Risks
- Accidental loss
- Theft and fraud
- Loss of privacy or confidentiality
- Loss of data integrity
- Loss of availability

Outline
- Risk
- Multi-tiered approach
- Views
- Authentication
- Integrity
- Permissions
- Policies and procedures

Risk assessment
- Assets
- Threats
- Vulnerabilities
- Safeguards
- Consequences
- Likelihood

Reading
- Hoffer et al. [2005] Modern Database Management Chapter 12

Introducing security into database design
Security of databases can be defined as preventing unauthorized or accidental disclosure, alteration, or destruction of data. Confidentiality of data that exists in the database must be considered, as should the availability of that data. Securing a database by looking at the various layers that are involved is an effective approach.
Three-Tier Design [n-tier design]

- Three layers of a Web application
  - running on different servers,
  - usually set apart by firewalls that have specific rules to only let traffic through to the specific port on a specific server at whichever layer the user is trying to access:
    - Internet -> Firewall -> Web -> Firewall -> Application -> Firewall -> Database
- It is costly to implement such a design.
- Systems Admin may choose a compromise, combining application and database servers. This is better than leaving a database facing the Internet directly.
- If one of the layers closest to the Internet is compromised, then several more layers still need to be compromised before access to the vital information can be gained.

Web-based database access

- Web port access security
  - All applications are directed to listen at a specific port number on the server. Like any standard HTTP server, the Web Listener can be configured to restrict access.
- XML-based access security
  - The latest trend among web-enabled database is in the area of Web services, specifically the inter-communications between databases over the Internet.
  - Issue of managing security between web portals. Most of these use XML security to verify communications across an insecure network.

Oracle’s 4 tier access

Levels of vulnerability

- Internet access
  - If hackers know the IP address of a server, they can telnet to the server and get a login prompt. A user ID and password will gain access to the server.
- Port access
  - All Web applications are configured to listen on a predefined port for incoming connections, and they generally use a listener daemon process to poll for connections.
- Server access
  - A multi-tiered Web application incorporates Web servers, application servers, and database servers. Each presents a potential point of entry.
- Network access
  - OracleNet, as an example, allows for incoming connect strings directly to the Oracle listener process.
  - If hackers know the port, IP address, Oracle ID, and password, they can gain direct access to the database.
- These possible attack points must have restricted access and disabled external entry.

Access Control

- Access to information contained in the tables must be properly regulated either
  - with control over direct access to the tables, or
  - through views.
- Views and privileges assigned to the views can be created to limit users to only see specified portions of data contained within a table.
- Control over the read, insert, update and delete commands must also be assigned appropriately within those views.

Role-based authentication

Typical roles for access include

- administrator, user, programmer and
- operator.

- For the first three roles, it is fairly obvious what access should be granted

- Operators are expected to play an essential part in the production operation of a system, yet they are often restricted in what type of access they are granted.

- Segregate duties in the operator role, instead of just one operator control over an entire process.

- Operators’ roles need to be carefully defined.

- All roles should have logging enabled to keep track of what occurs.
### Authentication
- **3 level approach**
  - User ID/PIN/password
  - Swipe-card, smart-card, etc
  - Biometric characteristic
- **Mediated access through a 3rd party**

### Encryption
- The sensitivity of the data will logically determine the need for the use of encryption.
- There are a few things to consider:
  - Will the data stored in the database need to be encrypted or just the user passwords?
  - Encrypt the data only in the local instance of the database, or also in transit?

### Passwords
- **Password changing routines**
  - When a user is forced to provide multiple passwords for each component in a system, they often choose passwords that are cyclic in nature. For example, a user may rotate between the passwords “north”, “south”, “east”, and “west”.
- More sophisticated password routines
  - Specify passwords of a minimum length, (greater than five characters),
  - Prohibit the re-use of passwords and
  - Require that the passwords are changed on a periodic basis.
- **Automatic account disabling**
  - Suspend the server ID after e.g. three password attempts.
- **Random password generators**
  - This is not very helpful, and virtually guarantees that users will have written lists of passwords.
- **Use of family names, vehicle registrations etc.**

### Change Control
- It is important to monitor or log changes made to the database, whether structural or to the data itself.
- Changes must be tracked and regulated by interested parties.
- The process must be defined and followed by all roles defined in the database structure.

### Integrity
- **Integrity ensures that the data in the database is valid and accurate.**
- No matter the sensitivity of the data (credit card information vs. a CD collection), if the data isn't right, then what good is the database?
- Determine who owns the data and they are responsible for its quality.
- A good process for ensuring the integrity of the data includes
  - Understanding what is processed
  - Identifying what can be considered personal, critical, or proprietary.
  - Risk must be assessed
  - Accountability must be assigned

### The MySQL Permission Model
- Permissions use the GRANT and REVOKE commands.
- There are four privilege levels that apply:
  - **Global**: these privileges apply to all databases on a server.
  - **Database**: these privileges apply to all tables in a database.
  - **Table**: these apply to all columns within a table.
  - **Column**: these apply to individual columns in a table.
Planning access levels

- The privileges can get very granular, so it is important that they are used in a well planned fashion.
- The types of privileges include:
  - Alter
  - Delete
  - Create
  - Drop
  - Execute
  - Select
  - Update
- Once a database is completely set up, privileges should be reviewed prior to going to any usage of the database to ensure that the privileges were set up appropriately.

Use of the GRANT command as an example

```
GRANT priv_type [(column_list)] [, priv_type [(column_list)] ...]
ON {tbl_name | * | *.* | db_name.*}
TO user_name [IDENTIFIED BY [PASSWORD] 'password'] [, user_name
[IDENTIFIED BY 'password'] ...]
[REQUIRE
  NONE |
  {SSL| X509} |
  [CIPHER cipher [AND]
  [ISSUER issuer [AND]
  [SUBJECT subject]]
  [WITH [GRANT OPTION]
  MAX QUERIES PER HOUR # |
  MAX_UPDATES_PER_HOUR # |
  MAX_CONNECTIONS_PER_HOUR #]
```

Policies and procedures

- A system may be most vulnerable to “abuse” by those on the inside.
- Personnel controls include
  - Careful selection processes
  - Monitoring that procedures are followed
  - Suitable training and awareness
  - Separation of duties to ensure no one employee is responsible for entire process
  - Procedures to remove accesses of staff no longer employed.

Physical access

- Sensitive equipment should be accessed by only those who are authorised
- Swipe card, biometric and password access to sensitive areas
- Controlled access by visitors
- Concerns about remote access – what is the impact of a stolen executive laptop?

Other associated tools

- If the Web server and the database are on the same server (not recommended but very common), then tools like Apache, PHP or Java may be loaded on the same server.
- If these tools are not kept up-to-date (just as with the OS), then the server may be vulnerable to intrusion through Web associated applications.

Database Backups

- [for next lecture]
- Another area that often gets lost in the layers of security is the critical area of database backup and recovery.
- As a part of whatever backup type is used, testing recovery of data is mandatory.
Summary

- Risk
- Multi-tiered approach
- Views
- Authentication
- Integrity
- Permissions
- Procedures and policies