IIT Madras

BSc Degree

Security

Security

- Access Control
- Web-based Mechanisms
- Session management
- HTTPS
- Logs and Analysis

Access Control

What is access control?

- Access: being able to read/write/modify information
- Not all parts of application for public access
  - Personal, Financial, Company, Grades, …
- Types of access:
  - read-only
  - read-write (CRUD)
  - modify but not create
  - …

Examples

- Linux files:
  - owner, group: access your own files, cannot modify (or even read?) others
  - can be changed by owner
  - “root” or “admin” or “superuser” has power to change permissions

Examples

- Linux files:
    - owner, group: access your own files, cannot modify (or even read?) others
    - can be changed by owner
    - “root” or “admin” or “superuser” has power to change permissions
- Email:
    - you can read your own email
    - can forward an email to someone else - this is also access!

Examples

- Linux files:
  - owner, group: access your own files, cannot modify (or even read?) others
  - can be changed by owner
  - “root” or “admin” or “superuser” has power to change permissions
- Email:
  - you can read your own email
  - can forward an email to someone else - this is also access!
- E-commerce login:
  - shopping cart etc visible only to user
  - financial information (credit card etc.) must be secure

Discretionary vs Mandatory

- Discretionary:
  - you have control over who you share with
  - forwarding emails, changing file access modes etc possible

- Mandatory:
  - decisions made by centralized management - users cannot even share information without permission
  - Typically only in military or high security scenarios

Role-based access control

- Access associated with “role” instead of “username”

Role-based access control

- Access associated with “role” instead of “username”
- Example:
    - Head of department has access to student records
    - What happens when HoD changes?

Role-based access control

- Access associated with “role” instead of “username”
- Example:
  - Head of department has access to student records
  - What happens when HoD changes?
- Single user can have multiple roles
  - HoD, Teacher, Cultural advisor, sports club member, …

Role-based access control

- Access associated with “role” instead of “username”
- Example:
    - Head of department has access to student records
    - What happens when HoD changes?
- Single user can have multiple roles
    - HoD, Teacher, Cultural advisor, sports club member, …
- Hierarchies, Groups
    - HoD > Teacher > Student
    - HoD vs sports club member? - no hierarchy here

Attribute-based access control

- Attribute
  - time of day
  - some attribute of user (citizenship, age, …)
- Can add extra capability over role-based

Policies vs Permissions

- Permissions
  - Static rules usually based on simple checks (does user belong to group)?
- Policies
  - More complex conditions possible
  - Combine multiple policies
  - Example:
    - Bank employee can view ledger entries
    - Ledger access only after 8am on working days

Principle of least privilege

- Entity should have minimal access required to do the job
- Example: Linux file system
  - users can read system libraries but not write
  - some files like /etc/shadow not even readable
  - you can install Python to local files using “venv” but not to system path
- Benefits
  - better security - fewer people with access to sensitive files
  - better stability - user cannot accidentally delete important files
  - ease of deployment - can create template filesystems to copy

Privilege escalation

- Change user or gain an attribute
  - “sudo” or “su”
- Usually combined with explicit logging, extra safety measures
- Recommended:
  - do **not** sudo unless absolutely necessary
  - never operate as root in a Linux/Unix environment unless absolutely necessary

Context: Web apps

- Admin dashboards, user access, etc.
- Gradebook example:
    - only admin should be able to add/delete/modify
    - users should have read permissions only on their own data

Enforcing

- Hardware level
  - Security key, hardware token for access, locked doors etc

Enforcing

- Hardware level
  - Security key, hardware token for access, locked doors etc
- Operating system
  - filesystem access, memory segmentation

Enforcing

- Hardware level
    - Security key, hardware token for access, locked doors etc
- Operating system
    - filesystem access, memory segmentation
- Application level
    - DB server can restrict access to specific database

Enforcing

- Hardware level
  - Security key, hardware token for access, locked doors etc
- Operating system
  - filesystem access, memory segmentation
- Application level
  - DB server can restrict access to specific database
- Web application
  - Controllers enforce restrictions
  - Decorators in Python used in frameworks like Flask