Themen zur Computersicherheit

Autorisierung: theoretische Modelle

PD Dr. Reinhard Bündgen
buendgen@de.ibm.com

Ziel der Autorisierung

Zugriffskontrolle

- Kontrolle des Zugriffs auf Ressourcen
- Wer darf mit welchen Ressourcen was tun?
  - Subjekte: Teilnehmer, Rollen, Prozesse
  - Objekte: Dateien, Sockets, Programme, Prozesse
  - Zugriffsarten: lesen, schreiben, verändern, ausführen, öffnen, erzeugen, tracen, …
- Durchsetzung der Zugriffskontrolle
  - von Nutzer bestimmt (discretionary access control – DAC)
  - vom System bestimmt (mandatory access control – MAC)

Modelle zur Autorisierung

- Zugriffsmatrix
- Rollenbasierter Zugriff
- Bell-LaPadula

Zugriffsmatrix

Definition Seien

- t ein Zeitpunkt
- O_t eine Menge von Objekten zum Zeitpunkt t
- S_t eine Menge von Subjekten zum Zeitpunkt t mit $S_t \subseteq O_t$
- A eine endliche Menge von Zugriffsrechten

dann ist M_t: S_t x $O_t \rightarrow 2^A$ eine *Zugriffsmatrix zum Zeitpunkt t* wobei $M_t(s,o) = \{a_1,...,a_n\}$ heißt, das Subjekt s hat zum Zeitpunkt t die Zugriffsrechte $a_1,...,a_n$ für Objekt o.

Notation
2^A: Potenzmenge von A

Objekte Subjekte	Datei 1	Datei 2	Uwe	Prozess 1	Prozess 2	Prozess 3	Prozess 4
Uwe						execute	
Prozess 1					control, send		control
Prozess 2				wait, signal		control	
Prozess 3	read, write	write, owner			receive		send
Prozess 4			create, delete			send receive	

Dynamische Zugriffsmatrizen

- Statische Zugriffsmatrix: $M = M_t$ für alle Zeitpunkte t
- Dynamische Zugriffsmatrix: M_t kann sich mit der Zeit ändern
  - Änderung der Rechte:
    - für t' > t: $M_{t'}$= change (M_t,M^+ ,M^-) mit $M_{t'}(s,o) = M_t(s,o) \cup M^+(s,o) \setminus M^-(s,o)$ für alle s,o
  - Änderung der Subjekt und Objektmengen
  - Beispiele
    - Änderung von Dateizugriffsrechten (chmod)
    - Erzeugung / Löschung von Dateien, Prozessen

Beschreibung von Sicherheitseigenschaften

Safety-Problem

- Die Relation $\models_{S'}$ beschreibt eine mögliche Änderung einer Zugriffsmatrix dadurch, dass ein Subjekt $s \in S' \subseteq S$ eine ihm erlaubte Operation ausführt.
- Sei $\models_{S'}^*$ die reflexiv-transitive Hülle von $\models_{S'}$
- Frage: Sei a eine unzulässiges Zugriffsrecht auf das Objekt o für das Subjekt s. Gibt es zu gegebener Zustandsmatrix M_t zum Zeitpunkt t einen Zeitpunkt t' und erlaubte Operationen von Subjekten in S', so dass $M_t \models_{S'}^* M_{t'}$, und $a \in M_{t'}(s,o)$?
- Das Safety-Problem ist unentscheidbar.

Soll-Ist Vergleiche

- Gegeben eine möglicherweise informelle Spezifikation der gewünschten Zugriffsrechte und Zugriffsbeschränkungen: Erfüllt ein System mit gegebener Zugriffsmatrix diese Spezifikation?
- Zugriffsmatrix beschreibt keine Zugriffsbeschränkungen.
- Zugriffsrechte können in Zugriffsmatrix implizit beschrieben werden
  - execute $\in$ M(Uwe, Process3), read $\in$ M(Prozess3, Datei1) => Uwe kann Datei1 lesen

Rollenbasiertes Zugriffsmodell

- role based access control (RBAC)

Definition Seien

- U eine Menge von Teilnehmern,
- O eine Menge von Objekten,
- R eine Menge von Rollen,
- A eine Menge von Rechten,
- ur: $U \rightarrow 2^{R,}$ mit s darf Rolle r einnehmen wenn $r \in ur(u)$,
- ra: $R \rightarrow 2^{A}$ mit r hat Recht a wenn $a \in ra(r)$,
- ses $\subseteq U \times 2^{R}$ die Menge der aktuellen Sitzungen, wobei $R' \subseteq ur(u)$ für alle $(u,R') \in ses$

Dann gilt für die Sitzung (u,R'), dass u die Rechte $\cup_{r \in R'} ra(r)$ hat.

RBAC Beispiel

Beispiel : Bank

- Teilnehmer: Sonja, Uwe, Klaus
- Rollen: Kunde, Kassierer, Kundenbetreuer, Zweigstellenleiter, Kassenprüfer
- Objekte: Kundenkonten, Personaldaten, Kundendaten; Kreditdaten
- Rechte: Geld einzahlen, Geld abheben, Konto sperren, Kreditrahmen erhöhen
- ur(Sonja) = {Zweigstellenleiter, Kunde}
- ur(Uwe) = {Kassierer, Kundenbetreuer, Kunde}
- ur(Klaus) = {Kassenprüfer}
- Einschränkungen:
  - kein Teilnehmer darf sowohl die Rolle eines Kassierers als auch die eines Kassenprüfers haben (statischen Aufgabentrennung)
  - keine Sitzung darf sowohl die Rollen eines Kundenbetreuers als auch die eines Kunden haben (dynamische Aufgabentrennung)

Hierarchische rollenbasierte Modelle

- Gegeben eine partielle Ordnung ≥ über den Rollen mit aus $r_1 \geq r_2$ folgt, dass r_1 alle Zugriffsrechte von r_2 hat.
- Beispiel:
  - Zweigstellenleiter ≥ Kassierer
  - Zweigstellenleiter ≥ Kundenbetreuer
  - Kassenprüfer ≥ Kassierer

partielle Ordnung:
- reflexiv,
- transitiv,
- anti-symmetrisch

Das Bell-LaPadula Modell

- 1973 von D. E. Bell und L. J. LaPadula (im Auftrag der US Air Force) entwickelt
- Oft auch mit Multi Level Security (MLS) bezeichnet
- Ziel: Schutz vertraulicher Information
- Implementierungen:
  - trusted Solaris
  - SELinux
  - z/OS - RACF

Bell-La Padula Modell - Definitionen

- aufbauend auf dynamischen Zugriffsmatrixmodell
- Zugriffsrechte: A = {read-only, append, execute, read-write, control}
- Geordnete Menge von Sicherheitsmarken M
  - z.B. Zugriffsklassifizierung (Ermächtigung/Einstufung)
- Menge von Sicherheitskategorien K
  - z.B. Rollen mit Zugriff (Zuständigkeitsbereich)
- Geordnete Menge von Sicherheitsklassen $C = M \times 2^K$
  - mit $(m,k) \leq (m',k') \iff (m \leq m' \wedge k \subseteq k')$
- sc: $S \rightarrow C$ ist die Clearance eines Subjekts
- sc: $O \rightarrow C$ ist Klassifikation eines Objekts
- Ein Subjekt kann sich mit der aktuellen Sicherheitsklasse $sc_{akt}(s)$ einloggen wenn $sc_{akt}(s) \leq sc(s)$

BLP Beispiel Krankenhaus

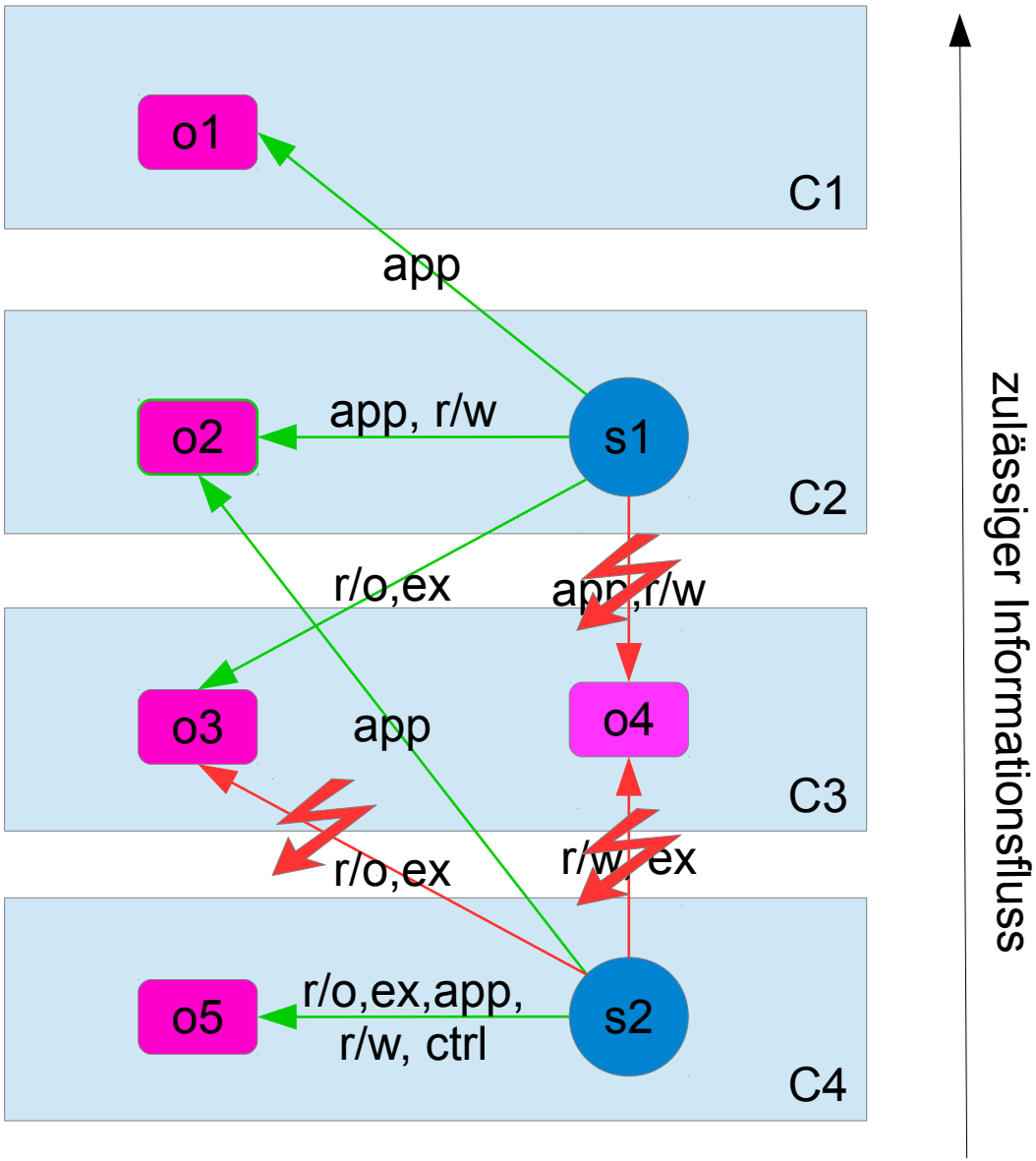
- Subjekte S = { Achim, Claudia, Lotte, Frieder, Hans }
- Objekte O = { Krankenakte, Forschungsergebnis, Behandlungsplan, Gehaltsabrechnung }
- Marken M = { geheim, vertraulich, unklassifiziert } mit geheim ≥ vertraulich ≥ unklassifiziert
- Sicherheitskategorien K = { Chefarzt, Arzt, Schwester, Patient, Verwaltung, Besucher }
- Sicherheitsklassen C = { (geheim, ∅), (geheim, {Chefarzt}), (geheim, {Chefarzt, Arzt}), (vertraulich, ∅), (vertraulich, {Arzt, Schwester}),(vertraulich, {Schwester}), (vertraulich, {Verwaltung}), … }
- Ordnung über C:
  - (geheim, ∅) ≥ (vertraulich, ∅),
  - (vertraulich, {Arzt, Schwester}) ≥ (vertraulich, {Arzt})
  - ¬ ((vertraulich, {Schwester}) ≥ (vertraulich, {Verwaltung}))
  - ¬ ((vertraulich, {Schwester}) ≤ (vertraulich, {Verwaltung}))

BLP Zugriffsregeln

- Ziel: verhindere unzulässige Informationsflüsse
- Simple security / no-read-up
  - $s \in S$ darf zum Zeitpunkt t den Zugriff $a \in$ {read, execute} auf Objekt $o \in O$ durchführen, wenn $a \in M_t(s,o) \wedge sc(s) \geq sc(o)$
- ✱-Eigenschaft / no-write-down
  - $s \in S$ darf zum Zeitpunkt t den append Zugriff auf Objekt $o \in O$ durchführen, wenn append $\in M_t(s,o) \wedge sc(s) \leq sc(o)$
  - $s \in S$ darf zum Zeitpunkt t den read-write Zugriff auf Objekt $o \in O$ durchführen, wenn read-write $\in M_t(s,o) \wedge sc(s) = sc(o)$

Beispiel BLP Zugriffsregeln

- C1, C2, C3, C4 ∈ C
- C1 > C2 > C3 > C4
- s1, s2 ∈S
- o1, o2, o3, o4, o5 ∈O
- Zugriffe
  - app = append
  - r/w = read-write
  - r/o = read-only
  - ex = execute



Diskussion von BLP

- Information sammelt sich „oben“
  - vertrauenswürdige Subjekte (trusted subjects) dürfen ✱-Eigenschaft verletzen
- append wird Subjekt mit niedrigster Klassifikation nie verboten (außer über Zugriffsmatrix)
- Informationsfluss über verdeckte Kanäle
  - z.B. über (Nicht)Existenz von Objekten auf die nur mit append zugegriffen werden kann
- Modellierung unterschiedlicher Sitzungen/Rollen eine Teilnehmers zu unterschiedlichen Zeitpunkten
  - Chinese Wall Model

Übung

Fallbeispiel Universität

- Rollen: Professoren, Wiss. Mitarbeiter, Prüfungsausschuss, Sekretäre, Studierende, Angestellte im Studentensekretariat, Eltern der Studierenden, Polizisten, Journalisten, ...
- Objekte: Forschungsergebnisse, Vorlesungsfolien, Übungsblätter, Noten Studierendenausweise, e-mail-Adressen, Fachbereichsbudget, …
- Wer soll Zugriff auf welche Daten (Objekte) haben?
- Welche Zugriffsrechtmodellierung passt am besten?