

Modell-basierte Sicherheit: Sicheres Konfigurations- und Änderungsmanagement

Jan Jürjens

Software & Systems Engineering

Informatik, Technische Universität München



juerjens@in.tum.de

<http://www4.in.tum.de/~juerjens>



Problem: Sicherheit

Sicherheit (**Security**): Schutz von Daten oder Systemen gegen **mutwillige Angriffe**.

Inhärent schwierig (zielorientierter Angreifer). Beispiel (1997):

NSA hacker team bricht in U.S. Department of Defense Computer und U.S. Stromversorgungssystem ein. Demonstriert Strom- und Notrufausfälle in Washington, D.C..

Sichere Geschäftsprozesse

Analyse von sicherheitskritischen Geschäftsprozessen **schwierig** (komplexe organisatorische Abläufe).
Viele entwickelte und eingesetzte Systeme genügen **nicht** Sicherheitsanforderungen.
Sichere Produkte oft auf **unsichere** Weise eingesetzt.
Viele z.T. spektakuläre **Angriffe**.
Problem: **Qualität** vs. **Kosten**.

Gründe I

Entwurf sicherer Systeme ist **schwierig**, sogar für Experten.

Sicherheit im Nachhinein:
„Penetrate-and-patch“
(aka „Bananenstrategie“)

- **unsicher**
- **störend**
- **Vertrauensverlust.**



Gründe II

Ganzheitliche Eigenschaft

- Sicherheitsmechanismen umgehbar ?
- Systemumgebung ?
- Kein Add-on im Nachhinein

„Those who think that their problem can be solved by simply applying cryptography don't understand cryptography and don't understand their problem“ (R. Needham).

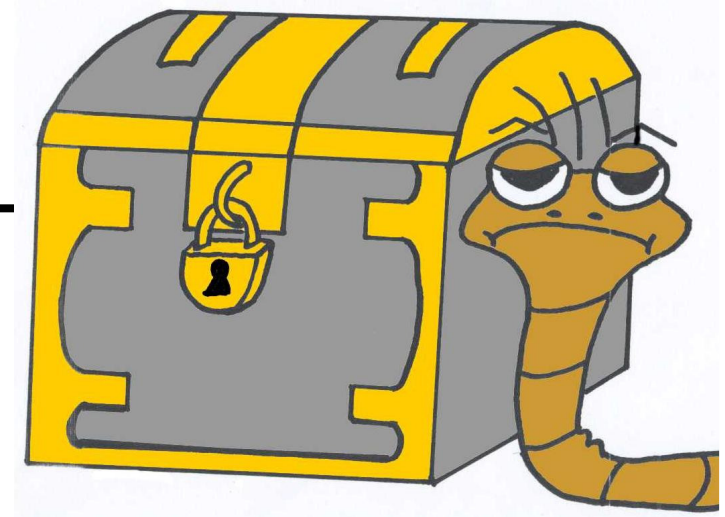


Security Engineering

Sicherheit erhöhen bei begrenzter
Zeit, Kosten.

Lösungsansatz:

- aus **Artefakten in industrieller Entwicklung und Betrieb sicherheitskritischer Software: Modelle** extrahieren (UML, Quellcode, Konfigurationen)
 - **Werkzeugunterstützung** für theoretisch fundierte effiziente (automatische) Sicherheitsanalyse
- *Modell-basiertes Security Engineering*



Modellbasierte Sicherheitsanalyse

Modellbasierte Sicherheitsanalyse von Geschäftsprozessen mit der Unified Modeling Language (UML):

- Einfache, intuitive Notation
- Komfortable Werkzeugunterstützung
- Automatische Sicherheits- und Risikoanalyse der modellierten Geschäftsprozesse unter Einbezugnahme des Unternehmensumfeldes
- Automatische Checks von Systemkonfigurationen (z.B. SAP-Berechtigungen, ...)

Einsatzfelder

- **Systementwurf, -einrichtung, -konfiguration**
 - z.B. Architekturbewertung (Beispiel: Teleworking), Plattformenwahl, Altsystemeinbindung.
 - spezielle Sicherheitsmechanismen wie Smartcards (Versicherungskarte).
- **Laufender Betrieb**
 - z.B. Konfigurationsmanagement, Überprüfung von Berechtigungen, Einrichtungen von Firewalls
- **Sichere Geschäftsprozesse / Behördenvorgänge**
- **Einsatz in Sicherheitsaudits**

Vorteile

- Verwendung **bewährter Regeln** für sichere Geschäftsprozesse.
- Verwendbar ohne **spezielle Ausbildung**.
- Berücksichtigung von Sicherheit ab **Geschäftsprozessentwurf**.
- Erhöht Vertrauen in **Korrektheit** und **Vollständigkeit** von **Audits**.
- Unterstützt **Zertifizierungen**.

Return on Security Investment (RoSI)

Wann lohnt eine Investition in Sicherheit ?

→ Wenn Reduktion im statistisch jährlich zu erwartendem Schaden Investition aufwiegt.

Problem: statistisch zu erwartender Schaden schwierig zu berechnen (Kumulschäden, abhängige Wahrscheinlichkeiten etc.).

Lösung: **werkzeugbasierter** Ansatz.

IT-Risiken vs. KontraG/Basel II

Basel II (bis 2006): **risikogerechtere**
Regelung der Eigenkapitalanforderungen

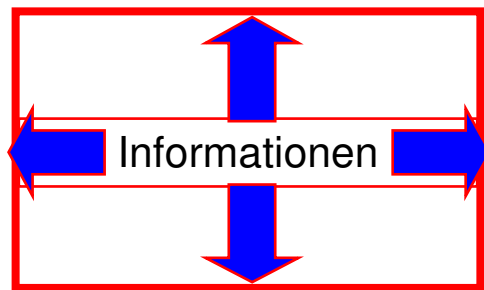
Genauere Analysemethoden (Kreditrisiko,
operationelles Risiko, *internal-ratings-based*). Offenzulegen.

Insbesondere **IT Risiken** (*unexpected loss*,
z.B. Virenbefall, Hackerangriff, ...)

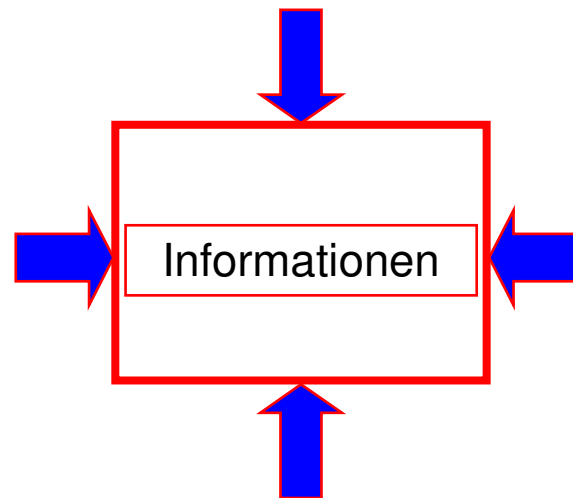
➔ **modellbasierte IT-Risiko-Bewertung**

Sicherheitsanforderungen I

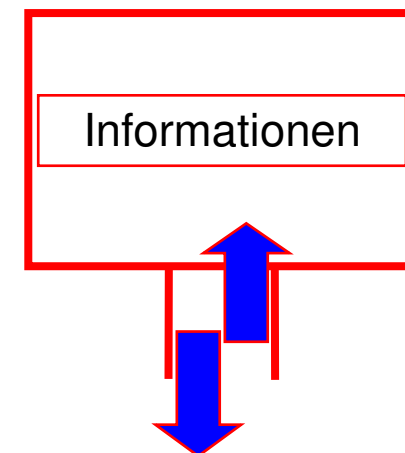
Vertraulichkeit



Integrität

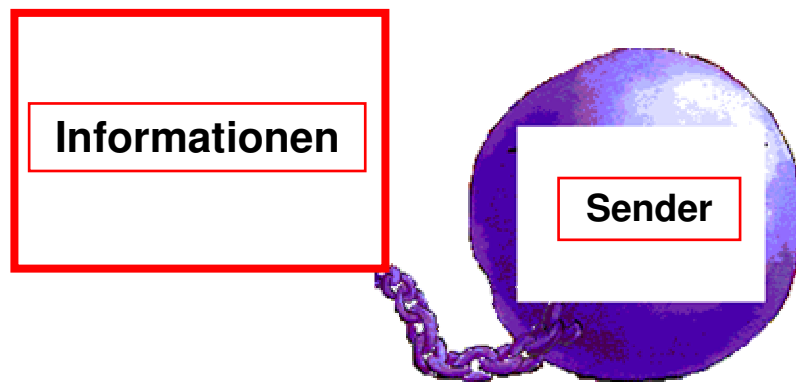


Verfügbarkeit

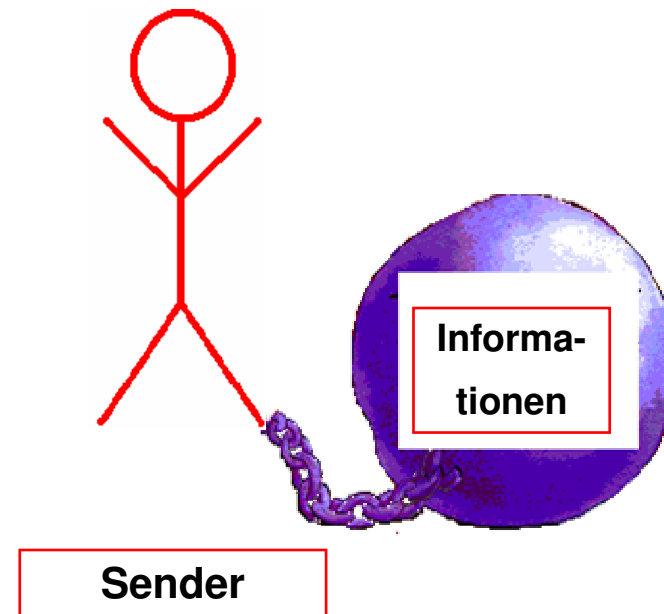


Sicherheitsanforderungen II

Authentizität



Nichtabstreitbarkeit



Sicherheitsanforderungen

Jeweils

- verschiedene Sicherheitsstufen (Gewichtungen) bzgl. einzelner Daten
- Berücksichtigung verschiedener möglicher Angreifer

Prinzipien für sichere Geschäftsprozesse

Saltzer, Schroeder (1975):

Design Prinzipien für sicherheitskritische Systeme.

- Anwendung auf Geschäftsprozesse ?
- Werkzeugunterstützung ?

“In dubio pro securitate”

Gründen Sie Zugriffsentscheidungen eher auf Verweigerung als auf Erteilung von Rechten.

Wenn eine Sicherheitsregel vergessen wird, führt dies nicht zu einer Sicherheitslücke.

Einhaltung der Regel automatisch überprüfen.

Vollständige Kontrolle

Jeder Zugriff auf jedes sicherheitskritische Objekt muss auf Berechtigung überprüft werden.

Sicherheit ist immer eine umfassende Eigenschaft – eine Kette ist so stark wie ihr schwächstes Glied.

Zugriffskontrollen können automatisch in Geschäftsprozessmodell eingefügt werden.

4-Augen-Prinzip

Ein Schutzmechanismus, der zwei Schlüssel erfordert, ist robuster und flexibler als einer, der den Zugriff mit einem einzigen Schlüssel erlaubt.

Beispiel: Erteilung größerer Kredite im Bankenbereich nur möglich durch zwei Angestellte. Vermindert Missbrauchsrisiko. Berücksichtigt in Analyse.

Minimale Berechtigungen

Jedes Programm und jeder Benutzer des Systems sollte nur die zur Erledigung seiner Aufgaben nötigen Berechtigungen erhalten.

Jede unnötige Berechtigung verleitet zum Missbrauch.

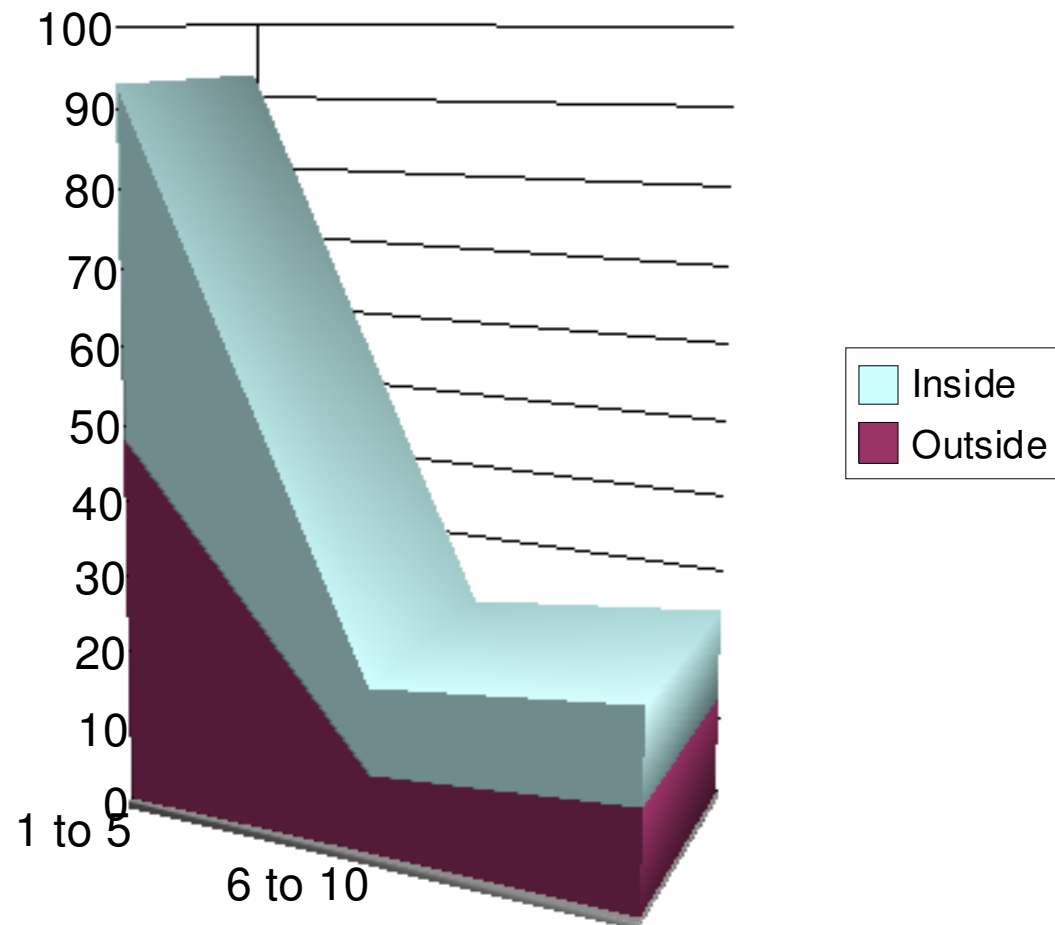
Notwendige Berechtigungen können automatisch generiert werden.

Konfigurationsdaten

Computer Crime
Studie (FBI
2004): 50%
Insider Angriffe

Konfigurationen
(Berechtigungen
etc.) korrekt ?

Where Attackers come from



Konfigurationsdaten überprüfen

Beispiel: SAP **Berechtigungen** auf **Sicherheitsregeln** überprüfen. Nicht von Hand machbar:

- grosse **Datenmenge** (60.000 Datensätze)
- komplexe **Querverbindungen** zwischen Berechtigungen auf verschiedenen Ebenen
- **dynamische** Änderungen, **Delegation**
- manuelle Analyse vertrauenswürdig ?

Automatische Überprüfung erhöht Sicherheit an zentraler Stelle.

Automatische Prüfung: Features

- Konfiguration von Geschäftsanwendung einlesen
- Bericht mit Schwachstellen generieren
- Flexible Konfiguration der Berichtsdaten
- Leicht konfigurierbare für verschiedene Geschäftsanwendungen / Use Cases
- Analyse großer Datenmengen
- Checks nach frei konfigurierbaren Regeln

Beispiel: 4-Augen-Prinzip

Karin: Kredit anlegen (K1)

Susanne: Kredit genehmigen (K2)

Stefan: Urlaubsvertretungen einteilen (U)

Ziel: kein Mitarbeiter hat jemals K1 und K2
für gegebenes Datenobjekt.

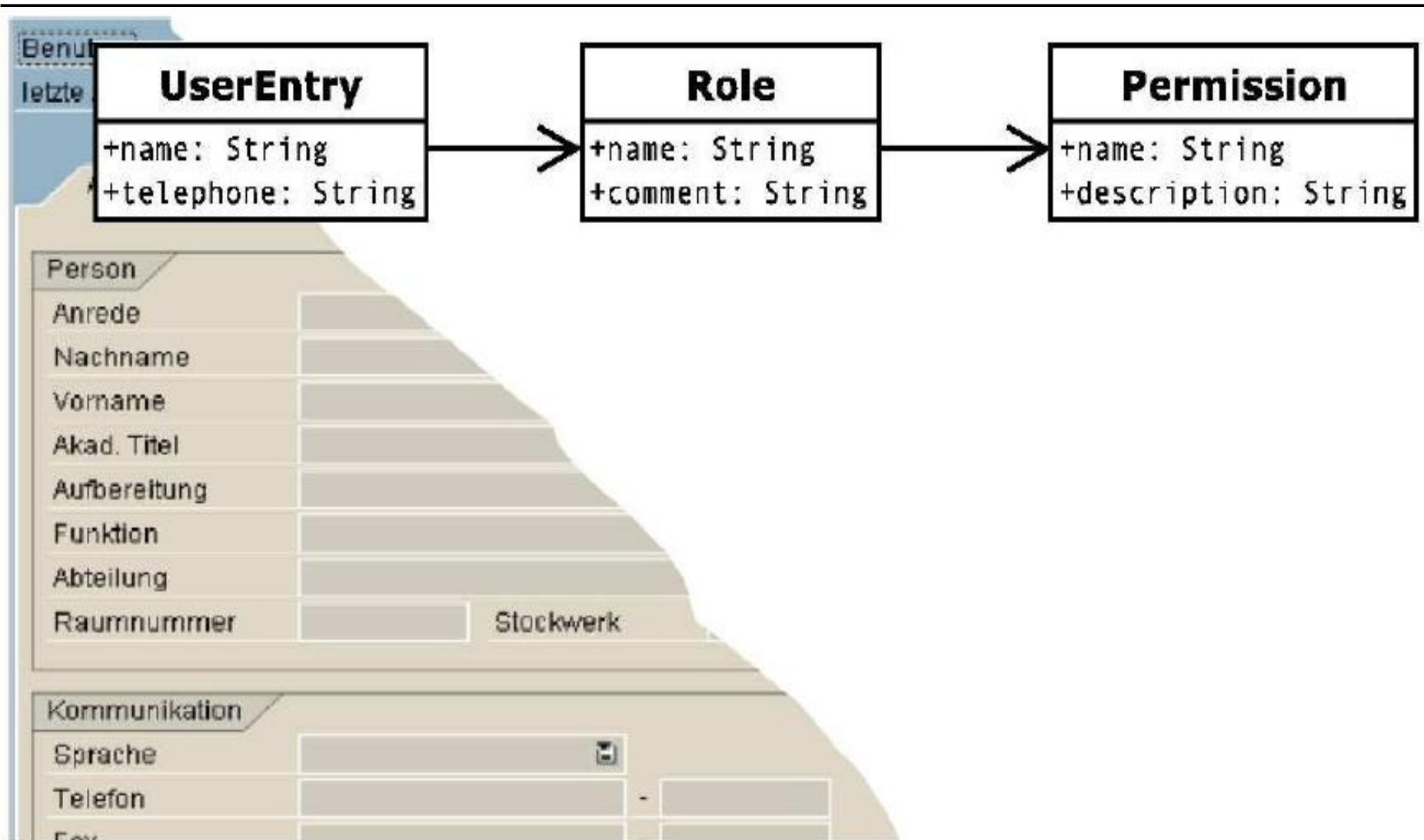
Beispiel: Transitive Berechtigungen

SAP Transaktionen können implizit Zugang zu Sub-Transaktionen geben.

„Transitive“ Berechtigungen schwierig zu durchschauen.

Ziel: Benutzer mit indirektem Zugang zu bestimmter Berechtigung finden.

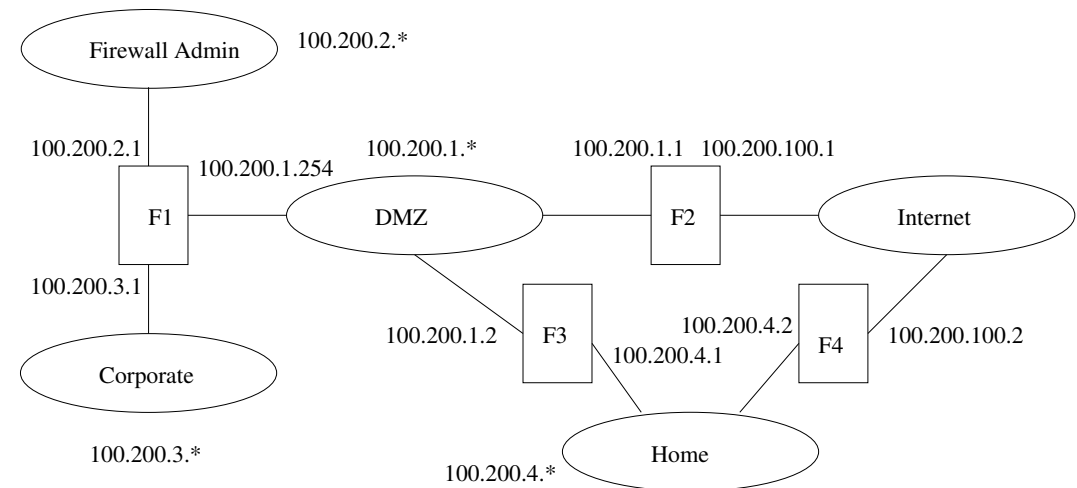
Screenshot Analysewerkzeug



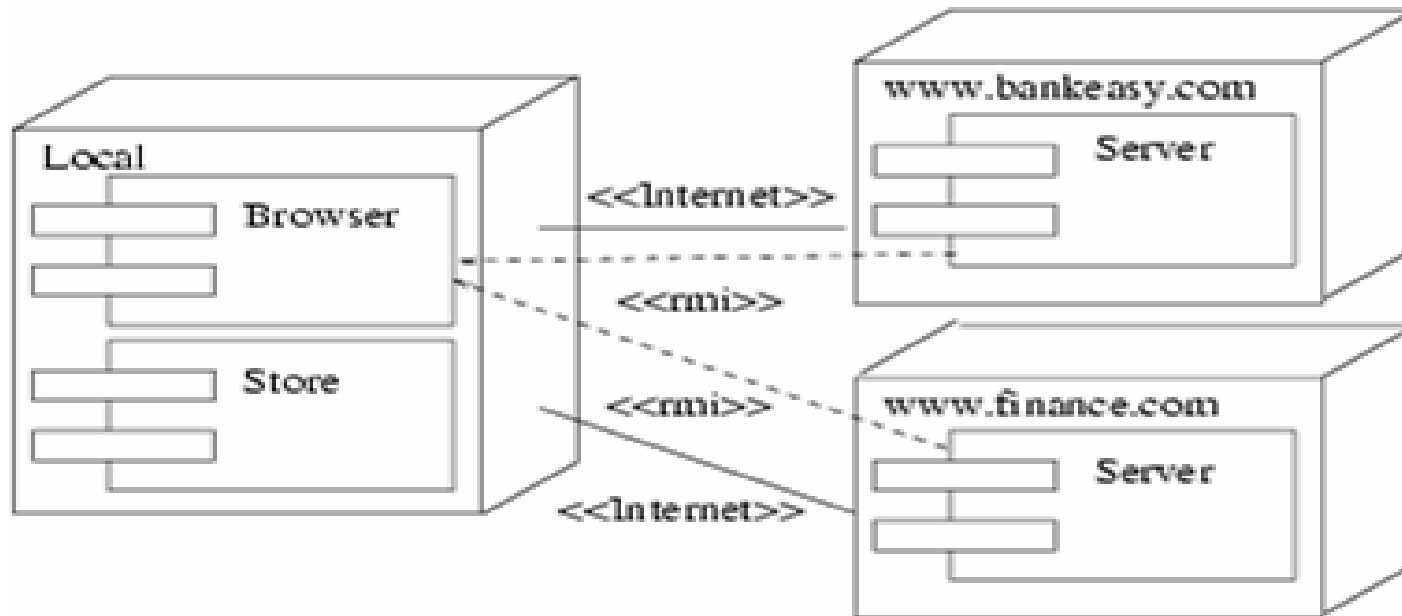
Firewallkonfigurationen

Fehlkonfiguration häufige Schwachstelle beim Einsatz von Firewalls.

Modell-basierte Analyse: Mit **Netzwerkmodell** automatisch überprüfen, dass Konfigurationen **Security Policy** umsetzen.



CORBA-Anwendung



Objektbasierte Zugangskontrolle, abhängig vom **Ausführungszusammenhang**.

Schwer nachzuvollziehen → **Automatische** Analyse von Modell / Implementierung

Fazit

Model-based Security Engineering:

- wissenschaftlich fundiert
 - mathematisch präzise Sicherheitsdefinitionen
- praktisch anwendbar
 - automatische, effiziente Analyse
- integrativer Ansatz (Geschäftsprozesse, Konfigurationsdaten).

Weitere Informationen

Jan Jürjens, Secure Systems Development with UML, Springer-Verlag, 2004

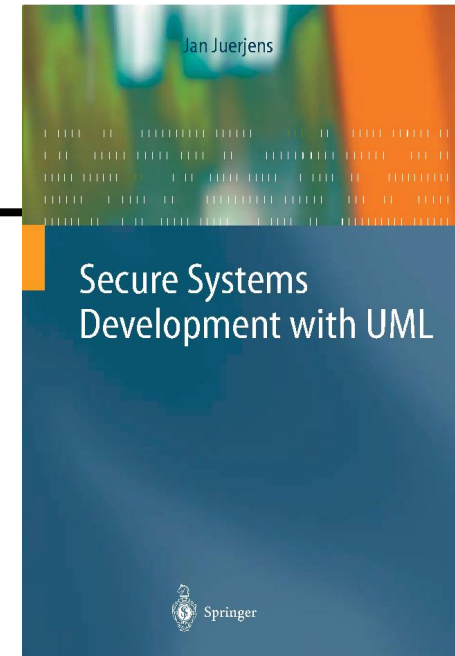
Seminar, Deutsche Informatik Akademie (Anfang 2005)

Gesellschaft für Informatik Fachgruppe: Formale Methoden und Software Engineering für Sichere Systeme (FoMSESS)

Konferenz: Sicherheit 2005

Mehr Informationen:

<http://www4.in.tum.de/~juerjens>



Zum Abschluss

Wir sind immer an **Problemen aus der Praxis** für unsere **Werkzeuge** und **Methoden** interessiert.

Mehr Info: <http://www4.in.tum.de/~secse>

Kontakt: hier oder via Internet.

Danke für Ihre Aufmerksamkeit !