Data Access and Integrity

Database Connectivity and Access

Authentication vs. Authorization

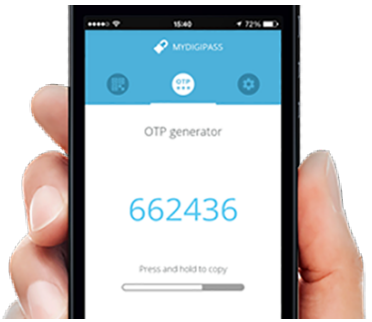
- Authentication – Who are you?
- Authorization – What can you do?

Authentication

- Good authentication requires all three of:
  - Something you know (e.g., password)
  - Something you have (e.g., dongle, cryptographic key)
  - Something about you (e.g., fingerprint, other biometric)

- Id and password
  - How good is this?

- One time password (OTP)



Authentication

- Who does authentication?
  - Application
  - "System"

Authorization

- Authorization can be determined at different levels

- Database level
  - Use of GRANT statement
  - DBA will assign access to users at database level
  - We have to understand the *role to resource* assignments for users

- Application-level
  - Application has an id(s) to access database
  - Developer (via biz rules) will assign access at application level
  - Use of WHERE clause

Application-Level Authorization

- Varied means of accomplishing authorization
  - Application-level
    - Username, password provided to application
    - All users interact with application in same way
    - This approach is frowned upon (effectively sharing a password)
  - User-level
    - Username, password provided to user
    - More complex but more common

Task-Level Authorization

- What can the user do?
  - Task-Based
    - Very granular
    - User is assigned (or not) permission to each and every task
      - May include read/write/create distinction
    - Difficult to maintain
      - Can create features to make it easier

Application-Level Authorization

- Role-Based
  - Username, password provided to user
  - Each user is assigned a role which in turn has certain privileges associated with it
  - Typical roles:
    - Admin
    - Editor
    - General User
    - Public

Application-Level Authorization

- Code considerations
  - Must ask the question:

Is the user authorized to perform this task?

  - Who is the user?
  - What are their rights?
  - Can they do this task?
  - What if they can't?
    - How did they get this far?
    - Why did they get this far?

Business Rules

- Who
- What
- When
- From Where

Checking Authorization

- Method 1
  - Determine user
  - Query database for user access rights
  - Check rights against requested action
- Method 2
  - Issue user "ticket" upon login
  - Check "ticket" for user access rights
  - Check rights against requested action

Tokens

- What is a token?

- Hash
  - One-way access only
  - Very secure
  - Salt: fixed or variable
  - Iteration count

- Encryption
  - Two-way Access
  - Embedded info
  - Less secure, but more useful?

Credential Storage

- User credentials
  - OS service
  - Application
- Application credentials
  - Code
  - File

SOA Considerations

- Where are authentication and authorization determined?
- Is one token passed through or are multiple tokens used?
- What about multiple business layers?

Auditing the Database

- Tracking activity can be as important as data itself
  - Connections, queries, errors, performance…
- Important for record keeping
  - What was the change?
  - Who made the change?
  - Why was the change made?
  - When was the change made?
  - How was the change made?
- May be logged to the same or another table

Auditing the Database

- Auditing types
  - "Snapshot"
    - Use fields in each table to record critical information
  - "full" audit trail
    - Use a separate table to store old and new information as well as metadata about the change

Data Integrity

- Garbage in = Garbage out
- Aggregating can be made difficult or impossible
  - And cleansing is expensive!
- RIT can be written as:
  - "RI of T"
  - "Rochester Institute of Technology"
  - "Roch. Inst. Of Tech."
  - "Rochester Inst. Of Tech."
  - Etc.

Form Validation

- Make data field do the work!
- Can be encouraged:
  - Appropriate form controls
- Can be enforced:
  - At query submission
  - At form processing
  - At information entry
  - At form construction

Handling Data Modification

- Must ensure data consistency, referential integrity
- Cascading of actions
  - Programmatically or at database level
  - Nullification or deletion of children
- Recall discussion on proper *deletion*!

Double Entry Systems

- Accuracy of inserted data may be critical
- What if data is free-form and not easily tested?
- Enter the data twice!
  - Operator A enters data
    - Status flag set to Hold
  - Operator B enters data
    - Data is compared
      - Adjusted (if necessary) and saved
      - Status flag set to Confirmed