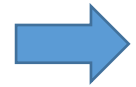


Security concepts

Frank Piessens

Overview



- Basic security concepts
 - Definitions
 - Examples
- Case study: e-mail security
- Conclusions

Cyber security / computer security

- The art, science and engineering of protecting computer-related **assets**
- Such assets include:
 - Data, information
 - Computer hardware, software or services
 - Electronic communication
 - Computer-controlled physical world devices
 - ...
- We discuss the main security properties that computer systems can provide to protect these assets (the **fundamental goals of computer security**)

Security goals

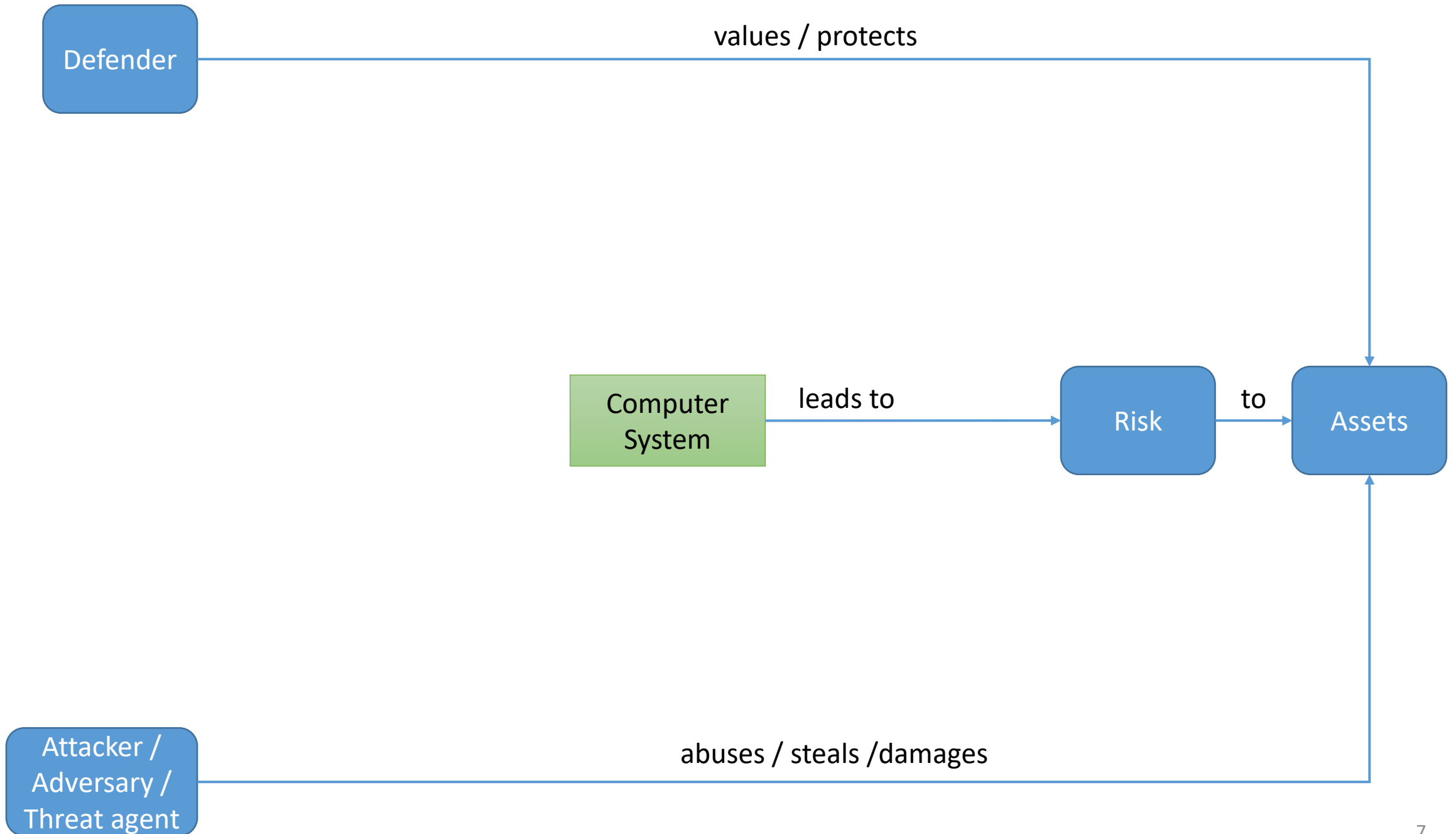
- **Confidentiality**: the property that information is only accessible to specific authorized parties
- **Integrity**: the property that data, software or hardware can only be modified by specific authorized parties
- **Authorization**: the property that services, communication or devices are only accessible by specific authorized parties
- **Availability**: the property that assets *remain* accessible for authorized use

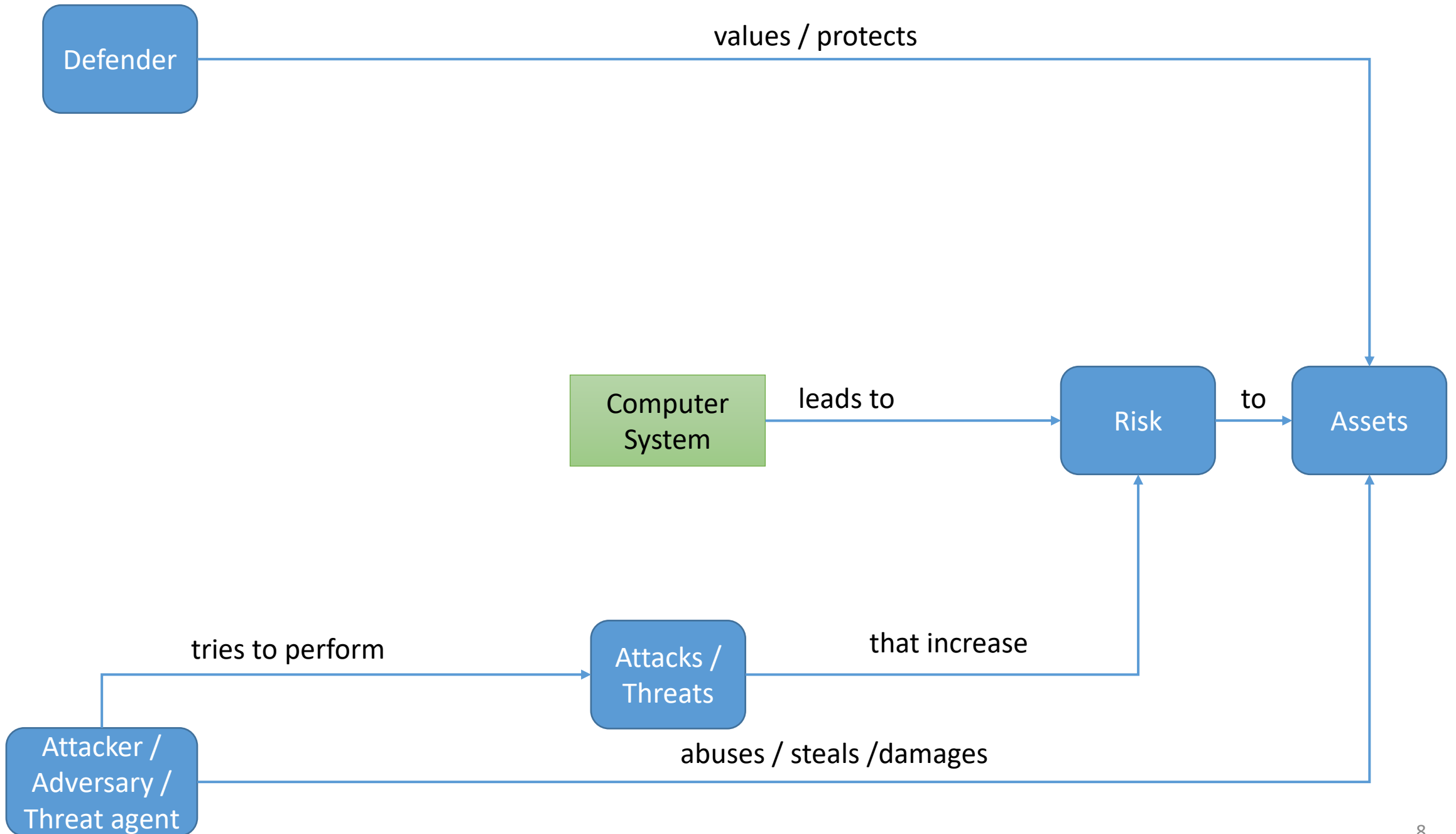
Security goals (continued)

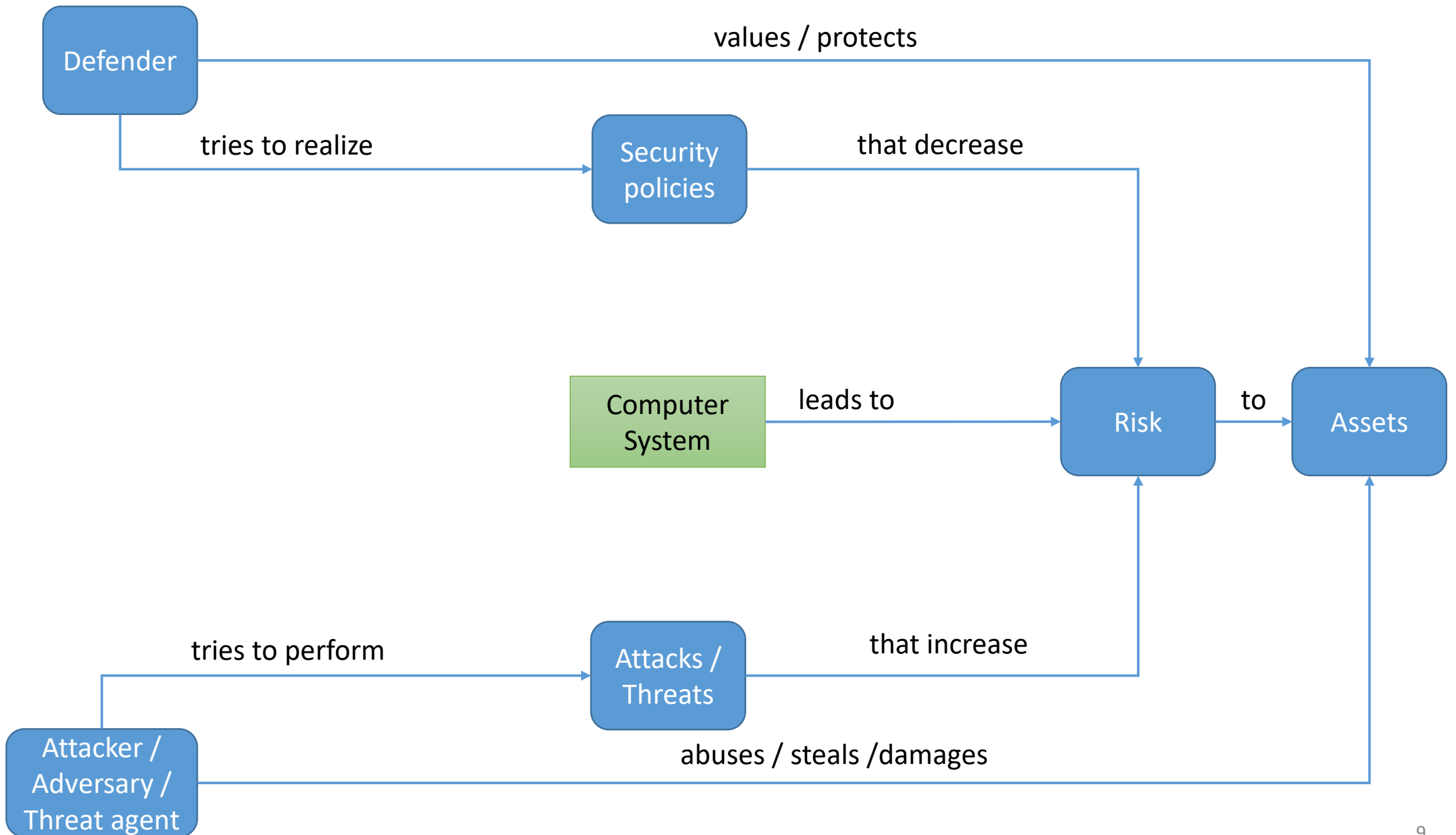
- The security goals on the previous slide often require verified identities of people, processes or devices (jointly called **principals**)
- **Authentication**: assurance that an asserted or expected identity is valid
 - Entity authentication: assurance that an asserted identity in a transaction is valid
 - Data origin authentication: assurance that the source of data or software is as asserted or as expected
- **Accountability**: the property that past actions can be correctly attributed to specific principals

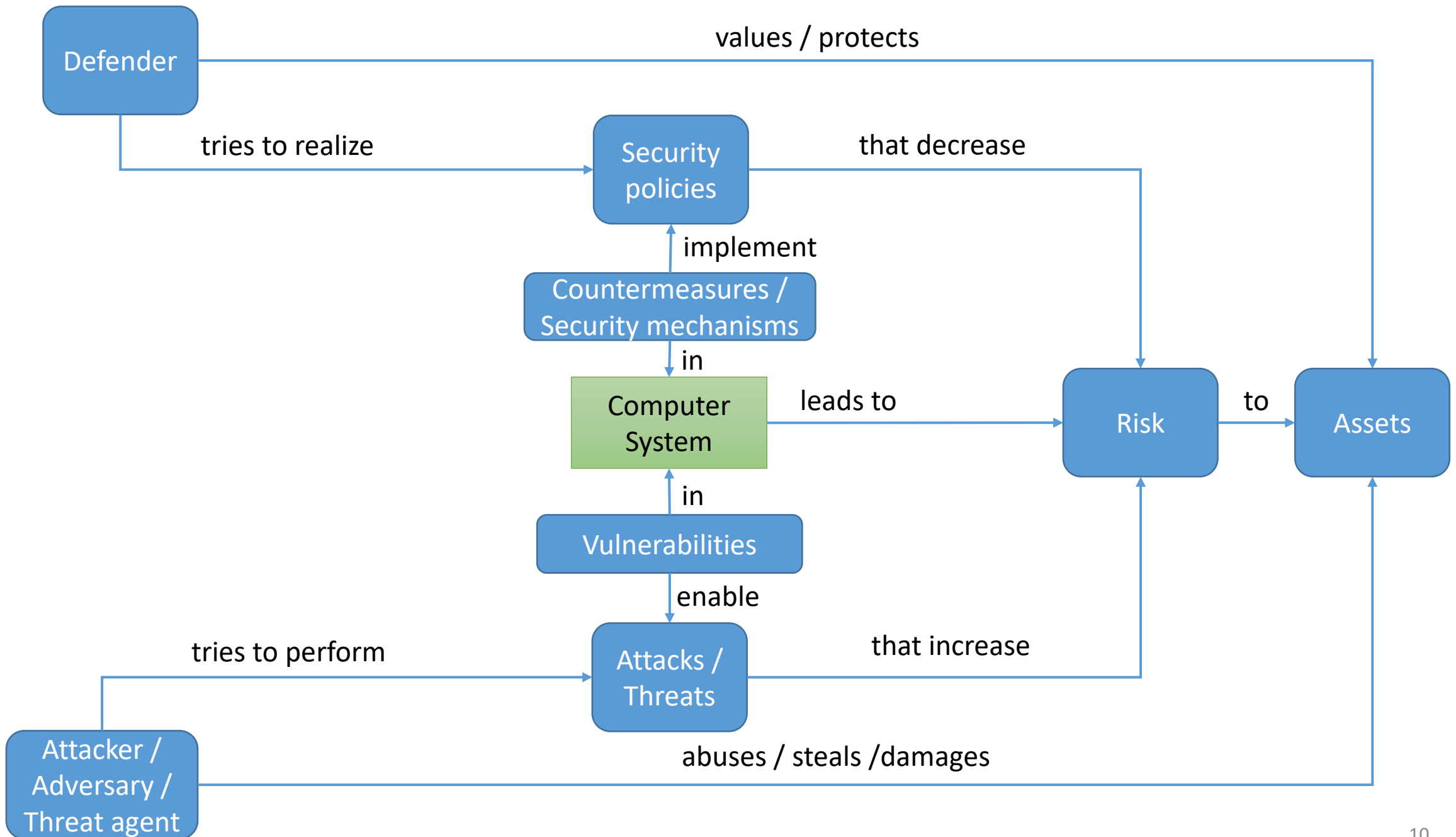
Intelligent adversaries

- Cybersecurity is about how computer systems can achieve these security goals in the presence of **intelligent adversaries**
 - This is related to but harder than achieving these goals in the presence of accidents, random errors or non-malicious mistakes
- For real-world systems, full protection is impossible to achieve: in practice cybersecurity is about managing **risk**, and balancing costs and benefits of additional protection









Key concept: vulnerability

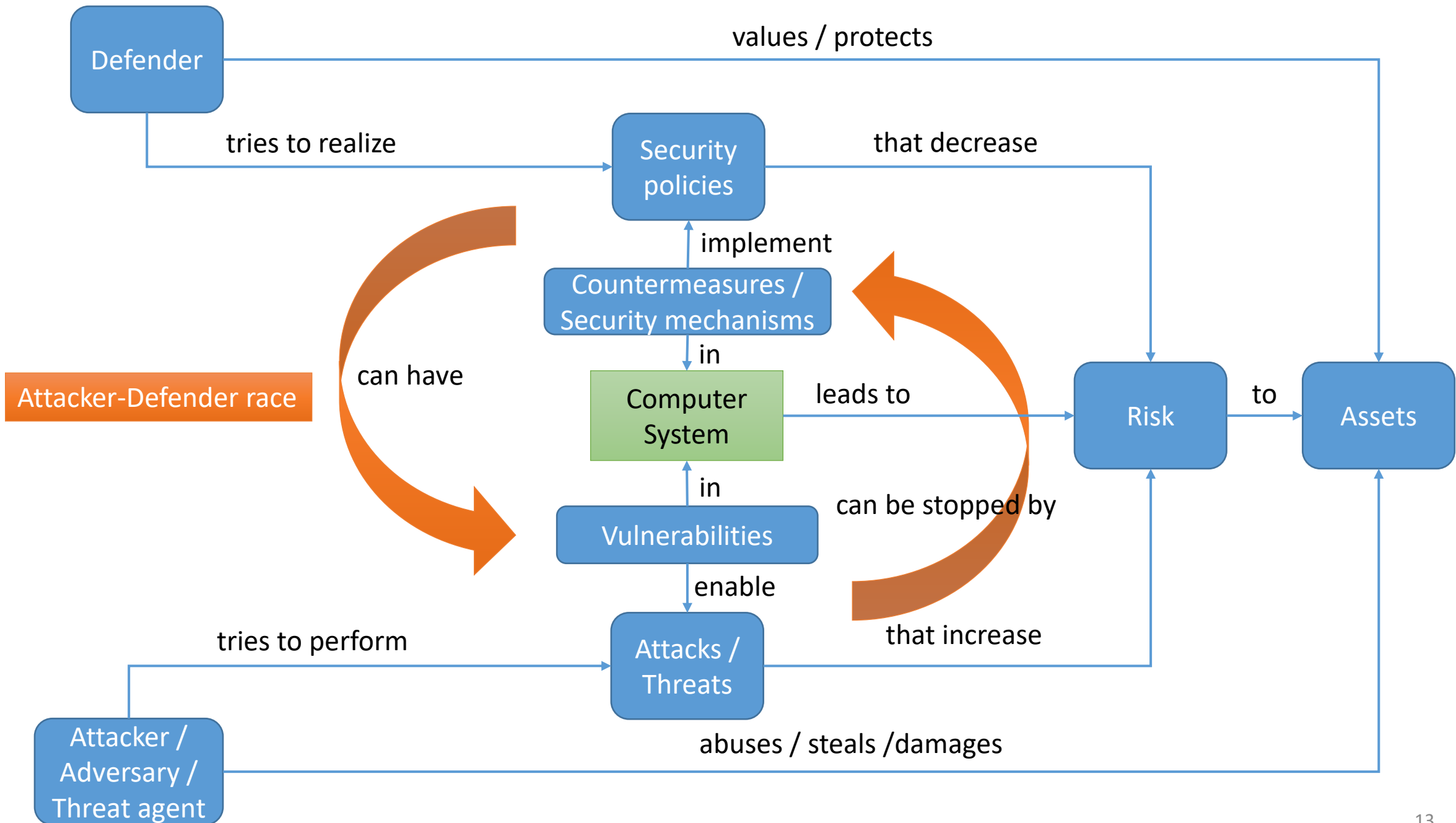
A **vulnerability** is an aspect of the system that allows an attacker to break a security policy.

- Examples:
 - A memory management vulnerability like Heartbleed
 - A structured output generation vulnerability enabling SQL injection
 - The absence of access control for system administrators
 - The incorrect configuration of an access control policy
- Vulnerabilities can be introduced in all phases of the development life-cycle.
- Vulnerabilities are relative to an attacker model

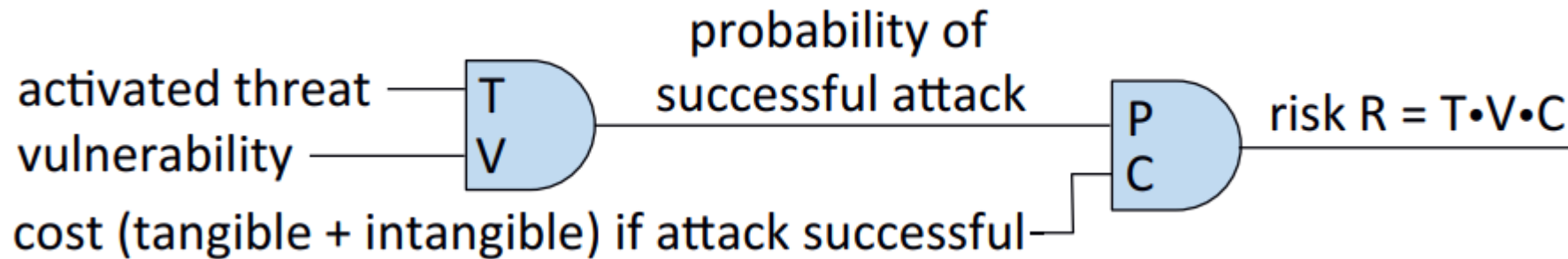
Key concept: countermeasure

A **countermeasure** (or a **security mechanism**) is a technique used to counter a threat or realize a security policy.

- Examples:
 - The use of encryption to counter eavesdropping
 - The use of Java instead of C to avoid memory-management vulnerabilities
 - Providing good documentation to avoid configuration errors
 - The use of specific testing techniques to detect vulnerabilities
 - The installation of a firewall to reduce attack surface
- Countermeasures can be *preventive, detective, or reactive*
- Countermeasures can be taken by different stakeholders: software developer, system administrator, system user, ...



Risk and risk assessment



(Source: Paul Van Oorschot, Computer Security and the Internet)

- Risk assessment helps in:
 - balancing the costs and benefits of the deployment of specific security countermeasures
 - ranking the priority of security countermeasures
- But precise risk assessment is hard in security, estimating T, V and C is difficult if not impossible

Adversary modeling

- Understanding the adversaries that you are facing helps making better security decisions
- What group of adversaries:
 - Nation-state intelligence services, hacktivists, cybercriminals, industrial espionage agents, ...
- What do you know about them in terms of:
 - Objectives
 - Methods
 - Capabilities and funding level
 - ...

Security analysis

- A rigorous analysis of the security of a system requires three ingredients:
 - **A system model:**
 - a rigorous description of the system we are trying to secure.
 - One uses different system models to study for instance the security of web applications or the security of digital documents.
 - **A security policy:**
 - what good properties of the system do we want to maintain?
 - Or what bad things do we want to prevent?
 - **A threat model / attack model:**
 - a precise definition of the power of the threat agents and threats in scope.
 - It can make sense to study security of the same system under different attack models.

Security policies

The **security policy** (synonym: security objective) specifies the good properties of a system that must be maintained.

- Examples:
 - A users posts on Facebook should only be visible to his friends
 - The website should be available 99.9% of the time
 - Players can not cheat in this online game
 - Students should not be able to change their marks
 - Users can only install software approved by Apple
- Security policies are often instantiations of security goals (like **Confidentiality, Integrity** or **Availability**) for specific **assets**

Where do security policies come from?

- From business considerations:
 - “The blueprints of our new product should only be accessible to engineers working on the product”
 - “Prices of products can only be set by the store administrator”
- From legal considerations:
 - “Personal data must be protected against unauthorized processing, and against accidental loss, destruction or damage”
- From technical considerations:
 - “Cryptographic keys can only be accessed by the smartcard, and can only be installed by an authorized representative”

Where do security policies come from?

CONFIDENTIALITY

- From business considerations:
 - “The blueprints of our new product should only be accessible to engineers working on the product”
 - “Prices of products can only be set by the store administrator”
- From legal considerations:
 - “Personal data must be protected against unauthorized processing, and against accidental loss, destruction or damage”
- From technical considerations:
 - “Cryptographic keys can only be accessed by the smartcard, and can only be installed by an authorized representative”

Where do security policies come from?

INTEGRITY

- From business considerations:
 - “The blueprints of our new product should only be accessible to engineers working on the product”
 - “Prices of products can only be set by the store administrator”
- From legal considerations:
 - “Personal data must be protected against unauthorized processing, and against accidental loss, destruction or damage”
- From technical considerations:
 - “Cryptographic keys can only be accessed by the smartcard, and can only be installed by an authorized representative”

Where do security policies come from?

AVAILABILITY

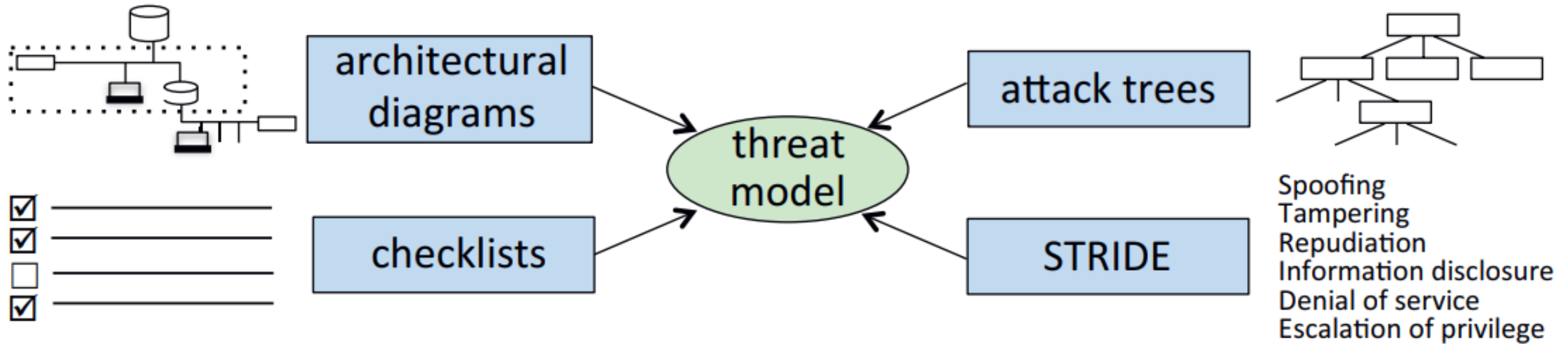
- From business considerations:
 - “The blueprints of our new product should only be accessible to engineers working on the product”
 - “Prices of products can only be set by the store administrator”
- From legal considerations:
 - “Personal data must be protected against unauthorized processing, and against accidental loss, destruction or damage”
- From technical considerations:
 - “Cryptographic keys can only be accessed by the smartcard, and can only be installed by an authorized representative”

Threat model

A **threat model** (synonym: attack model) defines the kinds of attacks and attack techniques that are in scope, the power / resources / capabilities of the attackers, and possible attack objectives.

- Examples:
 - The attacker can observe all the network traffic
 - The attacker can observe and alter all the network traffic
 - The attacker has no physical access to the web server
 - The attacker can not forge digital signatures
- An important point of the model is to make explicit the assumptions of what an attacker can **not** do

Threat modeling approaches

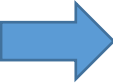


(Source: Paul Van Oorschot, Computer Security and the Internet)

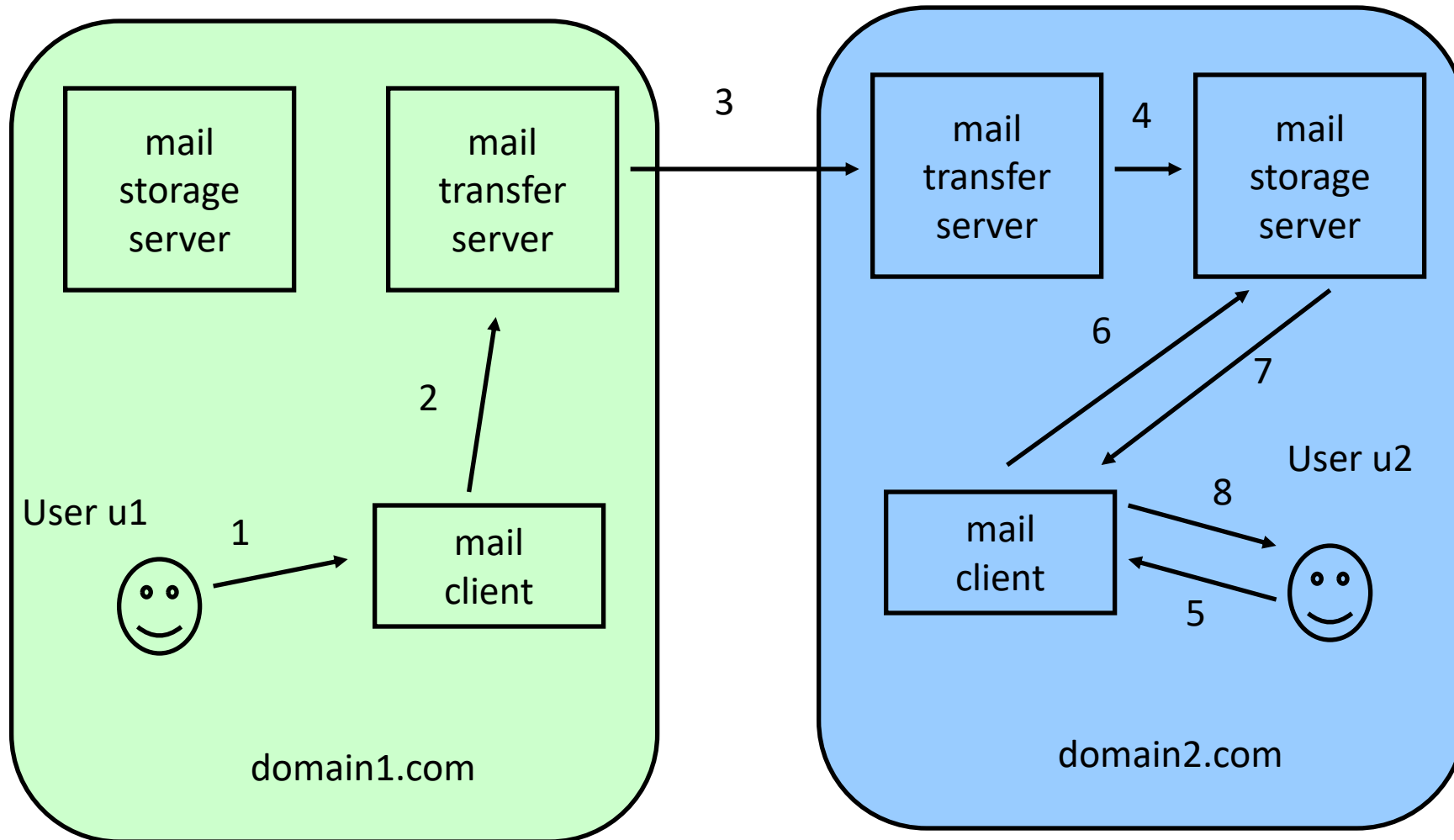
Important observations

- The same system can be studied for different security objectives and under different attacker models
- System models and attacker models can evolve as we learn about more realistic ways to attack the system
- Security objectives can evolve as we learn about more ways to abuse a system

Overview

- Basic security concepts
 - Definitions
 - Examples
-  • Case study: e-mail security
- Conclusions

Case study: Simplified e-mail system



Discussion

- What are the assets at stake?
- What adversaries should we consider?
- What are relevant security policies for this system?
- What are relevant attacks / threats for this system?
- What security mechanisms could be used to enforce the policies?
- What vulnerabilities could be exploited to launch the attacks?

Overview

- Basic security concepts
 - Definitions
 - Examples
- Case study: e-mail security
- • Conclusions

Conclusions

- Rigorous study of security centers around adequate system and attacker models, and:
 - Either an understanding of the security objectives,
 - Or an understanding of security failures to avoid
- Attack research: given a system, find new relevant attacks?
 - By finding vulnerabilities
 - By showing the attacker model is too limiting in practice
 - By showing the security objective is insufficient
- Defense research: achieve security objectives under a given attacker model
 - Implement well-understood security mechanisms or come up with new ones
 - Build a **security argument** for the system
 - What design or implementation activities lead to more secure systems?
- Mandatory reading:
 - Paul Van Oorschot, Computer Security and the Internet (2nd edition), Chapter 1, Sections 1.1-1.6
 - <https://people.scs.carleton.ca/~paulv/toolsjewels/TJrev1/ch1-rev1.pdf>