Module 8: Access Control and Authentication

Disclaimer: large parts from Stefan Katzenbeisser, Günter Schäfer

Dresden, WS 14/15
Security goals and services describe and implement protection from threats

History has been an arms race between cryptography and cryptanalysis

Each success for the cryptanalysis community has helped make ciphers more secure

Different flavors of ciphers with different properties aim at confidentiality

Secure pseudo-random numbers are essential for the security of ciphers

MACs and signatures aim at providing integrity

Keys can be agreed upon apriori, exchanged, or agreed upon online

Stream- and block ciphers are commonly symmetric and have different properties

Asymmetric crypto allows for public keys and has many different applications
Recall security goals of confidentiality and integrity

So far, using crypto:

• Conceal information in seemingly random noise
• Prove absence of tampering by signature

• How does this solution relate to „real life“?
Confidentiality and Integrity „in RL“
Physical access control

Objects vs. Subjects

Subjects have controlled access to objects
- Prevents information disclosure
- Prevents tampering

Requires some gatekeeper:
- Identification of subjects (Authentication)
- Explicit instructions (Policy, <policy descriptions>, authorization)
- Controlling (and granting) access
Def: **Access control** comprises mechanisms to enforce mediation on subject requests for access to objects as defined in a security policy.

Def: A **subject** is an active entity that can initiate a request for resources and utilize these resources to complete some task.

Def: An **object** is a resource that is used to store, access, or process information.

Def: An **operation** (action) is an instance of access, commonly a utilization, retrieval, or manipulation event, of a subject on an object.

*Objects* historically had the notion of files, or repositories

*Subjects* commonly processes (local or remote)

*Operations* historically: “r,w,x”
“Reference monitor“ is a concept to detail decision process:

RM not necessarily a physical/logical component in the system

AC/RM may be implemented on different levels:
- **Online application**: control access to functions/data
- **Databases**: control access to tables, columns
- **OS**: control access to resources (files, devices)
A **security objective** is a *statement of intent* to counter a given threat or enforce a given organisational **security policy**.

A **security level** is defined as a *hierarchical attribute* with entities of a system to denote their degree of sensitivity

- Examples:
  - Military: unclassified < confidential < secret < top secret
  - Commercial: public < sensitive < proprietary < restricted

A **security category** is defined as a nonhierarchical *grouping of entities* to help denote their degree of sensitivity

- Example (commercial): department A, department B, administration, etc.

--> Security categories facilitate the “Need-to-know” principle
A *security label* is defined as an *attribute* that is *associated with system entities* to denote their hierarchical sensitivity level *and* security categories.

Security labels that denote the security sensitivity of:

- Subjects are called *clearances*
- Objects are called *classifications*

The *security policy* of a system defines the *conditions* under which *subject accesses to objects* are mediated by the system reference monitor functionality:

- To be derived from the organizational policy (IPRs, procedures)
- Compliance to be monitored (on introduction, regularly)
Classes of Security Models

Access control models

- Identity-based access control (IBAC)
- Role-based access control (RBAC)
- Attribute-based access control (ABAC)

Information flow models (e.g. Chinese Wall model)

- Multilevel security models (e.g. Bell-La Padula model)

Non-interference models

General types of access control:

- Discretionary
- Mandatory
Discretionary Access Control

- Owner is responsible for security of her objects
- Authorization per object
- No system-wide security properties
- Rights commonly to be granted: read, write, execute (*NIX, win)
- -> commonly challenged by lack of competence, overview

Mandatory Access Control

- System-wide (usually: rule-based) security policy configuration
- User may change authorization, but system policy dominates
- -> commonly challenged by lack of overview, BOfH

26.01.2015
**IBAC: Access Control Matrix**

**Task:** Configuration of authorizations (rights of **subjects** on **objects**)

Define: Set of objects $O$, set of subjects $S$, set of rights $R$ (e.g. rwx...)

Access Control Matrix defines mapping $M : S \times O \rightarrow 2^R$ (e.g.: \{true,false\})

**Advantages of ACM:**
- Intuitive, flexible
- Easy to implement

**Disadvantages of ACM:**
- Huge, sparse
- Static

<table>
<thead>
<tr>
<th></th>
<th>o1</th>
<th>o2</th>
<th>o3</th>
<th>o4</th>
<th>o5</th>
</tr>
</thead>
<tbody>
<tr>
<td>s1</td>
<td>{ read, write }</td>
<td></td>
<td>{ read, write }</td>
<td></td>
<td>{ send, receive }</td>
</tr>
<tr>
<td>s2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>s3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

26.01.2015
Access Control Description Schemes

Access Control Lists (ACLs)

- Columns of the ACM: list of authorizations on an object
  - ACL(o1) = {(s1,\{r,w\}), (s2,\{r\})};  ACL(o2) = {(s3,\{r,w,x\})};  ...
  (*NIX: subjects only identified as owner, group, others)

- Assessing authorizations to an object is simple
- Assessing authorizations granted to a subject is difficult

Capabilities

- Rows of ACM: list of objects and rights granted to a subject
  - CL(s3) = {(o2,\{o,x\}), (o4,\{s,r\})};  ...
- Advantages/disadvantages inverse to ACLs...
From IBAC to RBAC

Complexity of IBAC yields problems of overview and adaptation
Subjects usually act in „roles“ (specificly in organizations)
Introduce indirection of the role abstraction:

Dr. Brains
Dr. Bones
Nurse Kathy
Carer Tuck

physician

↔

nurse

User↔Role - Relation  Role↔ Right - Relation

Read patient information
Write diagnosis
Read prescriptions
Write blood values
Role-based Access Control

Extend IBAC:
- Set of subjects $S$
- Set of roles $R$
- Set of objects $O$
- Set of permissions $P$

Define mappings $sr: S \rightarrow 2^R$; $pr: R \rightarrow 2^P$

*Sessions* are dynamic role assignments (a subject is active in a role)
Subject is assigned permissions from role for the session accordingly

*Role hierarchies and constraints extend RBAC*