Secure Software: Current Research

Jan Jürjens

Competence Center for IT Security
Software & Systems Engineering
Informatics, Technical University Munich

juerjens@in.tum.de

http://www4.in.tum.de/~juerjens
Secure Software

Today: strong security mechanisms: crypto, protocols,…

Still a problem: Security requirements.
- Hard to get right.
- How to transform into secure systems (out of (in)secure components)?
Security Requirements

Security is **holistic** property:
- Attackers often **circumvent** (not: **break**) mechanisms.
- Rely on system **context**.

„Those who think that their problem can be solved by simply applying cryptography don`t understand cryptography and don`t understand their problem“  
(B. Lampson / R. Needham).
Analyze Artefacts for Security

To enforce security requirements:
Extract models from artefacts in development and use of software and analyze against security requirements:

- specifications (e.g. UML models)
- source code
- configuration data

Tool-supported, theoretically sound, efficient automated security analysis.
Model-based Security

Requirements

Weave in
Analyze against

Models

Reverse Engin.
Verify.
Gener.

Configurations

Configure

Code-
Testgen.

Code

Jan Jürjens, TU Munich: Secure Software: Current Research
Secure System Lifecycle

Requirements and use cases

Abuse cases

Risk analysis

Security requirements

External review

Risk-based Security tests

Static analysis (tools)

Penetration testing

Security breaks

Field feedback

Design: Encapsulate prudent security engineering rules.

Analysis: Formally based, automated, efficient tools.

UMLsec

[McGraw 2003]
Tool Support
Security Analysis: Model or Code?

Model:
+ earlier (less expensive to fix flaws)
+ more abstract $\Rightarrow$ more efficient
- more abstract $\Rightarrow$ may miss attacks
- programmers may introduce security flaws
- even code generators, if not formally verified

Code:
+ „the real thing“ (which is executed)

$\Rightarrow$ Do both: verify code against interface spec.
Some Applications

Analyzed designs / implementations / configurations for
- biometry, smart-card or RFID based identification
- authentication (crypto protocols)
- authorization (user permissions, e.g. SAP systems)

Analyzed security policies, e.g. for privacy regulations.
Questions?

juerjens@in.tum.de

http://www.umlsec.org