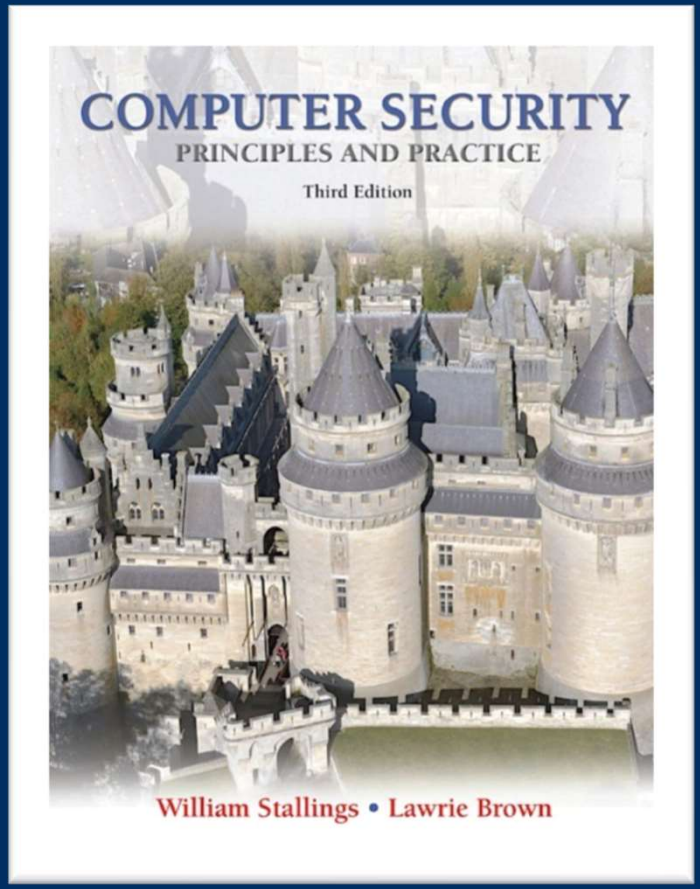
Database Security

Georgia Tech College of Computing



modified from slides of Lawrie Brown

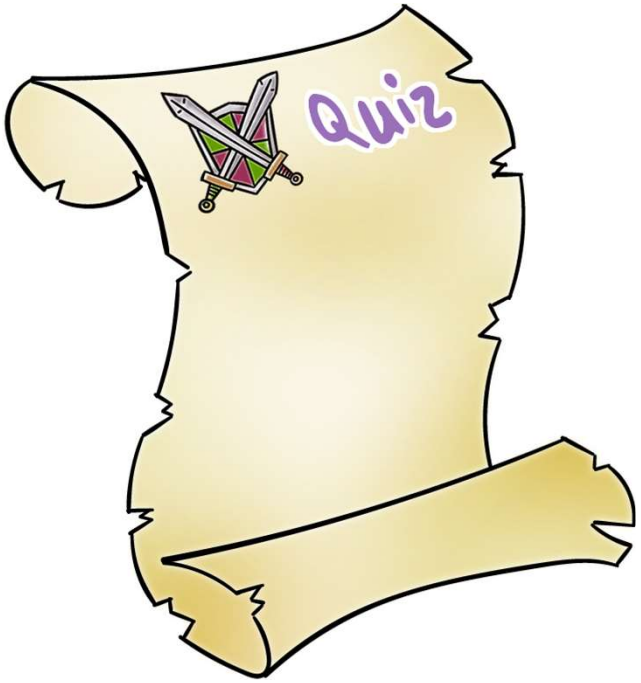
Revisiting Assurance

Assurance: Ways of convincing ourselves
that a model, design, & implementation are correct

Methods of assurance validation:

- Testing /Penetration testing
- Formal verification Validation
- Checking that developers have implemented all requirements
- Requirements checking, design & code reviews, system testing

Revisiting Assurance



Testing:

- Demonstrate existence of problem
- Cannot demonstrate absence of problem
- **Regression testing**: ensure that alterations do not break existing functionality / performance (regression: “going backwards”)

Revisiting Assurance

Challenges:

- Test case generation
- Code coverage
- Exponential number of different executions
- Different execution environments

Penetration testing:

- Ethical hackers attempt to defeat security measures
- Cannot demonstrate absence of problem

Revisiting Assurance

Formal verification: Checking a mathematical specification of program to ensure that security assertions hold.

- **Model checking**, automated theorem proving
- State variables w/ initial assignment, program specification describing how state changes, boolean predicates over state variables
- **Difficulty:** exponential time & space worst case complexity
- Model checking pioneers won the 2007 Turing Award

Security Evaluations

Government Security Evaluations

- **U.S. Orange Book (late 1970's)**
- D < C1 < C2 < B1 < B2 < B3 < A1
  - **D:** no protection
  - **C:** discretionary protection
  - **B:** mandatory protection
  - **A:** Verified protection
- C1, C2, B1: security features common to commercial OSes
- B2: Proof of security of underlying model, narrative spec of TCB
- B3, A1: Formal design & proof of TCB

Government Security Evaluations

Common Criteria (2005) international standard replaced orange book

- Originated out of European, Canadian, and US standards
- **Idea:** users specify system needs, vendors implement solution and make claims about security properties, evaluators determine whether vendors actually met claims
- **Evaluation assurance level** (EAL) rates systems
  - EAL1 most basic, EAL7 most rigorous

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
- database management system (DBMS)
  - suite of programs for constructing and maintaining the database
    - ad hoc query facilities to multiple users and applications
  - Query language
    - provides a uniform interface to the database

Relational Databases

- table of data consisting of rows and columns
  - each column holds a particular type of data
  - each row contains a specific value for each column
  - ideally has one column where all values are unique, forming an identifier/key for that row
    - enables the creation of multiple tables linked together by a unique identifier that is present in all tables
- use a relational query language to access the database
  - request data that fit a given set of criteria

Relational Database Elements

- 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

Structured 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 statements

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

Injection Technique

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

IMPORTANT ! ! ! ! ! ! ! ! !

SQLi Attack Avenues

User input

- Attackers inject SQL commands by providing suitable crafted user input

Server variables

- Attackers can forge the values that are placed in HTTP and network headers and exploit this vulnerability by placing data directly into the headers

Second-order injection

- A malicious user could rely on data already present in the system or database to trigger an SQL injection attack, so when the attack occurs, the input that modifies the query to cause an attack does not come from the user, but from within the system itself

Cookies

- An attacker could alter cookies such that when the application server builds an SQL query based on the cookie's content, the structure and function of the query is modified

Physical user input

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

SQLi Example

Incorrectly filtered escape characters

Statement = "**SELECT*** **FROM** users **WHERE** name='" + userName + "';"

- If the following is entered for the username

' OR '1' ='1

- You get the following SQL statement

you get True all the time:
get the whole list of users!

SELECT * **FROM** users **WHERE** name = ' ' **OR** '1'='1';

SQLi Example

Incorrect type handling

statement := "**SELECT** * **FROM** userinfo **WHERE** id = " + a_variable + ";”

- If a string is entered instead of an integer

1;DROP TABLE users

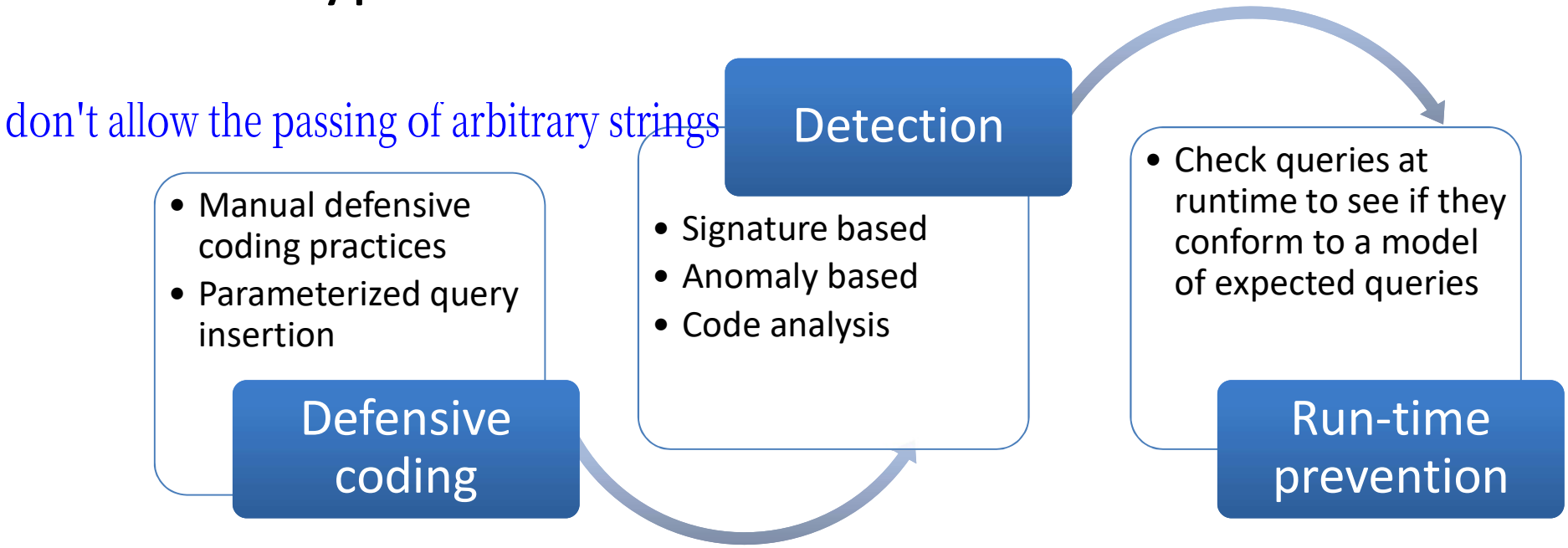
- You get the following SQL statement

"DROP TABLE" command should be protected by administrator rights. Not everyone can execute DROP TABLE

SELECT * **FROM** userinfo **WHERE** id = 1; **DROP TABLE** users;

SQLi Countermeasures

Three types:



Anomaly: looks for anything unusual

Database Access Control

database access control system determines:

if the user has access to the entire database or just portions of it

what access rights the user has

(create, insert, delete, update, read, write)

can support a range of administrative policies

centralized administration

- small number of privileged users may grant and revoke access rights

ownership-based administration

- the creator of a table may grant and revoke access rights to the table

decentralized administration

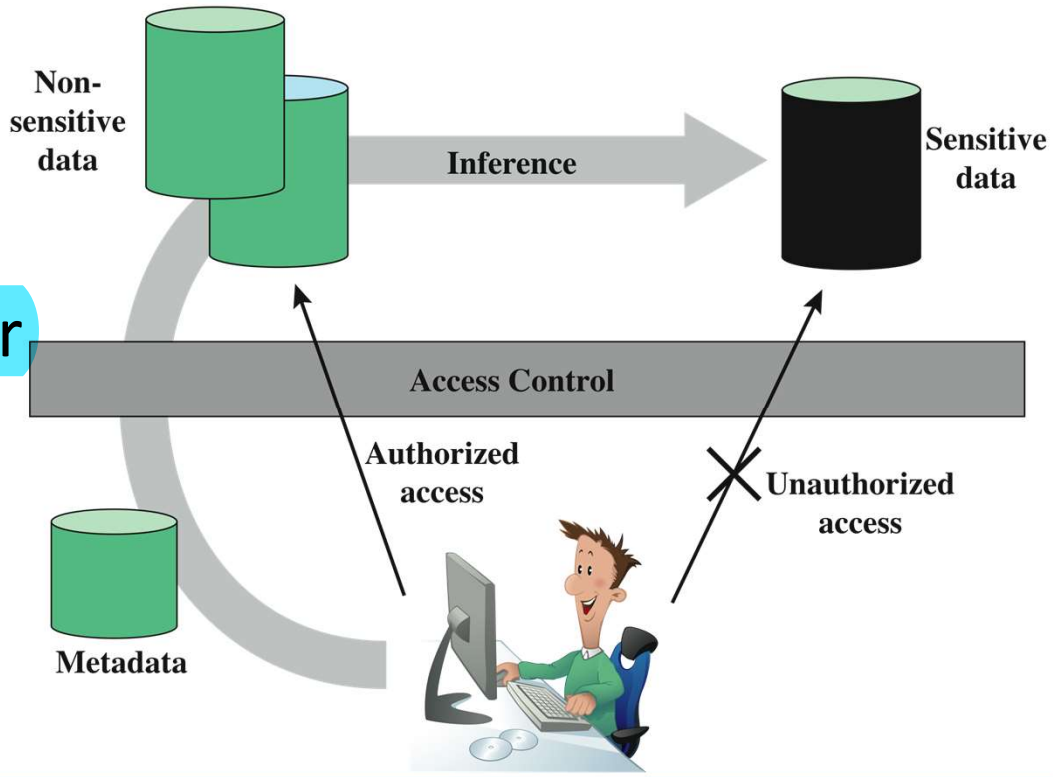
- the owner of the table may grant and revoke authorization rights to other users, allowing them to grant and revoke access rights to the table

SQL Access Controls

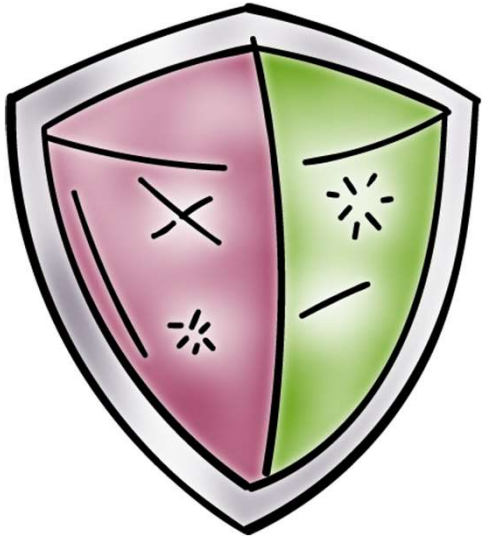
- two commands for managing access rights:
  - grant
    - used to grant one or more access rights or can be used to assign a user to a role
  - revoke
    - revokes the access rights
- typical access rights are:
  - select, insert, update, delete, references

Inference

- Performing queries to deduce unauthorized information from the legitimate responses received
- inference channel
  - information transfer path by which unauthorized data is obtained



Defenses Against Inference Attacks



- Do not allow aggregate query results when the set of tuples selected is either too small or too large

- Transform data by removing identifying information
  - Deindentification
  - Anonymization
  - This has to be done with care

Statistical Databases (SDB)

- provides data of a statistical nature such as counts and averages
  - pure statistical database
  - ordinary database with statistical access
- access control objective
  - provide users with the needed information
  - without compromising the confidentiality
- security problem is one of inference

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	$\|A^j\|$
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

Latanya Sweeney

- Professor in Residence at Harvard
- Director of Data Privacy Lab at Harvard
- Chief Technologist at FTC
- “87% of U.S. population is uniquely identified by DOB, gender, and postal code”

Latanya Sweeney's Finding

- In Massachusetts, the Group Insurance Commission (GIC) is responsible for purchasing health insurance for state employees. GIC has to publish the data:

GIC(**zip, dob, sex**, diagnosis, procedure, ...)

- Sweeney paid $20 and bought the voter registration list for Cambridge Massachusetts:

VOTER(name, party, ..., **zip, dob, sex**)

Latanya Sweeney's Finding

GIC(**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 VOTER share his **dob**
- only 3 of them were man (same **sex**)
- Weld was the only one in that **zip**

- Sweeney learned Weld's medical records

Perturbation

- data perturbation technique
  - data swapping
  - generate statistics from probability distribution
- output perturbation technique
  - statistic adjustment
  - random-sample query
- *goal* is to minimize the differences between original results and perturbed results
  - *challenge* is to determine the average size of the error to be used

Other Query Restrictions

- query set overlap control
  - limit overlap between new and previous queries
- partitioning
  - cluster records into a number of mutually exclusive groups
  - query the statistical properties of each group as a whole
- query denial and information leakage
  - denials can leak information
  - to counter, must track queries from user

Database Encryption

- database is typically the most valuable information resource for any organization
  - protected by multiple layers of security
    - firewalls, authentication, O/S access control systems, DB access control systems, database encryption
- encryption is often implemented with particularly sensitive data
  - at record, attribute, or individual field levels
- disadvantages to encryption:
  - key management
  - inflexibility

Summary

- database
  - structured collection of data
- database management system (DBMS)
  - programs for constructing and maintaining the database
- structured query language (SQL)
  - language used to define schema/manipulate/query data in a relational database
- SQL injection attacks
  - A typical SQLi attack
  - The injection technique
  - SQLi attack avenues and types
  - SQLi countermeasures
- relational database
  - table of data consisting of rows (tuples) and columns (attributes)
  - multiple tables tied together by a unique identifier that is present in all tables

- database access control
  - centralized/ownership-based/decentralized administration
- role-based access control (RBAC)
  - application owner/end user other than application owner/administrator
- inference channel
  - information transfer path by which unauthorized data is obtained
- statistical database (SDB)
  - query restriction/perturbation/data swapping/random-sample query
- database encryption
- cloud computing/security/ data protection
  - multi-instance/ multi-tenant model
  - Cloud security as a service