Modellierung und formale Analyse von Betriebssystem-Sicherheitspolitiken

Fachgruppentreffen Betriebssysteme

Peter Amthor

Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Fachgebiet Verteilte Systeme und Betriebssysteme

peter.amthor@tu-ilmenau.de

2018-10-19

Überblick

Sicherheitsmechanismen in Betriebssystemen

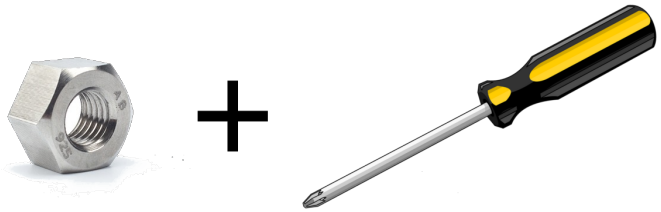
Was uns zur Verfügung steht ...

- Zugriffssteuerungsmechanismen
- kryptographische Mechanismen
- μKernel-, Exokernel-, (Para-) Virtualisierungsarchitekturen
- Codeverifikation, typsichere Sprachen
- Trusted Hardware

Sicherheitsmechanismen in Betriebssystemen

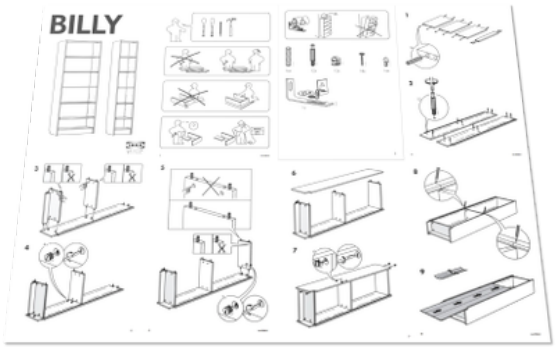
... und wie wir es einsetzen

- Wann **brauche** ich all dies?
  → *TCB-Größe* [1]

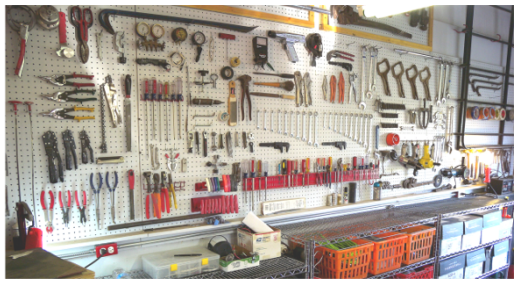


- Wie **interagiert** all dies?
  → *Sicherheitsarchitektur*

  [nächster Vortrag ...]



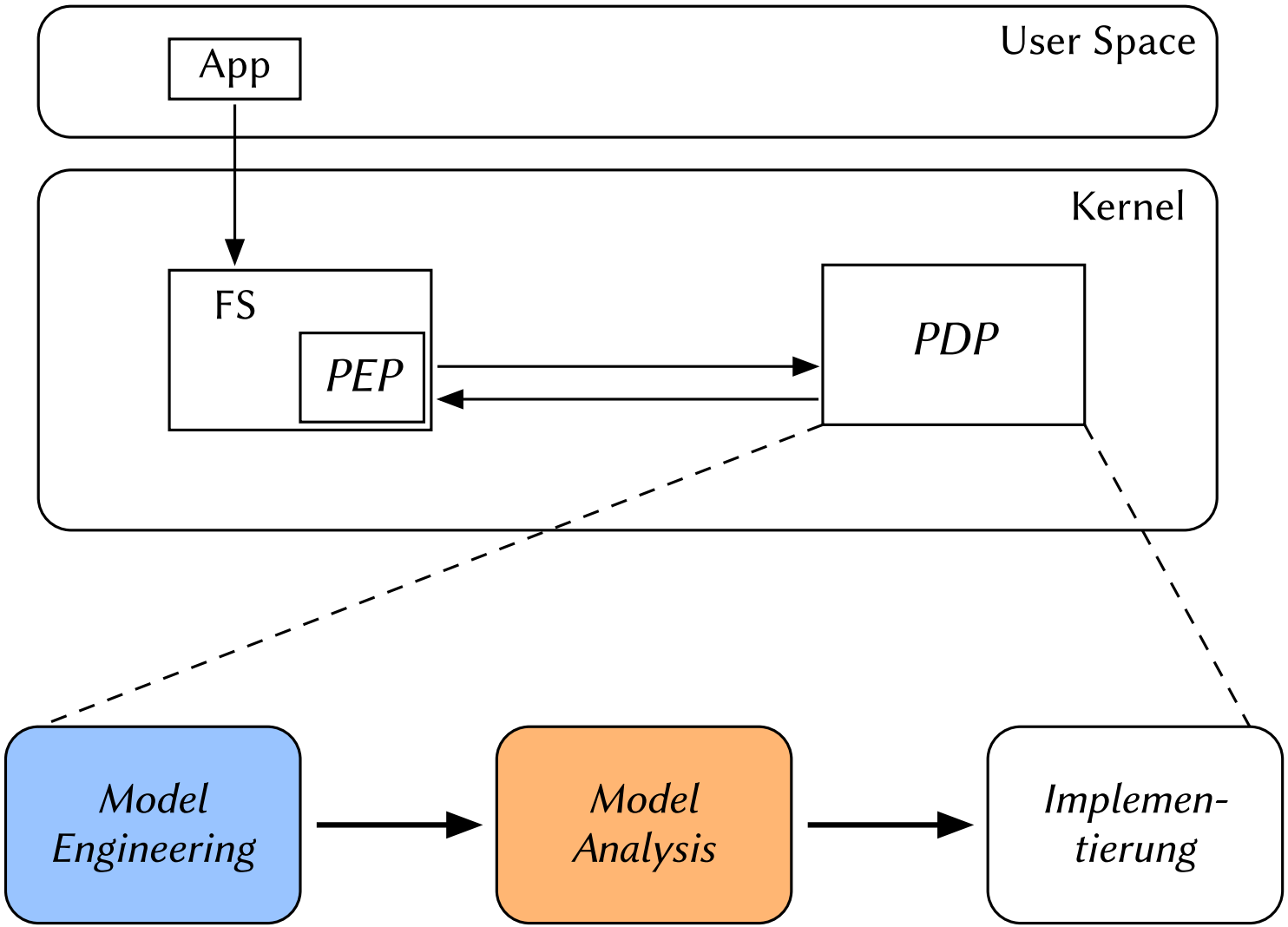
- Wie **steuere** ich all dies?
  → *Sicherheitspolitik*



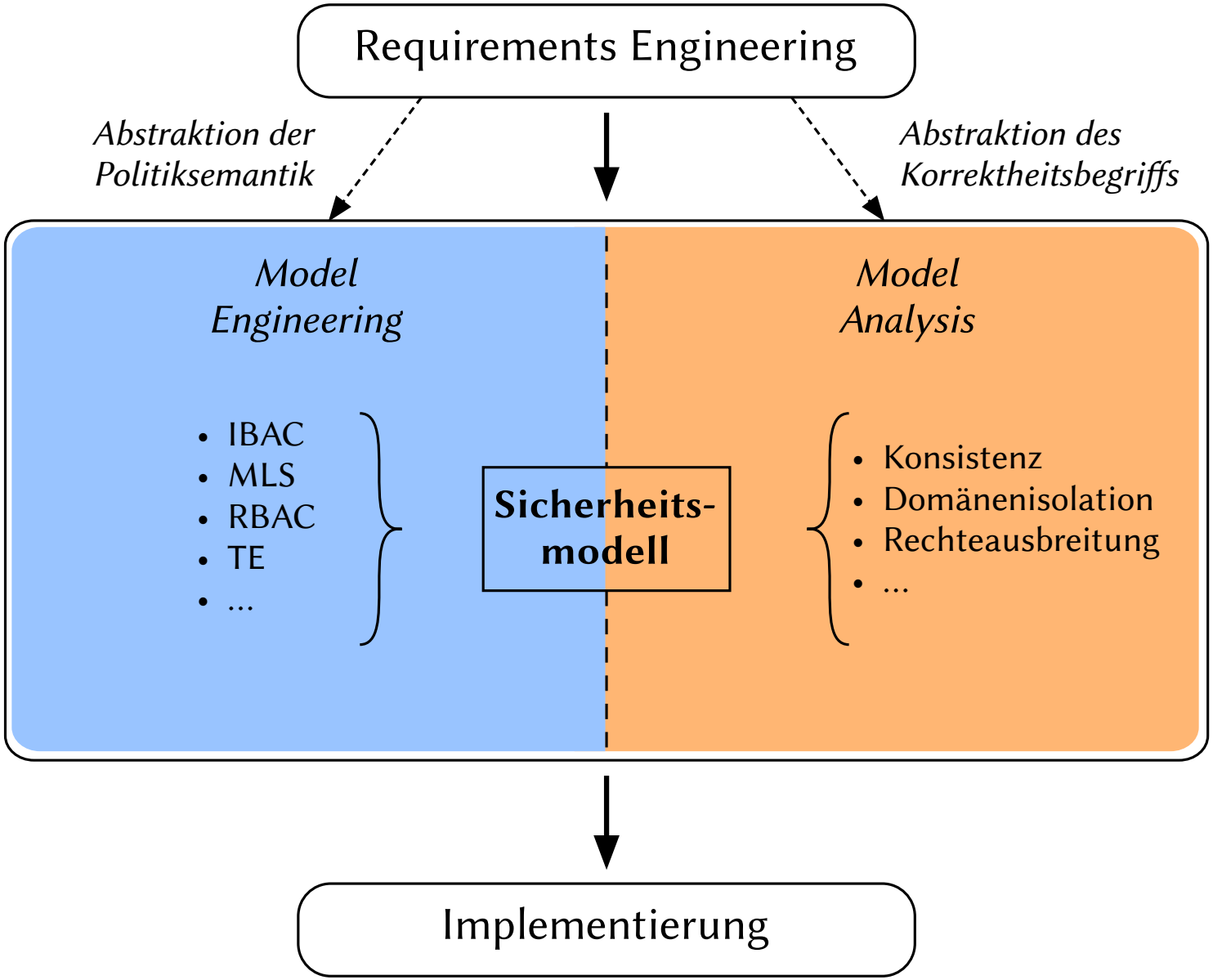
[1] Anja Pölck. *Small TCBs of Policy-controlled Operating Systems.*

Korrektheit einer Sicherheitspolitik

Der modellbasierte Weg



Modellbasiertes Security Engineering



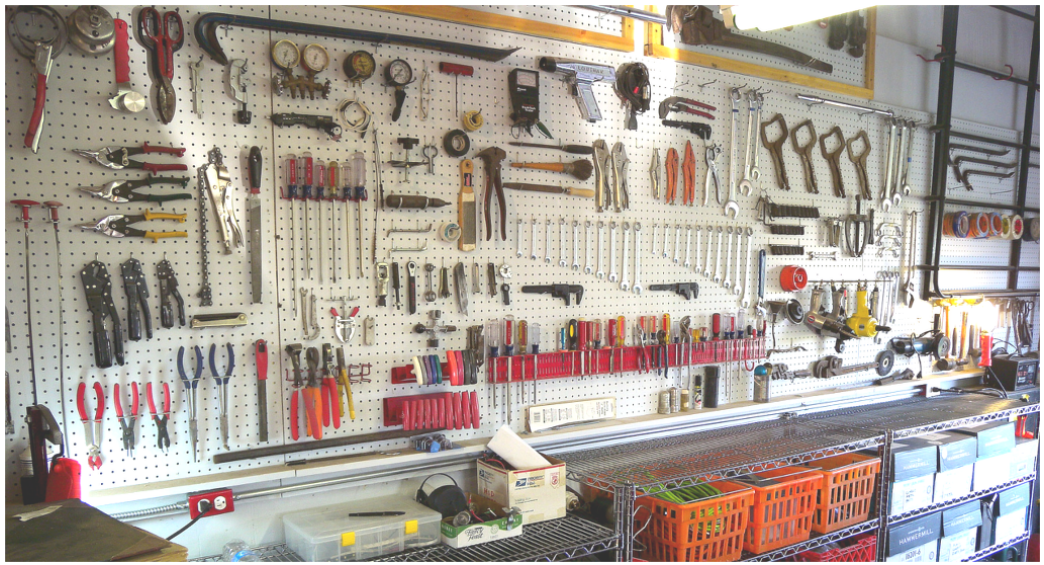
Modellbasiertes Security Engineering

In der Praxis

Welche **Modellabstraktionen** brauche ich zur präzisen Formalisierung

1. der *Politiksemantik*?
2. des Korrektheitsbegriffs (*Sicherheitseigenschaft*)?

→ Ordnung im Werkzeugkasten



Idee

~~Baukasten~~ Werkzeugkastenprinzip:

Ordnung schaffen ...

...unter Modellabstraktionen

1. Politiksemantik-Klassen ($\longrightarrow$ *Model Engineering*)

IBAC: $access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow \langle user(\text{proc}_i), \text{write} \rangle \in acl(\text{file}_j)$

MLS: $access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow sensitivity(\text{file}_j) \leq clearance(\text{proc}_i)$

```
-rwxr-xr-x 1 peter peter 6604 1 Oct 2017 a.out
```

$allow(domain(\text{proc}_i), type(\text{file}_j), class(\text{file}_j))$

Ordnung schaffen ...

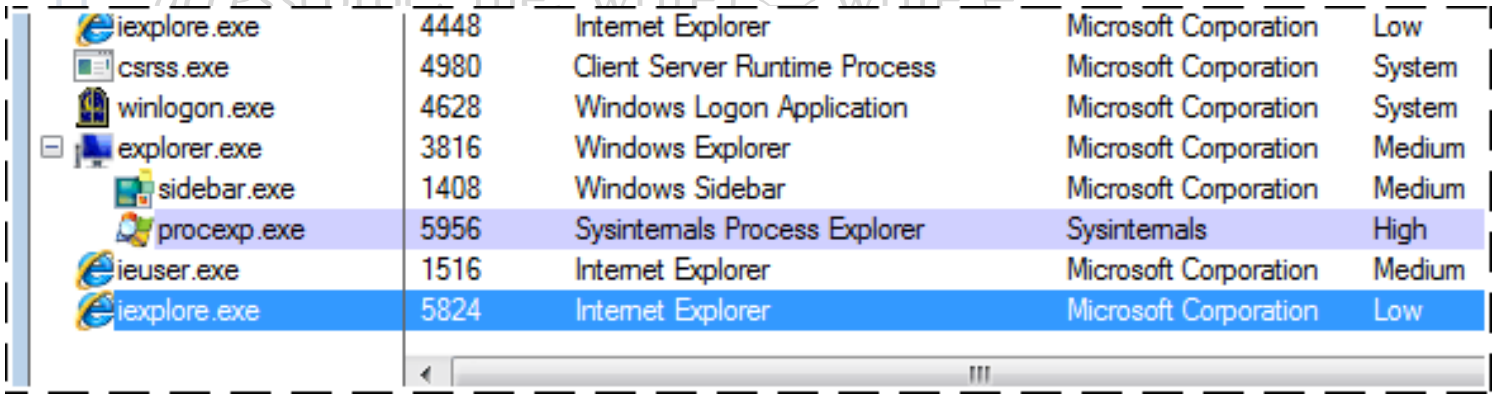
...unter Modellabstraktionen

1. Politiksemantik-Klassen ($\longrightarrow$ *Model Engineering*)

IBAC: $access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow \langle user(\text{proc}_i), \text{write} \rangle \in acl(\text{file}_j)$

MLS: $access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow sensitivity(\text{file}_j) \leq clearance(\text{proc}_i)$

TE: $access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow \text{write} \in$



Ordnung schaffen ...

...unter Modellabstraktionen

1. Politiksemantik-Klassen ($\longrightarrow$ *Model Engineering*)

IBAC: $access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow \langle user(\text{proc}_i), \text{write} \rangle \in acl(\text{file}_j)$

MLS: $access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow sensitivity(\text{file}_j) \leq clearance(\text{proc}_i)$

TE: $access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow \text{write} \in allow(domain(\text{proc}_i), type(\text{file}_j), class(\text{file}_j))$

```
allow system_t etc_t : file {read execute}
```

Ordnung schaffen ...

...unter Modellabstraktionen

1. Politiksemantik-Klassen ($\longrightarrow$ *Model Engineering*)

$$\text{IBAC:}\quad \mathit{access}(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow \langle \mathit{user}(\text{proc}_i), \text{write} \rangle \in \mathit{acl}(\text{file}_j)$$

$$\text{MLS:}\quad \mathit{access}(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow \mathit{sensitivity}(\text{file}_j) \leq \mathit{clearance}(\text{proc}_i)$$

$$\text{TE:}\quad \mathit{access}(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow \text{write} \in \mathit{allow}(\mathit{domain}(\text{proc}_i), \mathit{type}(\text{file}_j), \mathit{class}(\text{file}_j))$$

$\longrightarrow$ **Attributierungsprinzip** verallgemeinern!

Ordnung schaffen ...

...unter Modellabstraktionen

2. dynamische Sicherheitseigenschaften ($\longrightarrow$ *Modellanalyse*)

$$access(\text{proc}_i, \text{file}_j, \text{write}) = false$$

$$\downarrow \delta^*$$

$$access(\text{proc}_i, \text{file}_j, \text{write}) = true$$

$\longrightarrow$ **Zustandsübergänge** verallgemeinern!

Entity Labeling

Formalisierung der Politiksemantik

- **Ziel:** Verallgemeinerung des *Attributierungsprinzips* in Betriebssystem-Zugriffssteuerungspolitiken
- **Kategorien** von Modellabstraktionen:

  Entity Sets (ES) : Endpunkte eines Zugriffs (*Entitäten*)
  Label Sets (LS) : sicherheitsrelevante Metainformationen (*Labels*)
  Label Assignments (LA) : $ES \rightarrow LS$
  Authorization Rules (AR) : $LS \rightarrow \mathbb{B}$

Model Engineering mit Entity Labeling

Beispiel: MLS-Politik

$$access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow sensitivity(\text{file}_j) \leq clearance(\text{proc}_i)$$

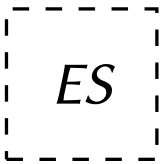
ES: $P = \{\text{proc}_i \mid i \in \mathbb{N}\}, F = \{\text{file}_j \mid j \in \mathbb{N}\}$

LS: $C = \{\text{low}, \text{med}, \text{high}\}$

LA: $clearance : P \to C, sensitivity : F \to C$

AR: $\leq \subseteq C \times C$

$\mathbb{B} = \{true, false\}$



Model Engineering mit Entity Labeling

Beispiel: MLS-Politik

$$access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow sensitivity(\text{file}_j) \leq clearance(\text{proc}_i)$$

ES: $P = \{\text{proc}_i \mid i \in \mathbb{N}\}, F = \{\text{file}_j \mid j \in \mathbb{N}\}$

LS: $C = \{\text{low}, \text{med}, \text{high}\}$

LA: $clearance : P \to C, sensitivity : F \to C$

AR: $\leq \subseteq C \times C$

$\mathbb{B} = \{true, false\}$

Model Engineering mit Entity Labeling

Beispiel: MLS-Politik

$$access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow sensitivity(\text{file}_j) \leq clearance(\text{proc}_i)$$

ES: $P = \{\text{proc}_i \mid i \in \mathbb{N}\}, F = \{\text{file}_j \mid j \in \mathbb{N}\}$

LS: $C = \{\text{low}, \text{med}, \text{high}\}$

LA: $clearance : P \to C, sensitivity : F \to C$

AR: $\leq \subseteq C \times C$

$\mathbb{B} = \{true, false\}$

Model Engineering mit Entity Labeling

Beispiel: MLS-Politik

$$access(\text{proc}_i, \text{file}_j, \text{write}) \Leftrightarrow sensitivity(\text{file}_j) \leq clearance(\text{proc}_i)$$

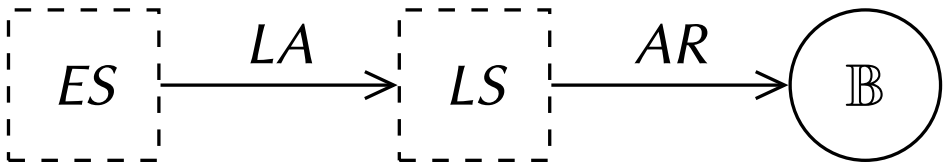
ES: $P = \{\text{proc}_i \mid i \in \mathbb{N}\}, F = \{\text{file}_j \mid j \in \mathbb{N}\}$

LS: $C = \{\text{low}, \text{med}, \text{high}\}$

LA: $clearance : P \to C, sensitivity : F \to C$

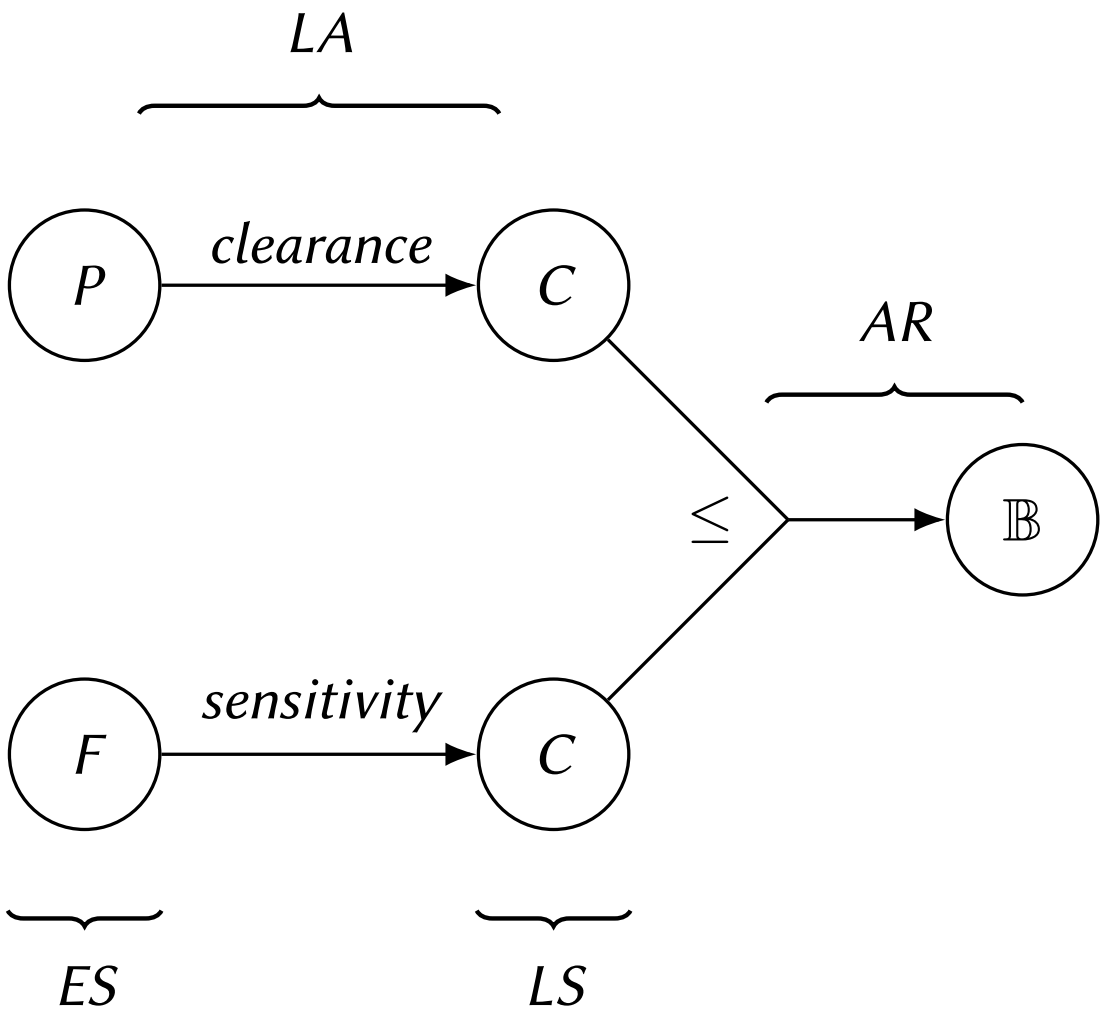
AR: $\leq \subseteq C \times C$

$\mathbb{B} = \{true, false\}$



Model Engineering mit Entity Labeling

Ergebnis



Modellanalyse

Formalisierung dynamischer Sicherheitseigenschaften

- **Ziel:** Verallgemeinerung von *Zustandsübergängen* in Betriebssystem-Zugriffssteuerungspolitiken
- **zusätzliche Kategorien** von Modellabstraktionen:

  Relabeling Rules (*RR*) steuern erlaubte Label-Änderungen
  Model Constraints (*MC*) schränken sie ein
- Analysemodell: Zustandsautomat

$$\langle Q, \Sigma, \delta, \lambda, Ext \rangle$$

  - Zustand $q_i \in Q : ES \times LS \times LA$
  - Zustandsübergang $q_i \xrightarrow{\delta} q_j$: *AR*, *RR* und *MC*

$\longrightarrow$ was wir damit tun können ...

Machbarkeitsstudie

Analyse von SELinux-Sicherheitspolitiken

Szenario:

- Zustandsextraktion:
  - 390 000 Entitäten
  - 2 900 Labels
- Politikgröße:
  - 131 000 *Authorization Rules*
  - 4 330 *Relabeling Rules*

Ergebnisse:

- konkretes Entity-Labeling-Modell: *SELX*
- konkretes Analyseziel (potenzielle Rechteausbreitung): *Type Safety*
- Verfahren: heuristisch gesteuerte Simulation, angepasst für SELinux

Zusammenfassung

1. **Entity Labeling:** Kategorisierung von Modellabstraktionen
2. **Modellanalyse** auf potenzielle Rechteausbreitung:
   - Grundlage zur Definition der Sicherheitseigenschaft
   - Grundlage einer simulativen Analysemethode
3. **Analyseergebnisse:** bislang quantitativ

~~Future~~ Ongoing Work:

- Model Engineering: methodisches Regelwerk, Visualisierungs- und Beschreibungssprachen, durchgehende Werkzeugunterstützung
- Modellanalyse: heuristische Strategien, entscheidbare Modellklassen
- Anwendung: praktische Szenarien erweitern (*Android*)

Modellierung und formale Analyse von Betriebssystem-Sicherheitspolitiken

Fachgruppentreffen Betriebssysteme

Peter Amthor

Technische Universität Ilmenau
Fakultät für Informatik und Automatisierung
Fachgebiet Verteilte Systeme und Betriebssysteme

peter.amthor@tu-ilmenau.de

2018-10-19