Secure Software Architecture Description using UML

Jan Jürjens
Competence Center for IT Security
Software & Systems Engineering
TU Munich, Germany

juerjens@in.tum.de
http://www.umlsec.org
Problems, Causes

Many flaws found in designs of security-critical system architectures, sometimes years after publication or use. Main reason: Security often compromised by circumventing security mechanisms within the architecture.
Proposed Solution

Increase security of architectures with bounded investment in time, costs (crucial for industry).

• Consider architectural design artefacts arising in industrial development of security-critical systems (e.g. UML models).

• Tool-supported theoretically sound efficient automated security analysis.

⇒ Model-based Security Engineering
Model-based Security Engineering

Combined strategy:
• Analyze models automatically against security requirements.
• Generate code (or tests) from models automatically.
• Generate models from code to get changes (or analyze legacy systems).

Goal: model-based = source-based.

Idea notation-independent. Here: use UML.
Why UML?

Seemingly de-facto standard in industrial modeling. Large number of developers trained in UML.

Increasingly used as architecture description language (ADL).

Relatively precisely defined (given the user community).

Many tools in development (also for code-generation, testing, reverse engineering, simulation, transformation).
UMLsec: Goals

Extension for secure systems development.

- evaluate UML specifications for weaknesses in design
- encapsulate established rules of prudent secure engineering as checklist
- make available to developers not specialized in secure systems
- consider security requirements from early design phases, in system context
- make certification cost-effective
UMLsec: How

Recurring security requirements, adversary scenarios, concepts offered as stereotypes with tags on component-level. Use associated constraints to verify specifications using automated theorem provers and indicate possible weaknesses. Ensures that UML specification provides desired level of security requirements. Link to code via round-trip engineering etc.
Secure Architecture Patterns

Architectural design patterns (Buschmann et al. 1996). Apply to security.
Example: Architectural primitive: Secure channel.
• Define a secure channel abstraction.
• Define concrete secure channel (protocol).
• Show simulates the abstraction.
Give conditions under which it is secure to substitute channel abstractions by concrete protocols.
Secure Channel Pattern: Problem

To keep $d$ secret, must be sent encrypted.
Secure Channel Pattern: (Toy) Solution

Exchange certificate and send encrypted data over Internet.
Here: Bank application

Security analysis of web-based banking application, to be put to commercial use (clients fill out and sign digital order forms).

Layered security protocol (first layer: SSL protocol, second layer: client authentication protocol)

Security requirements:
- confidentiality
- authenticity
Further Applications to Architectures

Secure Architectural Design Principles by Saltzer, Schroeder
Variant of the Internet security protocol TLS (SSL)
Common Electronic Purse Specification
Biometric authentication protocol for German Telekom
Analysis of SAP access control configurations for German bank
Telematic automobile emergency application of German car company
Electronic signature architecture of German insurance company
Electronic purse for Oktoberfest
Jan Jürjens, TU Munich: Secure Software Architecture Description using UML 13
Conclusions

Model-based Secure Software Architectures using UML:

- formally based approach
- automated tool support
- industrially used notation
- integrated approach (source-code, configuration data)
Resources

Jan Jürjens, Secure Systems Development with UML, Springer 04
Tutorials: e.g. WICS 04. Nov.: SISBD (Malaga), ISSRE (Rennes).
Spring School: May 2005, Carlos IV Univ. Madrid
Workshops: WITS05@POPL05, CSDUML05

More information (papers, slides, tool etc.):
http://www.umlsec.org