

# Software Security

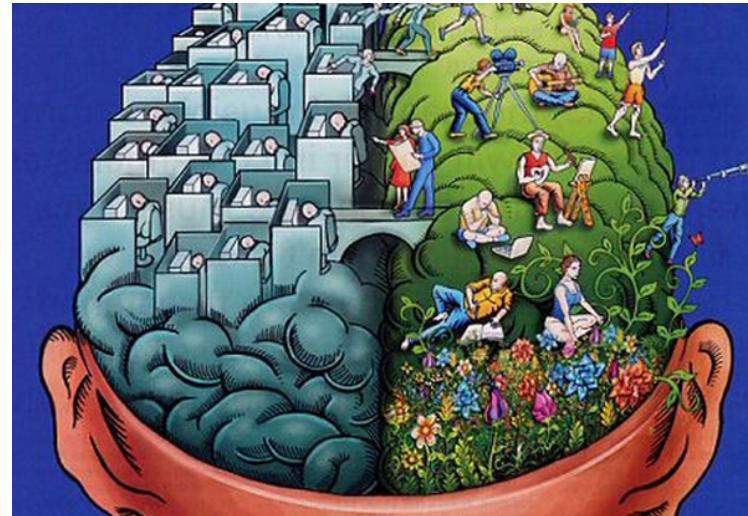
Steffen Helke

Chair of Software Engineering

15th October 2018



Brandenburgische  
Technische Universität  
Cottbus - Senftenberg



*Out of the crooked timber of humanity, no straight thing was ever made.*

Note: Ross Anderson's book *Security Engineering* opens with this quotation written by a German philosopher.  
Original source: Immanuel Kant's Essay, *Idea for a General History with a Cosmopolitan Purpose*, 1784.

## The Importance of Security

→ Almost every day, we see reports in the news about hacker attacks



UK hospitals hit with massive  
ransomware attack

Sixteen hospitals shut down as a result of the attack  
by Russell Brandom | May 12, 2017, 11:36am EDT

[SHARE](#) [TWEET](#) [LINKEDIN](#)



A massive ransomware attack has shut down work at 16 hospitals across the United Kingdom. According to *The Guardian*, the attack began at roughly 12:30PM local time, freezing systems and encrypting files. When employees tried to access the computers, they were presented with a demand for \$300 in bitcoin, a classic ransomware tactic.



Germans Don't WannaCry

The global ransomware virus WannaCry that made headlines over the weekend left Germany relatively unscathed. Do Germans know something the rest of the world does not?



The global computer virus WannaCry – nicknamed WannaCry – shut down more than 230,000 computers in 150 countries over the weekend. But in Germany, at least so far, it has been a nuisance rather than a catastrophe. At train stations, some arrival and departure screens have malfunctioned, a few ticket machines stopped working, and so forth. Deutsche Bahn, Germany's state-owned railway operator, said that actual traffic was not affected. No sweat.

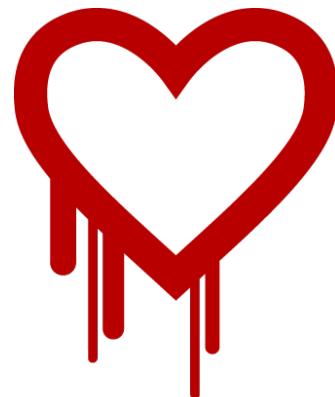
## WannaCry Ransomware



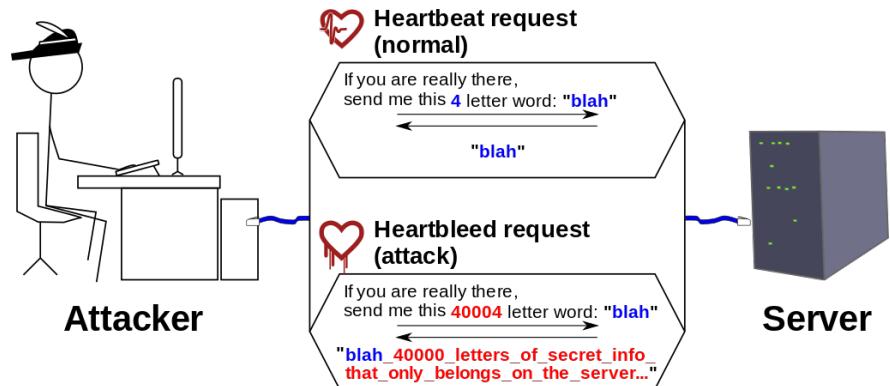
**Don't pay, it's better to submit a complaint to the police.**

## Exploit: OpenSSL Implementation of 2012

### Heartbleed Bug



- Heartbeat functionality for encryption protocols, e.g. SSL/TLS
- Poor input validation allows an unauthorized information flow

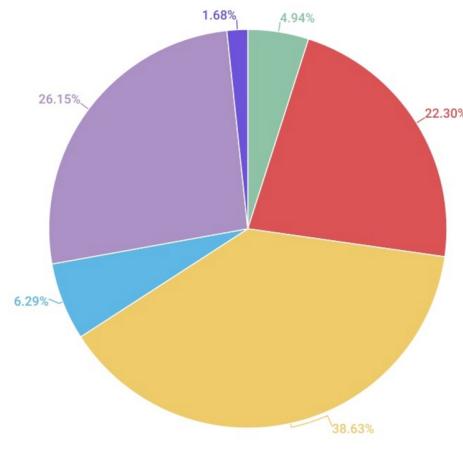


Gives the attacker the possibility to access confidential data

Source: <http://de.wikipedia.org/wiki/Heartbleed> [Accessed May 2018]

### Vulnerable apps exploited by cybercriminals

→ current statistic, the second quarter of 2017 (Q2 2017)



Adobe Flash    Android    Browser    Java    Office    PDF

Source: Kaspersky Lab, <https://securelist.com/it-threat-evolution-q2-2017-statistics/79432/> [Accessed September 2017]

### Objectives of today's lecture

- Understanding the *rules and conditions* of this course
- Getting familiar with the *content overview* and some *basic terms*
- Being able to name important *standard literature*

- Understanding the weaknesses of the *trojan horse of the german government*



- Being able to judge the *personal suitability* for this course

# Who is giving this lecture?

---

## ■ Contact Person

Steffen Helke, Assistant Professor, Software Engineering  
Development of Safe and Secure Software Systems

## ■ Email Address

steffen.helke@b-tu.de

## ■ Consultation Hour

Wednesday, 1pm-2pm

## ■ Room

VG1c 2.13

## ■ Secretary's Office

VG1c 2.20

# Attributes of this course

---

- Master module *Software Security*
- Study program Cyber Security M.Sc.: compulsory elective module (complex “Cyber Security Methods”)
- also suitable for the following study programmes: Computer Science M.Sc., Information and Media Technology M.Sc., eBusiness M.Sc. and Applied Mathematics M.Sc.

## Forms of Teaching and Proportion

- Lecture (2 Hours per Week) + Exercise (2 Hours per Week)
- Lectures and exercises will be mixed as desired

## Preconditions

- Basic knowledge in software engineering and
- mathematics, e.g. logic, algebra and number theory

# Prerequisite and Module Examination

---

## Exercises

- Successful passing of two exercise sheets

## Presentation

- Successful presentation on one technical topic,  
four students per one event

## Final Module Examination

- Written examination, 90 min or
- Oral examination, 30-45 min  
(with small number of participants)

# Where do I register and find more information?

---

## Website

- <https://www.b-tu.de/elearning/btu/course/view.php?id=3649>
- Collection of slides, exercise sheets, etc.
- Regular visit to this Moodle course website is strongly recommended

## Overview of Contents

### 1. Foundations and Motivation

- Examples of vulnerabilities in software, e.g. buffer overflows
- Basic terms, definitions, protection goals
- Malware categorisation, virus vs. worm, trojan horse, etc.
- Anonymity, remailer and anonymous web browsing, e.g. TOR

### 2. Security Analysis

- Process models for security engineering
- Misuse cases, attack trees, multilateral security
- Conflicting protection goals and compromises

### 3. Security Design

- Security models (MLS and BIBA)
- Security Patterns
- Information flow control using JiF

---

Steffen Helke: Software Security, 15th October 2018

13

## Part II: Encryption Techniques

---

### 4. Foundations

- History of cryptology, monoalphabetic and polyalphabetic cipher, one-time pad, Enigma
- Prime numbers, prime factorisation, number theory, Chinese remainder theorem (CRT)

### 5. Asymmetric Encryption

- Encryption and digital signatures using RSA
- Pseudo-one-time pad using  $s^2$ -mod-n generator
- GMR - strong cryptographic signature system

### 6. Symmetric Encryption

- Functional principle of DES, Triple-DES and AES
- Cryptographic systems in operation: block cipher vs. stream cipher, ECB, CBC, CFB, OFB, CTR

## Part III: Security Protocol Engineering

---

### 7. Foundations

- Needham-Schroeder protocol, typical attacks, man-in-the-middle attack and replay attack
- Kerberos protocol, differences between v4 and v5, time stamps and nonces to prevent replay attacks

### 8. Verification

- Analysis using Burrows-Abadi-Needham logic (BAN logic)
- Specifying protocols using CSP and verifying using FDR
- Verifying using Proverif based on the Dolev-Yao attacker model

### 9. Summary and Outlook

- Repetition of important content and preparation for the exam
- Research topics for master theses

# What shall you read for this course?

## Script

- Andreas Pfitzmann: *Security in IT-Networks*  
→ only the first *three* chapters are relevant!

## Books

- R. Anderson: *Security Engineering*, Wiley, 2001.
- C. Eckert: *IT-Sicherheit* (in German), Oldenburg-Verlag, 2014 (9. Auflage).
- B. Schneier: *Applied Cryptography*, Wiley, 1996.

## Collection of Articles

- addressing special topics, e.g. for presentations, available on our Moodle website

Steffen Helke: Software Security, 15th October 2018

16

# Example: Trojan Horse of the German Government



Bundestrojaner

## Background to the Trojan Horse of the German Government

### General remarks

- Increasing communication via VoIP or Skype
- Usually this communication will be encrypted
- Hence, source tracking is necessary for the purposes of spying



### Constraints (German Laws)

- Source tracking using trojan horses in general is not allowed
- The observation is only permitted in certain situations, e.g. if the life of a person is in danger

## Design of the Bundestrojaner, Version 2010

### Features (some critical)

- Creating screenshots
- Observation of VoIP or Skype
- Reloading of any software modules
- Transporting observed data via a server from the USA to Germany



### Weaknesses

- Weak authentication mechanism
- Encryption using AES is good, however the encryption was combined with a weak block cipher (ECB)
- ECB – Electronical Code Book

# What is the Difference between Safety and Security?

---

## Basic Terms

### Functional Reliability (Safety)

- Idea: Protecting the environment from the system
- Functions should be implemented correctly according to the specification



### Information Security (Security)

- Idea: Protecting the system from its environment
- Only authorised modification or retrieval of information is allowed



Steffen Helke: Software Security, 15th October 2018

21

## Different Types of IT Systems

---

### Closed Systems

- Homogeneous hardware, similar operating systems
- Technology often only from one supplier
- Limited number of participants
- Same location
- Central administration

### Open Systems

- Heterogeneous hardware and operating systems
- Networked and physically distributed systems
- Decentralized administration

### Socio-technological Systems

- Understanding IT systems as a part of society
- Compliance with legal requirements
- Taking into account user acceptance of security concepts

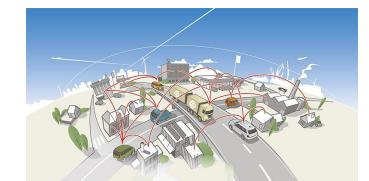
## Open Computer Networks in our Everyday Life

---

- ➔ Main communication medium of industrialized countries
- ➔ Almost all computers are integrated into networks



- ➔ Formerly closed systems are now also integrated into networks, e.g. embedded systems
- ➔ Useful for software updates, maintenance and error analysis during operation



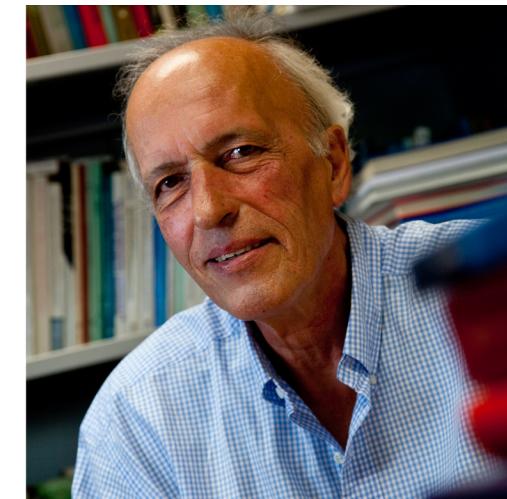
*All people in this world increasingly depend on the correct functioning of software systems that surround them personally, socially, economically and politically.*

## ■ What are the challenges?

We need to develop *dependable* software systems

## ■ How this can be implemented?

We need to establish high quality standards for such software systems



## What does dependability means?

---

### Quelle

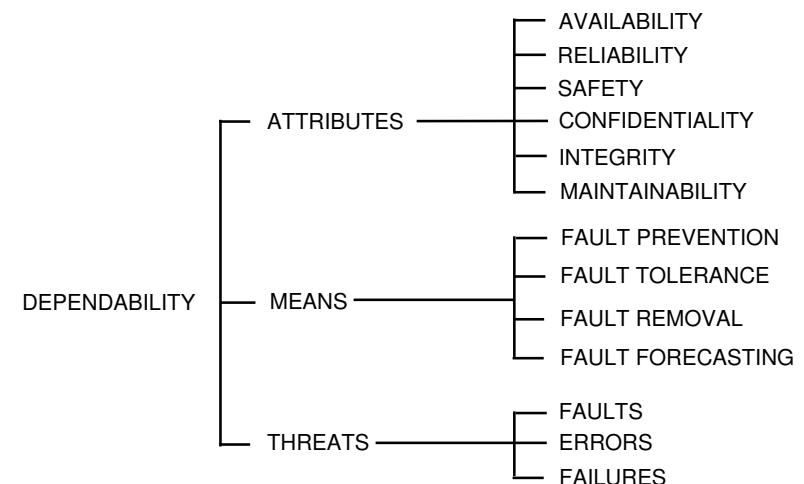
- Jean Claude Laprie (ed.): *Dependability: Basic Concepts and Terminology*. Springer, 1992.
- ➔ Objective of this document is to give an informal and precise definition of dependable systems

### Background

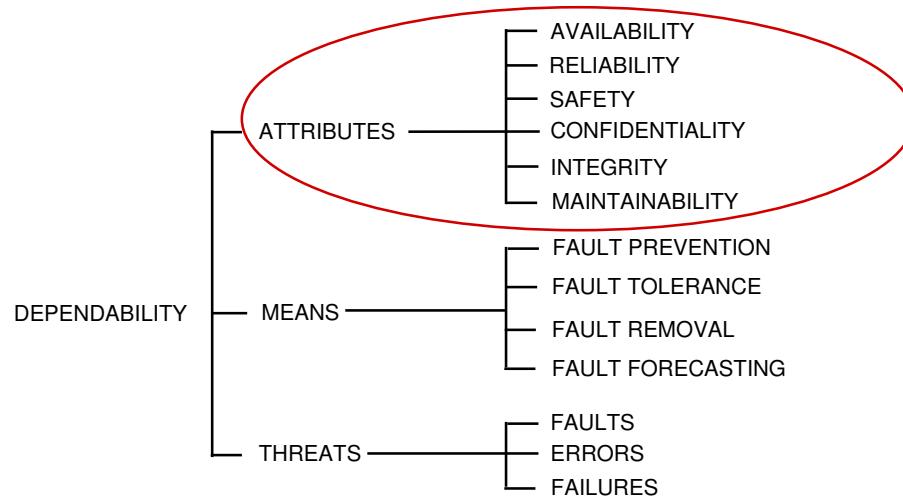
- Results of an exhaustive discussion on the characteristics of fault-tolerant systems
- Translations in many different languages
- Common reference for quality standards to clarify terms

## Dependability defined by Laprie

---



# Dependability defined by Laprie



# Attributes of Dependable Systems

- **Availability**  
readiness for correct service at a certain point in time
- **Reliability**  
continuity of correct service over a certain time interval
- **Safety**  
absence of catastrophic consequences on the environment
- **Confidentiality**  
absence of unauthorized disclosure of information
- **Integrity**  
absence of unauthorized system modifications
- **Maintainability**  
ability to undergo modifications and repairs

## Attributes of Dependable Systems

- **Availability**  
readiness for correct service at a certain point in time
- **Reliability**  
continuity of correct service over a certain time interval
- **Safety**  
absence of catastrophic consequences on the environment
- **Confidentiality**  
absence of unauthorized disclosure of information
- **Integrity**  
absence of unauthorized system modifications
- **Maintainability**  
ability to undergo modifications and repairs

## Three important protection goals (CIA)

Note: This does not refer to the Central Intelligence Agency ;-)  
It's just a kind of mnemonic

- 1 **Confidentiality**  
absence of unauthorized disclosure of information
- 2 **Integrity**  
absence of unauthorized modifications of information
- 3 **Availability**  
readiness for correct service at a certain point in time

# Reliability vs. Availability

## → Availability

readiness for correct service at a certain point in time

## → Reliability

continuity of correct service over a certain time interval

## → Safety

absence of catastrophic consequences on the environment

## → Confidentiality

absence of unauthorized disclosure of information

## → Integrity

absence of unauthorized system modifications

## → Maintainability

ability to undergo modifications and repairs

# Reliability vs. Availability

## Reliability

- Measure for continuity of correct service
- Estimated value for the time period until the system next fails
- e.g. *Mean Time between Failures* (MTBF)

## Availability

- Measure for correct service at a certain point in time
- Definition is additionally influenced by  
*Mean Time to Repair* (MTTR)
- e.g., it could be calculated by  $\frac{MTBF}{MTBF+MTTR}$

⇒ Reduced downtime (MTTR) increases availability,  
but not system reliability!

| Reliability | Repairing capability | Availability |
|-------------|----------------------|--------------|
| constant    | Λ decreasing         | ⇒ decreasing |
| constant    | Λ increasing         | ⇒ increasing |
| increasing  | Λ constant           | ⇒ increasing |
| decreasing  | Λ constant           | ⇒ decreasing |

### Example 1

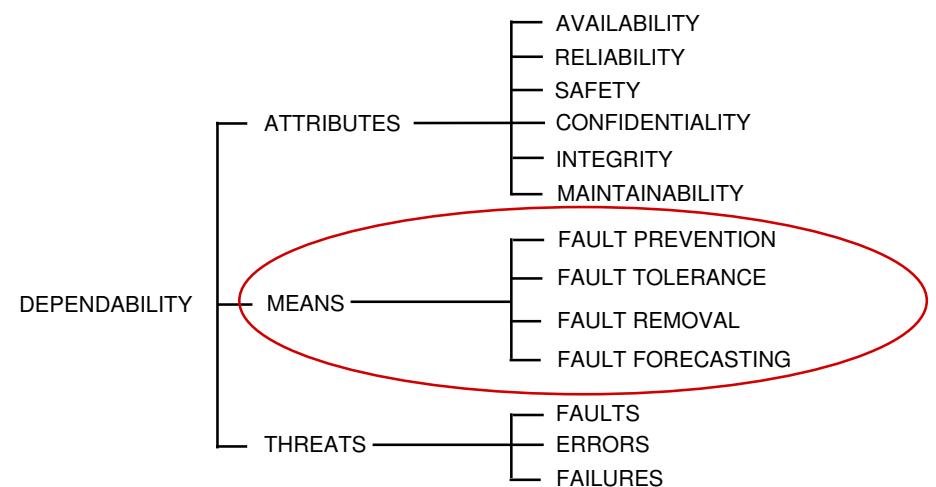
- System A fails *once a year*
- System B fails *once a month*

→ A is more reliable than B

### Example 2 (as an extension of Example 1)

- Restarting System A takes *3 days*
  - Restarting System B takes *10 minutes*
- The availability of B is better than that of A

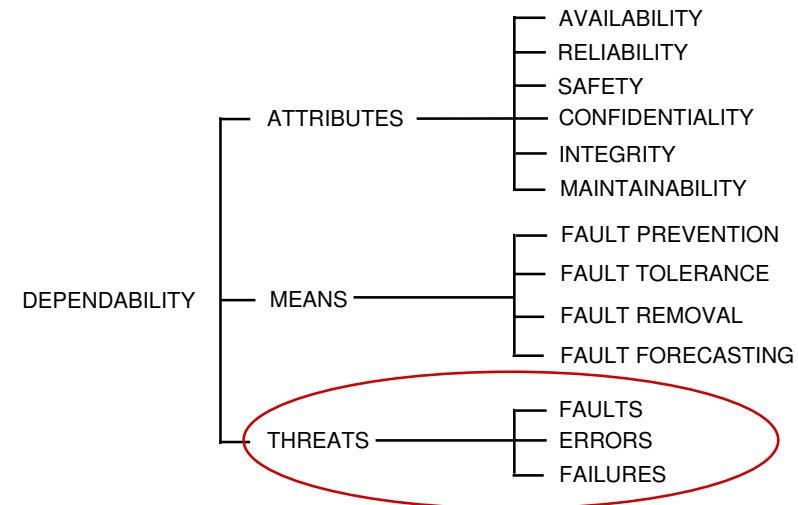
# Means to achieve Dependability



# Means to achieve Dependability

- **Fault Prevention**
  - to prevent the occurrence or introduction of faults
- **Fault Removal**
  - to reduce the number and severity of faults
- **Fault Tolerance**
  - to avoid service failures in the presence of faults
- **Fault Forecasting**
  - to estimate the present number, the future incidence, and the likely consequences of faults

# What are the threats to dependability?



# What are the threats to dependability?

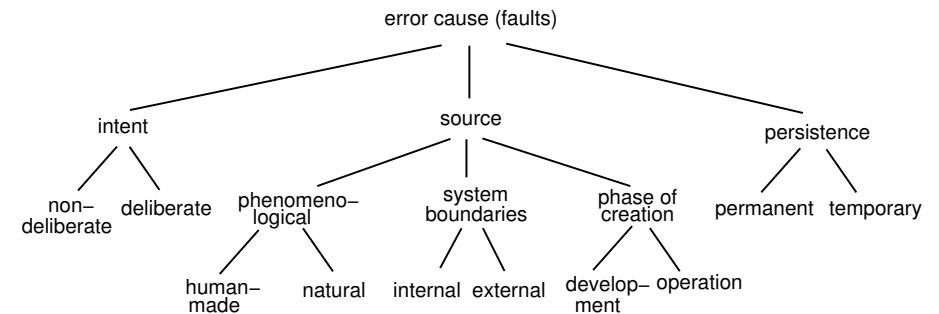
- **Faults**

faults may exist, e.g. inside of code,
  - but do not necessarily have to be activated
- **Errors**

transition from correct system state to an *error state*
  - the cause of an error is usually a fault
  - errors are observable by reading error states, however the services may not influenced
- **Failures**

transition from correct service to an *incorrect service*
  - the cause of a failure is usually an error
  - failures become observable at the service interface

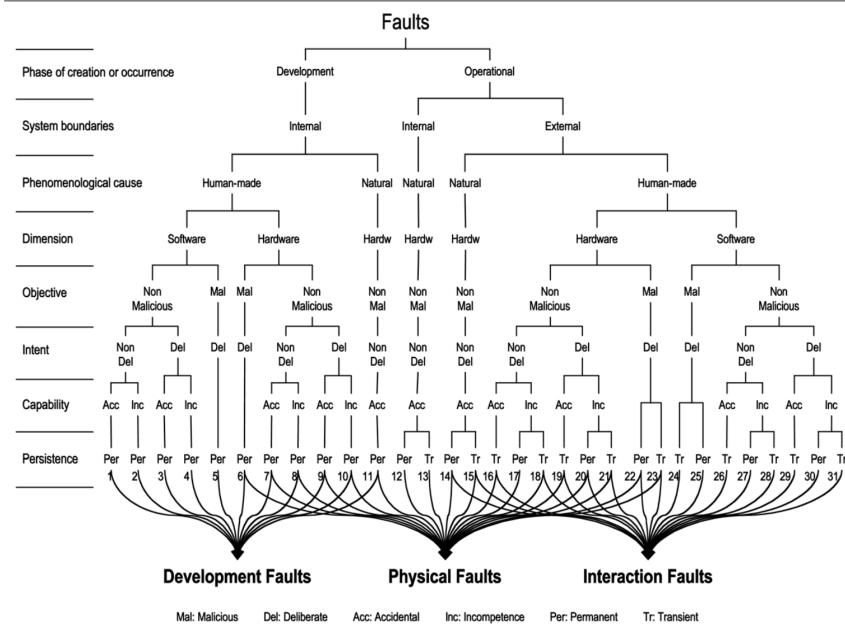
# How to classify faults?



## Examples

- 1 Malicious and incorrect function (e.g. Trojan Horse)
  - deliberate, internal, human-made, development, permanent
- 2 Incorrect computation caused by electromagnetic radiation
  - non-deliberate, external, natural, temporary, operation

## The classes of combined faults



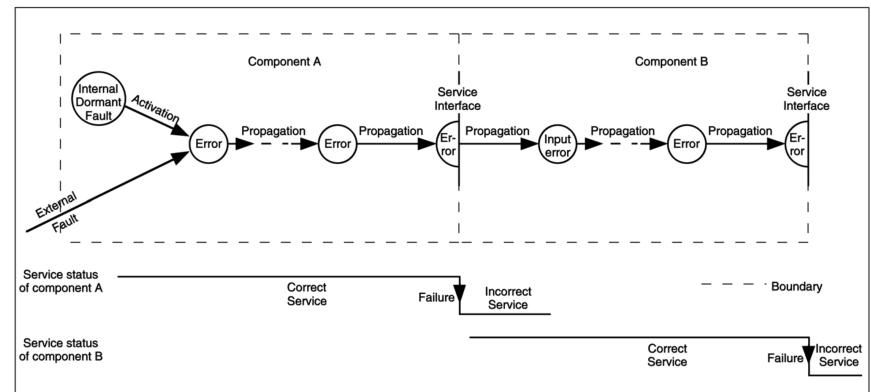
Quelle: Avizienis et. al: *Basic Concepts and Taxonomy of Dependable and Secure Computing*, IEEE Transaction on Dependable and Secure Computing, 2004.

Steffen Helke: Software Security, 15th October 2018

39

## Propagation between system components

- Faults activate errors inside of a component
- Errors are successively transformed into other errors
- Some errors cause failures at the service interface and failures of components can be external faults of other components



Quelle: Avizienis et. al: *Basic Concepts and Taxonomy of Dependable and Secure Computing*, IEEE Transaction on Dependable and Secure Computing, 2004.

Steffen Helke: Software Security, 15th October 2018

40

## What does **Security** mean?

→ 4 sub categories defined by Claudia Eckert [Eckert,2014]

Note: Security is a terribly overloaded word, which often means quite incompatible things to different people.

Quotation by Ross Anderson: *Security Engineering*, 2008.

## What does **Security** mean?

- 1 **Safety**
  - Protecting the environment from the system
  - Functions should be implemented correctly according to the specification
- 2 **Security**
  - Protecting the system from the environment
  - Only authorised modification or retrieval of information is allowed, focus on *protection of data currently being processed*
- 3 **Protection**
  - Similar definition to security, but with a focus on protecting archived data (e.g. backups)
- 4 **Privacy**
  - Ability and/or the right to control the transfer of your personal data

Steffen Helke: Software Security, 15th October 2018

42