UNIVERSITÄT PASSAU

Fakultät für Informatik und Mathematik

6090: Security of Computer and Embedded Systems

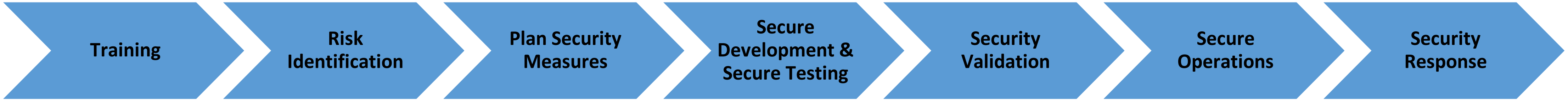
Week 2: Computer and Embedded Systems Security Fundamentals; Access Control

Elif Bilge Kavun

elif.kavun@uni-passau.de

October 26, 2021

Secure Design Flow



- The course roughly follows secure design flow/secure software lifecycle
  - Foundations and Security Technologies
    - Access Control
    - Cryptography
    - Security Protocols
  - Building Secure Systems
    - Risk Identification, Analyzing Systems
    - Analyzing Security Protocols
    - Application Security & Secure Programming
    - Security Testing

- In this lecture you will
    - learn fundamental security metrics and their definitions
    - learn basic access control mechanisms

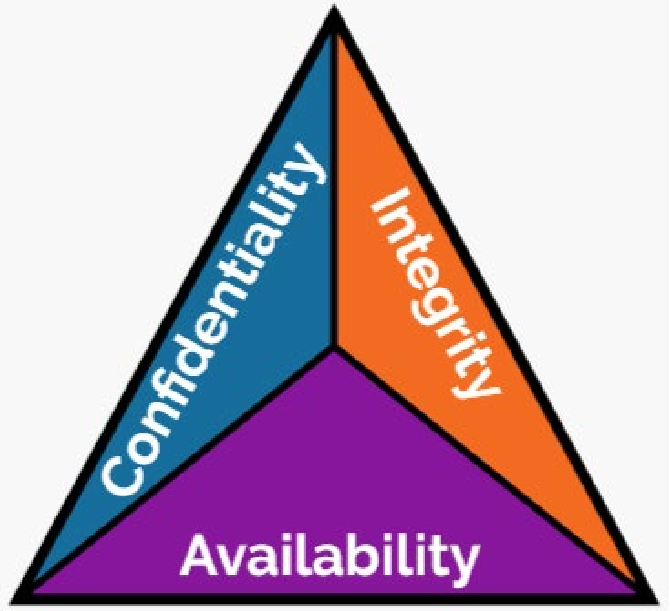
Identity and AAA
(Authentication, Authorization, and Access Control)

Identity and AAA
(Authentication, Authorization, and Access Control)

- The Three Fundamental Concepts of Security: **CIA**

Identity and AAA
(Authentication, Authorization, and Access Control)

- The Three Fundamental Concepts of Security: **CIA**
  - **C**onfidentiality
    - Protecting information from disclosure to unauthorized parties
  - **I**ntegrity
    - Protecting information from being modified by unauthorized parties
  - **A**vailability
    - Ensuring that information is available (accessible) to authorized parties



Identity and AAA (Authentication, Authorization, and Access Control)

- To decide if a subject (e.g., a human person) is a member of a authorized party that can access (i.e., execute an operation such as read, write, or execute on) an object (resource) (i.e., a physical object, a function call, data/information), we need to solve

Identity and AAA (Authentication, Authorization, and Access Control)

- To decide if a subject (e.g., a human person) is a member of a authorized party that can access (i.e., execute an operation such as read, write, or execute on) an object (resource) (i.e., a physical object, a function call, data/information), we need to solve
  - Identification
    - Associating an identity with a subject

Identity and AAA
(Authentication, Authorization, and Access Control)

- To decide if a subject (e.g., a human person) is a member of a authorized party that can access (i.e., execute an operation such as read, write, or execute on) an object (resource) (i.e., a physical object, a function call, data/information), we need to solve
  - Identification
    - Associating an identity with a subject
  - Authentication
    - Verifying the validity of something (usually the identity claimed by a system entity)

Identity and AAA
(Authentication, Authorization, and Access Control)

- To decide if a subject (e.g., a human person) is a member of a authorized party that can access (i.e., execute an operation such as read, write, or execute on) an object (resource) (i.e., a physical object, a function call, data/information), we need to solve
    - Identification
        - Associating an identity with a subject
    - Authentication
        - Verifying the validity of something (usually the identity claimed by a system entity)
    - Authorization
        - Granting (or denying) the right or permission of a system entity to access a object

Identity and AAA (Authentication, Authorization, and Access Control)

- To decide if a subject (e.g., a human person) is a member of a authorized party that can access (i.e., execute an operation such as read, write, or execute on) an object (resource) (i.e., a physical object, a function call, data/information), we need to solve
  - Identification
    - Associating an identity with a subject
  - Authentication
    - Verifying the validity of something (usually the identity claimed by a system entity)
  - Authorization
    - Granting (or denying) the right or permission of a system entity to access a object
  - Access Control
    - Controlling access of system entities (on behalf of subjects) to objects based on a access control policy ("security policy")

Mechanisms for Identity Authentication

- The most widely used mechanisms for authentication are
  - Something that you ~~forgot~~ know
    - E.g., a password or a PIN

Mechanisms for Identity Authentication

- The most widely used mechanisms for authentication are
  - Something that you ~~forgot~~ know
    - E.g., a password or a PIN
  - Something that you ~~lost~~ have
    - E.g., a smart card or a one-time password generator

Mechanisms for Identity Authentication

- The most widely used mechanisms for authentication are
  - Something that you ~~forgot~~ know
    - E.g., a password or a PIN
  - Something that you ~~lost~~ have
    - E.g., a smart card or a one-time password generator
  - Something that you ~~were~~ are
    - E.g, Biometric characteristics e.g., a facial scan/photograph

Mechanisms for Identity Authentication

- The most widely used mechanisms for authentication are
  - Something that you ~~forgot~~ know
    - E.g., a password or a PIN
  - Something that you ~~lost~~ have
    - E.g., a smart card or a one-time password generator
  - Something that you ~~were~~ are
    - E.g, Biometric characteristics e.g., a facial scan/photograph
  - Context location, e.g., ~~a place you visited~~ your current location
    - E.g,. Being physical close to an object, being in a secure building

Mechanisms for Identity Authentication

- The most widely used mechanisms for authentication are
  - Something that you ~~forgot~~ know
    - E.g., a password or a PIN
  - Something that you ~~lost~~ have
    - E.g., a smart card or a one-time password generator
  - Something that you ~~were~~ are
    - E.g, Biometric characteristics e.g., a facial scan/photograph
  - Context location, e.g., ~~a place you visited~~ your current location
    - E.g,. Being physical close to an object, being in a secure building

- Multi-factor authentication
  - Use more than one authentication mechanism (at the same time)

Example of Something That You Know: Passwords

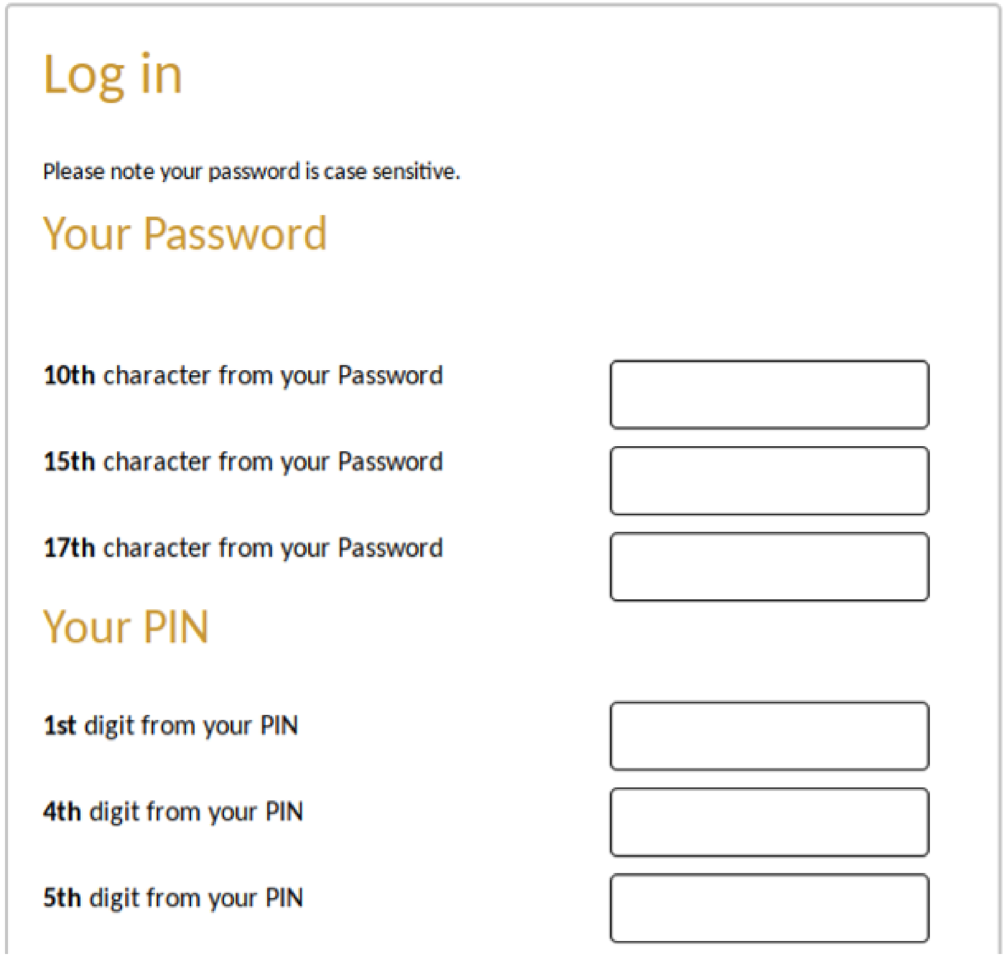
- Passwords
    - Widely used
    - Hard to remember
    - Not always kept secret (social engineering): https://www.youtube.com/watch?v=opRMrEfAIiI

Example of Something That You Know: Passwords

- Good passwords
  - Long and random
- Good systems
  - Allow for passwords of arbitrary length
  - Store passwords hashed and salted (following lectures for details)
- Does it really help enforcing users to
  - Change passwords frequently
  - Use a certain structure (e.g., upper and lower case characters, special characters)
- What could be the problems?

Passwords: Is This a Good 2-Factor Authentication?

- The password can be changed by the user
- The PIN was sent in a letter



Log in

Please note your password is case sensitive.

Your Password

10th character from your Password

15th character from your Password

17th character from your Password

Your PIN

1st digit from your PIN

4th digit from your PIN

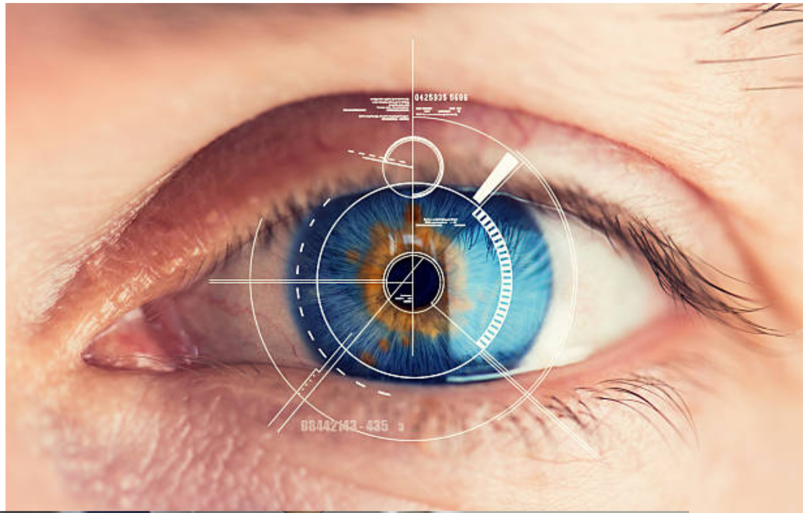
5th digit from your PIN

Example of Something That You Have: Hardware Tokens

- Examples something that you have
  - Chip cards
  - One-time password generators
  - Your CampusCard
  - Smartphone (working with apps, see below)
- We see a shift towards soft-tokens, e.g., a one-time password app on your mobile

Example of Something That You Are: Biometric

- Biometric
  - Uses characteristics of your body to authenticate the identity
    - Fingerprint
    - Retina scan
  - Very promising on the first sight
  - Downside: Check Hollywood movies :)
  - Many unsolved problems
    - Is fingerprint a secret protected by law?
    - Biometric sensors can be tricked



Access Control Models: Introduction

- Typical access control models focus on authorization
  - Specification of who is allowed to do what (permissions)
  - How to update/change permissions
- An example of a simple access control model is a relation

Subject X Object X Request

Access Control Models: Introduction

- Typical access control models focus on authorization
  - Specification of who is allowed to do what (permissions)
  - How to update/change permissions
- An example of a simple access control model is a relation

  Subject X Object X Request
- In reality, quite complex
  - Might depend on the system state (or context)
  - Subjects and permissions change over time
  - Access rights might require the fulfillment of obligations
  - Implementation bugs
  - Access control needs to be enforced

Forms of Access Control

- Access control might come in various forms
  - Physical protection
    - E.g., gates, turnstiles
  - Network traffic
    - E.g., firewalls
  - Hardware
    - E.g., memory management
  - Operating system
    - E.g., file system
  - Application level
    - E.g., Google login, databases

The Access Control Matrix Model
Introduction

- Based on the ideas of privileges of subjects on objects
  - *Subjects:* Users, processes, agents, groups, ...
  - *Objects:* Data, memory banks, other processes, files, ...
  - *Privileges:* Right to read, write, modify, ...
- Abstract
  - A model
- Implementation
  - A mechanism

The Access Control Matrix Model

Protection State

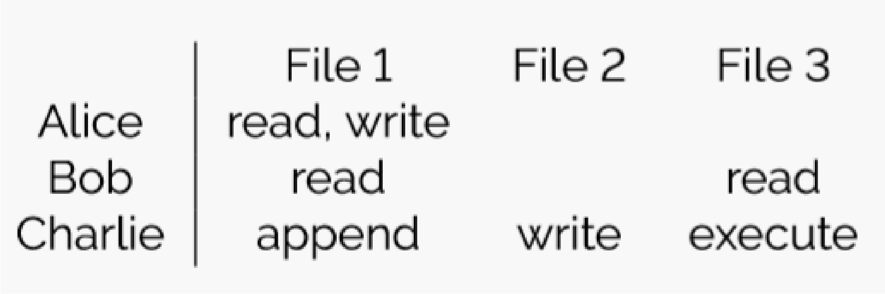
- A protection state (relative to a set of privileges *P* is a triple (*S*, *O*, *M*))
  - A set of current subjects *S*
  - A set of current objects *O*
  - A access control matrix *M*, defining
    - The privileges for each $(s, o) \in S \times O$, i.e.,
    - A relation $S \times O \times P$

The Access Control Matrix Model

Protection State

- A protection state (relative to a set of privileges *P* is a triple (*S*, *O*, *M*))
  - A set of current subjects *S*
  - A set of current objects *O*
  - A access control matrix *M*, defining
    - The privileges for each $(s, o) \in S \times O$, i.e.,
    - A relation $S \times O \times P$

- Example

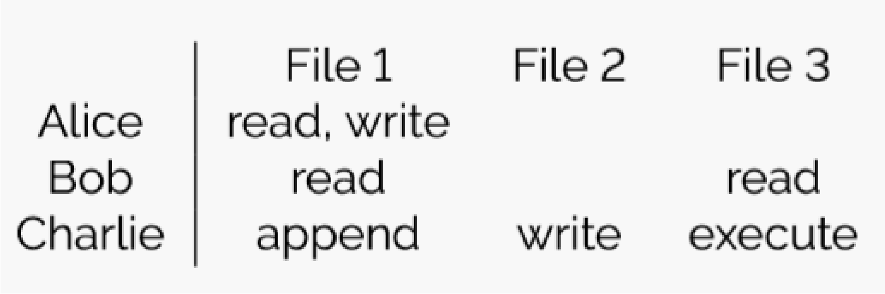


	File 1	File 2	File 3
Alice	read, write		
Bob	read		read
Charlie	append	write	execute

The Access Control Matrix Model
Protection State

- A protection state (relative to a set of privileges *P* is a triple (*S*, *O*, *M*))
  - A set of current subjects *S*
  - A set of current objects *O*
  - A access control matrix *M*, defining
    - The privileges for each $(s, o) \in S \times O$, i.e.,
    - A relation $S \times O \times P$

- Example



	File 1	File 2	File 3
Alice	read, write		
Bob	read		read
Charlie	append	write	execute

- Alice, Bob, Charlie are subjects
- File 1, File 2, File 3 are objects
- Matrix entries are set of privileges (rights)

- Does this scale? What about systems with thousands (millions) of subjects and objects?

Role-Based Access Control (RBAC)
Introduction

- How can we formalize a policy for more than
  - Thousands or millions of subjects
  - A similar number of objects
- Think of your bank as an example
- An access control matrix is most likely unmaintainable
- Observation
  - Subjects (users) often have roles
    - Customer, employee, student, etc.
  - Roles share the same rights
    - Students can attend lectures
- Core idea of RBAC
  - Create roles for job functions in enterprises
  - Assign users to roles (based on their responsibilities)
  - Assign a set of permissions to each role
- RBAC decouples users and permissions by introducing roles

Role-Based Access Control (RBAC)
Formalization

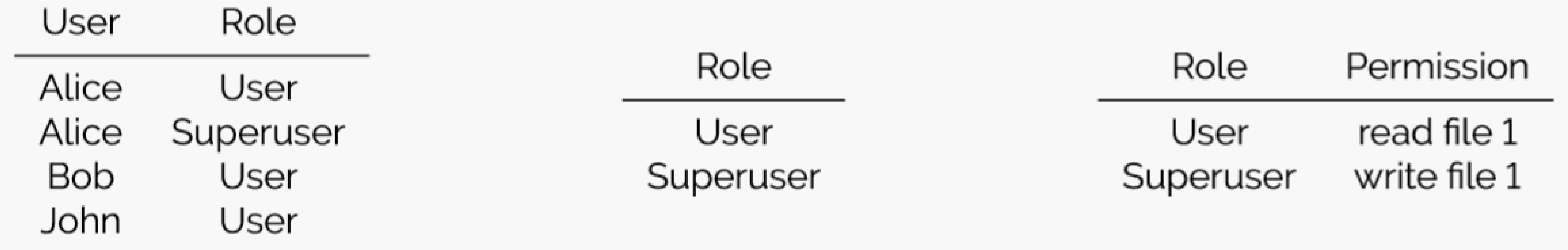
- RBAC is formalized by
  - A set *ROLES*
  - A set *USERS*
  - A relation $UA \subset USERS \times ROLES$
  - A relation $PA \subset ROLES \times PERMISSIONS$
- The access control model is

$$AC \coloneqq PA \circ UA$$

i.e.,

$$AC \coloneqq \{(u,p) \in USERS \times PERMISSIONS \mid \exists r \in ROLES\colon (u,r) \in UA \; \Lambda \; (r,p) \in PA\}$$

- Example



User	Role
Alice	User
Alice	Superuser
Bob	User
John	User

Role
User
Superuser

Role	Permission
User	read file 1
Superuser	write file 1

Beyond RBAC

- Most practical RBAC applications use extended/modified versions
- Widely used
    - XACML (a kind of attribute-based access control, very flexible)
- Other access control models
    - Discretionary access control (DAC)
        - Owners can chance permissions
        - Unix/Linux file system
    - Data classification: Instead of grouping subject, one can also group objects
        - Can be extended to information-flow models such as Bell-LaPadula
            - Hierarchy of data classifications
            - One can copy data from lower to higher classified documents
            - One can read only lower classified documents
        - How to re-classify information?

Next Generation Access Control

Usage Control

- Traditional access control focuses
  - Controlling access to documents/data/information
  - Decisions that are fast to evaluate/decide
  - Decisions that can immediately be enforced
- Today, we move in many areas towards Usage Control
  - Controlling the use of documents
    - You are allowed to read the book but not to give it to someone else
    - You are allowed to watch this movie three times within the next two weeks
  - You might encounter usage control in the form of DRM (Digital Rights Management)
    - The "media industry" likes DRM a lot
  - Techniques used for usage control/DRM
    - Watermarking (violations/misuse is pursued economically/legally)
    - Monitoring (easier in a closed/trusted environment, e.g., using a trusted OS and/or trusted viewer)

Bibliography

- Ross J. Anderson. Security Engineering: A Guide to Building Dependable Distributed Systems. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 2001.
  - The complete book is available at: http://www.cl.cam.ac.uk/~rja14/book.html
- Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. Handbook of Applied Cryptography. CRC Press, Inc., Boca Raton, FL, USA, 5th edition, 2001.
  - The complete book is available at: http://cacr.uwaterloo.ca/hac/
- D. Elliott Bell and Leonard J. LaPadula. Secure Computer Systems: A Mathematical Model, volume II. In Journal of Computer Security 4, pages 229–263, 1996. An Electronic Reconstruction of Secure Computer Systems: Mathematical Foundations, 1973.
- M Golla, M Wei, J Hainline, L Filipe, M Dürmuth. What was that site doing with my Facebook password? Designing Password-Reuse Notifications. Proceedings of the 2018 ACM SIGSAC Conference, 2018.
- Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D Ullman. Protection in Operating Systems. Communications of the ACM, 19(8):461–471, 1976.
- Konstantin Beznosov. Requirements for Access Control: US Healthcare Domain. In Proceedings of the 3rd ACM workshop on Role-based Access Control (RBAC), page 43, New York, NY USA, 1998. ACM Press.
- Achim D. Brucker and Helmut Petritsch. Extending Access Control Models with Break-glass. In Barbara Carminati and James Joshi, editors, ACM Symposium on Access Control Models and Technologies (SACMAT), pages 197–206. ACM Press, New York, NY USA, 2009.
- eXtensible Access Control Markup Language (XACML), version 2.0, 2005.
- Ravi S. Sandhu, Edward J. Coyne, Hal L. Feinstein, and Charles E. Youman. Role-based Access Control Models. Computer, 29(2):38–47, 1996.
- Ravi S. Sandhu, David F. Ferraiolo, and D. Richard Kuhn. The NIST Model for Role-based Access Control: Towards a Unified Standard. In ACM Workshop on Role-Based Access Control, pages 47–63, 2000.

Thanks for your attention!

- Any questions or remarks?