Database Management Systems

- Security

Doug Shook

Types of Security

- Ethical Considerations
- Legal Considerations
- System Considerations
- Multiple Security Levels

Types of Threats

- Loss of integrity
- Loss of availability
- Loss of confidentiality

Access Control

- Who can access what?
  - Based on user accounts
  - Tracks all operations by every user
- Two classifications necessary:
  - How sensitive is the data?
  - Who has the right to see sensitive data?

Sensitivity

- Many ways to classify:
  - Inherently sensitive
  - From a sensitive source
  - Declared sensitive
  - Partial sensitivity
- When is it safe to reveal sensitive data?
  - Data availability
  - Access acceptability
  - Authenticity Assurance

Privileges

- Used for discretionary access control
- Account Level
  - What operations can be performed?
- Relation Level
  - What data can be seen?
    - SELECT
    - Modify
    - References

GRANT

- GRANT CREATE ON db TO A1;
- GRANT INSERT, DELETE ON EMPLOYEE, DEPARTMENT TO A2;
- GRANT SELECT ON EMPLOYEE, DEPARTMENT TO A3 WITH GRANT OPTION;
- GRANT SELECT ON EMPLOYEE TO A4;
- REVOKE SELECT ON EMPLOYEE FROM A3;

Role Based Control

- Idea: permissions should be associated with roles instead of (or in addition to) users.

- Create roles
- Add users to roles
- Grant permissions on the roles

- What if a user permission conflicts with a role permission?
  - Or two role permissions conflict?

Common Attacks

- Unauthorized privilege escalation
- Privilege Abuse
- Denial of Service
- Weak Authentication
- SQL Injection

SQL Injection

- Our applications must accept input from users
  - What if that input isn't what we expect?
- SELECT * FROM users WHERE username = 'jake' and PASSWORD = 'jakespasswd'
- SELECT * FROM users WHERE username = 'jake' and (PASSWORD = 'jakespasswd' or 'x' = 'x')

SQL Injection

- Code Injection

- Function Call Injection
  - SELECT TRANSLATE (‘user input’, ‘from_string’, ‘to_string’) FROM dual;

  - SELECT TRANSLATE (“ || UTL_HTTP.REQUEST (‘http://129.107.2.1/’) || ”, '98765432’, ‘9876’) FROM dual;

SQL Injection Risks

- Database Fingerprinting
- Denial of Service
- Bypassing Authentication
- Identifying Injectable Parameters
- Executing Remote Commands
- Performing Privilege Escalation

Protecting Against SQL Injection

- Binding Parameters
  - PreparedStatement stmt = conn.prepareStatement("SELECT * FROM EMPLOYEE WHERE EMPLOYEE_ID=? AND PASSWORD=?");
    stmt.setString(1, employee_id);
    stmt.setString(2, password);
- Input Validation
- Function Security

SQL Injection Exercise

- http://sqlzoo.net/hack/

Encryption

- We are often communicating data
  - What if that data gets in the wrong hands?
- Encryption obscures data
  - Ciphertext
  - Plaintext
  - Encryption
  - Decryption

Encryption

- Symmetric Key
  - A secret key that both parties have
  - Used to encrypt/decrypt
  - Problems?

- Asymmetric (Public) key encryption
  - Two keys: public and private
  - Send the public key freely
  - Keep private key a secret

Asymmetric Encryption

- Generate two keys, mark one as public
- Encrypt the message with the other person's public key
- Use your secret key to decrypt the message
- Larger keys are more secure

RSA Encryption

- Choose a large value, n, that is the product of two prime numbers, p and q
- Compute the totient as lcm(p – 1, q – 1)
- e can be any value that is coprime to this totient
- d is computed as e mod totient
- Encrypt: m^e mod n
- Decrypt: m^d mod n
- How can we break this scheme?

Security Challenges

- Data Quality
- Intellectual Property Rights
- Database Survivability
  - Confinement
  - Damage Assessment
  - Reconfiguration
  - Repair
  - Fault Treatment