Security and Dependability Modelling

Presented by

Erland Jonsson
Department of Computer Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY

OUTLINE OF LECTURE.

- Dependability and its attributes
- Security (and its aspects)
- An integrated system model
- A biological analogy
- The time aspect
- A few observations
- Conclusions

GOAL OF LECTURE.

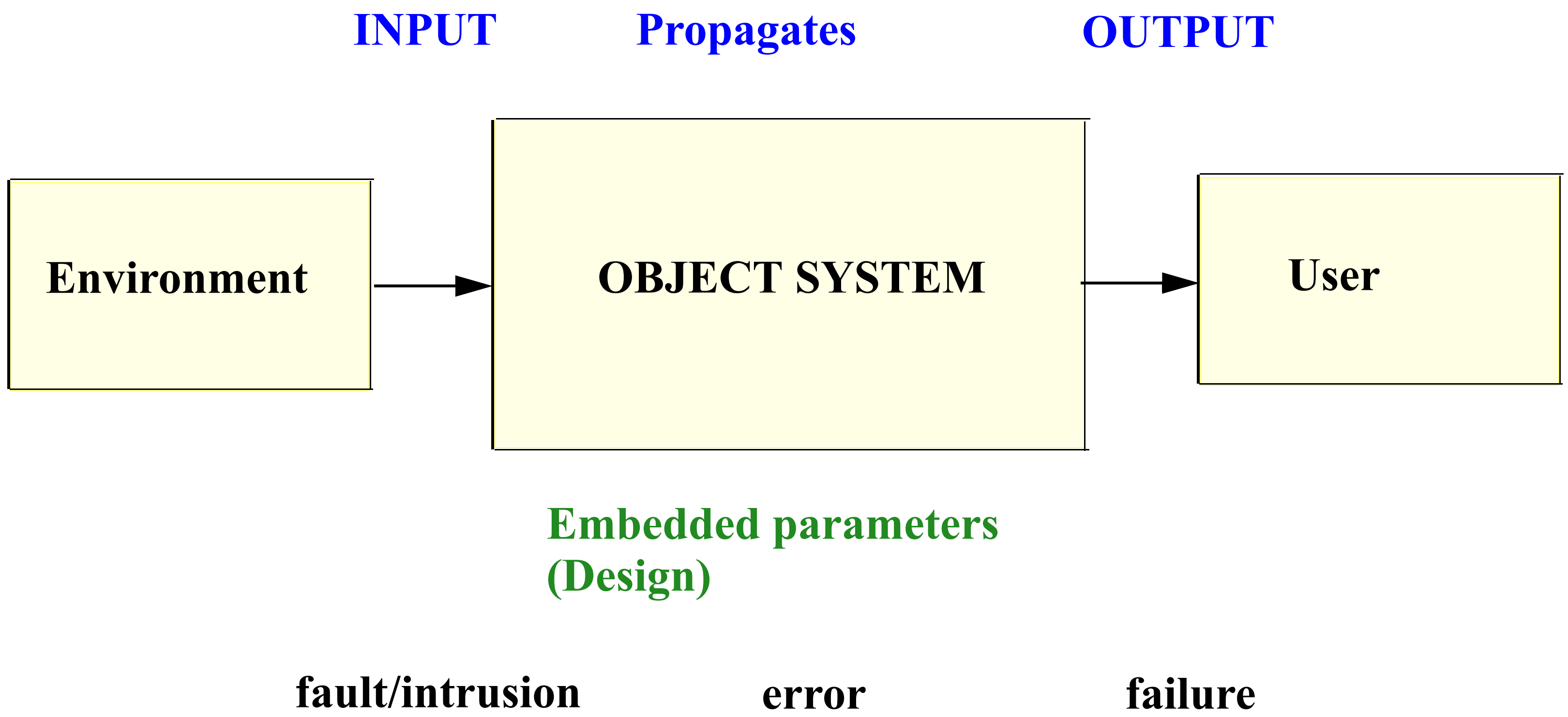
The goal of this lecture is to:

- answer the question: "What ***is*** SECURITY?"
- present a conceptual model of dependability and security, including a suggested terminology. Thus, dependability and security represent different aspects of a common meta-concept.
- clarify that security is multi-faceted and can not be treated as a clear-cut atomic concept.
- the conceptual model is aimed to facilitate metrication of security/dependability

All in all: **to give a profound understanding of the security/dependability area.**

CHALMERS
Department of ComputerScience
and Engineering

A VERY BASIC SYSTEM MODEL



DEPENDABILITY
and its attributes

Relation to the Dependability area:

DEPENDABILITY ATTRIBUTES

- Reliability
- **Availability**
- Safety
- Maintainability
- **Confidentiality**
- **Integrity**

"CIA" = SECURITY

DEPENDABILITY

- is a general, “umbrella” concept
- is not mathematically well-defined
- denotes the research area:
  Dependable Computing

- RELIABILITY *("continuity of service")*

The reliability R(t) of a system SYS can be expressed as:

R(t) = 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 λ 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) = Prob (SYS is fully functioning at time t)

A metric for the average, steady-state availability is $A(\infty) = \dfrac{MTTF}{MTTF + MTTR}$, where

MTTR = $\dfrac{1}{\mu}$, where μ 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) = 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"*)

ITSEC:

INFORMATION SECURITY

Confidentiality Sekretess	Integrity Integritet	Availability Tillgänglighet
prevention of the unauthorized dis-closure of information	prevention of the unauthorized modifi-cation of information	prevention of the unauthorized withholding of information or resources

AN INTEGRATED SYSTEM MODEL

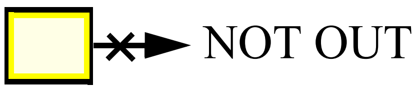
BEHAVIOURAL DEPENDABILITY ATTRIBUTES

- Reliability
- **Availability**
- Safety
- Maintainability → *“other” characteristic*
- **Confidentiality**
- **Integrity** → *protective characteristic*

Information security
Datasäkerhet

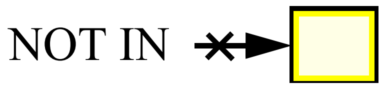
Confidentiality
Sekretess

prevention of the ***unauthorized*** disclosure of information



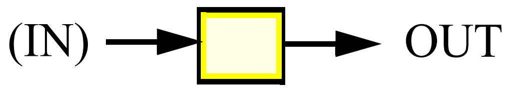
Integrity
Integritet

prevention of the ***unauthorized*** modification of information

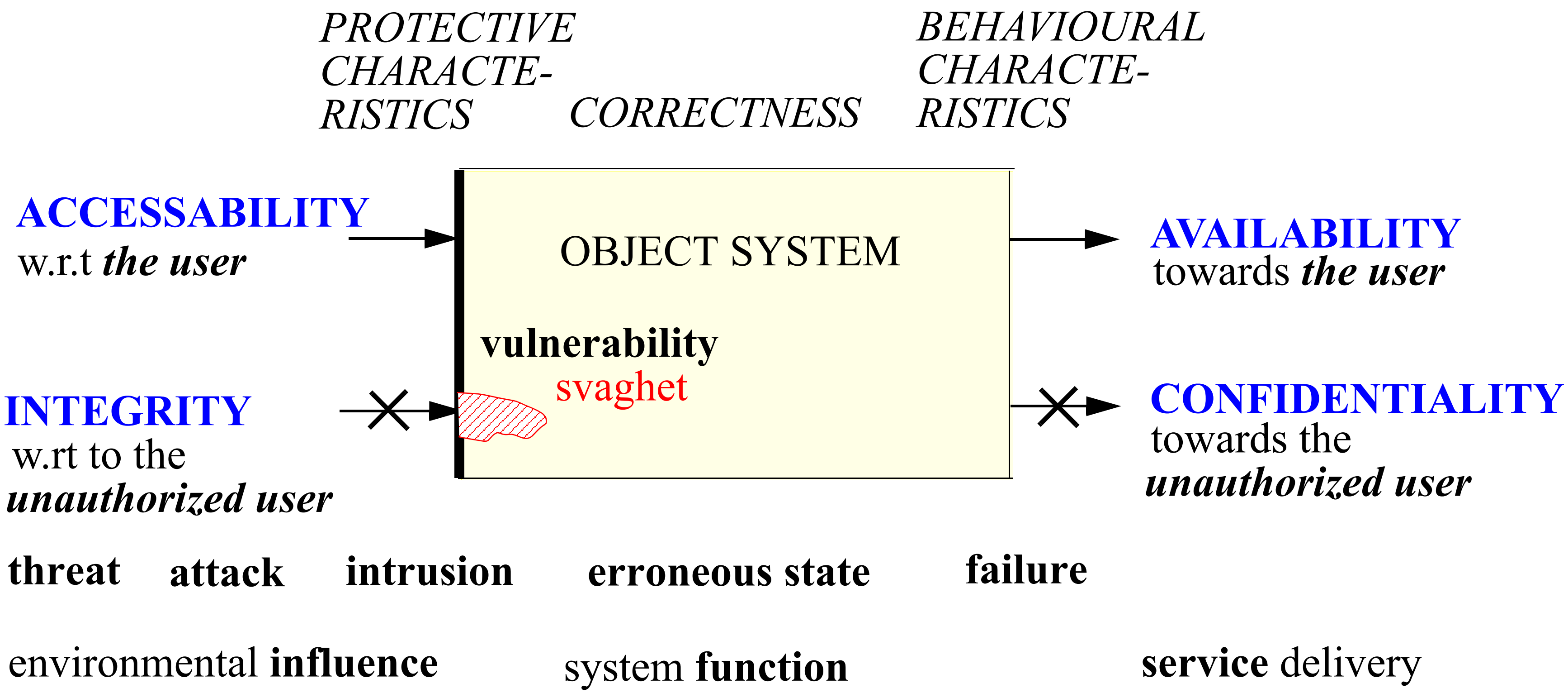


Availability ("CIA")
Tillgänglighet

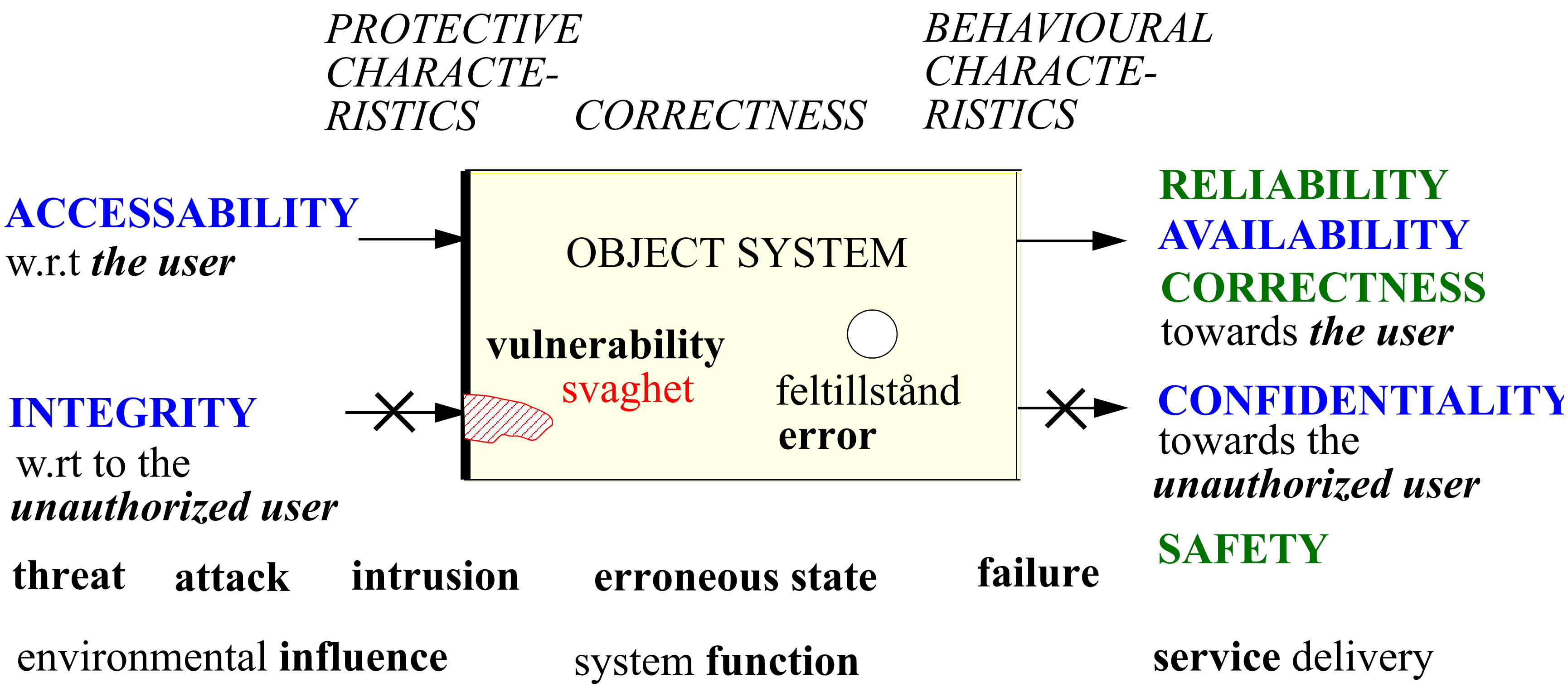
information must be available to the ***authorized*** user



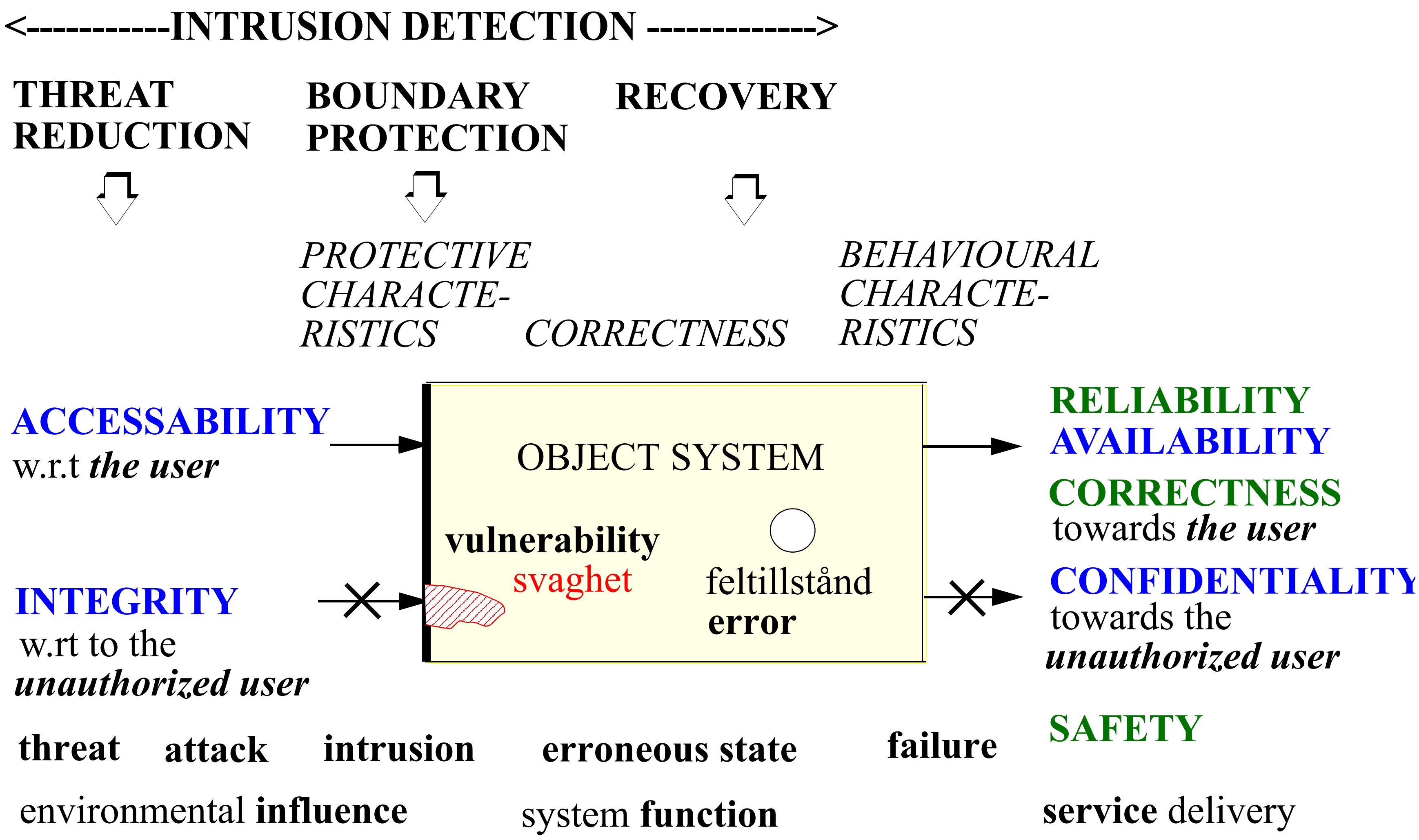
SECURITY ASPECTS vs OBJECT SYSTEM



DEPENDABILITY ATTRIBUTES vs OBJECT SYSTEM



A FUNDAMENTAL SYSTEM MODEL FOR DEPENDABILITY/SECURITY

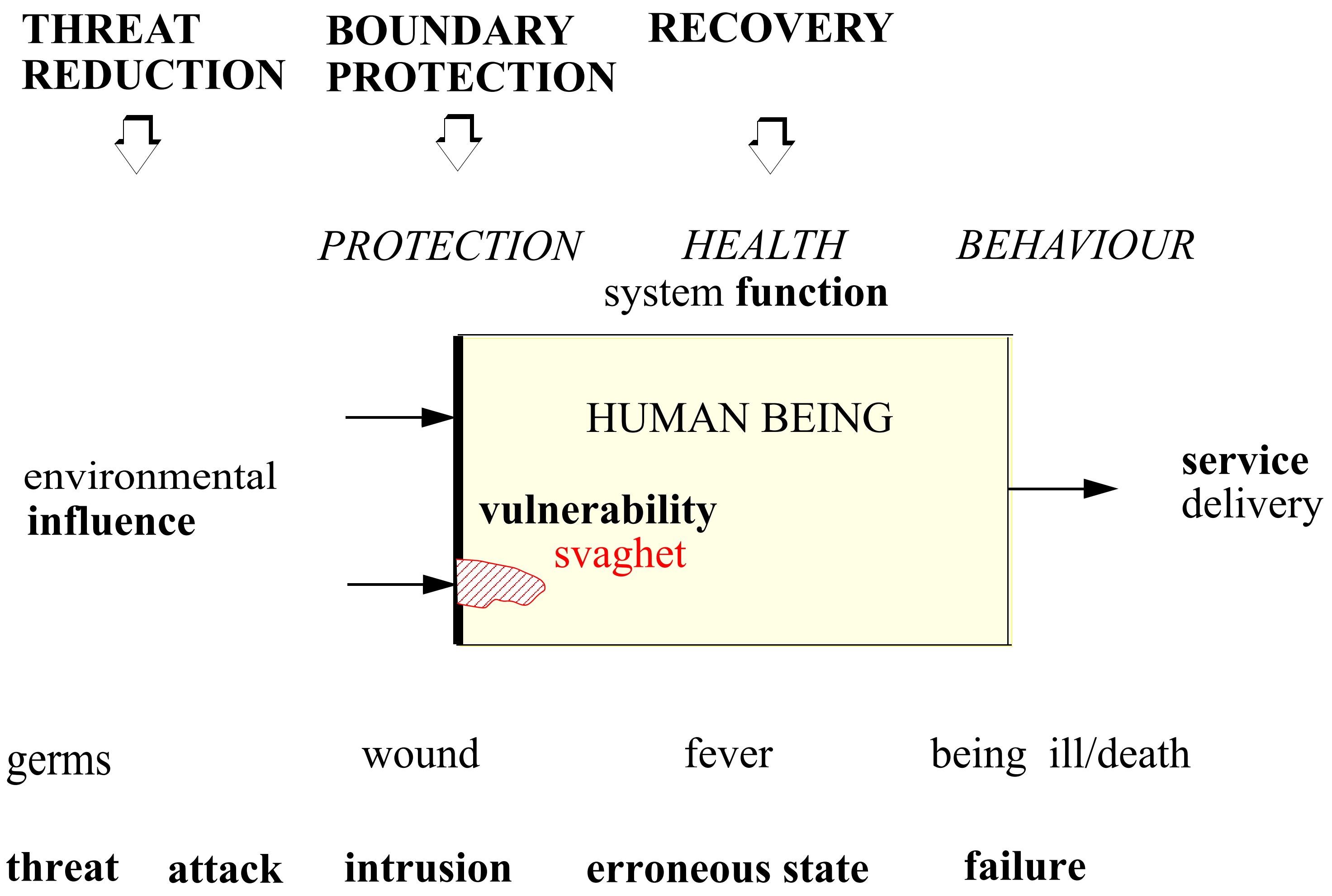


EXEMPLES of PROTECTION MECHANISMS - IN PRINCIPLE

- ***preventive protection - threat reduction:***
  - legal protection
  - reducing threats (e.g. "security check-ups")
  - **education / information / propaganda!**
- ***boundary protection:***
  - shield cables
  - encryption
  - physical protection (e.g. locks)
  - access control
- ***internal protection - recovery:***
  - (anti-)virusprograms
  - supervision mechanisms (with recovery capabilities)
  - encryption of stored data

A BIOLOGICAL ANALOGY

AN ANALOGY TO HUMAN BEINGS



- **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)

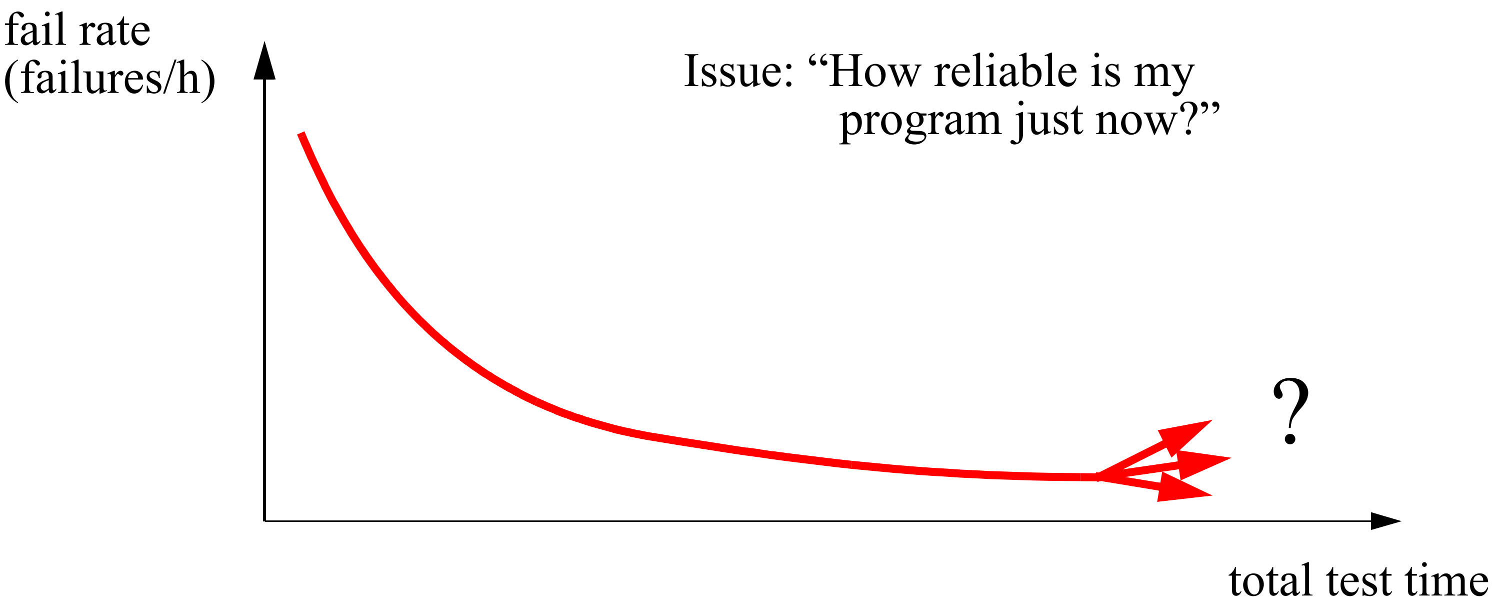
\+ **Hypothesis:**
Modern IT systems are so complicated so that a biological paradigm must be adapted. Thus, security protection must be a continouos process, taking place simultaneously on all protection levels. Security protection must be adaptive.

THE TIME ASPECT

THE TIME ASPECT - SOME OBSERVATIONS

- 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 do not protect the system as it stands. But it will give information for improving subsequent generations. (E.g. intrusion detection)

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



"regression testing"

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*

SUMMING UP

A FEW OBSERVATIONS

- Make a distinction between non-functional and functional attributes
- The desirable behaviour of a system depends on the intended user
- System latency affects system behaviour (e.g. reliability, availability, etc)
- Introduction of faults, errors and vulnerabilities can be throughout the system´s lifecycle
- a security problem is not the same as a reliability problem but they are related (in a complicated way)

CONCLUSIONS (general)

- The areas of Dependability and Security have traditionally evolved separately and there is still 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 could be defined in accordance
- Protection methods and mechanisms have been related to the system model
- Intrusion detection is a mechanisms that introduces the "product-in-a-process" concept for the system

EXTENSIONS/COMPLICATIONS

Why is this just part of the truth?

There are a number of issues that are not addressed and extensions to be made to make things more realistic:

- add feedback
- non-binary output (degraded performance)
- non-binary input ("gradual attack")
- multiple causes for an attack

- cascading of systems
- hierarchical systems ("systems-of-systems")