MAFTIA's Dependability Concepts

Yves Deswarte & David Powell
LAAS-CNRS, Toulouse, France

OASIS PI Meeting, Santa Rosa, August 19-21, 2002

MAFTIA

IST Dependability Initiative
Cross Program Action 2
Dependability in services and technologies

Malicious- and Accidental-Fault Tolerance for Internet Applications

University of Newcastle (UK)	Brian Randell, Robert Stroud
University of Lisbon (P)	Paulo Verissimo
DSTL + QinetiQ (ex-DERA) (UK)	Tom McCutcheon, Sadie Creese
University of Saarland (D)	Birgit Pfitzmann
LAAS-CNRS, Toulouse (F)	Yves Deswarte, David Powell
IBM Research, Zurich (CH)	Marc Dacier, Michael Waidner

c. 55 man-years, EU funding c. 2.5M€
Jan. 2000 -> Dec. 2002 (Feb. 2003)

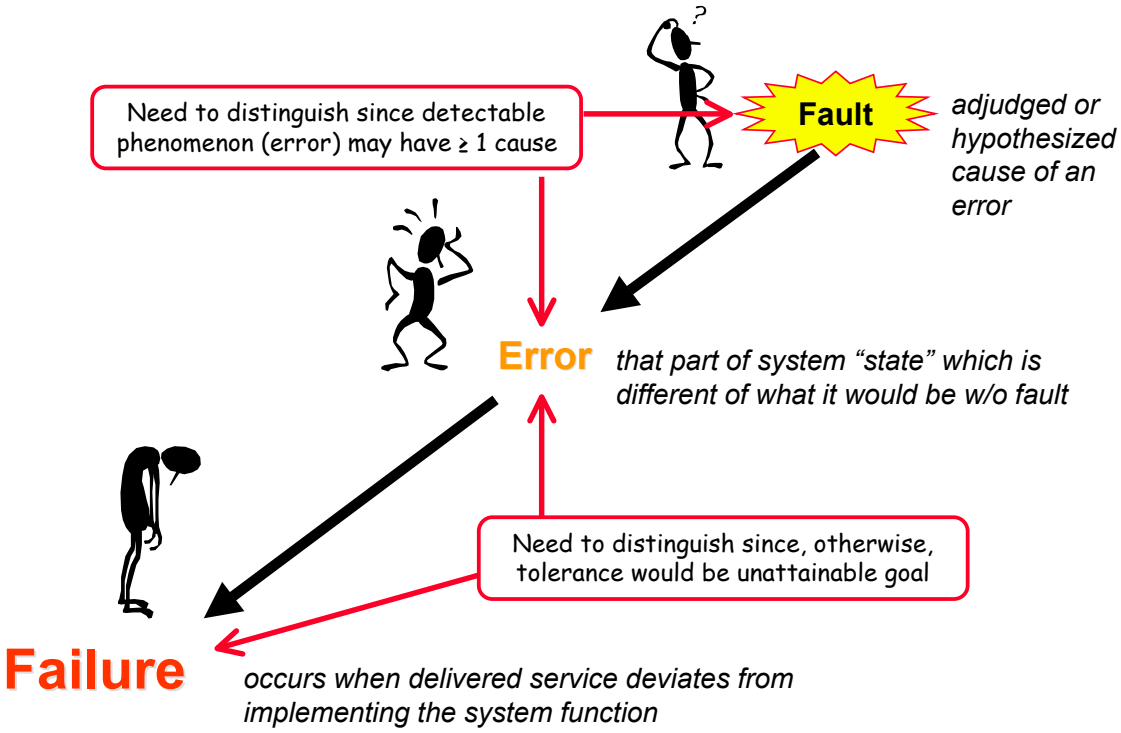
Objectives

- Architectural framework and conceptual model (WP1)
- Mechanisms and protocols:
  - dependable middleware (WP2)
  - large scale intrusion detection systems (WP3)
  - dependable trusted third parties (WP4)
  - distributed authorization mechanisms (WP5)
- Validation and assessment techniques (WP6)

Summary

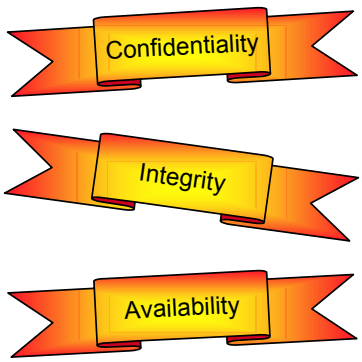
- Causal chain of impairments
- Security policy and security failure
- Intrusion, attack and vulnerability
- Security methods
- Fault tolerance
- Intrusion detection
- Integrated intrusion detection/tolerance framework

Causal Chain of Impairments

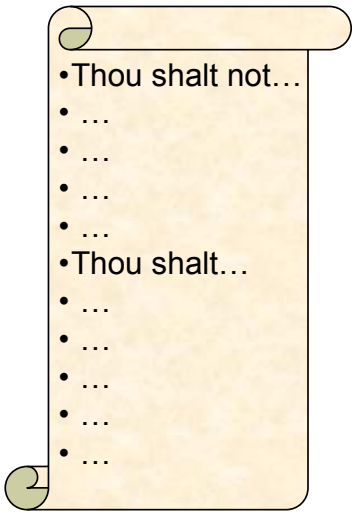


Security Policy

- Security properties which are to be fulfilled by the system

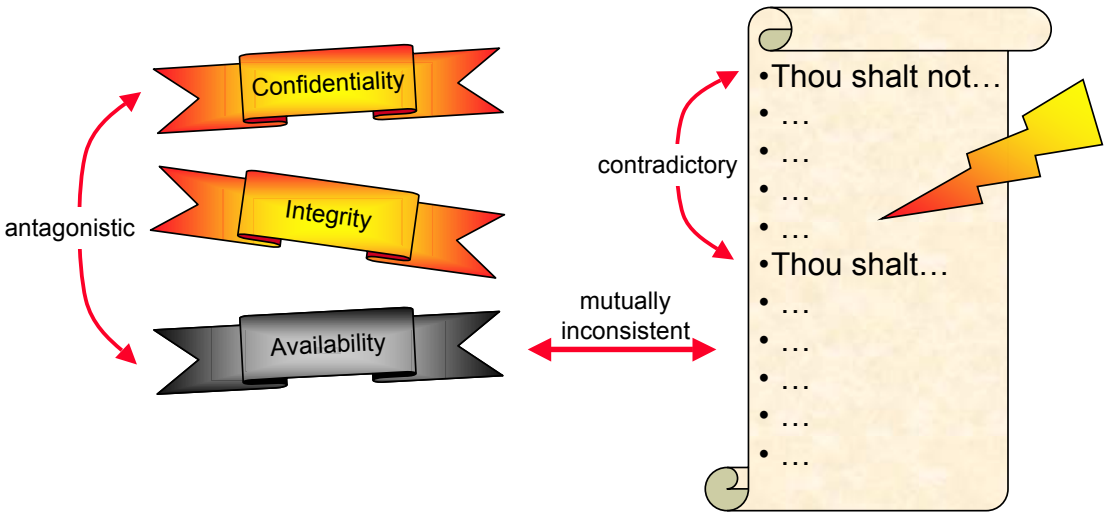


- Rules according to which the system security state may evolve

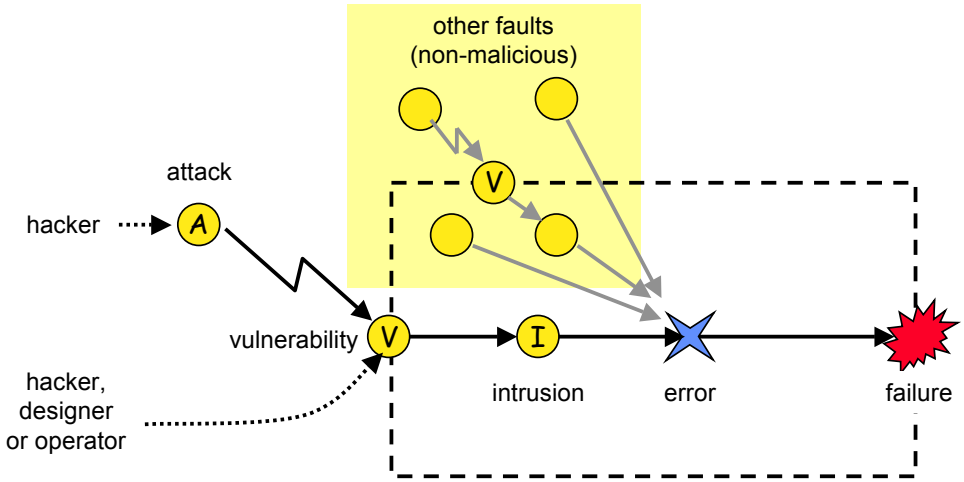


Security Failure

- Violation of a security property of intended security policy

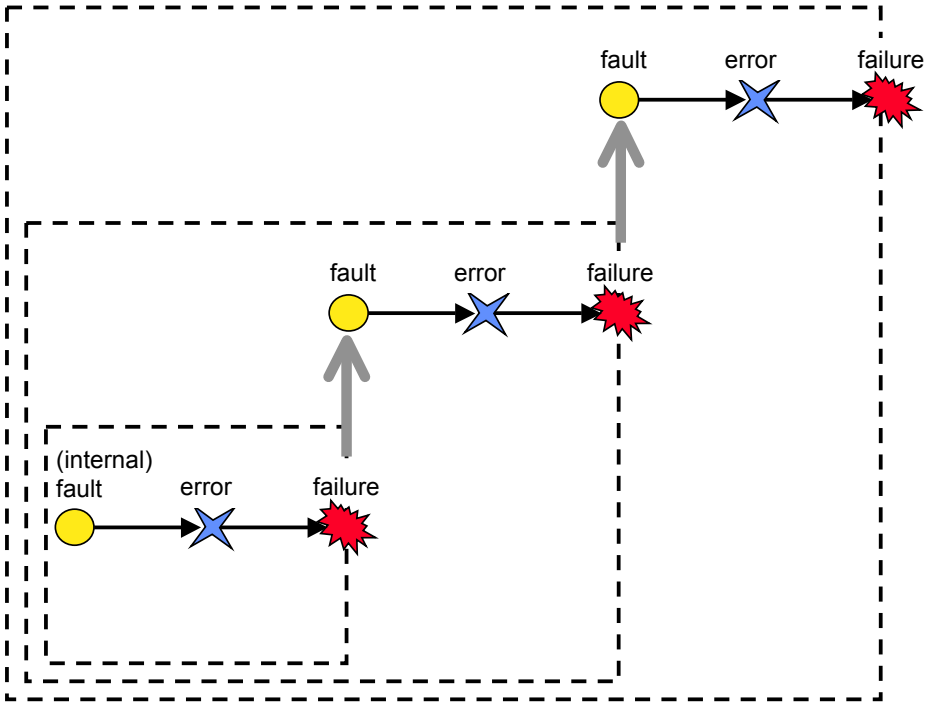


Fault Model

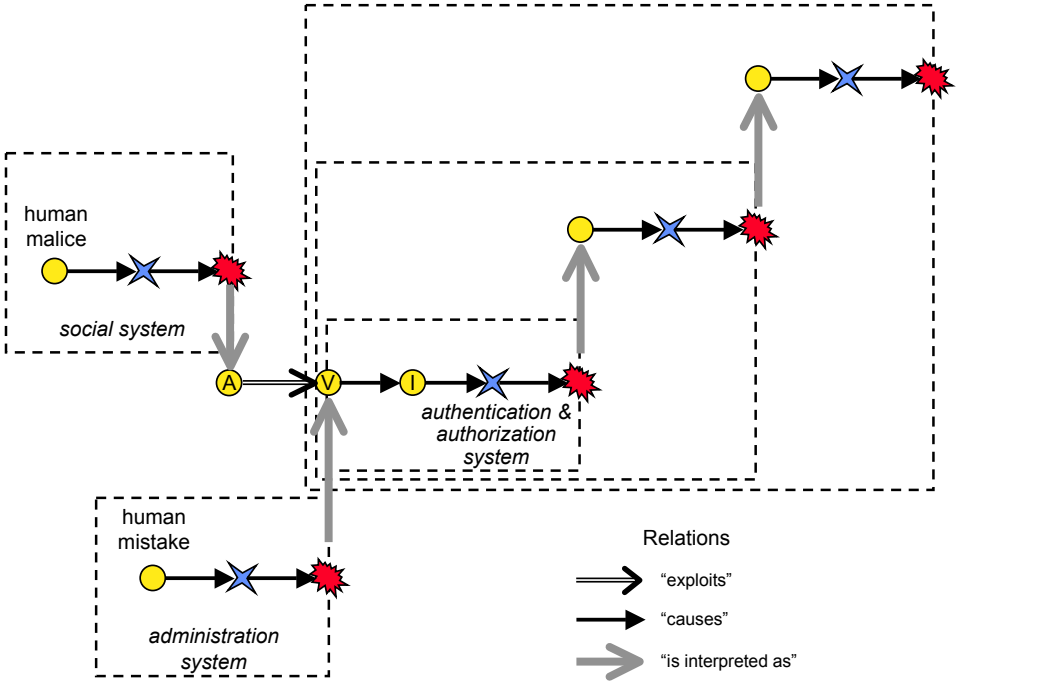


- **attack** - malicious external activity aiming to intentionally violate one or more security properties; an *intrusion* attempt
- **vulnerability** - a malicious or non-malicious fault, in the requirements, the specification, the design or the configuration of the system, or in the way it is used, that could be exploited to create an *intrusion*
- **intrusion** - a malicious fault resulting from an *attack* that has been successful in exploiting a *vulnerability*

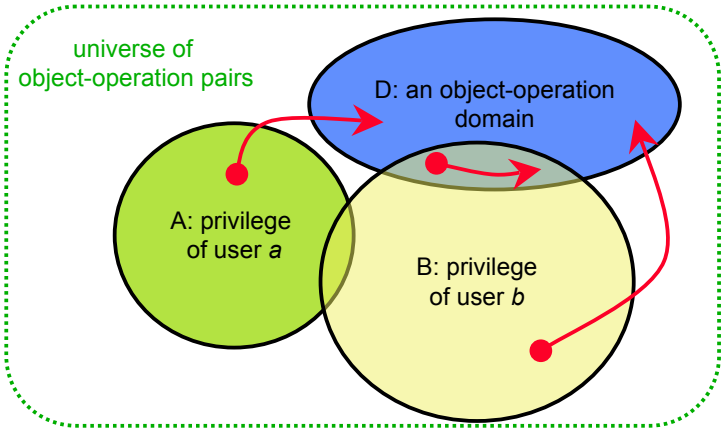
Fault Model: Recursion



Malicious Fault Model: Recursion?

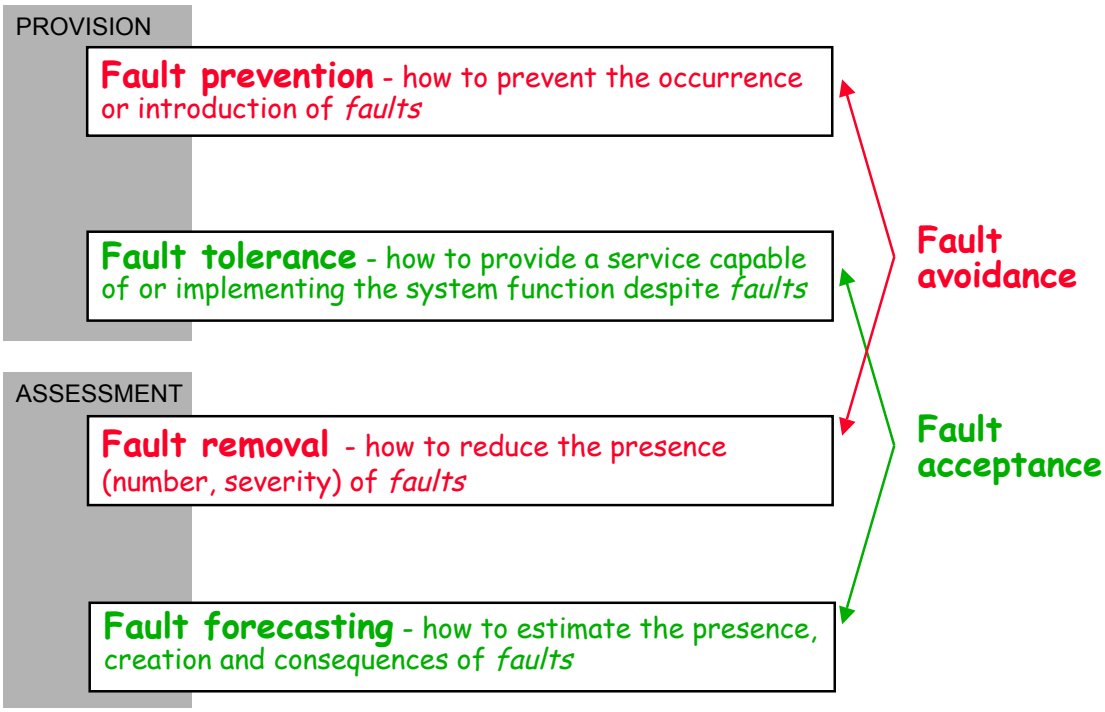


Outsiders or Insiders: Privilege

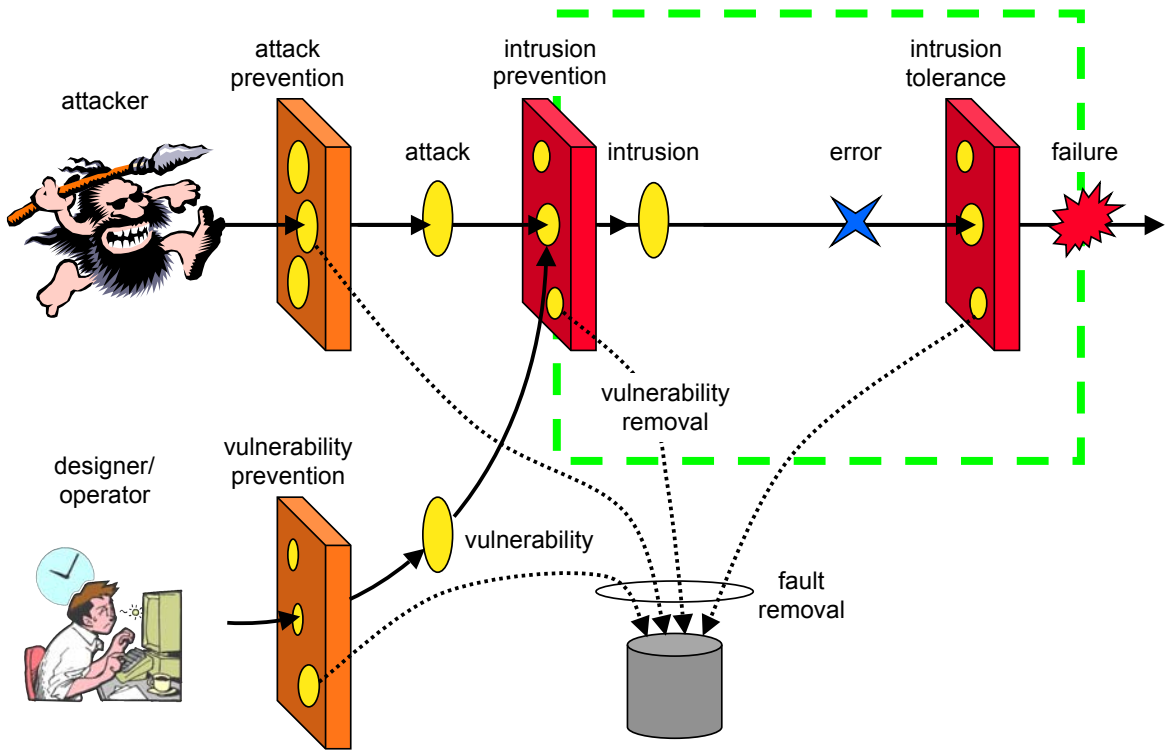


- Theft of privilege: unauthorized increase in privilege
- Abuse of privilege: improper use of authorized operations
- Outsider: current privilege does not intersect considered domain
- Insider: current privilege intersects considered domain

Dependability Methods



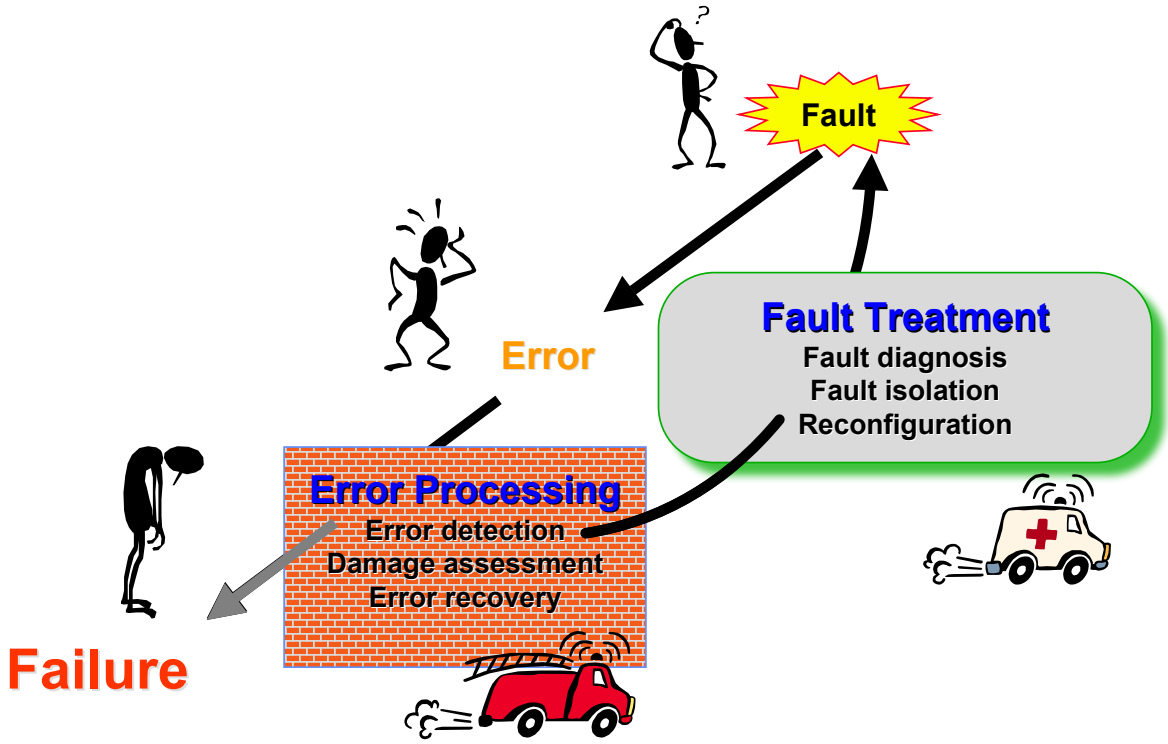
Prevention, Tolerance and Removal



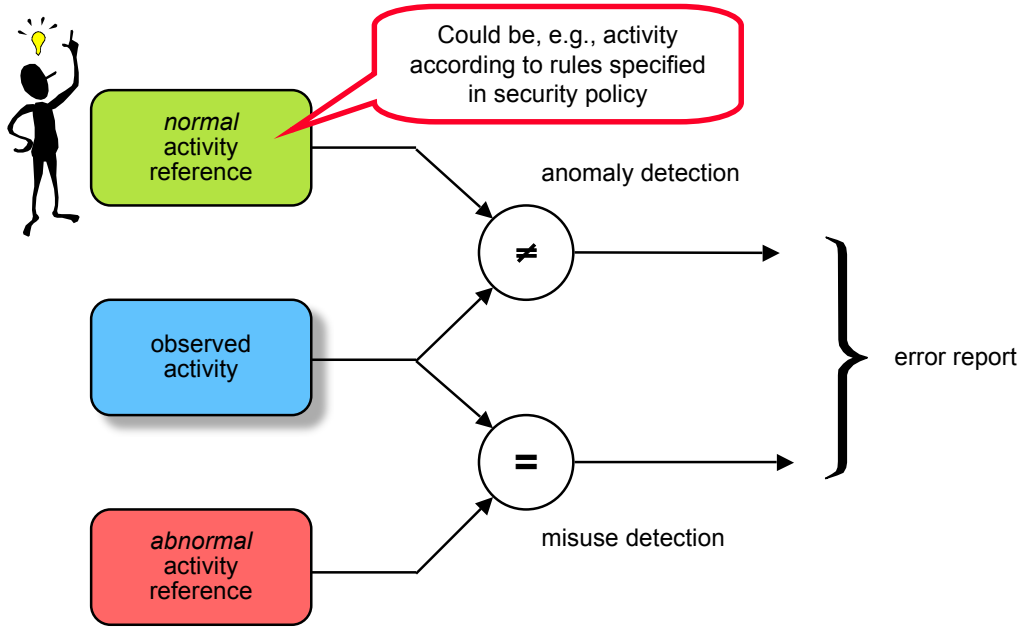
Security Methods

		Attack	Vulnerability	Intrusion
Prevention	*how to prevent the occurrence or introduction of…*	deterrence, laws, social pressure, secret service…	security policy, semi-formal and formal specification, rigorous design and management…	firewalls, authentication, authorization… (+ attack prevention vulnerability prevention)
Tolerance	*how to provide a service capable of or implementing the system function despite…*	vulnerability prevention vulnerability removal intrusion tolerance	= intrusion tolerance	confinement, detection/recovery, masking (e.g. FRS), + intrusion detection for fault treatment
Removal	*how to reduce the presence (number, severity) of…*	not applicable	formal proof, model-checking, inspection, test…	not applicable
Forecasting	*how to estimate the creation and consequences of…*	intelligence gathering, threat assessment, attack warning…	assess presence of vulnerabilities, exploitation difficulty, potential consequences	vulnerability forecasting, attack forecasting

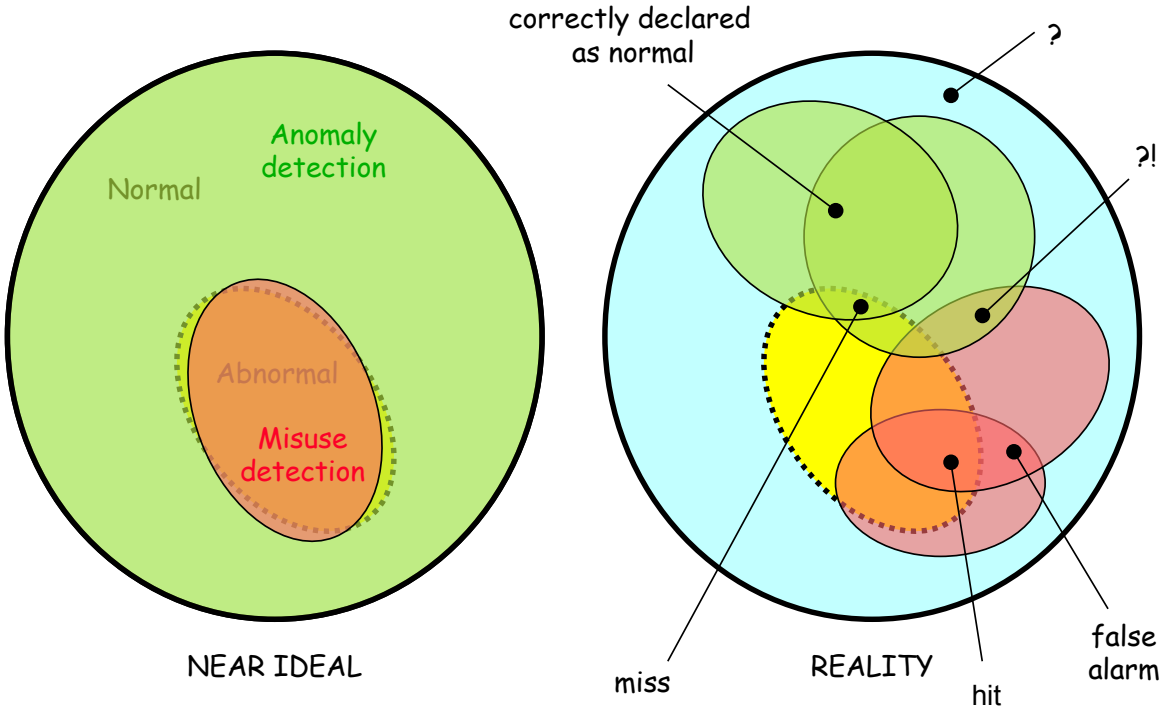
Fault Tolerance



Error Detection



Anomaly vs Misuse Detection



Preemptive Error Detection

[Avizienis, Laprie & Randell 2001]
(as opposed to concurrent error detection)

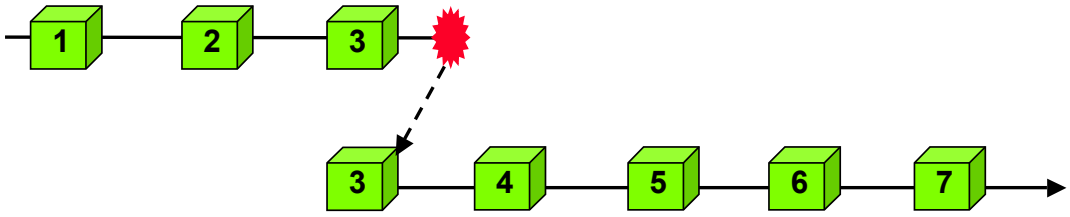
- Core concepts: AKA "built-in test"
  - -> e.g., Memory scrubbing
- Interpretation wrt malicious faults
  - Vulnerability scanning
  - Configuration checking

(Damage assessment)

- Core concepts: aims to evaluate extent of error propagation before initiating recovery
  - How many checkpoints to rollback?
  - How many processes affected before detection?
- Interpretation?
  - How many files have been corrupted by an intruder, and thus need to be restored *before use*?

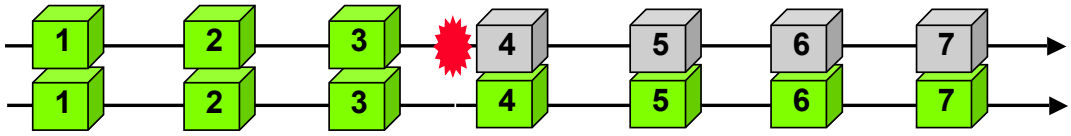
Error Recovery

Backward recovery



Forward recovery

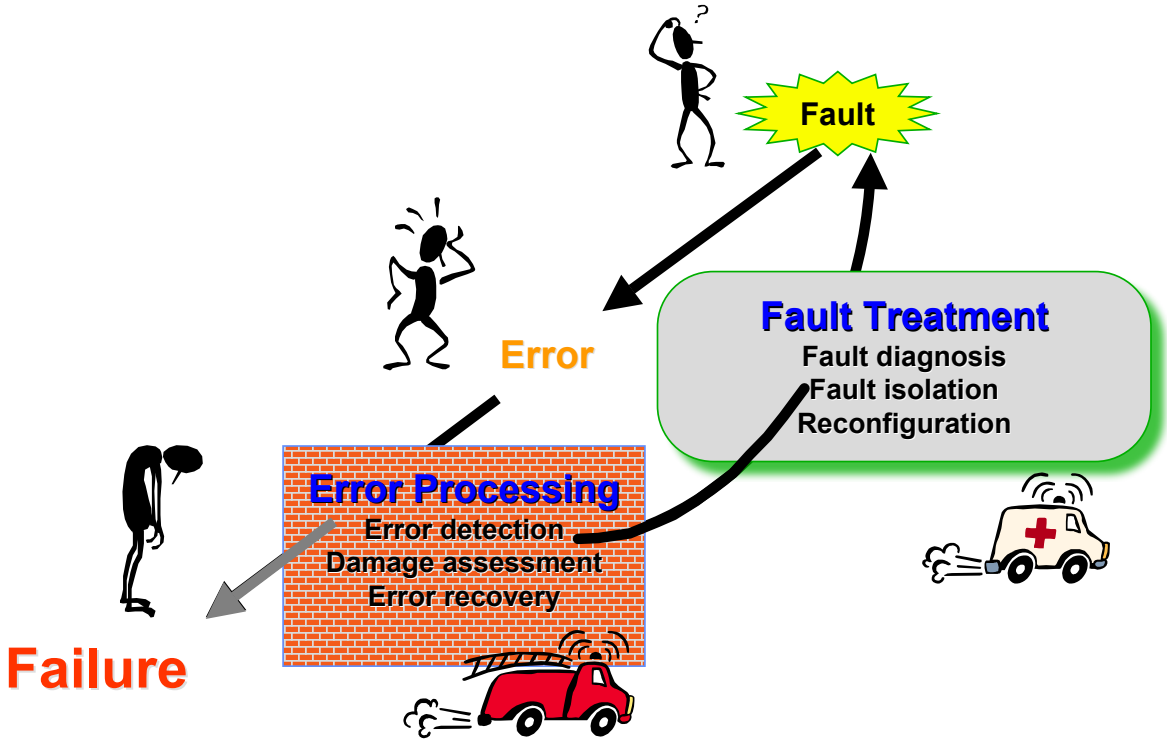
Compensation-based recovery (fault masking)



Error Recovery

- Backward recovery
  - Software rejuvenation
  - Operating system re-installation
  - TCP/IP connection resets
  - System reboots and process re-initialisation
  - Software downgrades
- Forward recovery
  - Automated re-keying procedures ("proactive security")
  - Switching to diminished "safe" mode.
  - Software upgrades
- Masking
  - Voting mechanisms
  - Fragmentation-Redundancy-Scattering
  - ID Sensor correlation

Fault Tolerance



Fault Diagnosis

- Core concepts: identification and locations of faults; prerequisite to isolation & reconfiguration
- **Intrusion diagnosis**, i.e., trying to assess the degree of success of the intruder in terms of system penetration
- **Vulnerability diagnosis**, i.e., trying to understand the channels through which the intrusion took place so that corrective maintenance can be carried out

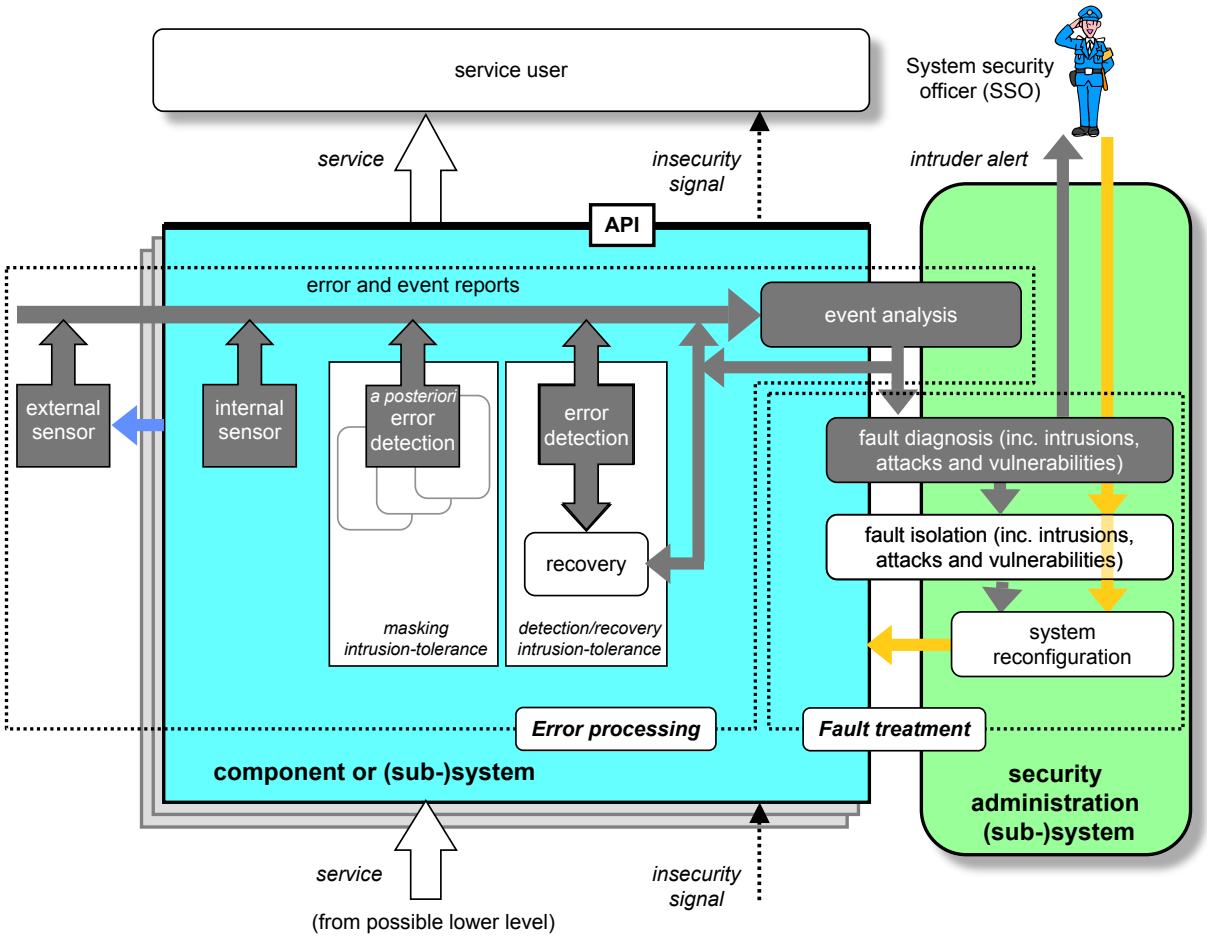
  (diagnosis immediate if errors signaled by vulnerability scanner or configuration checker)
- **Attack diagnosis**, i.e., finding out who or what organisation is responsible for the attack in order that appropriate litigation or retaliation may be initiated

Fault Isolation

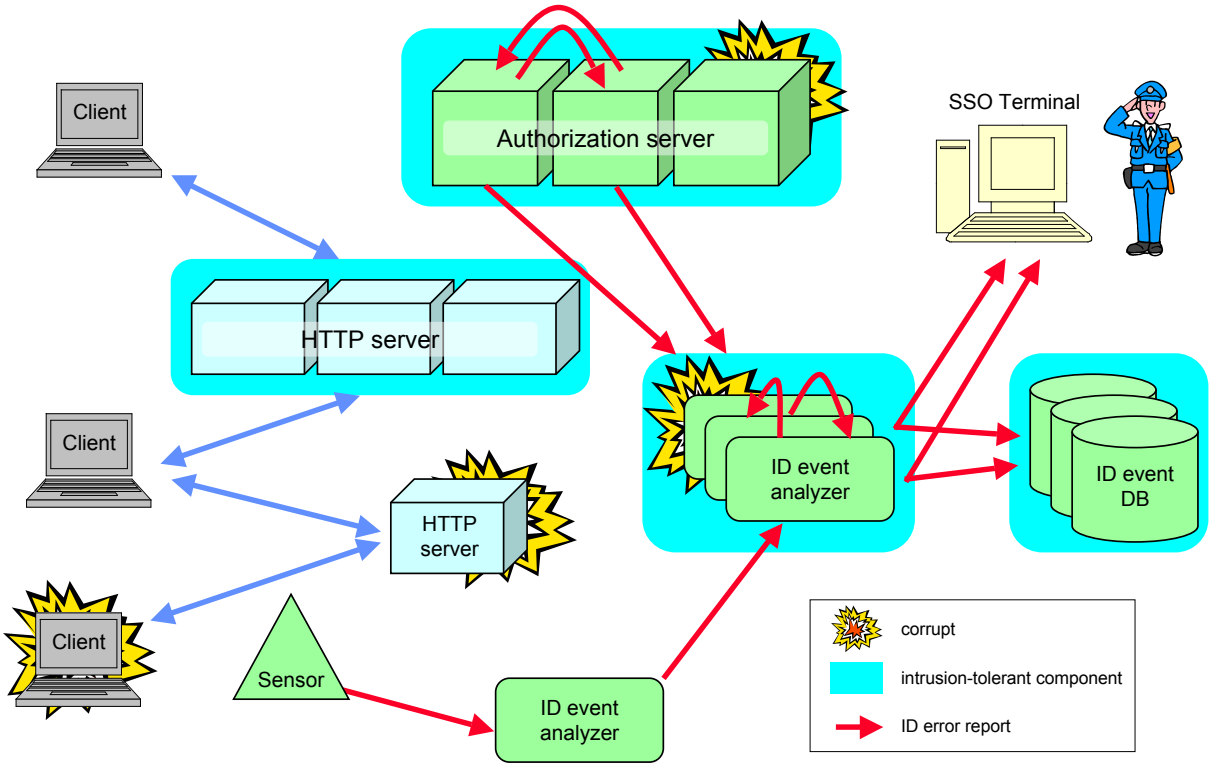
- Core concepts: needed to prevent further errors
- Interpretation wrt. intrusions
  - Blocking traffic from an intrusion containment domain that is diagnosed as corrupt, by, for example, changing the settings of firewalls or routers
  - Removing a corrupted file from the system
- Interpretation wrt. root causes (vulnerability/attack)
  - Taking off line software versions with newly-found vulnerabilities
  - Arresting the attacker

System Reconfiguration

- Core concepts: redeployment of fault-free resources + corrective maintenance
- Interpretation wrt. intrusions
  - Change a voting threshold, e.g., 3/5 => 2/3 after 2 corruptions
  - Deployment of countermeasures, inc. probes and traps
- Corrective maintenance actions
  - Vulnerability removal
    - software revision and upgrade
    - security patches
  - Attacker rehabilitation



A (very) Simple Example



http://www.research.ec.org/maftia/