



# Introduction to IT security

**Security and Privacy**  
**VSE Prag, 7 - 11.6.2010**

Institute for Information Processing and  
Microprocessor Technology (FIM)  
Johannes Kepler University Linz, Austria

E-Mail: [sonntag@fim.uni-linz.ac.at](mailto:sonntag@fim.uni-linz.ac.at)  
<http://www.fim.uni-linz.ac.at/staff/sonntag.htm>



- Security, safety and their combinations/differences
- Aims of protection in ICT systems
  - Confidentiality, integrity, availability
  - Non repudiation, authenticity, anonymity/pseudonymity
- Threats: Dangers and risks
  - Kinds of dangers
  - Internal vs. external threats
- Security terms:
  - Weakness, vulnerability, exploit
  - Threat and risk
- The security process

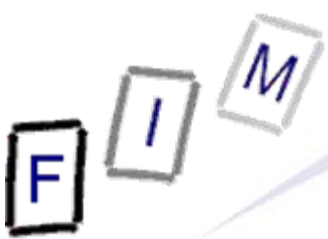


- What is "security"?
  - What does it mean at all?
  - Are there different types?
- When we say: "We secure an ICT system"
  - What are we actually going to do?
- What kinds of threats exist?
  - Internal vs. external
  - Gain vs. damage
- What can we do against these threats?
  - Encryption, signatures
  - Organisational measures
  - Laws and their enforcement
  - User education



# Security vs. Safety

- Even more problematic in German: A single word for both!
- Security: Protection against malicious acts
  - Example: Security check-in at the airport
- Safety: Protection against negative consequences
  - Example: Safety belt in in a car
- Both are completely different topics!
  - Or are they really?
  - Isn't the airport security check also providing for a safe flight?
    - » But the safety check of the pilot before starting has nothing to do with the security check of the passengers!
  - Might the safety belt ensure that we can (try to) control the car in exceptional situations as well?
- Interdependence of safety and security
  - Often used synonymous



# Safety vs. Security

- Safety

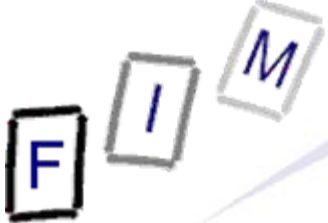
- The state of being protected against failure, damage, error, accidents, harm or any other event which could be considered non-desirable.
- Examples: No sharp edges, electronically insulated
- "If you are using it, no harm will come to you or others"

- Security

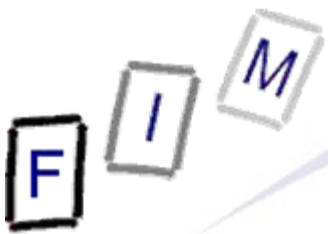
- A condition that results from the establishment and maintenance of protective measures that ensure a state of inviolability from hostile acts or influences.
- Examples: Unauthorized access, information theft
- "Nobody can harm it"

- Usually security is a narrower concept!

- Safety includes protection against well-intended actions

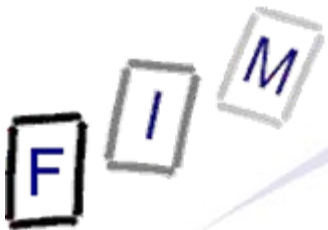


- **I**nformation and **C**ommunication **T**echnology
  - Includes storage, transmission, processing
    - » Routers, telephone line, computers, hard disks, USB sticks, ...
- Typically ICT occurs not alone, but in a system
  - I.e., the elements are interconnected, but can be identified as separate and separately useful components
- System = Several elements in a certain environment (context) building a single entity
  - It consists of
    - » Entities: Subject and objects
    - » Relations between entities
    - » Actions originating by entities and working on entities
  - Note: Humans may (but need not!) be part of a system
  - What is a subject and what an object can change over time
    - » Example: Server and client often change their roles!



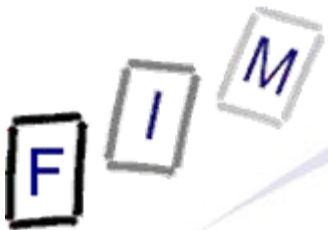
# IT system as defined by C. Eckert

- An IT system is a closed or open technical system with the ability to store and process information
  - Closed: Proprietary, no interaction with the environment or surrounding systems
  - Open: Interconnected, distributed system, communicates by open standards with other systems
    - » Typically heterogeneous hardware, various operating system, ...
- Note: "Closed" and "open" should rather be defined only in relation to the environment
  - Whether open or closed source software is used is a completely different aspect of "open" and "closed"
  - These should not be intermingled!
- Typically IT systems are socio-technical systems
  - Contains human beings
  - Embedded in business structures



# Orderly systems

- A system is orderly, if
  - All actions required from it are performed
    - » Correctly
    - » At the determined point in time
  - No other actions are performed
- This means:
  - It does what it should do and it does **only** these things
    - » No additional work, e.g. sending mails not written by the user
      - Trojan sending Spam
  - What it does, is what was planned
    - » No bugs: Saving a corrupted file
  - Everything happens soon enough, but not too early
    - » No delays, e.g. in VoIP



# The two views on safety/security (According to Dierstein)

---

- An ICT system is **secure** in the sense of **technically secure**, when its planned function is performed in a way, that the remaining risks can be tolerated (by the operator)
  - It must be resistant against threats as far as possible
  - It is very difficult, but not impossible, to misuse it
  - Excluding the remaining risk it is an orderly system
- An ICT system is **safe** in the sense of controllable when its environment (humans, society) is protected from undesirable effects during normal behaviour
  - Excluding the remaining risks
  - A technically secure system, e.g. a database, should have no additional side-effects, e.g. on individual persons
  - Especially important in machinery control systems



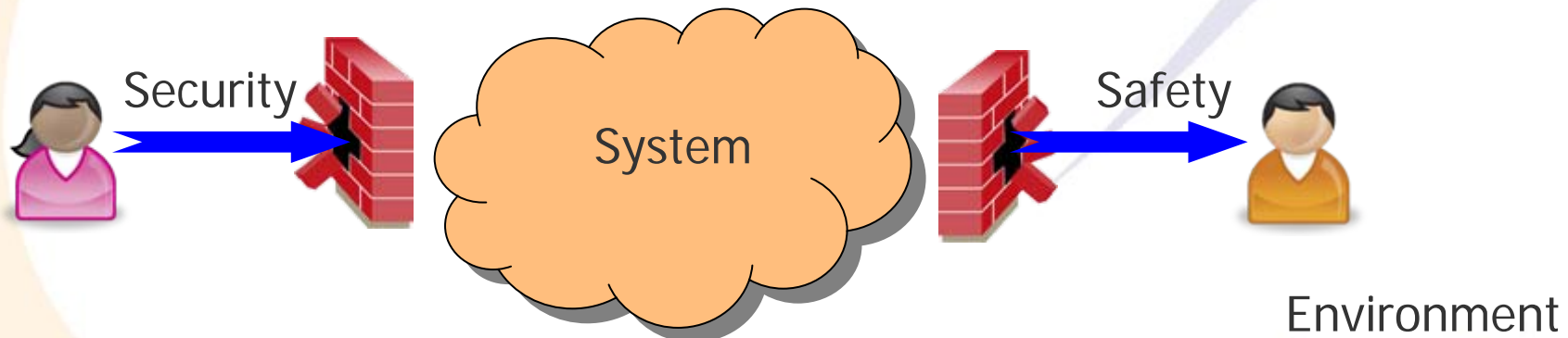
# Elements of controllability

- In addition to be secure, a safe system should also provide
  - **Accountability**
    - » Who initiated a certain action?
      - Typically: Logging of user/system actions
    - » What was the cause for a specific effect?
      - Traces of individual acts, exclusion of random components (or specific logging of everything not clearly pre-determined!)
  - **Legal liability**
    - » Can the actor deny acting?
      - Non-repudiation, e.g. username and password to log in
    - » Can we identify the person behind the actor?
      - Who is legally responsible for this entity
- It must always be possible to identify all effects and trace them back to a human being
  - This could be an operator, a programmer, a user, ...



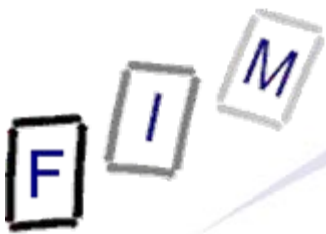
# Safety and Security: In the context of systems

- Safety: Protecting the environment of the system (incl. humans) against non-orderly behaviour of the system
  - Protecting the outside from the inside
- Security: Protecting the system from dangerous behaviour of the environment
  - » Includes all measures taken to achieve this protection
  - Protecting the inside from the outside
  - Note: Security is a process!
    - » It must be continually updated and revised

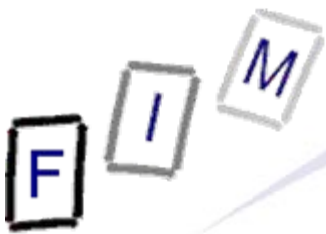




- Functional safety
  - The implemented functionality of the components matches the planned functionality
  - The system will not reach an unspecified state
    - » Protecting the environment from unplanned actions
- In a broader view
  - Safety from random or unintentional events
    - » Fire, earthquakes, car accidents, spilled liquid, ...
- The threat is (usually) unintentional, random
  - Not specifically targeted at the system
  - Does not exploit specific weaknesses
    - » But may occasionally target them (random!)
  - Just "happens" but chances are not increased
  - Insurance is typically no problem
    - » But perhaps costly and only available for specific risks

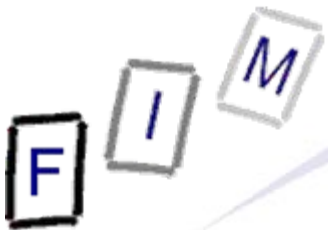


- Information system security:
  - The system will not reach a state leading to unauthorized change of information or disclosure
    - » Note: This cannot be ensured purely technical, as a decision (and then definition) is required, what is "authorized"!
- In a broader view
  - Safety from intentional or planned events
    - » Hacking, data theft, Virus sending, Phishing, ...
- The threat is (usually) planned and consciously executed
  - Protecting the system from the environments misbehaviour
    - » Note: This again requires a definition what is "normal" behaviour!
  - Chances are often carefully weighted and increased
  - Attack might be deliberately hidden
  - Insurance difficult



# Connection Safety ↔ Security

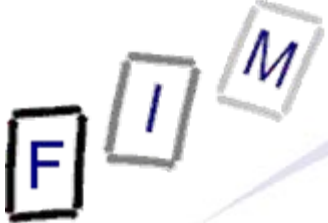
- Safety incidents can often lead to security problems
  - When the system crashes, it doesn't necessarily crash simultaneously in all parts
    - » In this period, some things, e.g. access right checks, might not be working completely or at all
    - » This can be exploited for attacks
  - Typical example: Buffer overflow!
    - » A bug, which may lead to unspecified behaviour → crash
    - » Most intruders crack a system through exploiting such buffer overflows by providing too long input data
    - » Then the safety issue becomes a security problem!
- Security problems often lead to safety issues
  - Not all intruders merely want to steal some data
  - Often they provide a lot of it → Denial of Service
  - Example: Crashing a system (Ping of Death)



# Security: Protection and Privacy

---

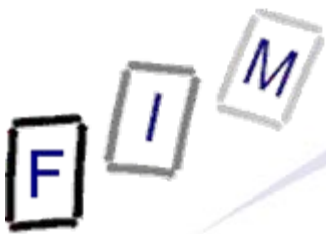
- Data protection: Making sure that data remains accessible and that its integrity is unaffected
  - No accidental changes and no intentional modification
  - Includes therefore both safety and security
- Privacy: Preventing unauthorized access/usage/disclosure
  - The data subject itself can decide, who may do what with his/her personal data
  - This is independent of security/safety!
    - » Planned and perfectly working functionality is no security or safety issue, but still an issue for privacy!
- Data protection is a prerequisite for privacy
  - Factually: Only what (still) exists, is protected
    - » Anonymous or deleted data is unprotected by privacy!
  - Legally: Personal data must be secured
    - » No security → Fine even before any unauthorized access



# Aims of protection in ICT systems

---

- Key concepts (stemming from secret services):
  - Confidentiality: Keep it secret
  - Integrity: Keep it intact
  - Availability: Keep it accessible
- Additional concepts (also important today/in business):
  - Non repudiation: Ability to tie actions and objects to a person
    - » For actions performed: Praise or blame
    - » For objects: Who created them, is responsible for the content, ...
  - Authenticity: Truthfulness of origin
    - » Who created/generated the information/Who is this
  - Privacy/Anonymity/Pseudonymity: Appear as someone else
    - » For respecting privacy
      - Note: This is not necessarily a problem regarding possession or non repudiation → Someone might be able to uncover the identity!



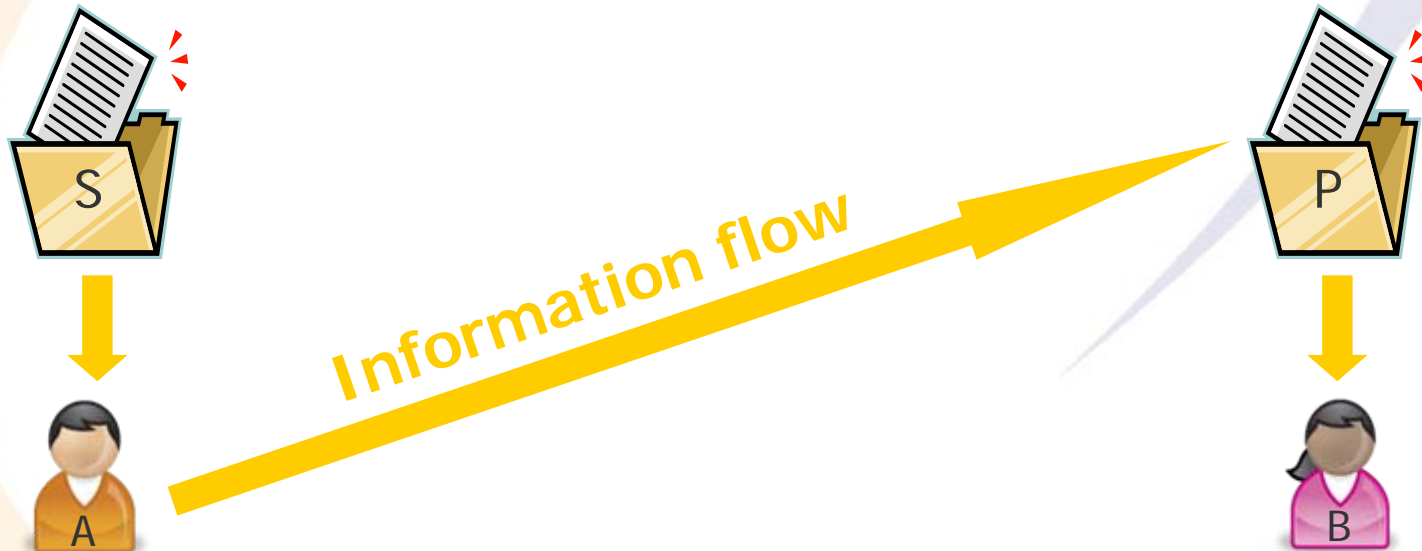
# Confidentiality

- Information must be available to those person **only**, who are authorized to do so
  - Everyone else shouldn't even see that the information exists!
  - Typically some metadata is still visible:
    - » Access to a directory → Names of all files within are visible, regardless whether they can actually be read or not
- This especially affects controlling information flows
  - This is called the "confinement problem"
  - Example:
    - » A may read a document S with secret data
    - » A may write to a document P with public data
    - » A can copy the information from S to P
    - » Result: The (supposedly) secret information is now public!
  - Attention: Most normal operating systems **do not solve this!**
    - » This is typically something for military-grade security to provide!



# Confidentiality: The confinement problem

- Person A can read document S
  - Person A can write document P
  - Person B can read document P
- ⇒ **A can transfer information to B within the system**
- Note:
    - Person A cannot send any data to person B
    - Person B cannot read document S





- Keeping information unchanged
  - Protects against transmission errors as well as attackers
    - » Examples: Forgery, disk crash, noisy lines
  - Includes the creation of data by unauthorized persons
- Usually much weaker in practice, but more useful:  
Only allow **authorized** modifications
  - Requires a definition of user rights and secure identification
- Even more weaker: Detecting modifications
  - That is what typically is present
    - » The original value is then just sent again (transmission), restored from a backup media, requested a second time, ...
- Typical examples:
  - Prevention: Write protection, WOM, backup
  - Detection: CRC/MAC codes, hash values, el. signatures



- If you are authorised (and authenticated), you can perform any actions you may without unauthorized interference
  - Authorized interference
    - » CPU scheduling and associated delays
    - » Locking of a document opened by another authorized person
  - Unauthorized interference
    - » Deliberate or accidental bandwidth consumption to delay the data transfer
    - » Denial of service attacks
- Typical solution:
  - Sufficient dimensioning of the system
    - » May include hot-spare, backups, ...
  - Exclusion of all unauthorized persons
  - Quotas to limit resource usage by authorized users



- To ensure than any action or object can be traced back to a human responsible for it
  - If someone performs an act, regardless of whether perfectly in order, mistakenly, or maliciously, it can be traced back to the person who initiated it
    - » This person cannot (credibly) deny performing this act
  - Objects must similarly be associated to persons
    - » Who created the content
- Important prerequisite for legal consequences!
  - Usually you have to prove what is good for you
    - » That someone else acted and caused the damage
    - » That you did act correctly, so you are not to blame
    - » That the content of the document is what was stated
- Note: This is not foolproof!
  - Depends on authentication and sometimes voluntariness

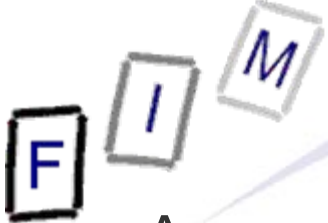


- Typical examples:
  - Extensive logging after authentication
    - » When did which user initiate what?
  - Webbugs
    - » Whether a webpage was displayed; also usable for E-Mail
  - Electronic signatures
    - » Depending on security (=availability to others) of the private key
  - Confirmation, like reply E-Mail
    - » Request an reply from the recipient
      - "Delivery notifications" of E-Mail are insufficient (only in closed systems they might provide non-repudiation)
  - Trusted Third parties
    - » Confirmation of someone else with no own interests
  - File permissions
    - » Who is the owner, who is member of the group
      - Take care of the individual permissions!



# Authenticity

- Information must be correct (and therefore believable)
- This applies to both subjects and objects
  - Subject: Is the person who he/she claims to be?
    - » Passwords, tokens, biometrics, ...
  - Object: Who entered this data?
    - » Ownership, Metadata, ...
- Ascertained by "authentication": Attempting to verify the identity of an entity ("principal"; person, program etc.)
  - Note: Access control is not necessarily authentication!
    - » "Bearer passes": Mere possession of a token can be enough
    - » The token need not disclose the identity of its owner
  - Something different is "identification"
    - » Someone is present → Who is he/she/it?
      - Person → ID
    - » Authentication: Person + ID → Do they match?

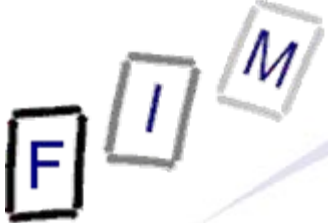


- Anonymity: Nobody can match the data to a certain person
  - Note: A person need not be identified by name or uniquely
  - If it can be picked from a group, this is sufficient
  - Example: Third person from left
    - » Anonymous in general
    - » Regarding a picture of a few persons this is not anonymous!
- Pseudonymity: Persons can still be identified, but this requires some additional information
  - Typical example: Assigning a random value to each person and keeping a separate file of the indentifying characteristics and this number
    - » Problem: From the rest of the data the person might perhaps still be identified → If too much information must be separated, this becomes unusable
  - Very good approach to improve privacy
    - » But depends on how this "add. inf." is stored/who may access it

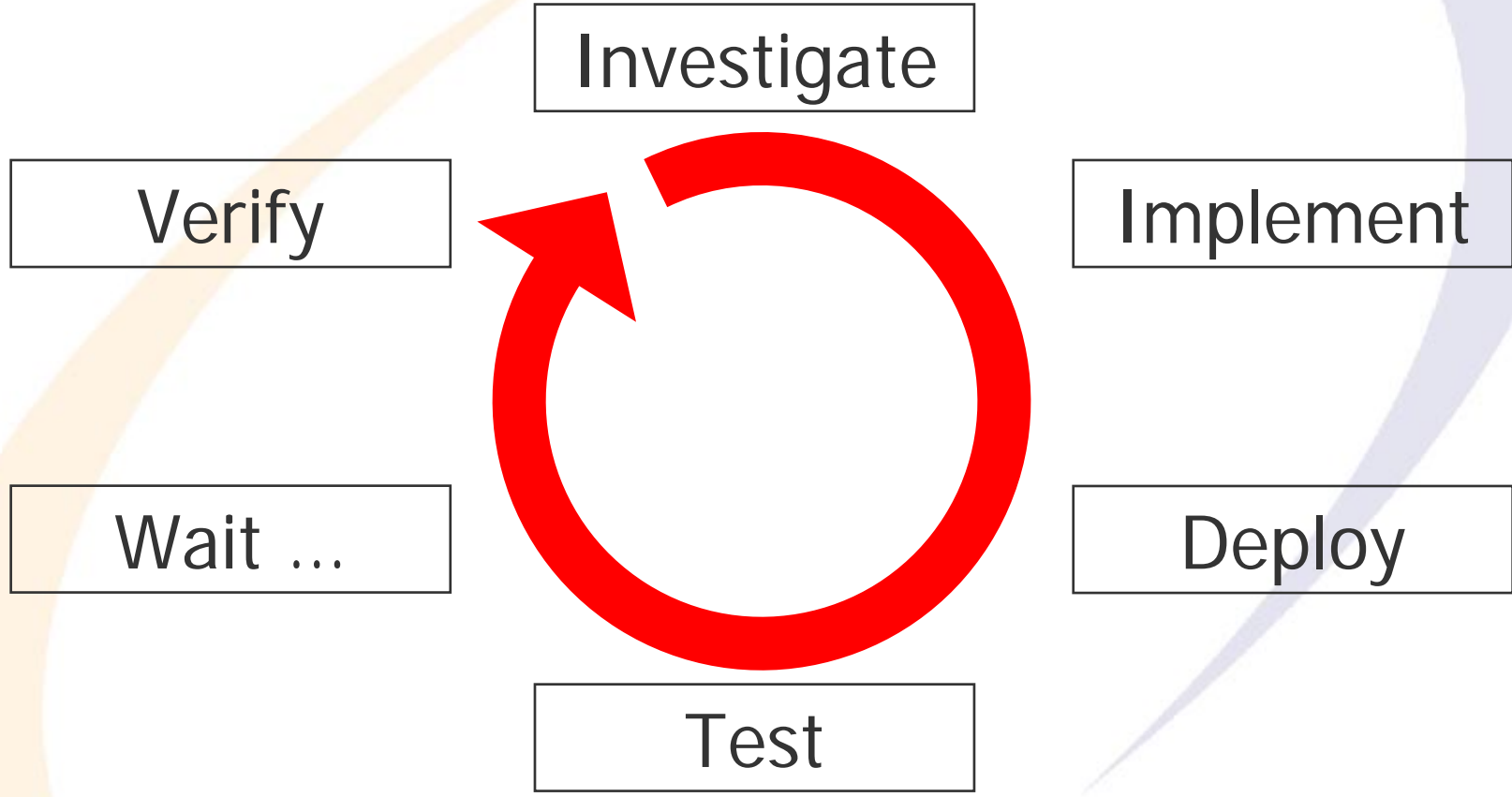


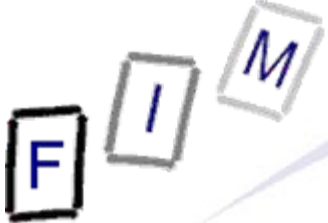
# Approaches to security

- Aims of protection in ICT systems can be reached twofold:
  - Preventive (Proactive): Trying to avoid security incidents by acting before they can take place
    - » Useful especially in the area of safety (security precautions)
      - Random events: Known, measures against them possible
    - + Avoiding the damage
      - Continuous costs, regardless of actual danger
      - Problem of identification of what is "erroneous" or "insecure"
  - Reactive: Handling security incidents after they occurred
    - » Limit consequences, repair damage
    - » Useful especially in the area of security
      - Human ingenuity: Not necessarily predictable
    - » Required for new and surprising attacks
    - + Cost minimization possible
      - No prevention; not necessarily "repairable"
      - » Problem of planning

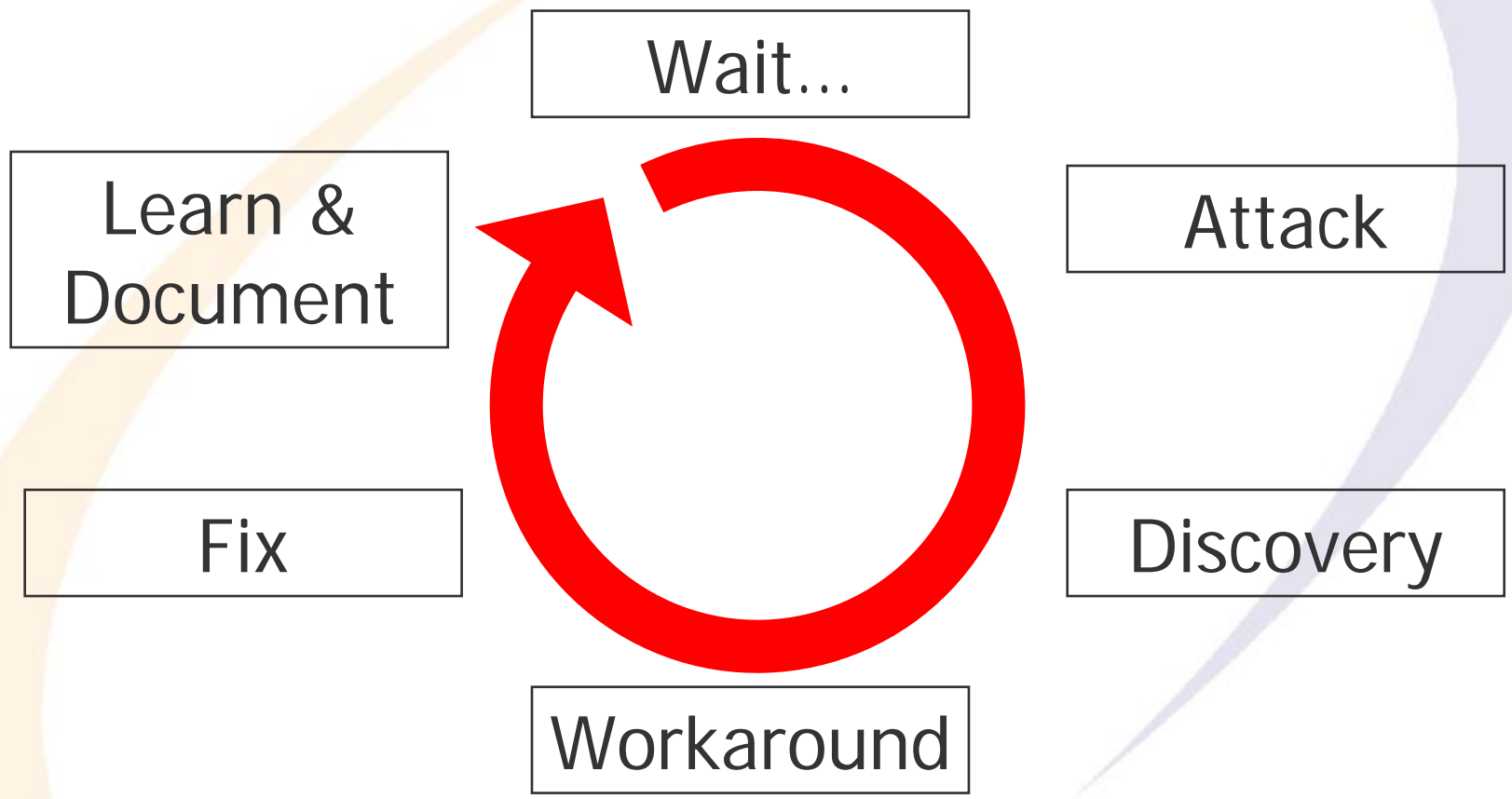


# Proactive security





# Reactive security





# Examples of preventive approaches

- Confidentiality: Encryption, access control, firewall, physical barriers
- Integrity: Access control
- Availability: Redundancy, backups
- Non repudiation: Auditing (logfiles), el. signature, video surveillance, data retention
- Authenticity: Access control
- Ano-/Pseudonymity: Splitting data, deletion
- Main avenues:
  - "Blocking": Checking whether something is "good" and only allowing this to "go through"
  - "Logging": Ensuring everything is documented, so attackers will not try at all
  - "Copying": Having a duplicate to compare/restore



# Examples of reactive approaches

- Confidentiality: Kill-switches, court orders, "hacking back", intrusion detection
- Integrity: Virus scanning, CRC/MAC, restoring backups
- Availability: Renting resources, hacks, restoring backups, intrusion detection
- Non repudiation: - (Computer forensics)
- Authenticity: Tracing actions, investigations
- Privacy: Court orders
- Main avenues:
  - Reactivity can be much more difficult for some areas!
  - "Detection": Try to identify attack as early as possible
  - "Replacement": Obtain a copy/similar entity on short notice
  - "Legal": Use legal system with possibility of physical force
  - "Investigation": Obtain data from various (offline) sources



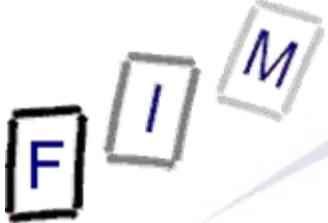
# Preventive or reactive?

- Depends heavily on the risk assessment
  - Will we be attacked regularly, or only accidentally?
  - Are we the aim or are we just a stepping stone?
  - Is the attacker's aim "gain" or "causing damage"?
- Prevention avoids problems at inconvenient times
  - And not everything is "repairable"
    - » See especially confidentiality and the Internet!
- Recovery should become more like a natural strategy than an emergency measure
  - Many zero-day attacks, which cannot reasonably be stopped
  - Some problems are very hard to protect against, but quite easy to cure when needed
    - » You must be prepared, because at some time you will need it!
- "One size fits all" → Usually fits exactly nobody!
  - Be wary of any "patent solutions"!

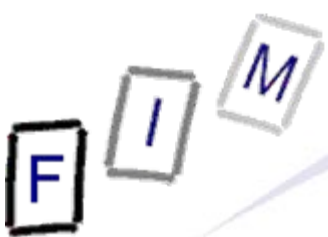


# Dangers and risks

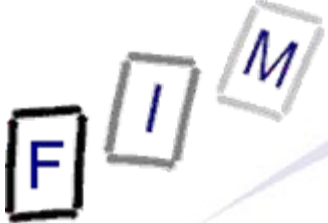
- If you want to protect something, there must be a danger!
  - What kinds of dangers do exist?
- If there is a danger, there should also be an assessment:
  - How likely is this danger?
    - » 1:10<sup>9</sup> or 1:10 ? In which period of time (per second/per year)?
  - What are the consequences of the danger comes true?
    - » In various terms: Money, legal proceedings, bad press, reputation loss, customers/inventors gone, ...
- If a danger has been realized and there is some damage:
  - How to identify the culprit/attacker?
    - » Note: Non-repudiation is required as well!
  - How to obtain evidence?
    - » See computer forensics!
  - How to repair the damage?
    - » In practice often at odds with obtaining evidence!



- Force majeure: You can't influence it at all
  - Lightning strike:
    - » Can be transported over power lines for long distances!
  - Fire: Take care of the extinguishing agent
  - Flooding: Can produce short circuits which "travel" upwards
  - Earthquake: Vibrations + things coming down from above
  - Strike: Power/communication outage
    - » Also note the problems if your own personnel is on strike:  
Regular maintenance might be missing!
- Carelessness: It just happens ... repeatedly
  - Avoiding thinking: Happens to many employees
  - Error: Everyone makes mistakes
  - Incorrect input: Make sure everything is labeled correctly
    - » Computer science: Use "normal" behaviour and handling!
  - Bad handling: Vandalism, anger, ...



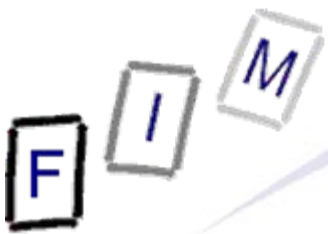
- Technical failure: Nothing ever works perfect
  - Power failure: Use UPS
  - Hardware failure: Replacements may be hard to come by
    - » Guaranteed service? Check reaction time!
  - Malfunctions: Can be hard- or software
    - » Every software has bugs → Appliances often have watchdogs
    - » Both: External monitoring (helps also against other problems!)
- Maliciousness: Someone is out to get you...
  - Hacking: For "fun", but continuously more for "profit"
    - » To acquire servers for fraud and bandwidth for spam
  - Vandalism: Website defacement
  - Espionage: Usually not the secret service, but competitors
    - » Copying files is easier today: Mobile phones, USB sticks, ...
  - Sabotage: Usually by disgruntled (former) employees



- Organisational deficiencies: Everyone tries, but ...
  - Unauthorized access: Missing/erroneous access control
    - » Terminated employees must have their account revoked
    - » Access rights must be fine-granular, even if this causes inconveniences and work for the administration
  - Installing software: Amusing games, helpful software
    - » Especially dangerous: Illegal copies of proprietary software
  - Deficiency in education of employees
    - » Similar to carelessness: Here they just don't know it any better
  - Delays/missing information:
    - » Example: New projects → No share set up → Reuse another

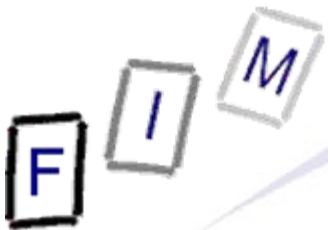


- "Natural enemies":
  - First and most important: The user
    - » "No software survives the first contact with real users"
      - Users will find all possibilities to "mis-"use the software
  - Complexity: (Non-computerized) Steel factories have fewer components than large computer programs!
    - » MS-DOS 1.0: 4000 lines of code
    - » Modern mobile phone: > 300.000 LoC
    - » Windows XP: ≈ 45.000.000 LoC
      - Estimate: 1 security relevant bug per 1000 lines → 45.000 bugs
      - Many are found, but not everyone installs all patches!
  - Speed of development: Testing = At end → Deadline approaching → Reduce testing!
    - » Time-to-market must be very short!
    - » Public "beta test" vs. systematic testing



# Internal vs. external threats

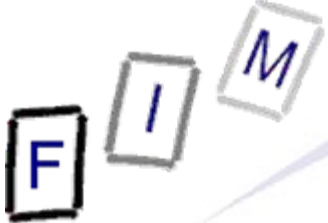
- "The dangerous people are the hackers!"
  - Are they really? 50% - 90% of all attacks come from insiders!
    - » I.e. employees or other persons inside your building/network
      - They know a lot about everything, including security precautions
    - » Example: Copying data you have (and should have!) access to before resigning (or for selling it)
- But also note the level of danger
  - Internal people are rarely specialists, but hackers are
  - Man security incidents from the inside are viruses, worms, ...
    - » These can be protected against comparatively easily, unlike a determined hacker exploiting a specific bug
- Result: Don't forget to protect the inside as well!
  - Firewall → Protection from outside
  - Access control, authentication → Protection from inside
    - » 802.1x, content filtering, IDS, organizational measures, ...



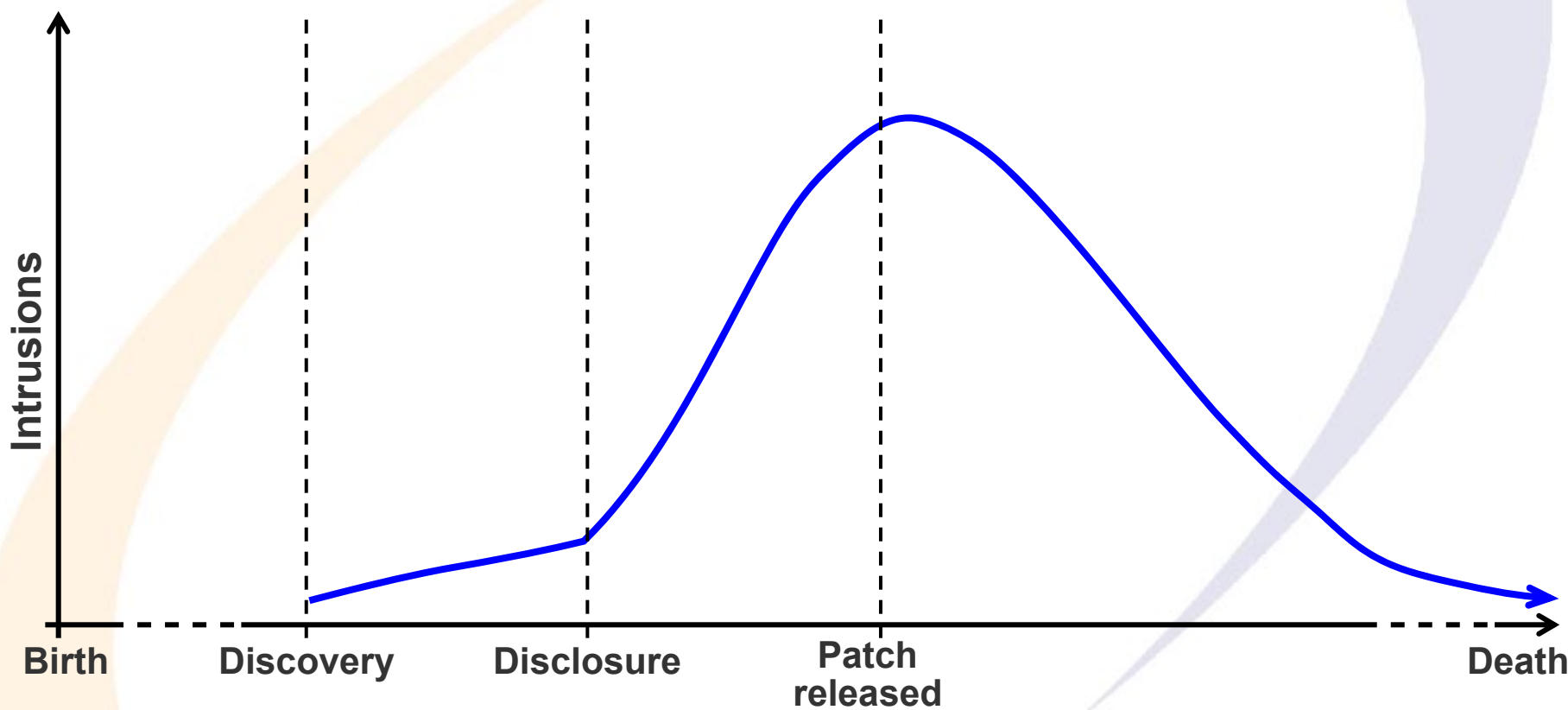
# Security terms: Weakness, vulnerability, exploit

---

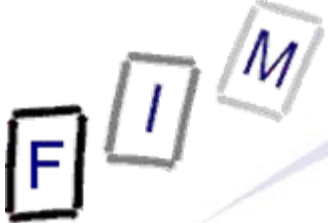
- Weakness: Technical aspect
  - Position in or part of a system where it can be "damaged" (attacked, exploited, ...)
- Vulnerability: Technical problem with security implications
  - Weakness, which can be used to circumvent, deceive or modify security services
- Exploit:
  - Developing an attack using a vulnerability
  - Typically not the theory, but a practical program for this
    - » Software to attain privileges, which should not be granted
  - Note: Exploits are often published widely to "encourage" vendors to fix the vulnerability
    - » But this also means, that the exploit can be readily (ab)used...
    - » "Good" hackers inform the company (set a deadline) and publish the exploit only some time afterwards



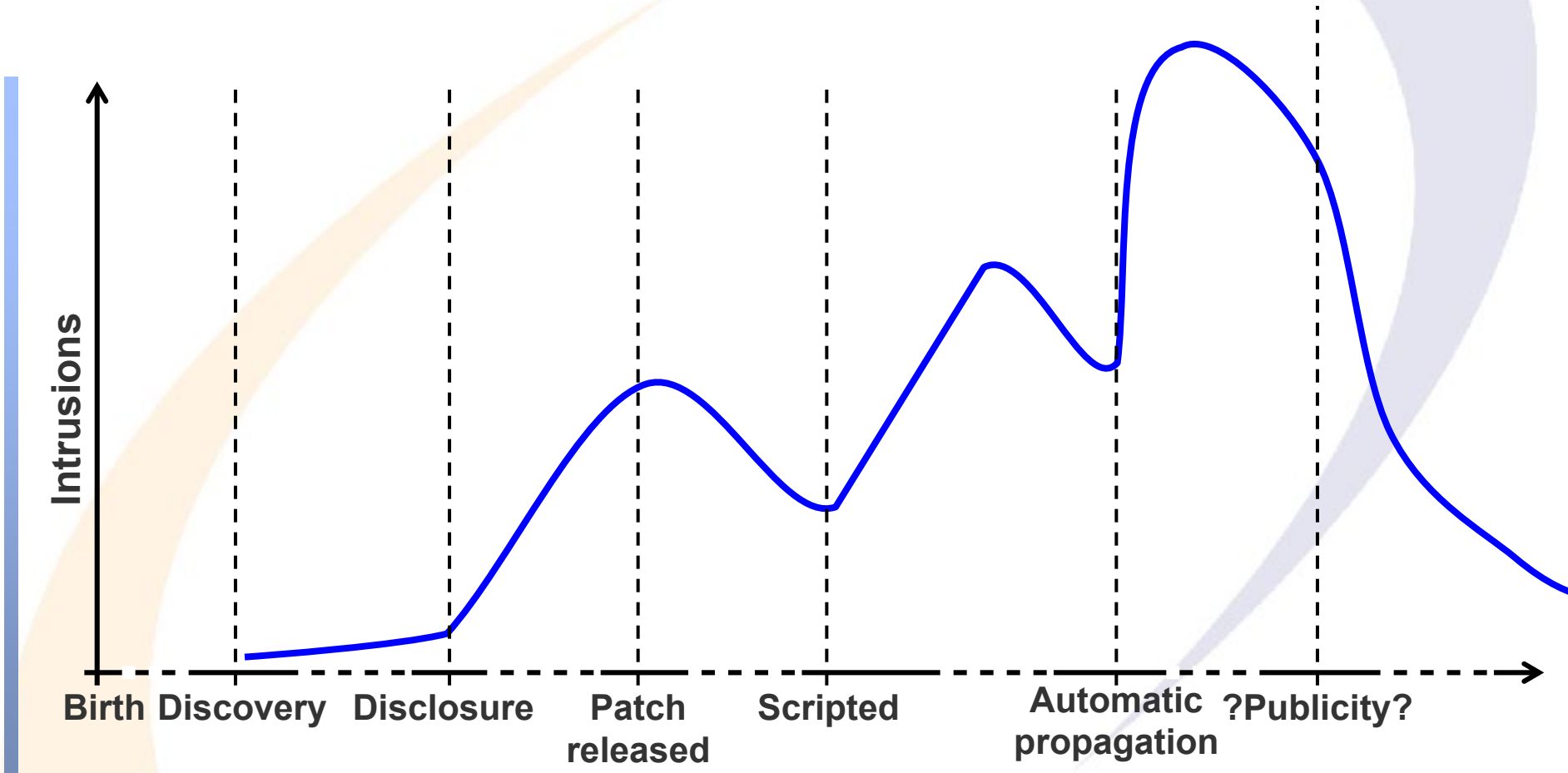
# Vulnerability lifecycle model

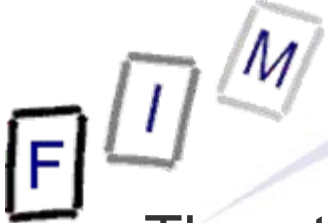


Intuitive life cycle of a system-security vulnerability. Intrusions increase once users discover a vulnerability, and the rate continues to increase until the vendor releases (and system administrators install) a patch or workaround.



# A possible curve ... (as source for discussion)





- Threat: Aims at exploiting a vulnerability to reduce or remove one of the aims of security
  - Attack: A threat is realized and put into motion
    - » Nothing problematic has happened yet
    - » Except perhaps filling the log and alerts!
  - Incident: An attack was successfully completed
    - » Security has been breached
    - » We have not necessarily "lost" anything valuable, but we could not have prevented it
- Risk: Probability of an (undesirable) event happening, times the anticipated damage caused by it (DIN, VDE Norm 31000)
  - » Other definition: „Something that may evolve from an issue to a problem if nothing is done about it“
  - Risk = Probability \* Damage
    - » The probability is very hard to estimate, especially for attacks
    - » The potential damage can also be very difficult to guess



# The security process

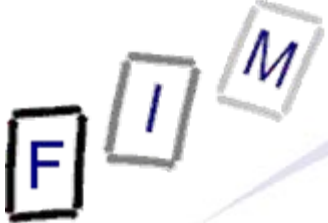
- Security is a continuous process
  - It allows an organization to design, implement and achieve its security aims
  - Identify, measure, manage and control risks
  - It is never complete
    - » Successful attacks are regrettable, but only enhance the importance of the circle!
    - » New developments (vulnerabilities) require new assessments
- The security process is not just an IT task
  - It affects the whole company and needs support by management, accounting, work council, legal etc.!
- Quantification is a major aspect in it
  - and a major problem ...

**What is needed: A security policy (+ strategy, ...)!**

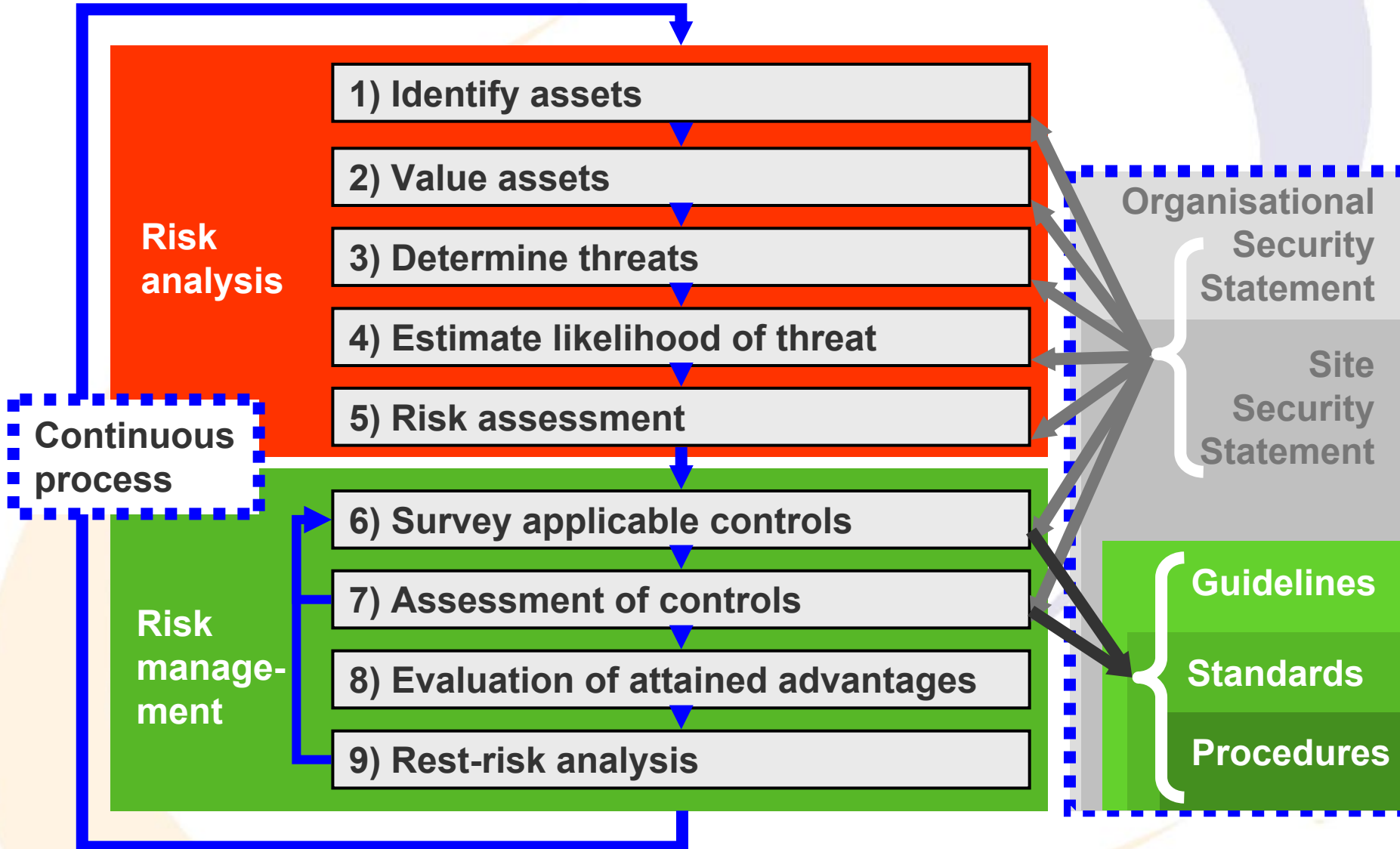


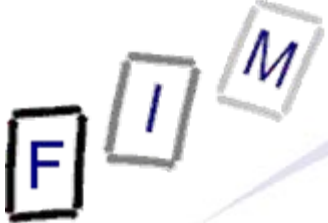
# Main elements of the security process

- **Asset analysis: What do we have that might be attacked?**
  - » Physical system, data contained therein, working state, ...
  - » How "valuable" are they?
    - Expenses, reduced income, "goodwill" etc.
- **Threat analysis: What dangers are possible for the system?**
  - » What may be their sources and reasons?
- **Risk analysis: Determine the risk of an asset**
  - » How probable does the damage occur and how big is it?
  - » Proportion between potential damage and costs for avoiding it?
    - For the decision: Prevention or mitigation (reactive behaviour)
- **Possible countermeasures: What can be done?**
  - » What will it help against?
  - » What will be prevented, what detected, what hindered, what will only help mitigate or remove a damage?
- **Risk coverage: Are all risks addressed by a countermeasure?**
  - » Minimum set of countermeasures (to reduce overlap)?

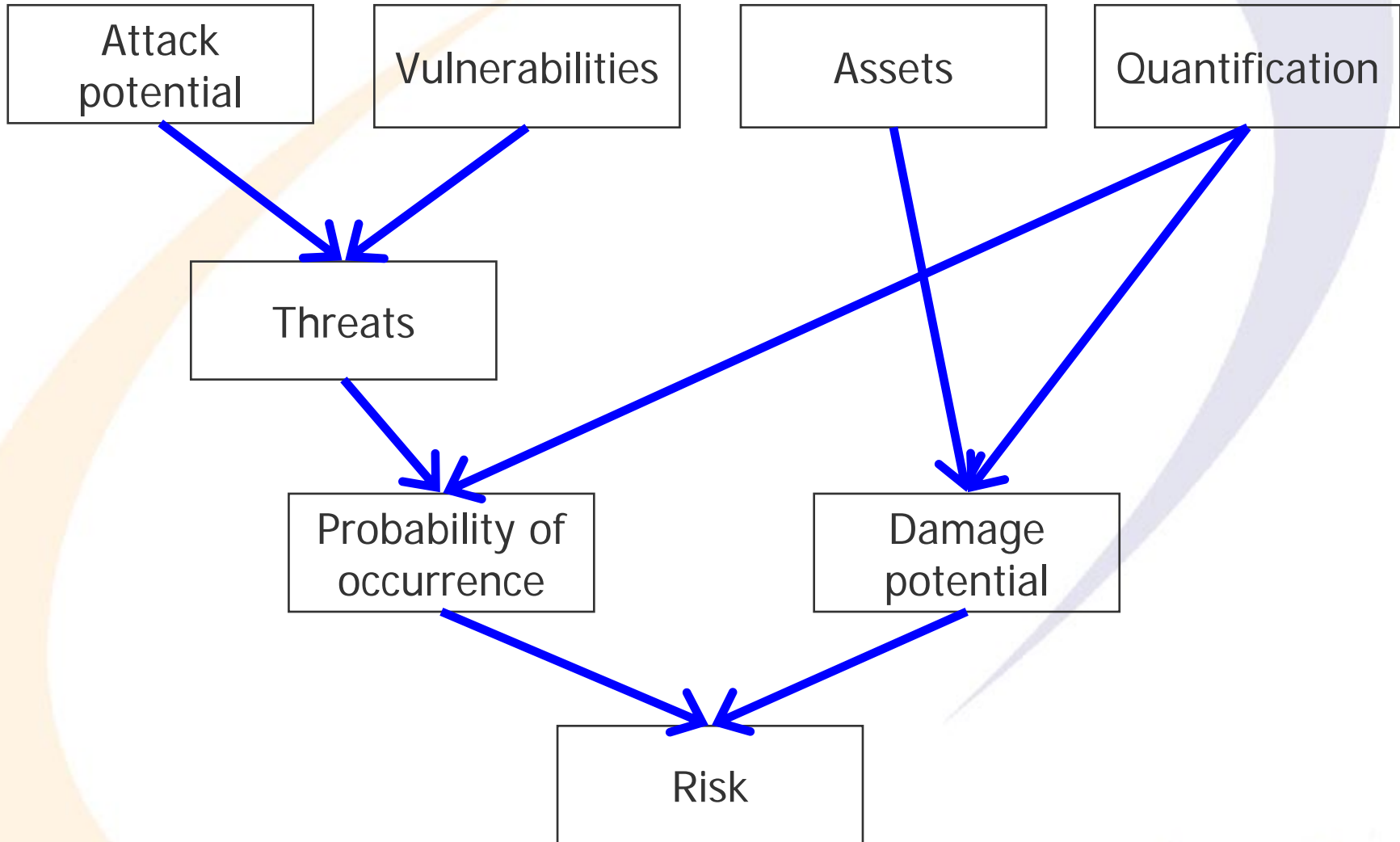


# The security process





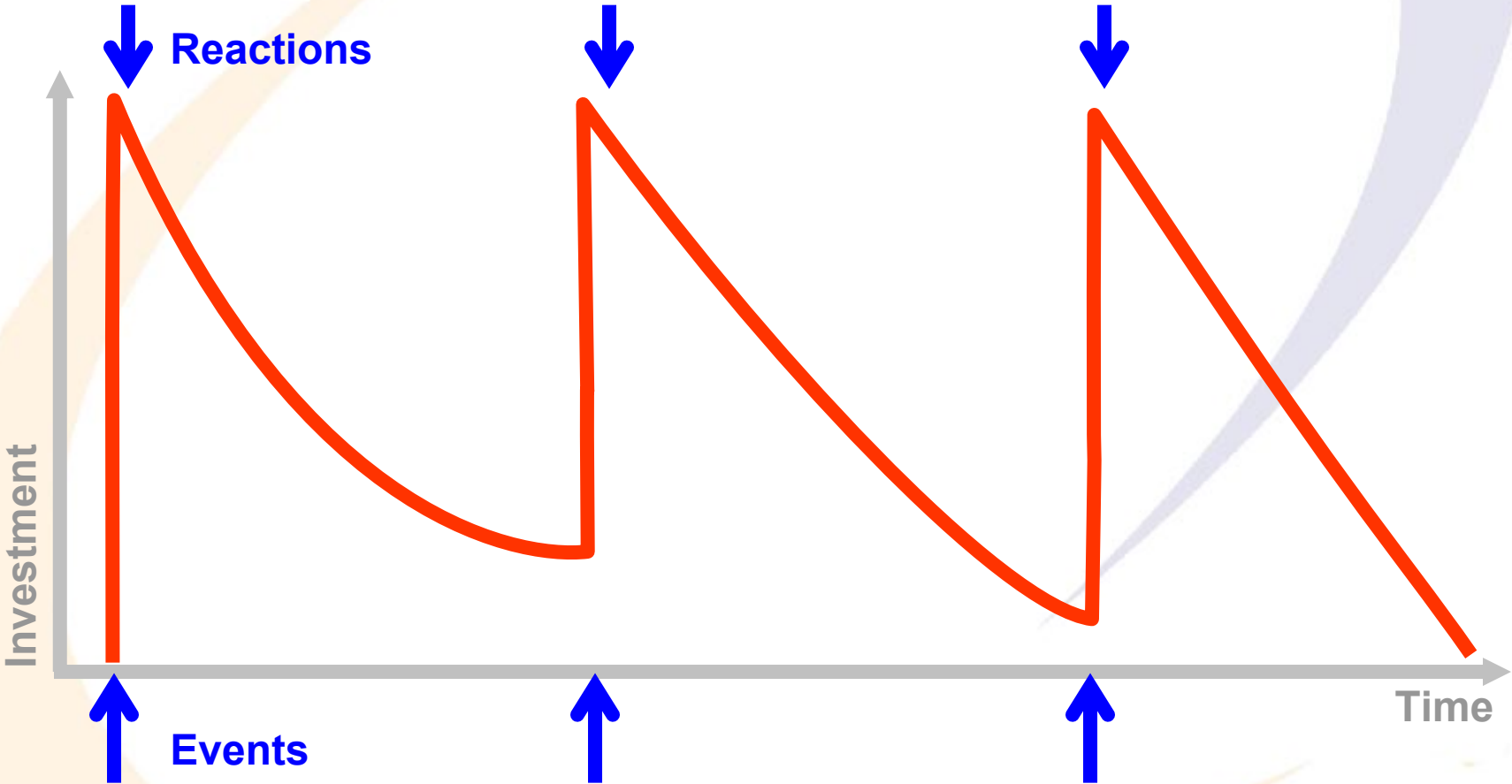
# Vulnerabilities, threats, risk





# Investment only after an incident

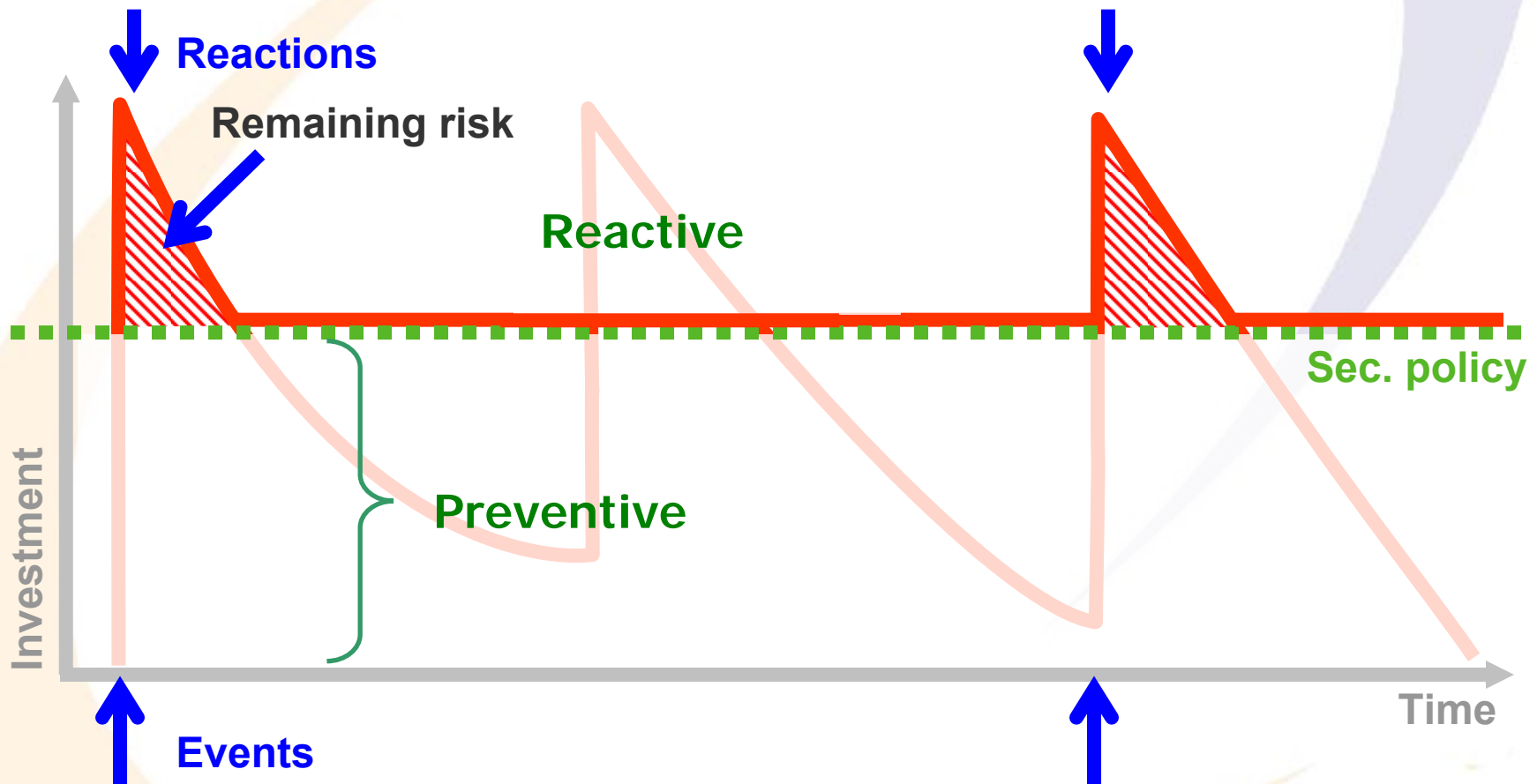
Please: Not like this!!!





# With security policy

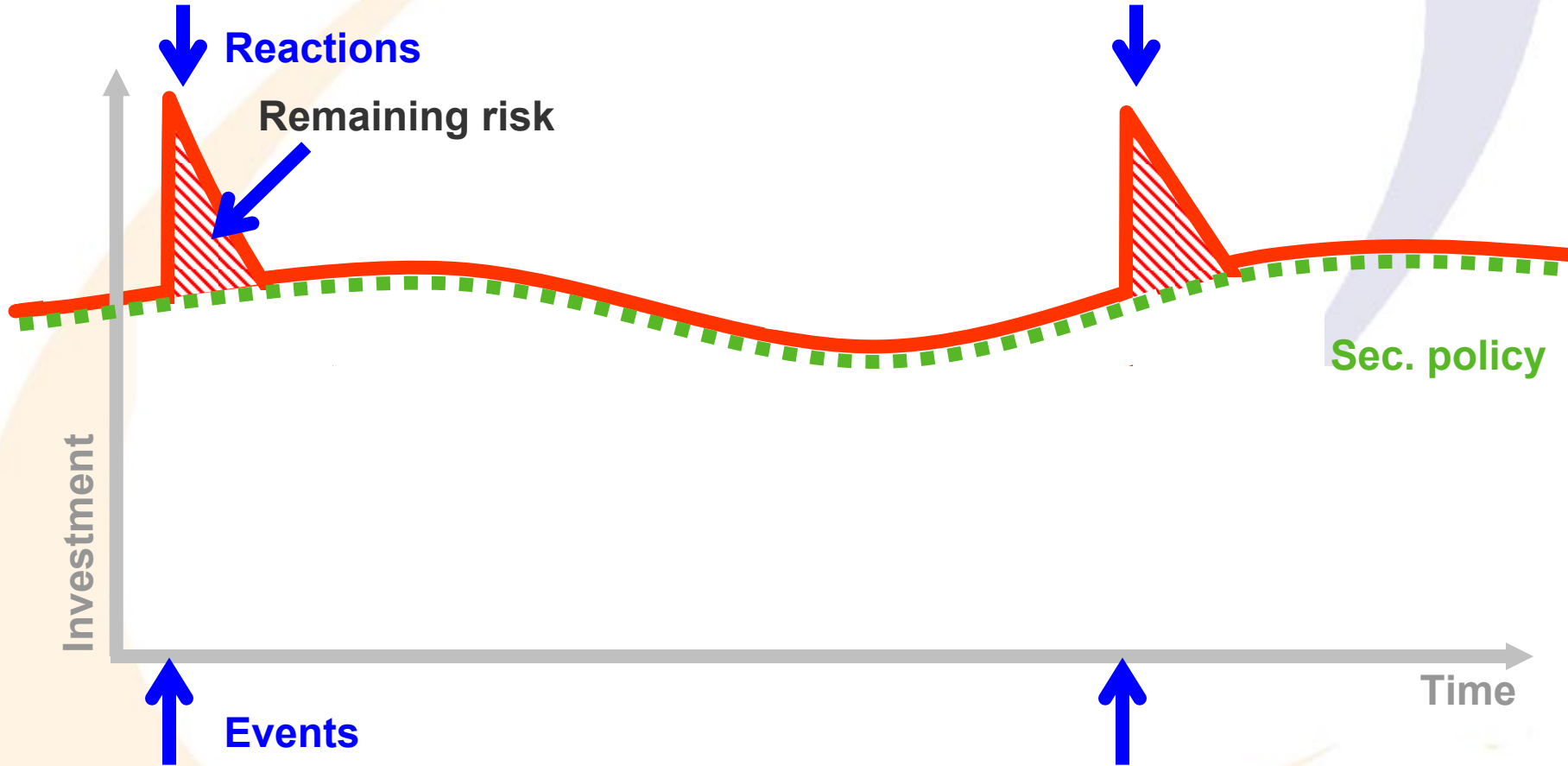
Security policy defines (minimum?) investment





# Modifications of the security policy

Continuous monitoring and modification of the policy is necessary/sensible





- Security is indispensable today
- Aims of security are very diverse
  - Every company must decide on its own, what is important
  - Every software developer should think about security:
    - » Is it possible to secure my program? Or is it inherently insecure?
    - » What are the implications for other programs/the OS?
- Security is not a product, but a process
  - If someone sells you "security" → Be very wary!
    - » What they usually are selling are individual products to prevent incidents or react to them!
  - Continuous monitoring and improvement required
    - » The attackers at least **will** do this ...

**Security is more than encryption and a firewall!**

F I M

# Questions?

Thank you for your attention!

F I M

Děkuji za Vaši pozornost