Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Modellierung von Sicherheitszielen im Softwareentwurf

OODACH 2015

Stefanie Jasser

Stefanie Jasser jasser@informatik.uni-hamburg.de

Motivation

- Verschiedene Ansätze zur Analyse von Sicherheitszielen
- Viele Security-Flaws auf Entwurfsebene
  - fehlende Security-Kenntnisse bei Architekten und Entwicklern
  - Security als nachträgliches Feature
- Herausforderung: Security by Design
  - wenige Arbeiten zur Überführung in den Entwurf
  - meist keine explizite Modellierung von Sicherheitszielen

Fragen

- Wie lassen sich Sicherheitsziele im Softwareentwurf explizit modellieren?
- Wie können Sicherheitsziele im Softwareentwurf systematisch berücksichtigt werden?
- Kann die Erfüllung der Sicherheitsziele durch einen Softwareentwurf überprüft werden?
- Kann die Einhaltung des Entwurfs bzw. der Sicherheitsziele durch das Softwaresystem überprüft werden?
- Wie kann die Konsistenz während der
  - Evolution des Softwaresystems
  - Evolution der Sicherheitsziele

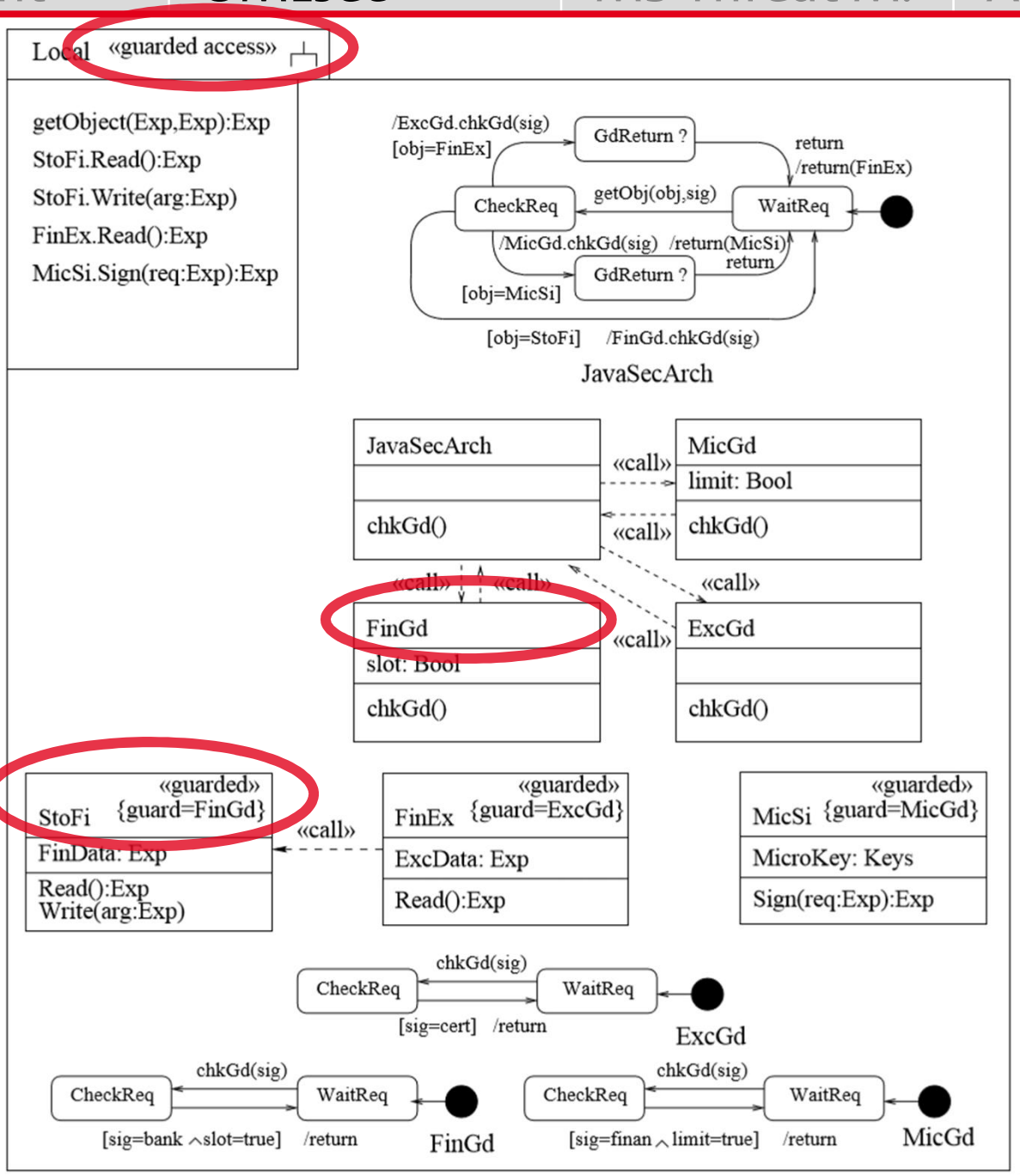
  sichergestellt werden?

Existierende Ansätze

- UML-basiert
  - UMLsec
  - Secure UML
  - ...
- Datenflussbasiert
  - Microsoft Threat Modeling
  - ...
- Weitere Ansätze
  - Secure xADL
  - BPMN-basiert

UMLsec [Jür02a, Jür02b]

- Ziel: Kapselung und Bereitstellung guter Sicherheitslösungen
- UML-Profil: nutzt Stereotypen, Tags, Constraints
  - Stereotypen und Tags
    für Sicherheitsziele und Fehler-/Angreiferszenarios
  - Zugeordnete Constraints
    für Identifikation potentieller Schwachstellen
- Erweiterung diverser Diagrammarten



In Anlehnung an [Jür06]

UMLsec: Bewertung

Basis: UML

- De-facto Standard der Industrie
- Relativ klar definiert
- Gute Tool-Unterstützung
- unterschiedliche Abstraktionslevel
- Größere Modelle schnell unübersichtlich

UMLsec

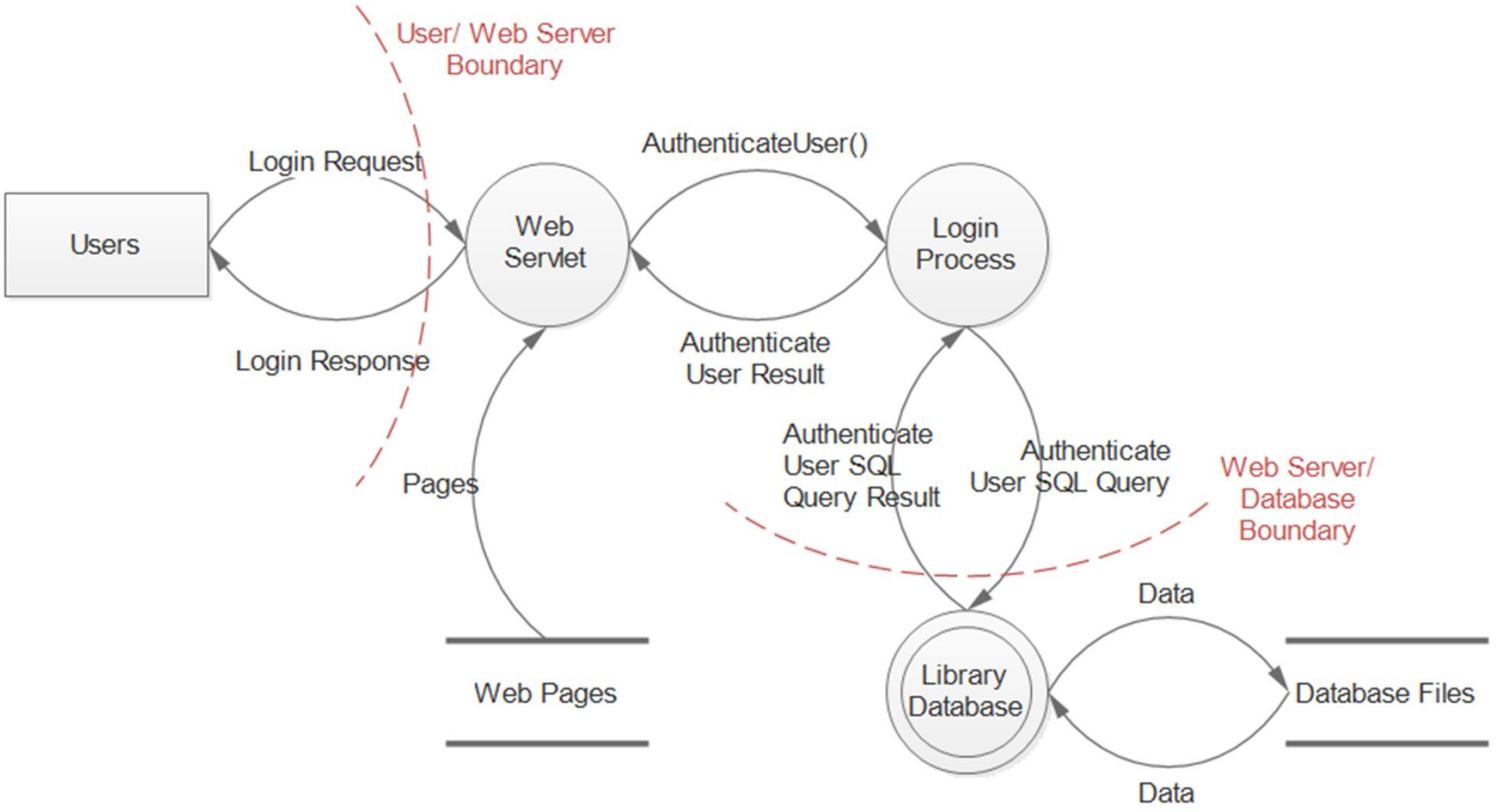
- Komplexität durch Stereotypen und Tags
- Anwendbarkeit ohne Expertenwissen fragwürdig
- Modellierung diverser Sicherheitsaspekte
- Constraints erlauben Validierung

Datenflussbasierte Modellierung

Beispiel: Microsoft Threat Modeling

- Empfohlen: Datenflussdiagramm
  - Zuordnung potentieller Bedrohungen zu Elementen: STRIDE
  - Übertragung auf Aktivitätsdiagramme [Joh10]
- Erweiterung um Vertrauensgrenzen
- Iterative Verfeinerung des Models

MS Threat Modeling: Beispiel



In Anlehnung an: https://www.owasp.org/images/1/16/Data_flow2.jpg

MS Threat Modeling: Bewertung

- Keine explizite Modellierung von
  - Sicherheitszielen
  - Bedrohungen und Gegenmaßnahmen
- Allgemeine Bedrohungskategorien:
  Experten für konkrete Analyse notwendig
- Bei notwendiger Detaillierung schnell komplexe Modelle
- Tool-Unterstützung zur Modellierung vorhanden

Ausblick

- (Leichtgewichtige) Modellierung von Sicherheitszielen im Softwareentwurf
  - Systematische Ableitung aus den Sicherheitszielen
  - Validierung des Entwurfs
  - Unterstützung von Entwicklern und Architekten mit Grundkenntnissen in Informationssicherheit
- Konsistenz des Entwurfs und der Sicherheitsziele während der Evolution

Quellen

[AWT07] Abi-Antoun, Marwan; Wang, Daniel; Torr, Peter (2007): Checking threat modeling data flow diagrams for implementation conformance and security. In: Proceedings of 22nd IEEE/ACM International Conference on Automated Software Engineering. ASE '07. New York, NY, USA, Los Alamitos u.a.: IEEE Computer Society; ACM, S. 393–396.

[FKK+14] Felderer, Michael; Katt, Basel; Kalb, Philipp; Jürjens, Jan; Ochoa, Martín; Paci, Federica et al. (2014): Evolution of Security Engineering Artifacts. A State of the Art Survey. In: *International Journal of Secure Software Engineering* 5 (4), S. 48–98. DOI: 10.4018/ijsse.2014100103.

[Joh10] Johnstone, Michael N. (2010): Threat Modelling with Stride and UML. Hg. v. Edith Cowan University. Research Online. Online verfügbar unter http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=1087&context=ism.

[Jür02a] Jürjens, Jan (2002): Principles for Secure Systems Design. Dissertation. Oxford University, University of Oxford. Computing Laboratory. Online verfügbar unter https://www-secse.cs.tu-dortmund.de/secse/pages/people/juerjens/publications/papers/Jur02.pdf.

[Jür02b] Jürjens, Jan (2002): UMLsec: Extending UML for Secure Systems Development. In: «UML» 2002 — The Unified Modeling Language, Bd. 2460. Berlin, Heidelberg: Springer (Lecture Notes in Computer Science), S. 412–425.

[Jür06] Jürjens, Jan (2006): Foundations for Designing Secure Architectures. In: *Electronic Notes in Theoretical Computer Science* 142, S. 31–46. DOI: 10.1016/j.entcs.2005.07.012.

[McG06] McGraw, Gary (2006): Software Security. Building Security In. Upper Saddle River, NJ: Addison-Wesley (Software Security Series).

[Mi10] Microsoft (2010): Security Development Lifecycle. Vereinfachte Implementierung des Microsoft SDL. Hg. v. Microsoft.

[NNY10] Nhlabatsi, Armstrong; Nuseibeh, Bashar; Yu, Yijun (2010): Security Requirements Engineering for Evolving Software Systems: A Survey. In: *IJSSE* 1 (1), S. 54–73. DOI: 10.4018/jsse.2010102004.

[RT05] Ren, Jie; Taylor, Richard N. (2005): A Secure Software Architecture Description Language. In: Proceedings of the Workshop on Software Security Assurance Tools, Techniques, and Metrics.