

# X. Zugriffskontrolle

(wirkt ein bisschen altmodisch, heute usage control)

## X.1. Bell-LaPadula-Modell

Das Bell-LaPadula-Modell ist ein statisches Zugriffskontrollmodell. Oberstes Schutzziel: Confidentiality

### X.1.1. Definition

- Subjektmenge  $\mathcal{S}$
- Objektmenge  $\mathcal{O}$
- Menge von Zugriffsoperationen  $\mathcal{A} = \{read, write, append, execute\}$
- Menge  $\mathcal{L}$  von Security Levels mit einer partiellen Ordnung  $\leq$

Dabei implizieren *write*-Rechte die *read*-Rechte. (Beispiel:  $unclassified \leq confidential \leq secret \leq top\ secret$ )

Im Bell-LaPadula-Modell ist der Systemzustand ein Element aus  $\mathcal{B} \times \mathcal{M} \times \mathcal{F}$ , wobei:

- $\mathcal{B} = \mathcal{P}(\mathcal{S} \times \mathcal{O} \times \mathcal{A})$  aktuelle Zugriffe beschreibt (wer hat Zugriff auf was mit welcher Zugriffsart)
- $\mathcal{M}$  die Menge der Zugriffskontrollmatrizen (ACM) bezüglich der Subjekte in  $\mathcal{S}$  und der Objekte in  $\mathcal{O}$  ist. Die Elemente von  $\mathcal{M}$  haben die Form  $M = (M_{so})_{s \in \mathcal{S}, o \in \mathcal{O}}$  mit  $M_{so} \subseteq \mathcal{A}$  für alle  $s \in \mathcal{S}, o \in \mathcal{O}$ .
- $\mathcal{F}$  Dreitupel aus Funktionen enthält, den sog. Security Level Assignments. Hier gilt  $\mathcal{F} \subseteq \mathcal{L}^{\mathcal{S}} \times \mathcal{L}^{\mathcal{S}} \times \mathcal{L}^{\mathcal{O}}$ . Ein Dreitupel  $(f_s, f_c, f_o)$  hat dabei folgende Form und Bedeutung:
  - $f_s: \mathcal{S} \rightarrow \mathcal{L}$  gibt für jedes  $s' \in \mathcal{S}$  den maximalen Sicherheitslevel an
  - $f_c: \mathcal{S} \rightarrow \mathcal{L}$  gibt den gegenwärtigen (current) Sicherheitslevel an
  - $f_o: \mathcal{O} \rightarrow \mathcal{L}$  gibt für jedes  $o' \in \mathcal{O}$  den Sicherheitslevel an

Dabei gilt:  $f_c$  muss von  $f_s$  dominiert werden:  $\forall s' \in \mathcal{S}: f_c(s') \leq f_s(s')$ .

Ein Systemzustand heißt sicher, wenn er die folgenden drei Eigenschaften erfüllt:

- Simple Security Property (ss-Eigenschaft):  
ein Zustand  $(b, M, f)$  genügt der ss-Eigenschaft, falls  $\forall (s', o', a') \in b$  mit  $a' \in \{read, write\}$  gilt:  $f_s(s') \geq f_o(o')$  („no read up“)

- Star Property (\*-Eigenschaft):  
ein Zustand  $(b, M, f)$  erfüllt die \*-Eigenschaft, falls  $\forall (s', o', a') \in b$  mit  $a \in \{append, write\}$  gilt:  $f_c(s') \leq f_o(o')$  („no write down“)  
Weiterhin, falls ein  $(s', o', a) \in b$  mit  $a \in \{append, write\}$  und ein  $(\hat{s}, \hat{o}, \hat{a}) \in b$  mit  $s' = \hat{s}$  und  $\hat{a} \in \{read, write\}$  existiert, dann muss  $f_o(\hat{o}) \leq f_o(o')$  gelten. („Kein Nachrichtenfluss von high level object zu low level object.“)
- Discretionary Security Property (ds-Eigenschaft):  
ein Zustand  $(b, M, f)$  erfüllt die ds-Eigenschaft, falls  $\forall (s', o', a) \in b$  stets  $a \in M_{s'o'}$  gilt

### X.1.2. Basic Security Theorem

Werden ausgehend von einem sicheren Initialzustand nur sichere Übergänge durchgeführt, so erhält man einen sicheren Systemzustand.

### X.1.3. Nachteile

Leider sammelt sich im Bell-LaPadula-Modell Information „oben“, ein Deklassifizieren ist nicht möglich. In der Praxis werden etwa „trusted subjects“ eingeführt, um ein Deklassifizieren von Daten zu erlauben.

Weitere Nachteile:

- Integrität der Daten wird nicht mitbetrachtet (niedrigstufige user/Prozesse können evtl. höher eingestufte Objekte verändern)
- keine Forderung an die ACM, etwa darf allen  $s \in \mathcal{S}$  alle Rechte gegeben werden
- verdeckte Kanäle, z.B. die (Nicht)Existenz von Dateien, bleiben unberücksichtigt

### X.1.4. Vorteile

- handhabbar
- formal (d.h. für Beweise geeignet)

## X.2. Chinese-Wall-Modell

Zugriffsrechte hängen von der Vergangenheit ab  $\rightarrow$  z.B. kein Informationsfluss zwischen konkurrierenden Firmen (z.B. bei Unternehmensberatung)

### X.2.1. Definition

- Menge  $\mathcal{C}$  von Firmen
- Objektmenge  $\mathcal{O}$  (jedes Objekt gehört einer Firma)
- Subjektmenge  $\mathcal{S}$  (die Berater)
- Funktion  $y: \mathcal{O} \rightarrow \mathcal{C}$ , welche jedem Objekt die zugehörige Firma zuordnet
- Funktion  $x: \mathcal{O} \rightarrow \mathcal{P}(\mathcal{C})$ , welche jedem Objekt eine conflict-of-interest-Klasse zuordnet

Die Sicherheitsmarke (security label) eines Objekts  $o \in \mathcal{O}$  ist das Tupel  $(x(o), y(o))$ . Im Falle  $x(o) = \emptyset$  spricht man von „sanitized information“.

Eine Matrix  $M$  enthält Informationen über Zugriffe  $M = (M_{so})_{s \in \mathcal{S}, o \in \mathcal{O}}$  mit

$$M_{so} = \begin{cases} true, & \text{falls } s \text{ Zugriff auf } o \text{ hatte,} \\ false, & \text{sonst.} \end{cases}$$

Der Initialzustand ist  $M = (false)_{s \in \mathcal{S}, o \in \mathcal{O}}$ .

### X.2.2. Eigenschaften

- Simple Security Property (ss-Eigenschaft):  
 $s \in \mathcal{S}$  erhält Zugriff auf  $o \in \mathcal{O}$  nur, falls  $\forall o' \in \mathcal{O}$  mit  $M_{so'} = true$  gilt:  $y(o) = y(o')$  oder  $y(o) \notin x(o')$ .
- Star Property (\*-Eigenschaft):  
 ein  $s \in \mathcal{S}$  erhält Schreibzugriff auf ein Objekt  $o \in \mathcal{O}$  nur, falls  $s$  aktuell keinen Lesezugriff auf  $o' \in \mathcal{O}$  hat mit  $y(o) \neq y(o')$  oder  $x(o') = \emptyset$ . (Die \*-Eigenschaft verhindert die Weitergabe von Daten über Dritte.)

