# Secure Software Design in Practice

## ARES – SECSE Workshop

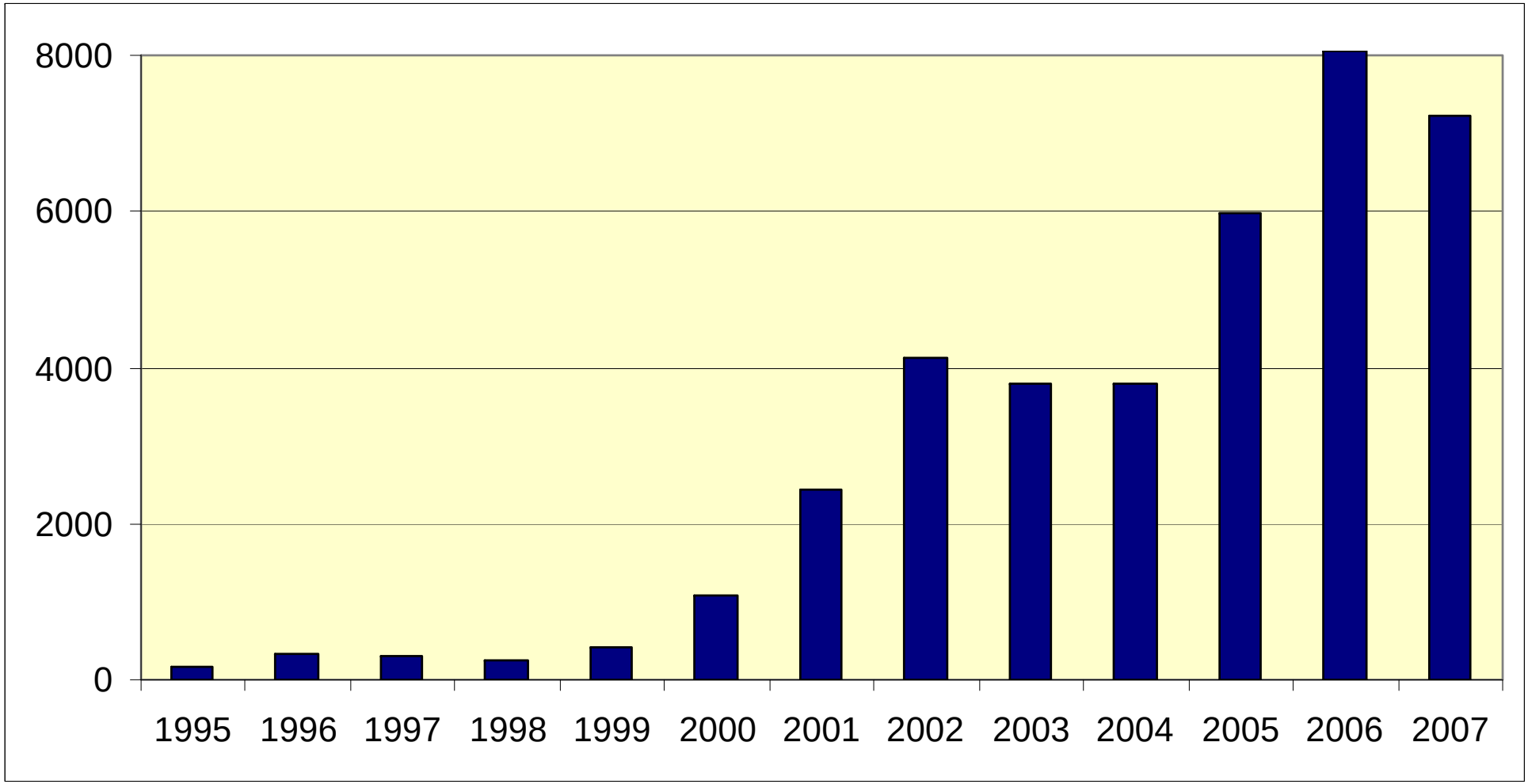
**Per Håkon Meland** and Jostein Jensen

*SINTEF Information and Communication Technology*

*Department of Security, Safety and System Development*

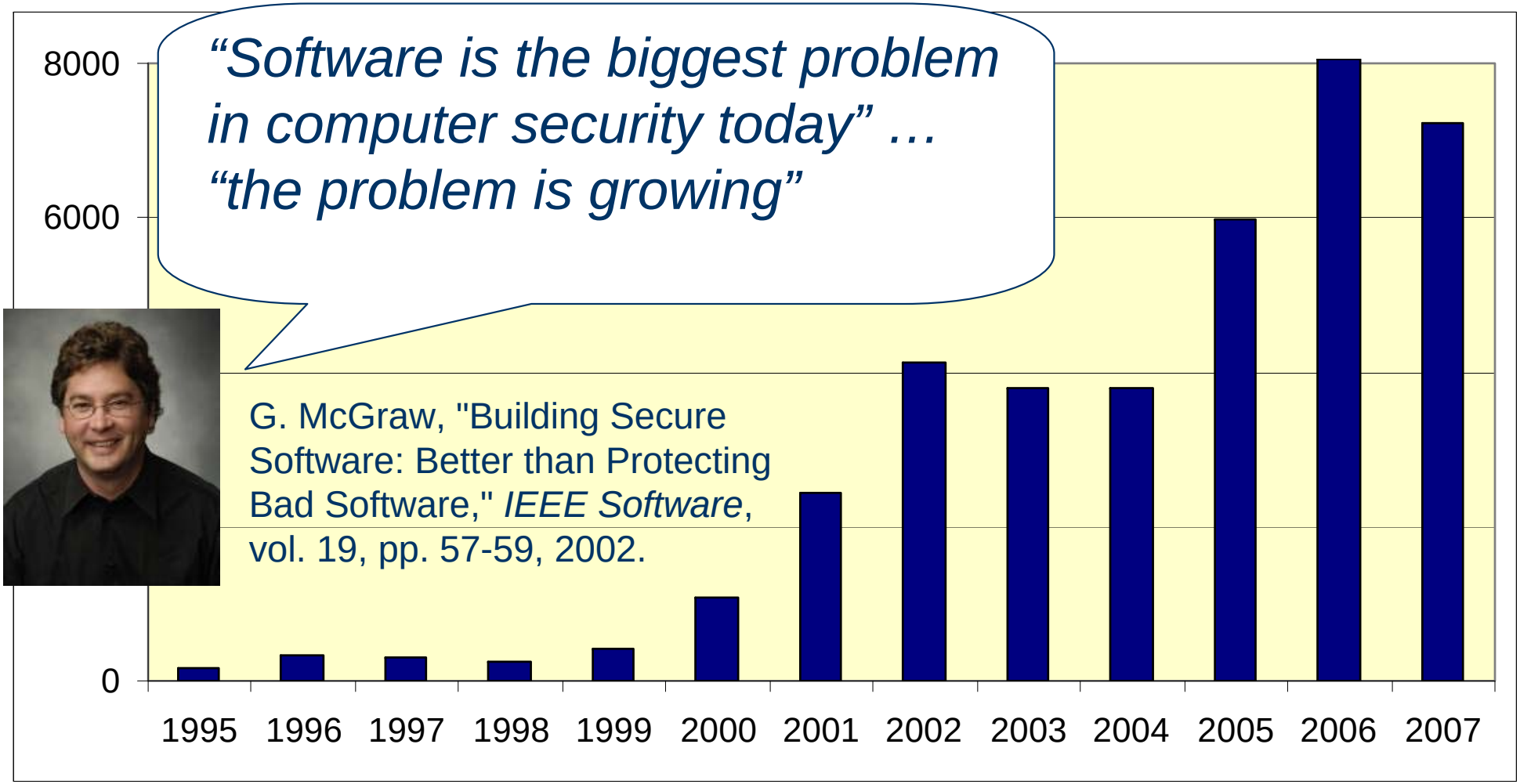
*{Per.H.Meland, Jostein.Jensen}@sintef.no*

# Increasing number of vulnerabilities in software



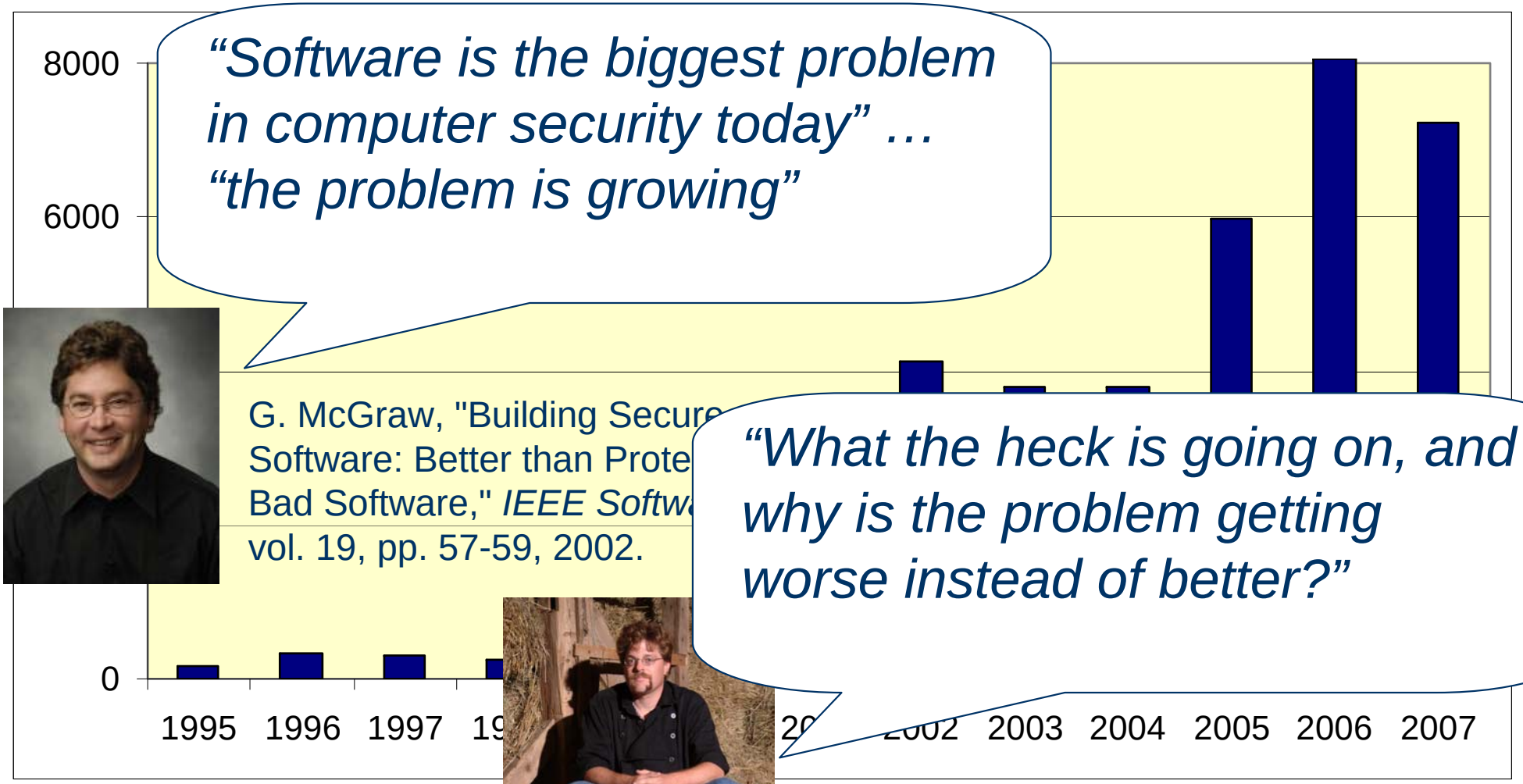

Vulnerability statistics
from CERT CC

# Increasing number of vulnerabilities in software




Vulnerability statistics
from CERT CC

# Increasing number of vulnerabilities in software




*"Software is the biggest problem in computer security today" …*
*"the problem is growing"*



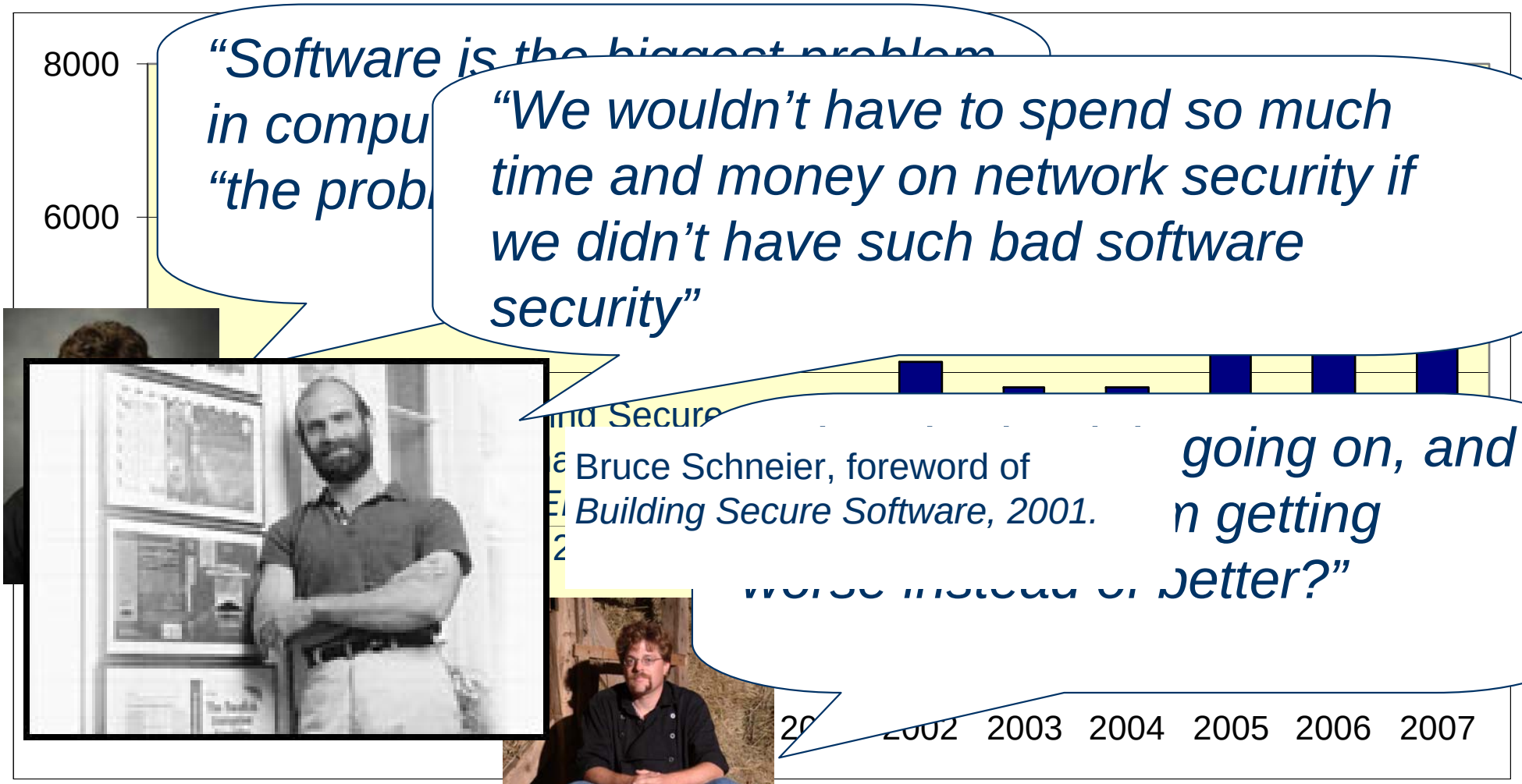
G. McGraw, "Building Secure Software: Better than Prote… Bad Software," *IEEE Softwa…* vol. 19, pp. 57-59, 2002.

*"What the heck is going on, and why is the problem getting worse instead of better?"*

Vulnerability statistics from CERT CC

M. J. Ranum, "Security: The root of the problem," in *ACM QUEUE*, vol. 2, 2004, pp. 45-49.

# Increasing number of vulnerabilities in software




*"Software is the biggest problem in compu… "the prob…*

*"We wouldn't have to spend so much time and money on network security if we didn't have such bad software security"*

Bruce Schneier, foreword of *Building Secure Software, 2001.*

*…going on, and …n getting …better?"*

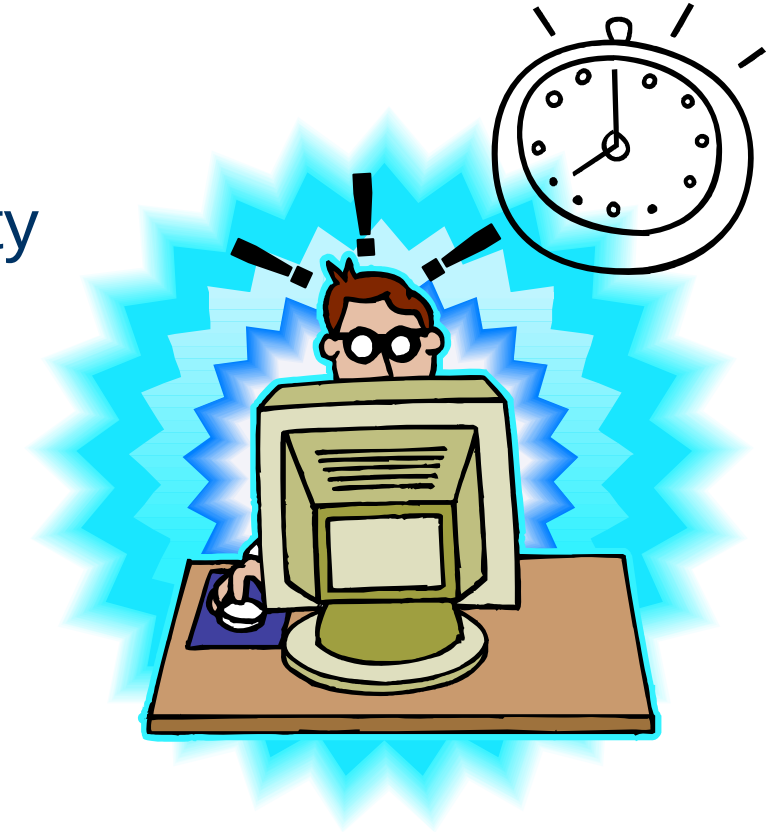


Vulnerability statistics from CERT CC

M. J. Ranum, "Security: The root of the problem," in *ACM QUEUE*, vol. 2, 2004, pp. 45-49.

# SODA - a Security-Oriented Software Development Framework

- The target group is the ordinary ”developer-on-the-street”
  - not primarily interested in (or knowledgeable about) security
  - must focus on designing/ implementing as much functionality as possible before the deadline and on budget.

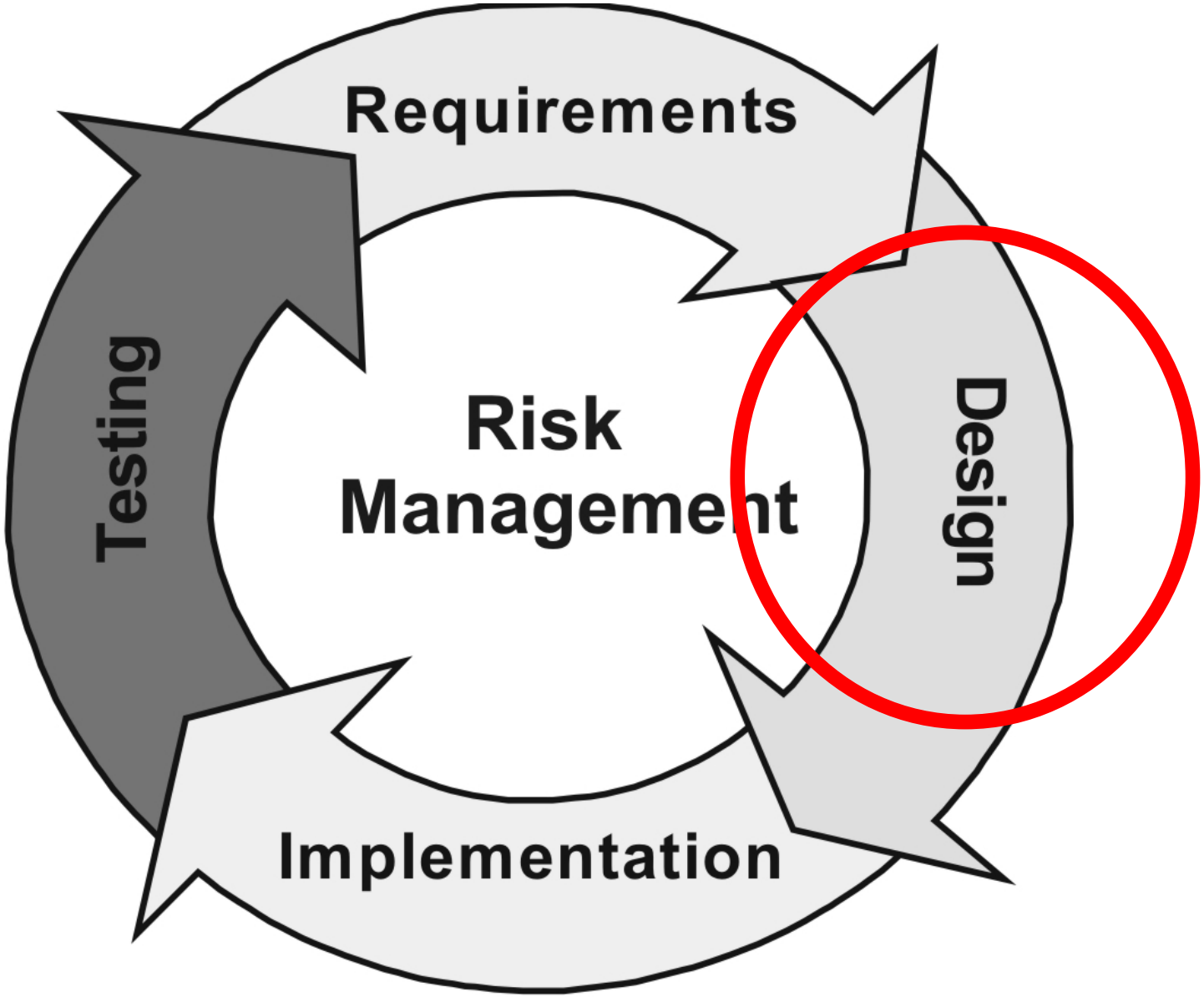


# The SODA assumptions

1. A developer will not try to learn or memorize security knowledge prior to starting the development.
2. There should be no significant change in the way developers work.
3. There must be good tool support that enhances security during development, preferably integrated into the current development tools.


Requirements
Design
Implementation
Testing
Risk
Management

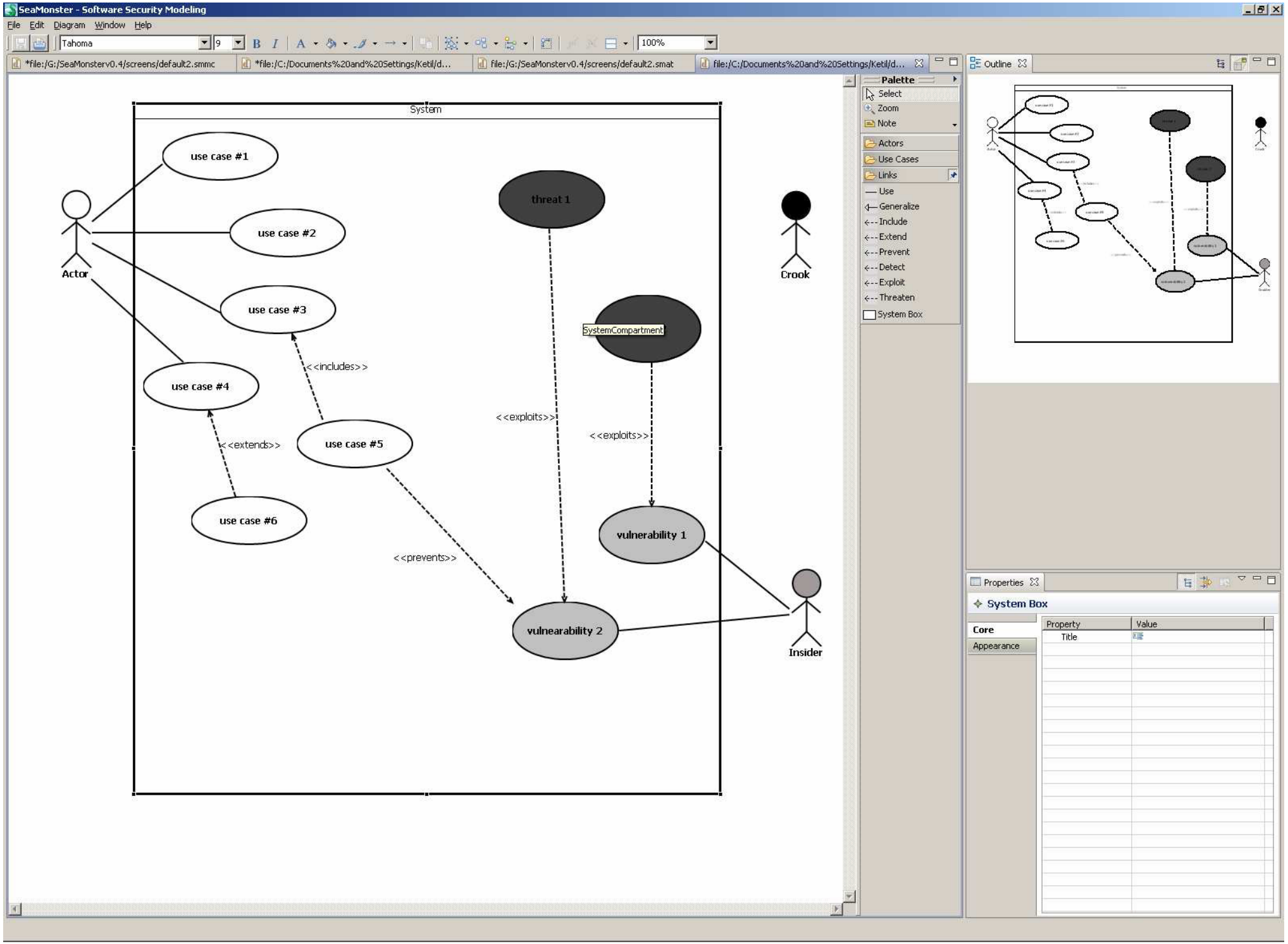
# SODA during architecture and design

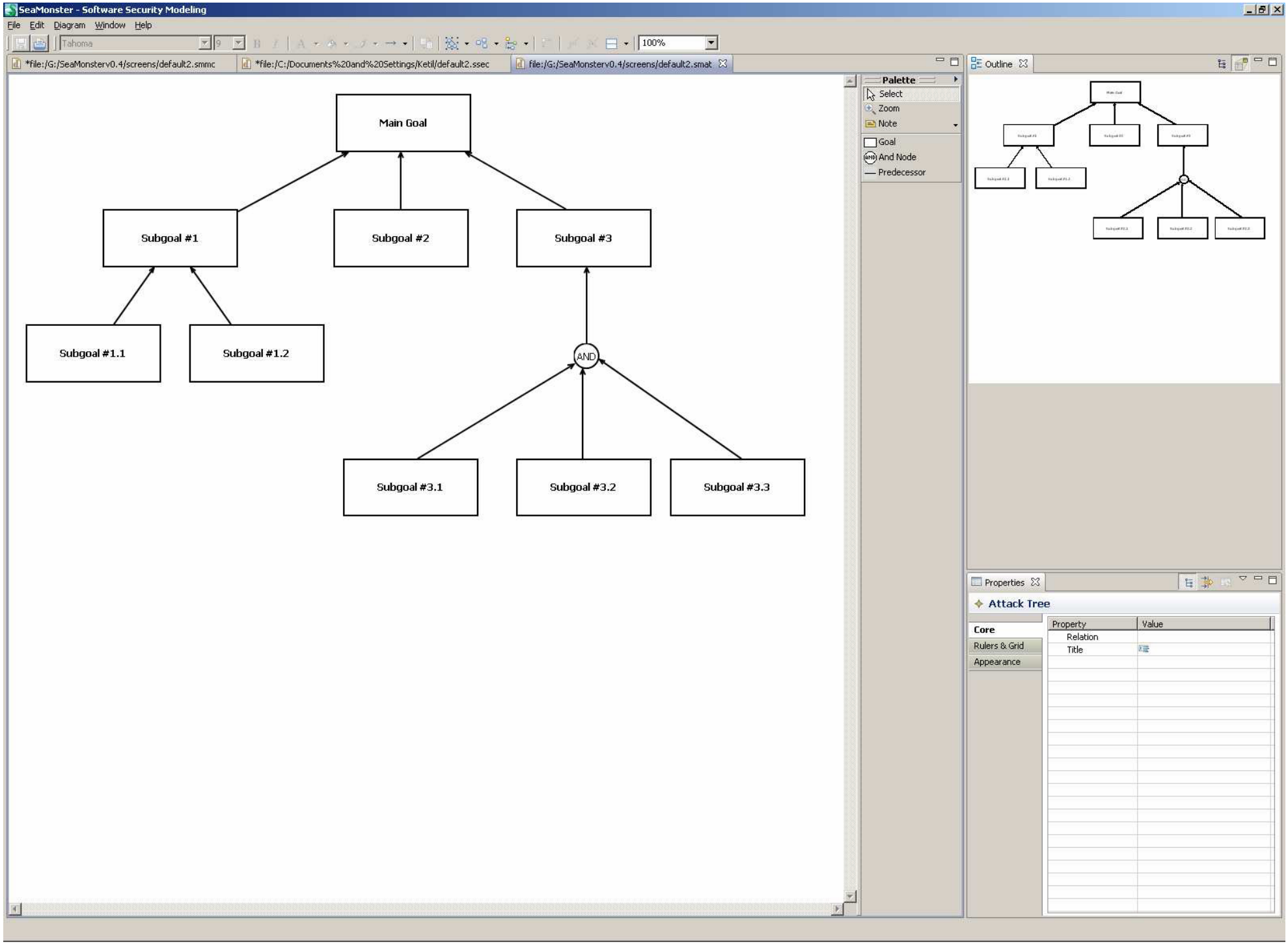


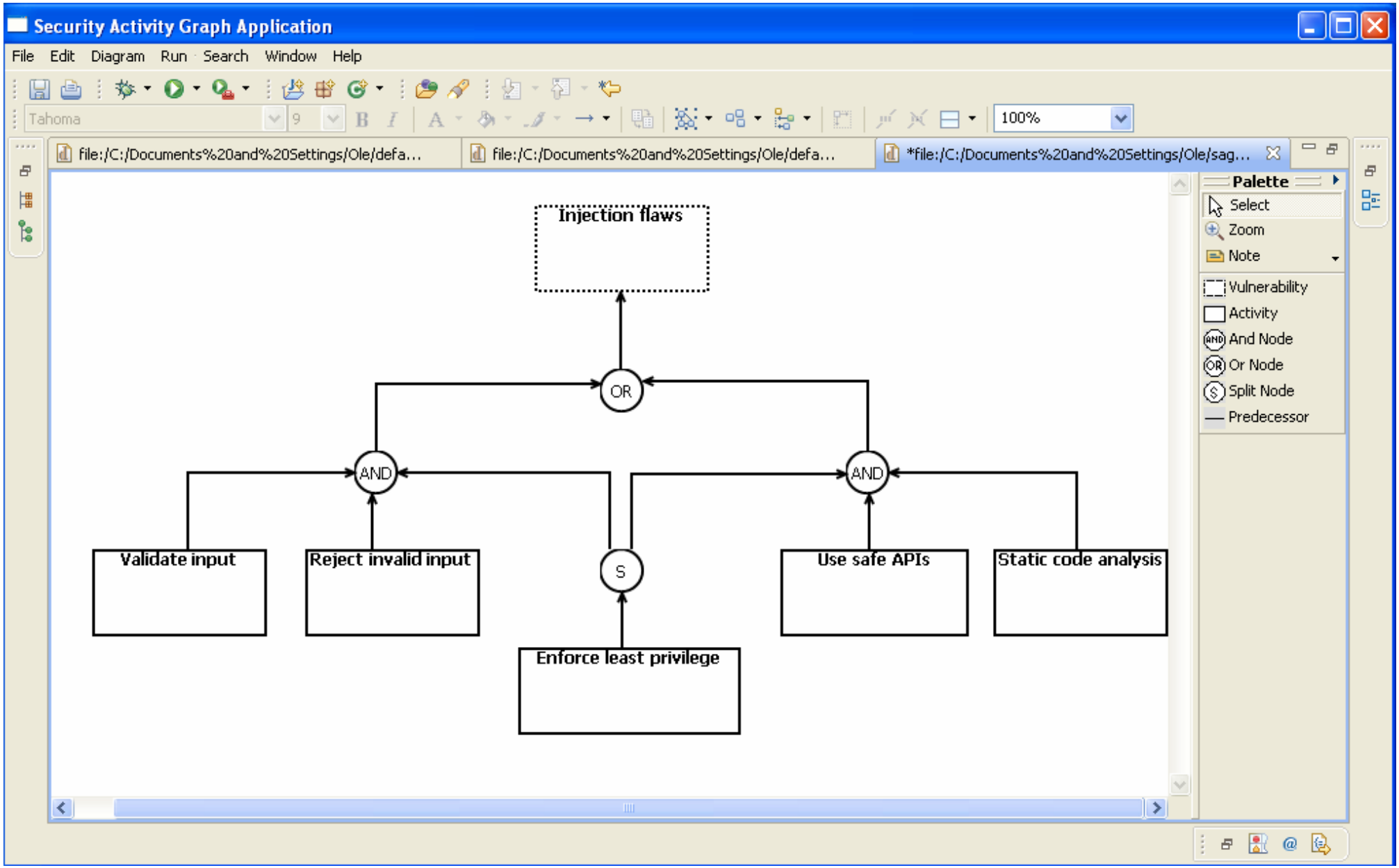
# Threat modeling

- Plan and evaluate from an attacker’s point of view and based on your assets.
- Results in a threat model document
- Not solely connected to the design phase


SeaMonster
SECURITY MODELING SOFTWARE


SeaMonster - Software Security Modeling
File Edit Diagram Window Help
Tahoma
100%
*file:/G:/SeaMonsterv0.4/screens/default2.smmc
*file:/C:/Documents%20and%20Settings/Ketil/d...
file:/G:/SeaMonsterv0.4/screens/default2.smat
file:/C:/Documents%20and%20Settings/Ketil/d...
System
use case #1
use case #2
use case #3
use case #4
use case #5
use case #6
threat 1
SystemCompartment
vulnerability 1
vulnearability 2
Actor
Crook
Insider
<<includes>>
<<extends>>
<<exploits>>
<<exploits>>
<<prevents>>
Palette
Select
Zoom
Note
Actors
Use Cases
Links
Use
Generalize
Include
Extend
Prevent
Detect
Exploit
Threaten
System Box
Outline
Properties
System Box
Core
Appearance
Property
Value
Title


SeaMonster - Software Security Modeling
File Edit Diagram Window Help
*file:/G:/SeaMonsterv0.4/screens/default2.smmc
*file:/C:/Documents%20and%20Settings/Ketil/default2.ssec
file:/G:/SeaMonsterv0.4/screens/default2.smat
Main Goal
Subgoal #1
Subgoal #2
Subgoal #3
Subgoal #1.1
Subgoal #1.2
AND
Subgoal #3.1
Subgoal #3.2
Subgoal #3.3
Palette
Select
Zoom
Note
Goal
And Node
Predecessor
Outline
Properties
Attack Tree
Core
Rulers & Grid
Appearance
Property
Value
Relation
Title


Security Activity Graph Application
File Edit Diagram Run Search Window Help
Tahoma
9
100%
file:/C:/Documents%20and%20Settings/Ole/defa...
file:/C:/Documents%20and%20Settings/Ole/defa...
*file:/C:/Documents%20and%20Settings/Ole/sag...
Palette
Select
Zoom
Note
Vulnerability
Activity
And Node
Or Node
Split Node
Predecessor
Injection flaws
OR
AND
AND
Validate input
Reject invalid input
S
Use safe APIs
Static code analysis
Enforce least privilege

# Security design guidelines

- Describes “good security hygiene”-knowledge[1]
  - Span from less formal best practices, principles and rules-of-thumb to different kinds of policies, rules, regulations and standards
- Forcing too much theoretical information about ways to incorporate security is not very efficient [2]
- We have applied the SODA assumptions on:
  - Security design principles
  - Security patterns

1. M. Howard and S. Lipner, *The Security Development Lifecycle*: Microsoft Press, 2006.
2. Apvrille and M. Pourzandi, "Secure Software Development by Example," in *IEEE Security & Privacy*, vol. 3, 2005, pp. 10-17.

# Security design principles

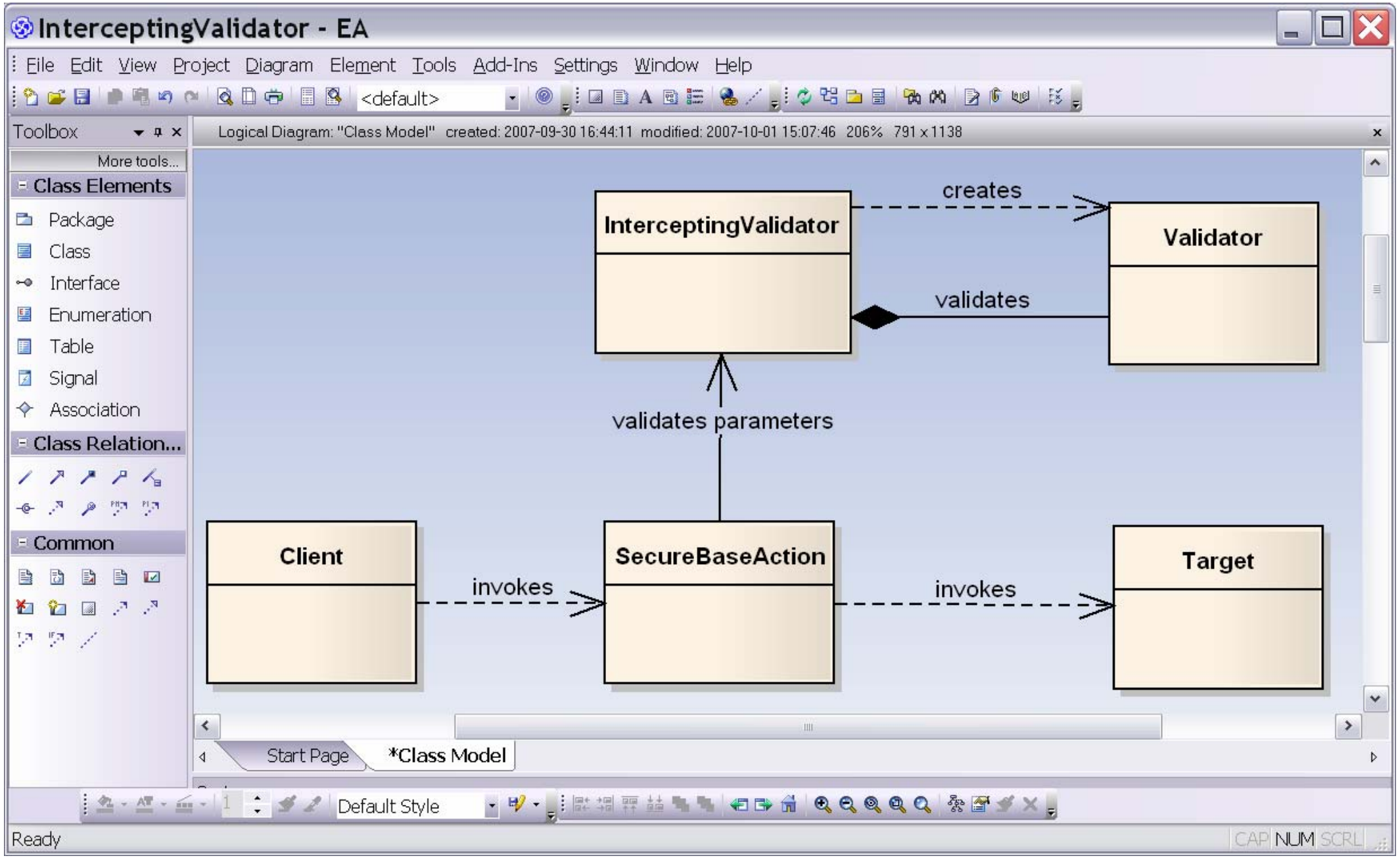
- Proven rules for improving the security posture of an application
  - E.g. the *principle of least common mechanism* states that mechanisms used to access resources should not be shared.

- **SODAWeb** is a tool that does a rough filtering based on your current project

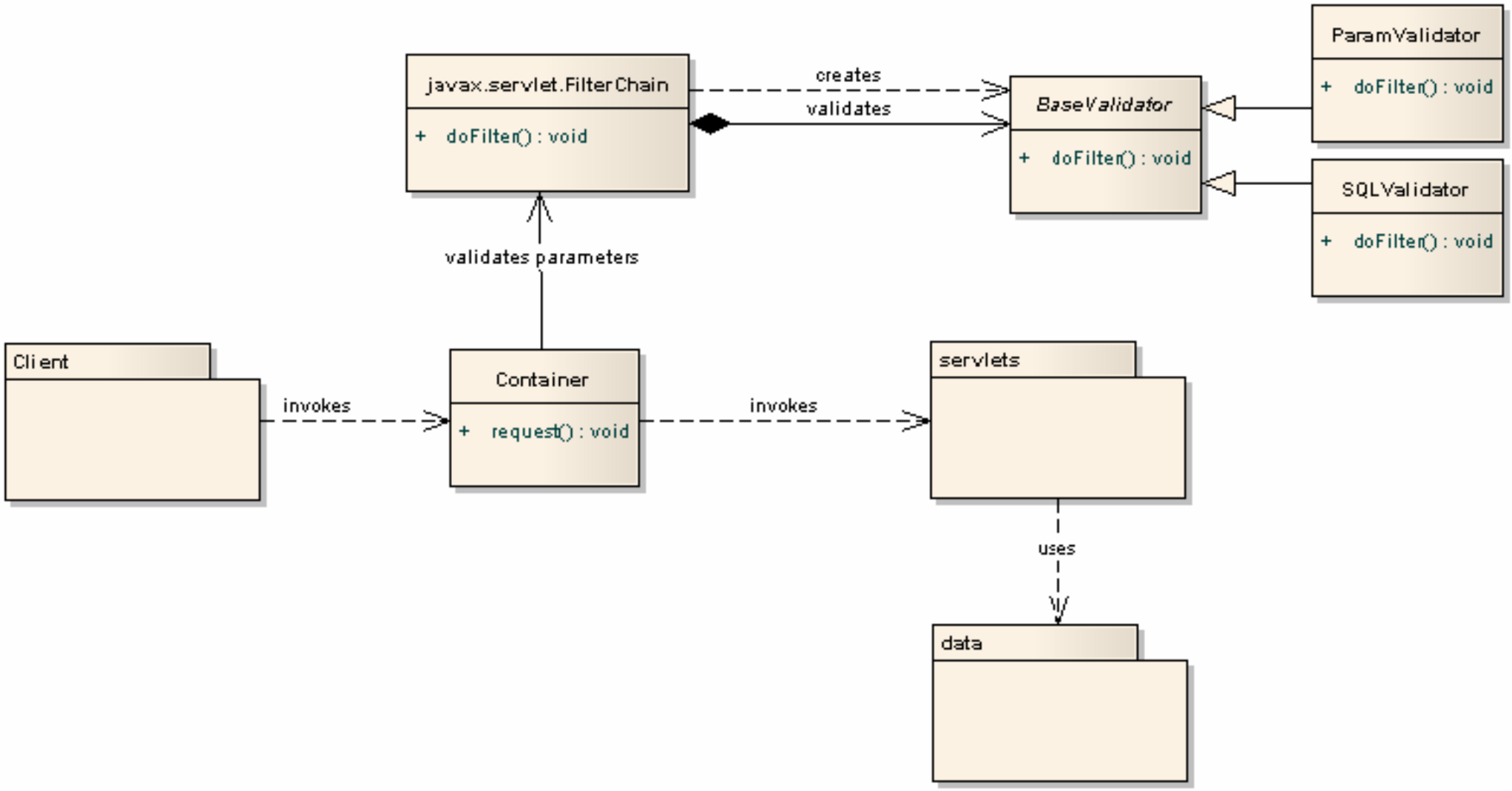
# Security patterns

- A security pattern is a well-understood solution to a recurring security problem
- Many types of patterns
  - Structural, behavioural and creational security patterns, antipatterns, mini-pattern, procedural patterns.
- **SODAWeb** provides:
  - An structured overview
  - XMI templates for **security design patterns**

# Security design pattern template



# Example: Instantiation in an EPR



# Security design review

- *"An architecture and design review helps you validate the security-related design features of your application before you start the development phase* "[1]
- Have fresh blood look at and question the design artifacts that have been produced so far.
- Use **SODAWeb** to find the most relevant checklists (see the example in Table 3)

1. J. D. Meier, A. Mackman, M. Dunner, S. Vasireddy, R. Escamilla, and A. Murukan, *Improving Web Application Security: Threats and Countermeasures*: Microsoft Corporation, 2003.

# Summary and further work

- Have a set of specific and hands-on techniques and tools
- We are pretty compliant with the SODA assumptions
- Need more tuning
  - Student experiments – effort vs effect
- We would like to share security models in a more automated way
  - EU FP7: SHIELDS