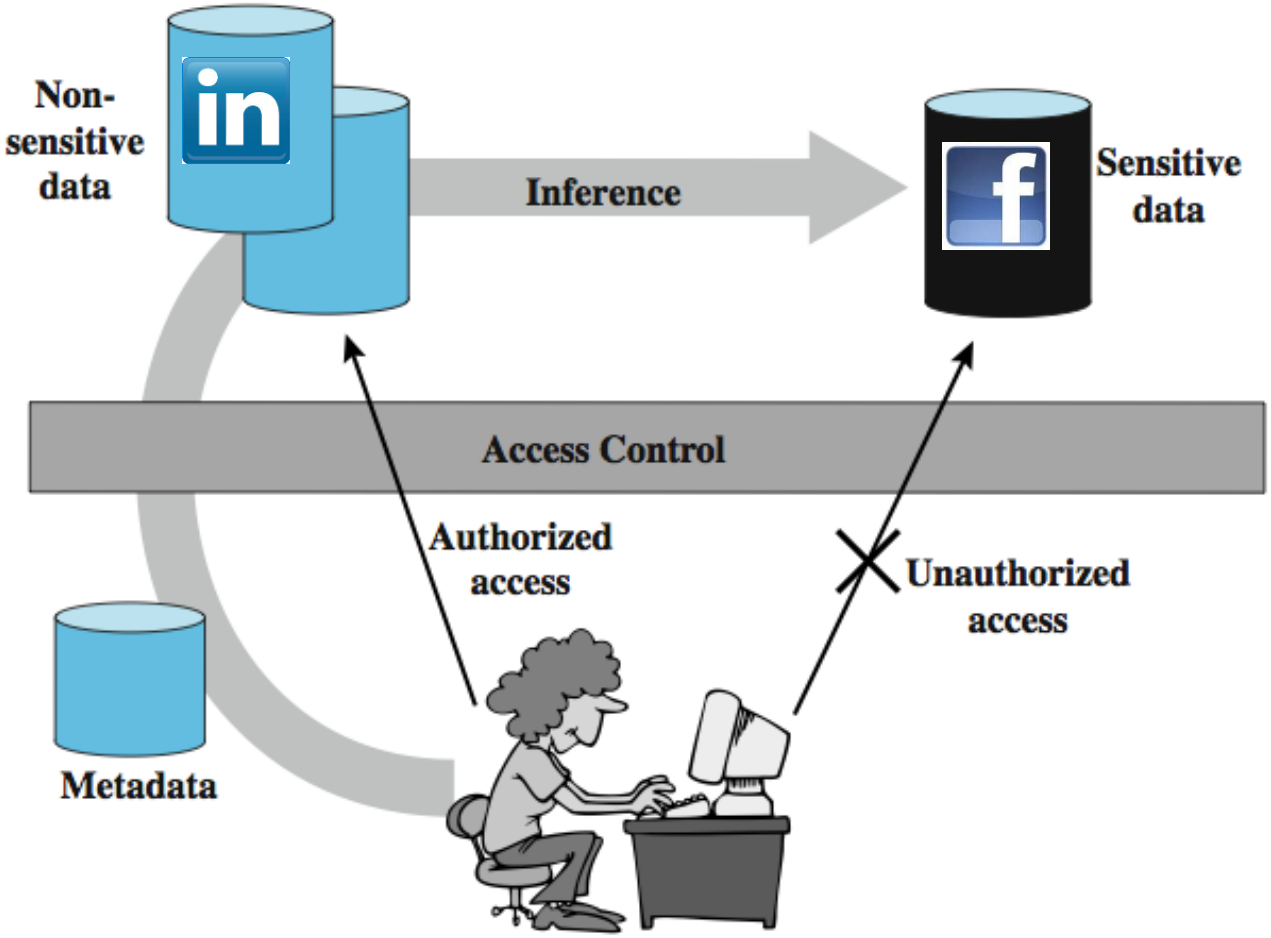
Role-Based Access Control

- role-based access control work well for DBMS
  - eases admin burden, improves security
- categories of database users:
  - application owner
  - end user
  - administrator
- DB RBAC must manage roles and their users
  - cf. RBAC on Microsoft's SQL Server

Inference Attacks



Inference Attacks



Inference Example

Name	Position	Salary ($)	Department	Dept. Manager
Andy	senior	43,000	strip	Cathy
Calvin	junior	35,000	strip	Cathy
Cathy	senior	48,000	strip	Cathy
Dennis	junior	38,000	panel	Herman
Herman	senior	55,000	panel	Herman
Ziggy	senior	67,000	panel	Herman

(a) Employee table

Position	Salary ($)
senior	43, 000
junior	35,000
senior	48,000

Name	Department
Andy	strip
Calvin	strip
Cathy	strip

(b) Two views

Name	Position	Salary ($)	Department
Andy	senior	43,000	strip
Calvin	junior	35,000	strip
Cathy	senior	48,000	strip

(c) Table derived from combining query answers

Inference Countermeasures

- inference detection at database design
  - alter database structure or access controls
- inference detection at query time
  - by monitoring and altering or rejecting queries
- need some inference detection algorithm
  - a difficult problem
  - cf. employee-salary example

Statistical Databases

- provides data of a statistical nature
  - e.g. counts, averages
- two types:
  - pure statistical database
  - ordinary database with statistical access
    - some users have normal access, others statistical
- access control objective to allow statistical use without revealing individual entries
- security problem is one of inference

Statistical Database Security

- use a characteristic formula C
  - a logical formula over the values of attributes
  - e.g. `(`*Sex*`=Male) AND ((`*Major*`=CS) OR (`*Major*`=EE))`

- query set X(*C*) of characteristic formula *C*, is the set of records matching C

- a statistical query is a query that produces a value calculated over a query set

Statistical Database Example

(a) Database with statistical access with $N = 13$ students

Name	Sex	Major	Class	SAT	GP
Allen	Female	CS	1980	600	3.4
Baker	Female	EE	1980	520	2.5
Cook	Male	EE	1978	630	3.5
Davis	Female	CS	1978	800	4.0
Evans	Male	Bio	1979	500	2.2
Frank	Male	EE	1981	580	3.0
Good	Male	CS	1978	700	3.8
Hall	Female	Psy	1979	580	2.8
Iles	Male	CS	1981	600	3.2
Jones	Female	Bio	1979	750	3.8
Kline	Female	Psy	1981	500	2.5
Lane	Male	EE	1978	600	3.0
Moore	Male	CS	1979	650	3.5

(b) Attribute values and counts

Attribute A_j	Possible Values	$\lvert A_j \rvert$
Sex	Male, Female	2
Major	Bio, CS, EE, Psy, ...	50
Class	1978, 1979, 1980, 1981	4
SAT	310, 320, 330, ..., 790, 800	50
GP	0.0, 0.1, 0.2, ..., 3.9, 4.0	41

Statistical Database Example

Grade of a student should not be revealed to Adv... (not even of Baker, EE student!)...

(a) Database with statistical access with $N = 13$ students

Name	Sex	Major	Class	SAT	GP
Allen	Female	CS	1980	600	3.4
Baker	Female	EE	1980	520	2.5
Cook	Male	EE	1978	630	3.5
Davis	Female	CS	1978	800	4.0
Evans	Male	Bio	1979	500	2.2
Frank	Male	EE	1981	580	3.0
Good	Male	CS	1978	700	3.8
Hall	Female	Psy	1979	580	2.8
Iles	Male	CS	1981	600	3.2
Jones	Female	Bio	1979	750	3.8
Kline	Female	Psy	1981	500	2.5
Lane	Male	EE	1978	600	3.0
Moore	Male	CS	1979	650	3.5

(b) Attribute values and counts

Attribute A_j	Possible Values	$\|A_j\|$
Sex	Male, Female	2
Major	Bio, CS, EE, Psy, ...	50
Class	1978, 1979, 1980, 1981	4
SAT	310, 320, 330, ..., 790, 800	50
GP	0.0, 0.1, 0.2, ..., 3.9, 4.0	41

Statistical Database Example

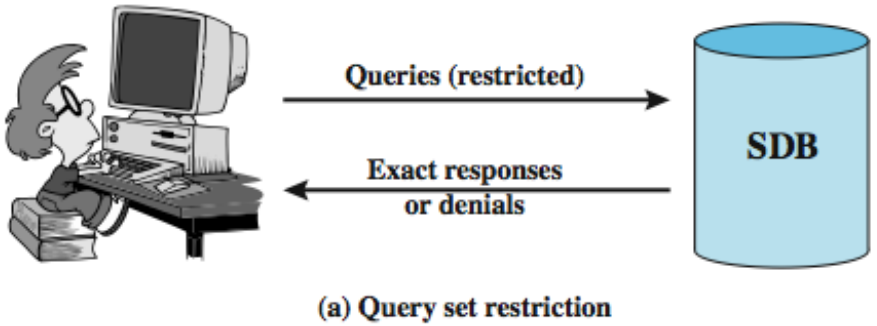
(a) Database with statistical access with $N = 13$ students

Name	Sex	Major	Class	SAT	GP
Allen	Female	CS	1980	600	3.4
Baker	Female	EE	1980	520	2.5
Cook	Male	EE	1978	630	3.5
Davis	Female	CS	1978	800	4.0
Evans	Male	Bio	1979	500	2.2
Frank	Male	EE	1981	580	3.0
Good	Male	CS	1978	700	3.8
Hall	Female	Psy	1979	580	2.8
Iles	Male	CS	1981	600	3.2
Jones	Female	Bio	1979	750	3.8
Kline	Female	Psy	1981	500	2.5
Lane	Male	EE	1978	600	3.0
Moore	Male	CS	1979	650	3.5

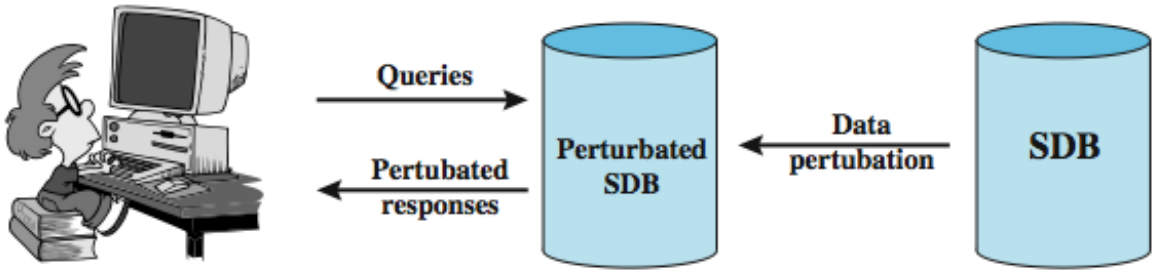
(b) Attribute values and counts

…
Count (EE*Female)=1
Sum(EE*Female,GP)=2.5

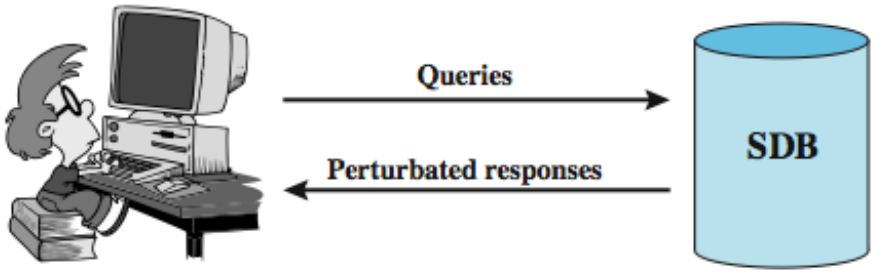
Protecting Against Inference



(a) Query set restriction



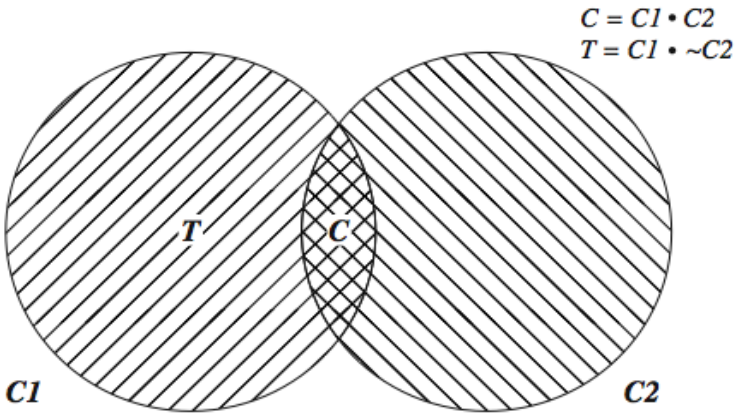
(b) Data perturbation



(c) Output perturbation

Tracker Attacks

- divide queries into parts
  - C = C1.C2
  - count(C.D) = count(C1) - count (C1. ~*C2*)
- combination is called a tracker
- each part acceptable query size
- overlap is desired result



Other Query Restrictions

- query set overlap control
  - limit overlap between new & previous queries
  - has problems and overheads
- partitioning
  - cluster records into exclusive groups
  - only allow queries on entire groups
- query denial and information leakage
  - denials can leak information
  - to counter must track queries from user

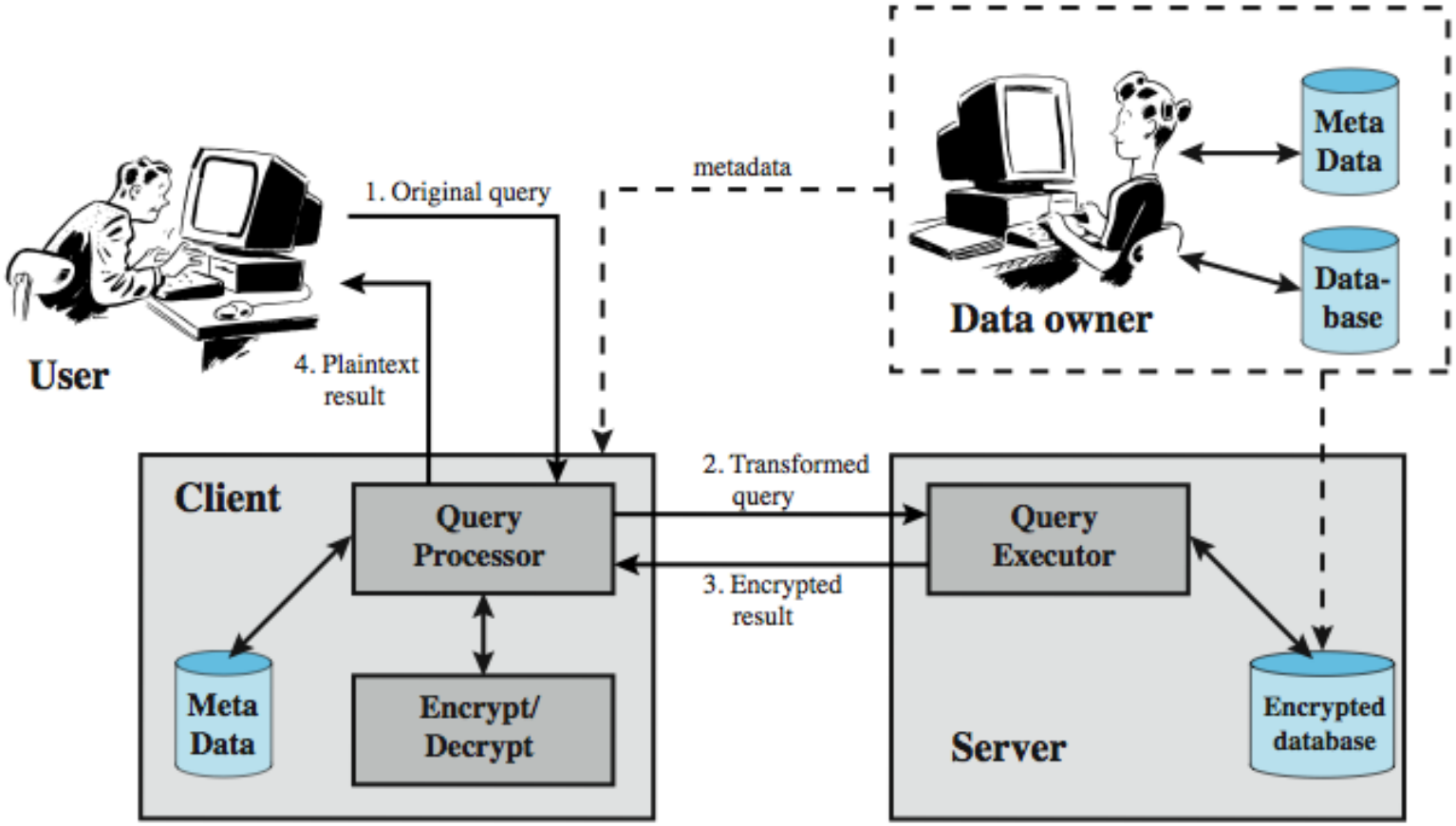
Perturbation

- add noise to statistics generated from data
  - will result in differences in statistics
- data perturbation techniques
  - data swapping
  - generate statistics from probability distribution
- output perturbation techniques
  - random-sample query
  - statistic adjustment
- must minimize loss of accuracy in results

Database Encryption

- databases typical a valuable info resource
  - protected by multiple layers of security: firewalls, authentication, O/S access control systems, DB access control systems, and database encryption
- can encrypt
  - entire database - very inflexible and inefficient
  - individual fields - simple but inflexible
  - records (rows) or columns (attributes) - best
    - also need attribute indexes to help data retrieval
- varying trade-offs

Database Encryption



Summary

- introduced databases and DBMS
- relational databases
- database access control issues
  - SQL, role-based
- Injection & inference attacks
- statistical database security issues
- database encryption