CONSTRUCTOR UNIVERSITY

Security & Authorization

Ramakrishnan & Gehrke, Chapter 21

Introduction

- Secrecy:
  Users should not be able to see things they are not supposed to
  - Ex: student can't see other students' grades
- Ex: *TJX*. owns many dept stores in US
  - Attacks exploited WEP used at branches
  - Over 47 million CC #s stolen dating back to 2002
  - *…sue filed by consortium of 300 banks*
- Ex: *CardSystems, Inc:* US credit card payment processing company
  - 263,000 CC #s stolen from database via SQL injection (June 2005)
  - 43 million CC #s stored unencrypted, compromised
  - *…out of business*

Introduction / contd.

- Secrecy:
  Users should not be able to see things they are not supposed to
  - Ex: student can't see other students' grades
- Ex: *Equifax 2017* [Siliconbeat]
  - Collecting most sensitive citizen data for credit assessment
    - ssn, name, address, birth dates, credit cards, driver's license, history, …
    - 143m customers affected
  - "maybe dozens" of breaches, fix only 6 months after warning
  - hacked due to insufficient internal security; known patch not installed
  - *BTW, senior execs sold 1.8m in stock*

It would be nice to think that perhaps the company was a victim […] of clever hackers using social engineering […], but it appears […] that there is gross incompetence involved.

Introduction / contd.

- Secrecy:
  Users should not be able to see things they are not supposed to
  - Ex: student can't see other students' grades
- Integrity:
  Users should not be able to modify things they are not supposed to
  - Ex: Only instructors can assign grades
- Availability:
  Users should be able to see and modify things they are allowed to
  - Ex: professor can see and set students' grades(but possibly not modify after release)

Database Access Control

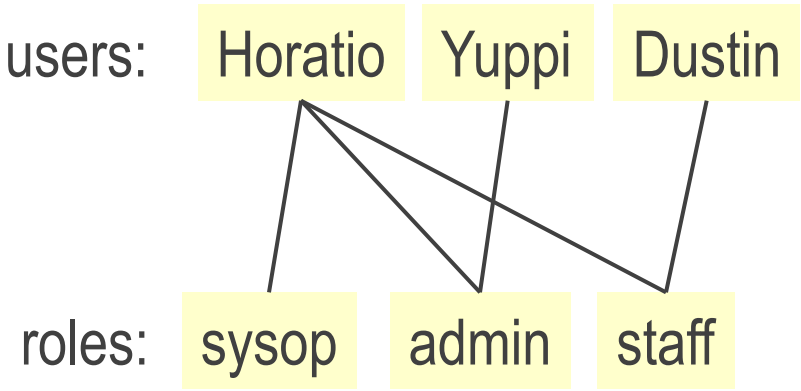
- A security policy specifies who is authorized to do what
- A security mechanism allows us to enforce a chosen security policy
- Two main mechanisms at DBMS level:
  - Discretionary access control (=security at users' discretion)
  - Mandatory access control (=security enforced)

Role-Based Access Control (RBAC)

- RBAC =
  - concept of access rights (privileges) for objects (tables and views)
  - mechanisms for granting & revoking privileges
- Creator of a table or a view automatically gets all privileges on it
- DBMS keeps track of who subsequently gains & loses privileges
- DBMS allows only requests from users with necessary privileges
- Auth/auth
  - Authentication = verifying user identity
  - Authorization = specifying user access rights/privileges to resources

Role-Based Authorization

- SQL-92: privileges assigned to authorization ids
  - single user or group of users
- SQL-99: privileges assigned to roles
  - Roles granted to users & other roles, recursively
  - Reflects real organizations
  - Illustrates how standards often catch up with “de facto” standards embodied in popular systems



GRANT Command

GRANT privileges ON object TO users [WITH GRANT OPTION]

- Privileges =
  - SELECT: Can read all columns
  - INSERT(col-name): Can insert tuples with non-null or non-default values
  - DELETE: Can delete tuples
  - REFERENCES(col-name): Can define foreign keys to this column
- WITH GRANT OPTION: can pass on to others
  - with or without passing on GRANT OPTION
- Only owner can execute CREATE, ALTER, DROP

GRANT and REVOKE of Privileges

- GRANT INSERT, SELECT ON Sailors TO Horatio
  - Horatio can query Sailors or insert tuples into it
- GRANT DELETE ON Sailors TO Yuppy WITH GRANT OPTION
  - Yuppy can delete tuples, and also authorize others to do so
- GRANT UPDATE (rating) ON Sailors TO Dustin
  - Dustin can update (only) the rating field of Sailors tuples
- GRANT SELECT ON ActiveSailors TO Guppy, Yuppy
  - This does NOT allow the 'uppies to query Sailors directly!
- REVOKE cascades: When a privilege is revoked from X, it is also revoked from all users who got it solely from X

Views and Security

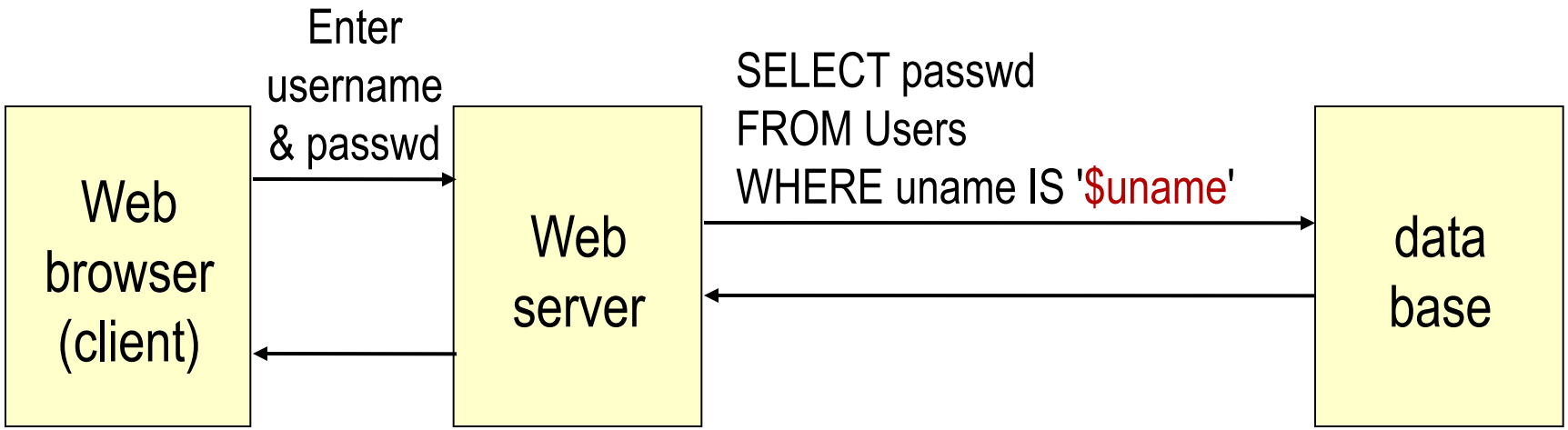
- Views for presenting only necessary information (or summary), hiding details in underlying relation(s)
  - Given ActiveSailors, but not Sailors or Reserves, we can find sailors who have a reservation, but not the bid's of boats that have been reserved
- Creator of view has privilege on view if has privilege on all underlying tables
- Together with GRANT/REVOKE commands, views are powerful access control tool

How to Expose Yourself

An error occured durring processing. Please call support.
Lost connection to MySQL server during query
SQL: select count(*) from LoginsActive where MacAddress=\'00:21:70:6E:04:AE\' and MacAddress!=\'\' and Iface=\'br0\' and PropertyID=\'51225\'
IP:sql.ethostream.com
DBU:remote
DB:

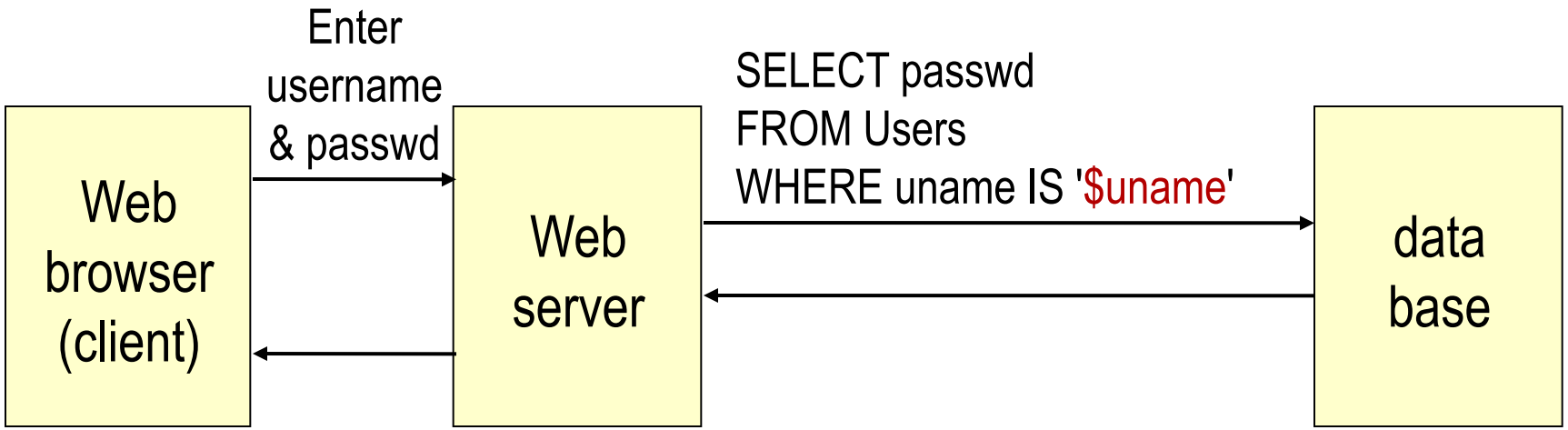
How To Hack a Database

- Most common: SQL injection
  - Compromise database query



How To Hack a Database (contd.)

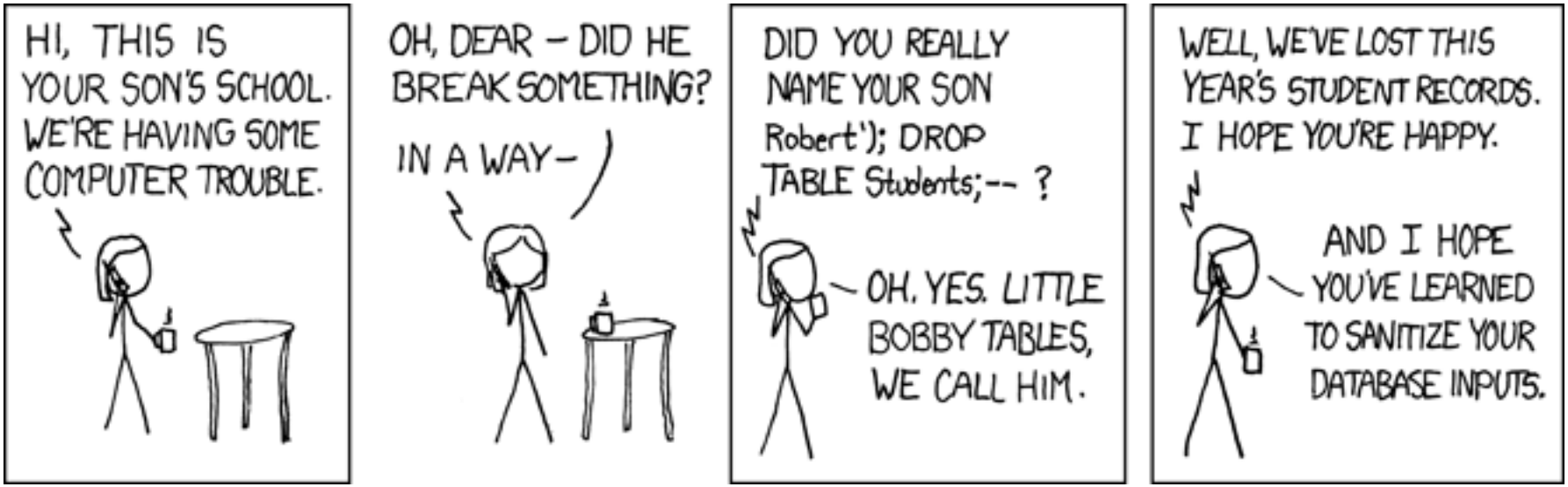
- Most common: SQL injection
  - Compromise database query



- What will happen at input of '; DROP TABLE Users; -- ? (keyword: DoS)
- *Name 2 independent techniques to prevent!*

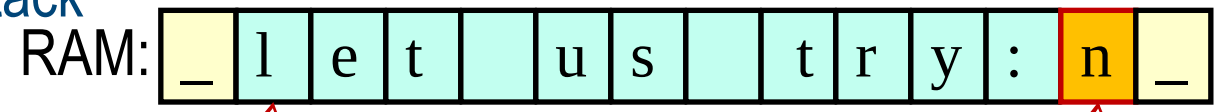
Mom 's a Hacker

[found by: Prashant Vaibhav]



Hacking, Generalized

- SQL injection generalizes to: Command injection
  - ...usually by abusing data paths as command paths
- Ex: buffer overflow attack




```
{    char inputData[11];
     char command;
     switch (command)
     {    case `s`: executeSelect( inputData ); break;
          case `u`: executeUpdate( inputData ); break;
          case `i`: executeInsert( inputData ); break;
          case `d`: executeDelete( inputData ); break;
          case `n`: detonateNuke(); break;
     }
}
```

Biggest Identity Leak to Date

- Discovered by Hold Security,
  reported in the New York times (Aug 5, 2014)
- 420,000 websites compromised,
  1.2 billion user password data, 500 million e-mail addresses
- presumably bots carrying out automated SQL injection attacks

- PS: https://sec.hpi.uni-potsdam.de/leak-checker/

Summary

- 3 main security objectives: secrecy, integrity, availability
  - DB / Web admin responsible for overall security
- DBMS security: role-based access control (RBAC)
  - GRANT, REVOKE
- Internet apps *heavily* increase playground for malicious attacks
  - Ex: SQL injection
  - Your responsibility to keep your site safe !