COMx501: Computer Security and Forensics

Achim D. Brucker
a.brucker@sheffield.ac.uk https://www.brucker.ch/

Software Assurance & Security Research
Department of Computer Science, The University of Sheffield, Sheffield, UK
https://logicalhacking.com/

March 7, 2018

COMx501: Computer Security and Forensics

Part 3: Authentication, Authorization, and Access Control

Achim D. Brucker

a.brucker@sheffield.ac.uk https://www.brucker.ch/

Software Assurance & Security Research
Department of Computer Science, The University of Sheffield, Sheffield, UK
https://logicalhacking.com/

February 6, 2018 (updated on March 7, 2018)

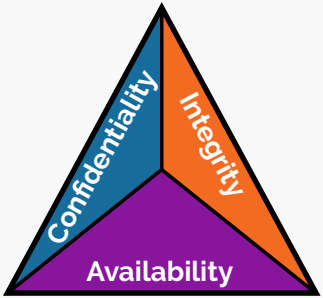
The Three Fundamental Concepts of Security: Discussion

Discuss with your neighbour:

1. What properties to you need to ensure the “security” of information
2. Prioritise your list of properties
3. What are your top three properties

The Three Fundamental Concepts of Security: CIA

- **Confidentiality**:
  Protecting information from disclosure to unauthorized parties.
- **Integrity**:
  Protecting information from being modified by unauthorized parties.
- **Availability**:
  Ensuring that information is available (accessible) to authorized parties.



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:

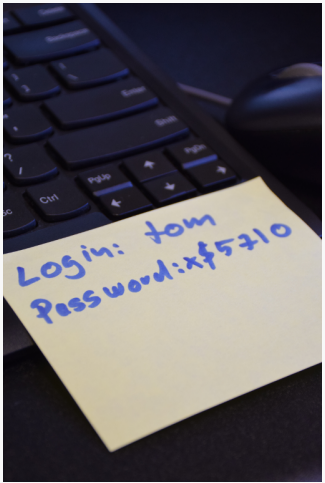
1. Something that you ~~forgot~~ know
   E.g., a password or a PIN
2. Something that you ~~lost~~ have
   E.g., a smart card or a one-time password generator
3. Something that you ~~were~~ are
   E.g, Biometric characteristics e.g., a facial scan/photograph
4. Context location, e.g., a place you ~~visited~~ 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)

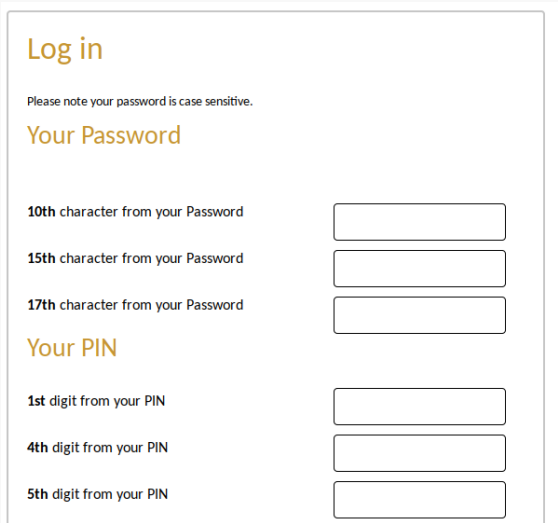
Example of Something That You Know: Passwords

- Passwords
  - are widely used
  - hard to remember
  - not always kept secret (social engineering):
    https://www.youtube.com/watch?v=opRMrEfAIiI
- Good passwords are: long and random
- Good systems:
  - allow for passwords of arbitrary length
  - store passwords hashed and salted
    (see following lectures for details)
- Not so clear, if enforcing users to
  - change passwords frequently
  - to use a certain structure
    (e.g., upper and lower case characters, special characters)

  really helps. What could be problems?



Passwords: Is This a Good 2-Factor Authentication?



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

Example of Something That You Have: Hardware Tokens

- Examples something that you have:
  - Chip cards
  - One-time password generators
  - Your UCard
- Today, we see a shift towards soft-tokens, e.g., a one-time password app on your mobile
- Is your UCard a good hardware token?

```
\documentclass{article}
\usepackage{pst-barcode}
\pagestyle{empty}
\begin{document}
  \begin{pspicture}(2in,.5in)
    \psbarcode{001xxxxxx}
      {width=2 height=.5}
      {code39}
  \end{pspicture}
\end{document}
```

Something that you are: Biometric

- Biometric:
  - Uses characteristics of your body, e.g..
    - fingerprint
    - retina scan

    to authenticate the identity
- On the first sight: very promising
- Clearly, the method of choice in Hollywood movies
- Many unsolved problems:
  - Is a fingerprint a secret protected by the first amendment, i.e., the protection of free speech
    (ongoing debate in the US, passwords are protected in the US)?
  - Biometric sensors can be tricked
    (and that might be good for your health)

Source: Spaceballs, 1987

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

$$Subject \times Object \times Request$$

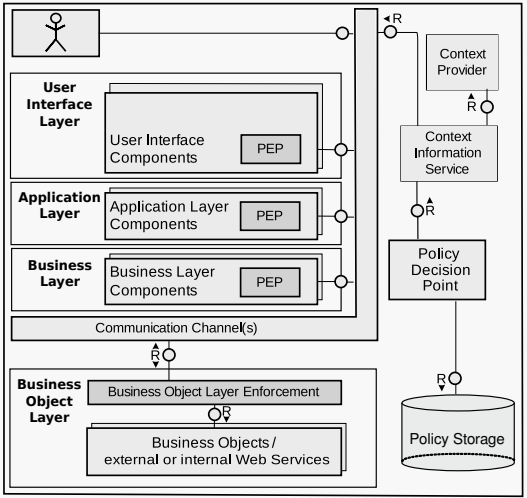
- In reality, quite complex
  - might depend on the system state (or context)
  - subjects and permissions change over time
  - access rights might require the fulfillment of obligations
  - implementation bugs
  - access control needs to be enforced

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

A Exemplary Infrastructure for Access Control Enforcement



- Policy Enforcement Point (PEP)
- Policy Decision Point (PDP)
- Authentication not shown

Policies vs. Models

- A **security policy** defines what is allowed (and/or forbidden).
  - It is analogous to a set of laws
  - Defined in terms of rules and/or requirements
- A **security model** is a (formal) representation of a class of systems (and their behavior)
  - highlights security features on a chosen level of abstraction
  - provides a vocabulary to develop specific policies

- 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

- 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 $S \times O \times P$
      (equivalently, a function $S \times O \to \mathcal{P}(P)$
- 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?

- 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, e.g.,
    - customer, employee, student
  - Roles share the same rights, e.g.,
    - 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

- RBAC is formalized by
  - a set *ROLES*
  - a set *USERS*
  - a relation $UA \subset USER \times ROLES$
  - a relation $PA \subset ROLES \times PERMISSION$
- The access control model is:

$$AC := PA \circ UA$$

i.e.,

$$AC := \{(u, p) \in Users \times Permissions \mid \exists r \in ROLES : (u, r) \in UA \wedge (r, p) \in PA\}$$

- 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

- Would recommend simple RBAC to your bank?
  - role hierarchies
  - who can change permissions
  - context information (constraints)
  - users switching roles
- 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
  - see the footer on this slide
  - can be extended to information-flow models a la Bell-LaPaduala
    - 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 (as discussed in this lecture) 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

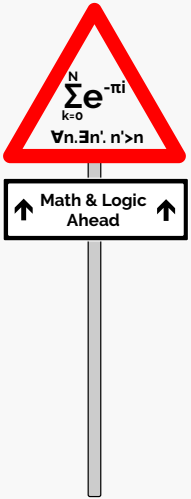
    You might encounter usage control in the form of DRM (Digital Rights Management)
    - The "media industry" likes DRM a lot
  - Techniques used for usage control/DRM:
    - watermarking (violations/misuse is pursued economically/legally)
    - monitoring (easier in a closed/trusted environment, e.g., using a trusted OS and/or trusted viewer)
  - Usage Control challenges and open questions:
    - Technical (examples):
      how to implement usage control efficiently
      how to implement usage control in an open environment
    - Ethical (examples):
      Richard M. Stallman. The Right to Read. Communication of the ACM. 1997.
      https://www.gnu.org/philosophy/right-to-read.html (highly recommended, takes only 5min to read!)

Next Lecture: An Introduction Into Cryptography

Up to now:

We built up the basic vocabulary – now comes the fun stuff!

- Problem sheet on MOLE (authentication & access control) (multi-factor authentication, RBAC, DAC, Bell-LaPadula)
- During the next three weeks, we will need (more) math!
  - fundamentals of cryptography
  - security protocols (how to use cryptography)
  - formal analysis of security protocols



Thank you for your attention!
Any questions or remarks?

Contact:

Dr. Achim D. Brucker
Department of Computer Science
University of Sheffield
Regent Court
211 Portobello St.
Sheffield S1 4DP, UK

a.brucker@sheffield.ac.uk
@adbrucker
https://de.linkedin.com/in/adbrucker/
https://www.brucker.ch/
https://logicalhacking.com/blog/

Bibliography I

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.

Konstantin Beznosov.

Requirements for access control: us healthcare domain.

In *Proceedings of the the third ACM workshop on Role-based access control (RBAC)*, page 43, New York, NY USA, 1998. ACM Press.

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.

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, 2009.

Bibliography II


Michael A. Harrison, Walter L. Ruzzo, and Jeffrey D Ullman.
Protection in operating systems.
Communications of the ACM, 19(8):461–471, 1976.


Roger M. Needham and Michael D. Schroeder.
Using encryption for authentication in large networks of computers.
Commun. ACM, 21:993–999, December 1978.


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.

Document Classification and License Information