Encapsulating Rules of Prudent Security Engineering

Jan Jürjens

Computing Laboratory, University of Oxford

jan@comlab.ox.ac.uk

http://www.jurjens.de/jan

Security Design

Saltzer, Schroeder (1975):

- "Expansive view of problem most appropriate to ensure no gaps appear in strategy".

- "No complete method applicable to construction of large general-purpose systems exists yet".

Problems in Security Design

Overall system design limitations and implementation errors

rather than direct attacks against security mechanisms

(Anderson 1994).

Addressing the problem

Use formal core of Unified Modeling Language (UML)
to encapsulate knowledge on prudent security engineering.

Much effort in verifying and implementing specifications wasted
since formulated imprecisely and unintelligibly.
Express security-relevant information in widely used notation.

May motivate that specifications be written more formally.

Usage

- reason formally
- specification-based testing

Economy of mechanism

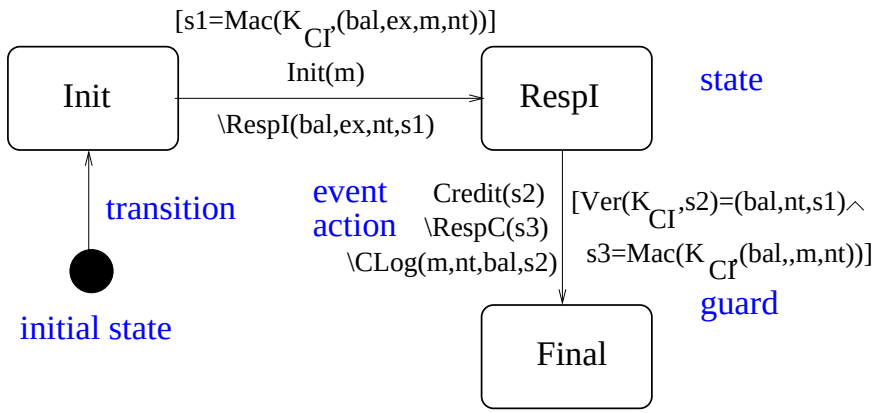
Guidance on employment of security mechanisms.

Otherwise more complicated mechanisms ("more secure").

Fail-safe defaults

Security-relevant invariants ensured throughout execution.

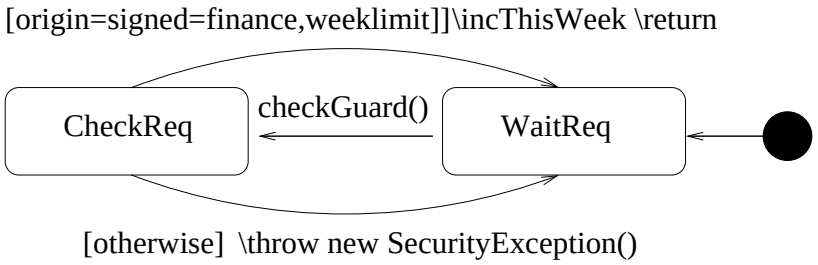
Example: secure log-keeping for audit control in Common Electronic Purse Specifications (CEPS).



Uncover unstated assumptions.

Complete mediation

Can enforce principle e.g. in Java using guarded objects.
Ensure proper use of guards.



Complete mediation: access to every object guarded.
More feasibly, mediation wrt. a set of sensitive objects.

Open design

Develop systems whose security does not rely

on the secrecy of its design.

Separation of privilege

Here specification satisfies separation of privilege wrt. privilege p if signature of two or more principals required to be granted p.

Formulate such requirements abstractly using activity diagrams. Verify behavioural specifications wrt. these requirements.

Least privilege

Least privilege: every proper diminishing of privileges gives system not satisfying functionality requirements.

Can formalise and verify this.

Least common mechanism

Object-orientation:

- data encapsulation
- data sharing well-defined (keep at necessary minimum).

Psychological acceptability

Wrt. development process:
ease of use in development of secure systems.

User side: e.g. performance evaluation
(acceptability of performance impact of security).

Protocol contexts

Underlying physical layer important for security of protocols.
Investigate security mechanisms in system context.

E.g. CEPS transactions secure because not feasible
for attacker to act as relay between card and terminal.
Not explicitely stated; planned to use CEPS over Internet.

Precisely formulate assumptions on protocol context.

Conclusion

"Expansive view of computer security."

Towards method for constructing secure systems.

Evaluate system design wrt. design guidelines.

Consider protocols in context.