OPERATING SYSTEMS SECURITY

- some basics

LAYERS OF A COMPUTER SYSTEM

Applications

Services

Operating system (OS)

OS kernel

Hardware

- Where should the security of the system be placed?
- The security of a layer could normally be compromised by attacks from lower layers!

The basis of OS protection is **separation**. The separation can be of four different kinds:

• physical

(physical objects, such as CPU's, printers, etc)

• temporal

(execution at different times)

logical

(domains, each user gets the impression the she is "alone" in the system)

cryptographic

(hiding data, so that other users can not understand them)

"Computing is sharing and non-location -

- security is separation"

In principle all objects in the OS need protection, but in particular those that are shareble, e.g.:

- memory
- I/O devices (disks, printers, tape drives, etc)
- programs, procedures
- data
- hardware, such as
 - normal operating system mechanisms (e.g. file management - logical, memory management - physical)
 - bus control
 - interrupt control
 - status registers

There are a few basic concepts that are fundamental when dealing with trusted OS:

• the **kernel**

is the part of the OS that performs the lowest-level functions

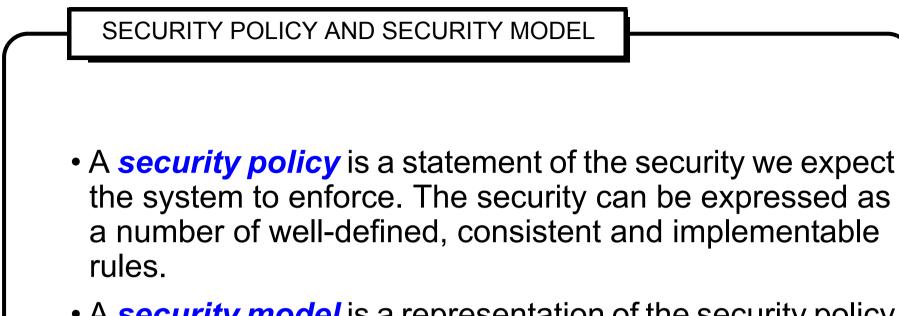
• the security kernel

is responsible for enforcing the security mechanisms of the entire OS

the reference monitor (RM)

is the part of the security kernel that controls access to objects

the trusted computing base (TCB) is everything in the trusted OS necessary to enforce the security policy



- A security model is a representation of the security policy for the OS.
- A *formal security model* is a mathematical description (formalisation) of the rules of the security policy. It could be used for formal proofs of security.

The development of secure OS can be made in six steps:

- analyze of the system
- choose/define a security policy
- choose/create a security model (based on the policy)
- choose implementation method
- make a (conceptual) design
- verify the correctness of the design
- make an *implementation*
- verify the implementation (?)

There are feed-back loops between all of the above steps Errors may occur in all above steps

