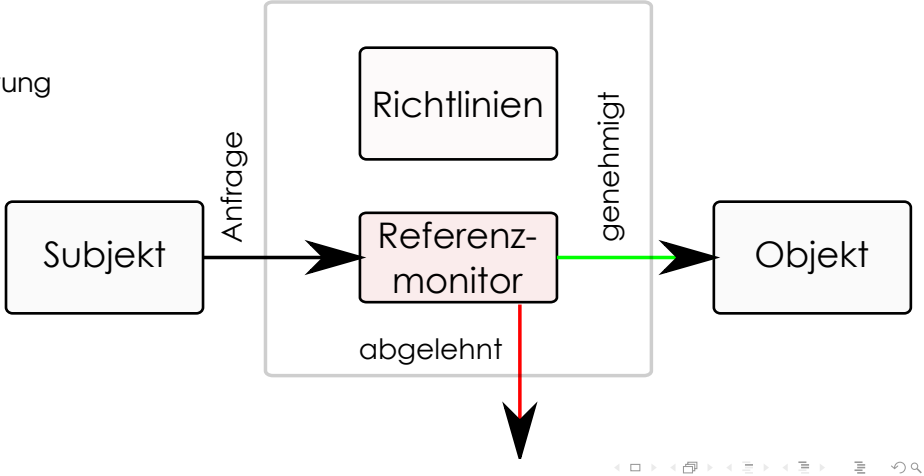
Informationssicherheit

2. Sicherheitsmodelle

Prof. Dr. Christoph Skornia
christoph.skornia@oth-regensburg.de

Ziele von Sicherheitsmodellen

- Abstraktion
- Vereinfachung
- Rahmen für Implementierung



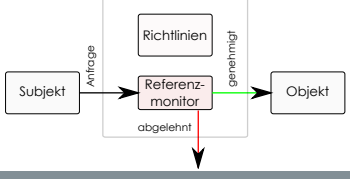
Komponenten

- **Subject**:

  Aktive Einheit, initiiert den Zugriff auf Objektressourcen z.B. handelnde Personen, Programme oder Prozesse

- **Objekt**:

  Soll geschützt werden, i.d.R. Information oder Ressource z.B. Drucker, Personaldaten, ...

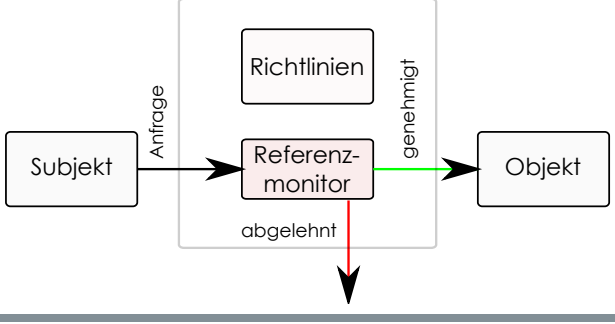


- **Referenzmonitor**:
  - Konzeptuelles Modell
  - Nicht unbedingt als physikalische Einheit im System vorhanden
  - Aufgabe:
    - kontrolliert jeden Zugriffsversuch
    - ggf. auch loggen von Zugriffen in Log-Datei
    - Zwischen Prüfung und Ausführung ist keine Änderung der Berechtigungen möglich
    - Referenzmonitor muss vor Manipulation geschützt werden

Komponenten

Security Policy/Richtlinie:

- Definiert die Bedingungen, unter denen ein Subjekt auf ein Objekt zugreifen darf
- Definiert eine Beziehung zwischen Subjekten, Objekten und Zugriffsrechten
- Beschreibt die erwünschten, zulässigen Zustände



Zugriffskontrolle

Discretionary Access Control (DAC)

- Benutzer-bestimmbare Zugriffskontrolle
- Eigentümer ist für den Schutz eines Objekts verantwortlich
- Rechte werden für einzelne Objekte vergeben
- Objektbezogene Sicherheitseigenschaften, aber keine systemweiten
- Problem:

  meist keine Betrachtung von Abhängigkeiten z.B.: implizite Vergabe von Leserechten durch die Ausführung einer Aktion, die das Lesen ansonsten vertraulicher Information erlaubt

Bem: Standardbetriebssysteme wie Unix/Linux oder Windows unterstützen Discretionary Access Control

Mandatory Access Control (MAC)

- Systembestimmte (regelbasierte) Festlegung von Sicherheitseigenschaften
- Benutzerdefinierte Rechte werden durch systembestimmte überschrieben (dominiert)
- Betriebssysteme oder Anwendungen müssen spezielle Maßnahmen und Dienste bereitstellen, um MAC-Policies durchzusetzen

Modelle für MAC - Zugriffsmatrix (ZM)

Komponenten einer ZM

- (Dynamische) Menge von Objekten O_t
- (Dynamische) Menge von Subjekten S_t mit: $S_t \subseteq O_t$
- Menge von Rechten R
- Zugriffsmatrix $M_t : S_t \times O_t \rightarrow 2^R$ (Schutz-Zustand zur Zeit t)

S_t	Datei 1	Datei 2	Datei 3	Prozess 1	Prozess 2
Prozess 1	{read, write}		{read, write}		{send, receive}
Prozess 2				{send, receive}	
Prozess 3		{owner, execute}		{signal}	

Modelle für MAC - Zugriffsmatrix (ZM)

Vorteile:

- sehr einfach und intuitiv nutzbar
- relativ flexibel, feingranulare Subjekte/Objekte und Rechte
- einfach zu implementieren, z.B. Rechtelisten
- Grundlage der Zugriffskontrolle aller Standard-OS!

Nachteile:

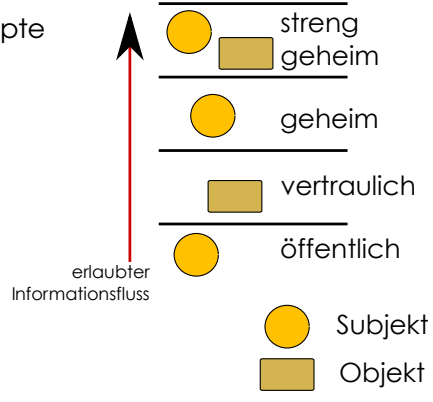
- Fehlende Typisierungskonzepte (aber Gruppenbildung)
- keine Rechtevergabe an Klassen mit Rechte-Vererbung
- Skaliert schlecht:
  - in der Praxis: mächtige dynamische Menge von Subjekten
  - aufwändige Rechtevergaben, bzw. –Rücknahmen
  - wenig geeignet für größere Unternehmen, Web-Services ..

Modelle für MAC – Bell-La Padula-Modell

- Bislang keine Kontrolle von Informationsflüssen
- Lösung: Multi-levelSecurity (MLS),Labeling-Konzepte
- erstes formalisiertes Modell: BLP
  - Zugriffsoperationen

    read, write, exec, append, control
  - Systembestimmte Regeln:
    - no-read-up
    - no-write-down
    - strong tranquility

      (keine Änderung der Klassifikation zur Laufzeit)



Modelle für MAC – Bell-La Padula-Modell

- Grenzen von Bell-La Padula:
  - sukzessive Höherstufung von Information/Objekten
  - Blindes Schreiben möglich
  - Keine Integrität

- Fazit:
  - wichtiges Modell zur strukturieren Klassifizierung von Information
  - einfach zu Implementieren
  - „nur" Teil von umfassenderen Sicherheitsregularien

Modelle für MAC – Rollen basiertes Modell

RBAC-Modell (Role-based Access Control)

- Aufgabenorientierte Rechtevergabe durch Rollen
- Rolle: beschreibt bestimmte Aufgabe mit damit verbundenen Verantwortlichkeiten und Berechtigungen
- Nachbilden von Organisationsstrukturen:
  Rechte und Verantwortlichkeiten sind häufig direkt aus den Organigrammen ableitbar
- Erfüllen der Prinzipien: need-to-know, separation-of-duty
- Weit verbreitet: u.a. integriert in gängige Systeme wie:
  - ERP (Enterprise ResourcePlanning)-Systeme (u.a. SAP)
  - CMS (Content-ManagementSysteme), ...

Modelle für MAC – Rollen basiertes Modell

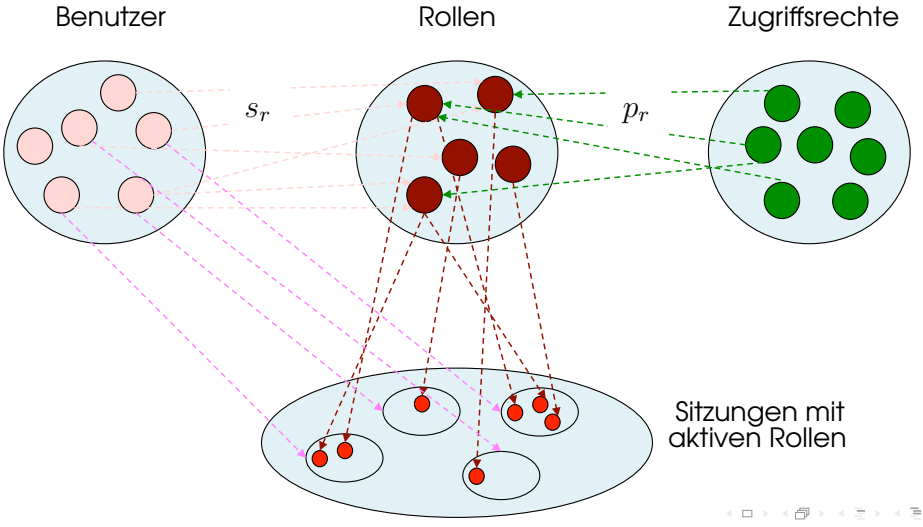
Komponenten eines (einfachen) RBAC-Modells

- Menge von Subjekten $=$ Benutzer
- Menge von Rollen Role, Rolle $r \in \mathrm{Role}$
- Menge von Zugriffsrechten P (permission) für Objekte
- Zwei Abbildungen:
  - Benutzer-Rollenzuordnung $s_r : S \to 2^{Role}$
  - Rechte-Rollenzuordnung $p_r : \mathrm{Role} \to 2^P$
- Sitzung: $\mathrm{session} \subseteq S \times 2^{Role}$, $(s, RL) \in \mathrm{session}$, dann ist RL die Menge der aktiven Rollen des Benutzers s, $RL \subseteq s_r(s)$
- $R_i \in \mathrm{session}(s)$, falls $(s, RL) \in \mathrm{session} \wedge R_i \in RL$

  D.h. s agiert in Rolle R_i, falls s Mitglied in der Rolle R_i ist u. diese Rolle in einer Sitzung aktiviert hat

Modelle für MAC – Rollen basiertes Modell

Zusammenhang zwischen den einzelnen Komponenten



Rollenhierarchien

Ziel: Vereinfachung von Verwaltungsaufgaben

Nachbilden hierarchischer Organisationsstrukturen

Plan:

- Definition einer partiellen Ordnung $\leq$auf Rollen:

  $R_i, R_j \in \mathrm{Role}$: falls $R_j \leq R_i$, so besitzt R_i alle Rechte von R_j

  und ggf. noch zusätzliche Rechte

- Beispiel: Software-Entwickler $\leq$ Projekt-Leiter:

  Rechte des Entwicklers: `r,w,x` auf Projekt-Dateien

  Rechte des Leiters: `r,w,x` auf Projekt-Dateien und
  `r,w,x` auf Projekt-Budget-Dateien, etc.

- Vererbung der Rollenmitgliedschaft: falls $R_j \leq R_i$, dann gilt:

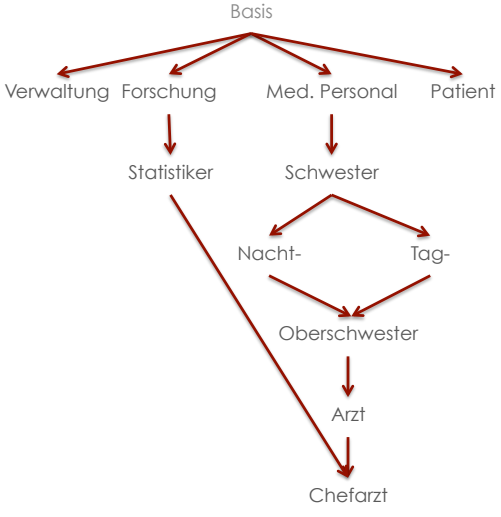
$$\forall s \in S : \; R_i \in s_r(s) \Longrightarrow R_j \in s_r(s)$$

Rollenhierarchien Beispiel: Krankenhaus

Rollen und deren Berechtigungen im Krankenhausszenario:

- Ärzte
  - ganze Patientenakte im Behandlungszusammenhang (außer besonders sensible Daten), (lesend, schreibend)
  - abteilungsinterne Daten aller Aufenthalte
- Pflegekräfte
  - Zugriff auf Krankenakte; Umfang durch Abteilungsleiter festgelegt
- Auszubildende
  - erforderlicher Umfang durch verantwortlich Lehrenden festgelegt(im Rahmen seiner eigenen Befugnisse).
- Verwaltungsmitarbeiter
  - Stammdaten, (lesend, schreibend)
  - abrechnungsrelevante Daten (u. U. auch besonders sensible!).

Rollenhierarchien Beispiel: Krankenhaus



Eigenschaften die ein RBAC-System zusätzlich garantieren muss!

- Ein Subjekt darf nur in solchen Rollen aktiv sein, in denen es Mitglied ist
- Ein Subjekt besitzt nur die Rechte seiner aktiven Rollen

Aufgabentrennung

Statische Aufgabentrennung

- Wechselseitiger Ausschluss von Rollenmitgliedschaften
- $R_1 =$ Kassenprüfer von Filiale$_A$
- $R_2 =$ Kassierer in Filiale$_A$
- Es gibt kein Subjekt welches in beiden Rollen Mitglied ist

Dynamische Aufgabentrennung

- Wechselseitiger Ausschluss von Rollenaktivitäten
- $R_3 =$ Kundenbetreuer
- $R_4 =$ Kontoinhaber
- Es gibt kein Subjekt welches in diesen beiden Rollen gleichzeitig aktiv sein kann.

Fazit:

- Rollenkonzepte sind sehr flexibel verwendbar, skalieren gut
- Modellierung zusätzlicher Zugriffsbeschränkungen durch Relationen auf Rollen möglich
- Direktes Nachbilden bekannter Organisations-und Rechtestrukturen in Unternehmen: gute Basis für ID-Mgmt
- intuitive und relativ einfache Abbildung der Rollen auf Geschäftsprozesse (Workflows): Need-to-know-Rechtvergabe
- Konsequenz: einfache und effiziente Rechte-Verwaltung automatischer Rechteentzug bei Mitgliedschafts-Ende

Weitere Entwicklung:

- Administrationvon RBAC-Systeme
- Modellierung von kontextabhängigen Rechten
- RBAC Policy Engineering
- RBAC und Workflows
- Delegationskonzepte
- Integration von RBAC in Betriebssysteme
- Kontrolle von Informationsflüssen in RBAC

Weitere Modelle:

- Conflict of Interest Modelle (Chinese-Wall-Modell)
  - Idee: Zugriff auf Information hängt davon ab, ob zugreifende Subjekte in Klassen mit kollidierenden Interessen enthalten sind (z.B. Banken, Autohersteller, Ölfirmen...)
- Non-Interference Modelle
  - Idee: Effekte von Aktionen sind nur für berechtigte sichtbar
- Fragen der Zukunft?
  - Vertrauensbasierte Modelle
    - Identitätsbasiert
    - Verhaltensbasiert
  - Kontext-abhängige Modelle

Danke für Ihr Interesse

Fortsetzung folgt