The goal of this lecture is to:

- answer the question: "What is SECURITY?"
- present a conceptual modelling of dependability and security, which should entail a new terminology or changed interpretation of the terminology. Thus, dependability and security represent different aspects of a common meta-concept.
- clarify security is multi-faceted and can not be treated as a clear-cut atomic concept.
- based on the conceptual model, suggest a structured way to measure security/dependability
• RELIABILITY ("continuity of service")
  The reliability $R(t)$ of a system $SYS$ can be expressed as:
  $R(t) = \text{Prob (SYS is fully functioning in } [0, t])$
  A metric for reliability $R(t)$ is MTTF, the Mean Time To Failure
  $MTTF = \int_{0}^{\infty} R(t) \, dt = \frac{1}{\lambda}$, where $\lambda$ is the constant failure rate. MTTF is normally expressed in hours.

• AVAILABILITY ("readiness for usage" - incorporates maintainability (repair))
  The availability $A(t)$ of a system $SYS$ can be expressed as:
  $A(t) = \text{Prob (SYS is fully functioning at time } t)$
  A metric for the average, steady-state availability is
  $A(\infty) = \frac{MTTF}{MTTF + MTTR}$, where
  $MTTR = \frac{1}{\mu}$, where $\mu$ is the constant repair rate. $A(\infty)$ is normally expressed in %.

• SAFETY ("avoidance of catastrophic consequences on the environment")
  The Safety $S(t)$ of a system $SYS$ can be expressed as:
  $S(t) = \text{Prob (SYS is fully functioning or has failed in a manner that does cause no harm in } [0, t])$
  A metric for safety $S(t)$ is MTTCF, the Mean Time To Critical Failure, defined similarly to MTTF and normally expressed in hours.

• SECURITY ("prevention of unauthorized access and/or handling")
  A system is considered Secure if it is can protect itself against intrusions
  There is no mathematical or formal definition of the Security of a system.
  There are no real metrics for security. Instead, there are a number of informal and/or subjective assessments or rankings.
  Security is normally defined by its three aspects: confidentiality, integrity and availability (the "CIA")

AN INTEGRATED SYSTEM MODEL
A BIOLOGICAL ANALOGY

• THREATS:
  Threats are there all the time.
  Threats change and evolve.

• PROTECTION MECHANISMS:
  Protection takes place at different levels.
  Protection mechanisms are active continuously.
  Protection mechanisms must also change and evolve
  according to the threats.
  Even anticipatory protection exists. (inoculation)

**Hypothesis:**
Modern IT systems are so complicated so that a biological
paradigm must be adapted. Thus, security protection must
be a continuous process, taking place simultaneously on all
protection levels. Security protection must be adaptive.

SOME OBSERVATIONS FROM THE BIOLOGICAL ANALOGY

SECURITY and DEPENDABILITY METRICS

Quotations:

• “Modelling is fundamental to measurement;
  without an empirical model or describing
  observations, measurement is not possible”
  (A. Kaposi 1991)

• “The history of science has been, in good part,
  the story of quantification of initially qualitative
  concepts” (Bunge 1967)

WHY MODELLING? - WHY METRICS?

METHODS FOR “MEASUREMENT” OF SECURITY

• Risk analysis:
  - estimation of the probability for specific intrusions
    and their consequences and costs. Trade-off towards
    the corresponding costs for protection.

• Evaluation/Certification:
  - classification of the system in classes based on
    design characteristics and security mechanisms.
    “The ‘better’ the design is, the more secure the
    system”

• Operational Metrics (based on the intrusion process):
  - a statistical metric of system security based on
    the effort it takes to make an intrusion.
    “The harder to make an intrusion, the more secure
    the system”
Basic methodology for risk analysis:

1. Identify **assets**
2. Determine **vulnerabilities**
3. Estimate **likelihood of exploitation**
4. Compute expected annual "loss" (due to intrusions)
5. Survey applicable **methods of protection** and their costs
6. Project annual savings (make **trade-off**)
The Common Criteria (CC) is aimed to be common to all countries. It defines a security evaluation methodology. It became the “official” evaluation standard in the USA in 1998. (TCSEC was discontinued in 2000.)

Central terms:

- **Target of Evaluation (TOE):** An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.
- **Evaluation Assurance Level (EAL):** A package consisting of assurance components that represent a point in the predefined assurance scale.

The Common Criteria comes in three (plus 1) parts:

1. **Introduction and general model** (79 pages)
   - general concepts, principles and evaluation model
2. **Security functional requirements** (127 pages)
   - describe the desired security behaviour expected of a Target of Evaluation (TOE) in order to meet the security objectives as stated in a Protection Profile (PP) or a Security Target (ST)
3. **Security assurance requirements** (242 pages):
   - defines a scale for measuring assurance - Evaluation Assurance Levels (EALs)
   - defines criteria for evaluation of Protection Profiles (PPs) and Security Targets (STs)

There is also a companion document to the Common Criteria:

4. **Common Methodology** for Information Technology Security Evaluation (CEM) (466 pages):
   - describes the minimum actions to be performed by an evaluator in order to conduct a CC evaluation.

CC URL: http://www.commoncriteriaportal.org/

There are three types of evaluation:

1. **PP evaluation**
   - is carried out against evaluation criteria for PPs
   - is to demonstrate that the PP is suitable as a statement of requirements for an evaluable TOE
2. **ST evaluation**
   - is to demonstrate that the ST properly meets the requirements of the PP
3. **TOE evaluation**
   - is to demonstrate that the TOE meets the requirements contained in the ST

The CC defines three types of requirements constructs:

- **package,** Protection Profile and Security Target
- **a component**
  - describes a specific set of security requirements
  - is the smallest selectable set of security requirements
- **a package**
  - an intermediate combination of components is termed a package
  - gives a set of functional or assurance requirements that meet a subset of security objectives
  - EALs are predefined assurance packages
There are seven predefined levels of assurance (EAL levels):

- **EAL1.** Functionally tested
- **EAL2.** Structurally tested
- **EAL3.** Methodically tested and checked
- **EAL4.** Methodically designed, tested and reviewed
- **EAL5.** Semiformally designed and tested
- **EAL6.** Semiformally verified design and tested
- **EAL7.** Formally verified design and tested

An evaluation may also be carried out against a user-defined level of assurance.

**COMMON CRITERIA**

**BEHAVIOURAL METRIC**

Behavioal metrics are well-known (except for confidentiality):

- A **behavioural metric** describes to what extent the system delivers its service to its User(s) or denies service to its Non-user(s).

Thus, **reliability, safety and confidentiality** could be covered by the same (vectorized) **metric** using Markov modelling.


**PROTECTIVE METRICS**

Protective metrics could be based on studies of the system in operation, e.g. the intrusion process:

- A **protective metric** describes the ability of a system to resist attacks during operation, i.e., to prevent faults from entering into the system, thus creating an error in the system.
- A deliberate intrusion or a security breach could normally be regarded as a fault.

**Hypothesis:**

A system is more secure the more effort it takes to make an intrusion into the system.


**METRICS OF CORRECTNESS**

Metrics of correctness:

- Measuring correctness is very hard
- Not only are there huge practical problems, but it is also a matter of lack of fundamental definitions

Thus, I know of no methods for measuring correctness.


**THE TIME ASPECT**

The time aspect is very often neglected in security analysis. It must be noted that:

- introduction of a fault into the system does not mean that the system fails immediately. It may never fail due to this fault. **The latency aspect - fault propagation.**
- the latency clearly affects metrics of system behaviour. There might be a substantial time between the original fault occurrence and the resulting (deficient) system behaviour.
- faults can be introduced into a system throughout its lifetime. Many faults are introduced during the design phase.
- Some security mechanisms does not protect the system as it stands. But it will give information for improving...
**THE TIME ASPECT - DEBUGGING (A software analogy)**

"the law of diminishing results" (regarding debugging of software): it will be increasingly hard to find the remaining faults.

**THE TIME ASPECT - LATENCY (Another software analogy)**

- A program can have many errors with very long MTTF.
- An investigation of an IBM-program showed that more than 30% of the errors had an MTTF > 5000 years! This means that if we test the system continuously, after 5000 years some 30% of the errors remain latent! (Ref: E. N. Adams: “Optimizing preventive service of software products”, IBM Journal of Research and Development, vol. 28, No. 1, pp. 2-14, 1984.)

- The same problem applies to security vulnerabilities.

---

**CONCLUSIONS (general):**

- The areas of Dependability and Security have traditionally evolved separately and there is a lack of coordination between them regarding concepts, terms, tools etc.
- Dependability and Security reflect two different approaches to the same fundamental research area.
- Dependability and Security must be integrated into one common context in order for us to be able to properly address the problems involved.

**CONCLUSIONS (specific):**

- We have suggested an **integrated system model** for Dependability and Security, describing the system in terms of **correctness** as well as **protective and behavioural characteristics**.
- Dependability and Security metrics can be defined in accordance.
- Protection methods and mechanisms have been related to the system model.
- Intrusion detection is a mechanism that introduces the "product-in-a-process" concept for the system.