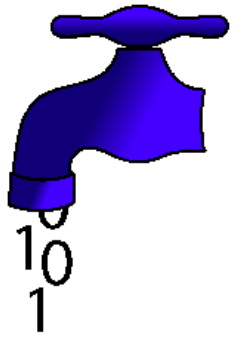
Schutz vor der ausführenden Maschine

Josef Spillner
js177634@inf.tu-dresden.de

17.06.2003

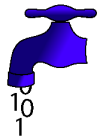


http://mindx.dyndns.org/uni/papers/

Thema

- **Untrusted Hosts and Confidentiality: Secure Program Partitioning**

- **von Steve Zdancewic, Lantian Zheng, Nathaniel Nystrom und Andrew C. Myers**



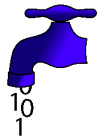
Problembeschreibung

- **Informationsverarbeitung erfordert verteiltes Rechnen**
- **Sicherheitspolicies zur Laufzeit schwer durchsetzbar**
- **Einsatz von security typing**

Herkömmliche Methoden

- **Zugriffskontrolle (ACLs) statt Informationsflusskontrolle**
- **Java: Sandbox, Stack-Kontrolle, private Variablen**
- **Laufzeitüberprüfung:**

```
H = 1
L = 0
if H == 1: L = 1
```

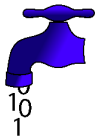


Implementierungen

- **Erweiterungen vorhandener Sprachen**
- **Splitter teilt Code je nach policies auf**
- **`jif` für Java-Programme**

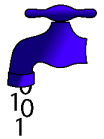
Beispiel

```
public class OTExample {
  int{Alice:; ?:Alice} m1, m2;
  boolean{Alice:; ?:Alice} isAccessed;
  int{Bob:} transfer{?:Alice} (int{Bob:} n)
  where authority(Alice) {
    int tmp1 = m1; int tmp2 = m2;
    if(!isAccessed) {
      isAccessed = true;
      if(endorse(n, {?:Alice}) == 1)
        return declassify(tmp1, {Bob:});
      else
        return declassify(tmp2, {Bob:});
    }
    else return 0;
  }
}
```



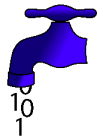
Verteiltes Rechnen

- `jif/split` berechnet Verteilung
- Überprüfung der Sicherheitslabels
- Zu wenige oder unpassende Hosts: Programm ist unsicher
- Compiler momentan noch in Trusted Computing Base; Zertifizierungen können Abhilfe schaffen

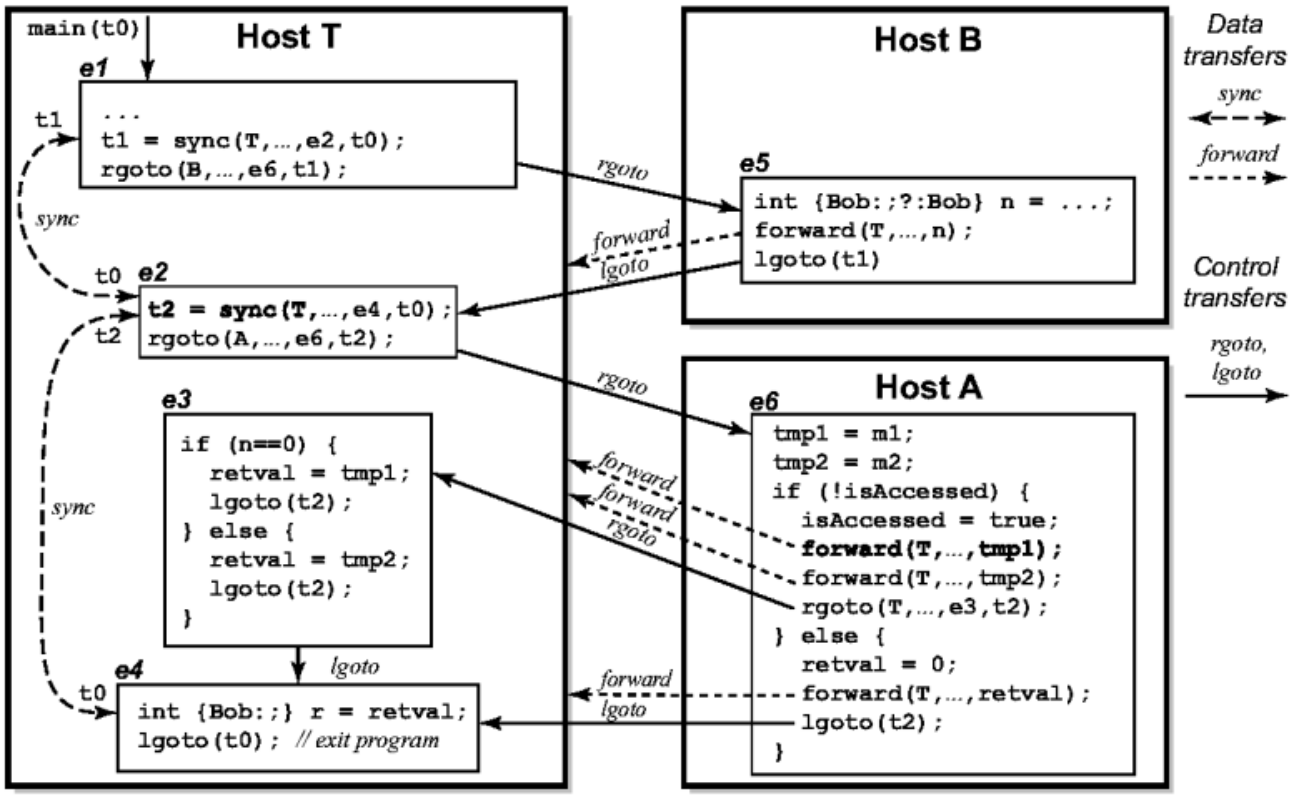


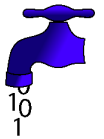
Bekanntes Problem

- Vertrauenswürdige Hosts H1, H2
- Nicht vertrauenswürdiger Host L dazwischen
- Problem: H2 kann Privilegien nicht nutzen (Angriff von L)
- Lösung: Einführung des ICS (integrity control stack)



Veranschaulichung



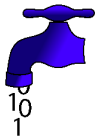


Optimierungen

- **Lokale Aufrufe werden nicht über das Netzwerk gesendet**
- **Keine Hashwert-generierung für lokale Aufrufe**
- **Aggregation mehrerer konsekutiver Nachrichten**

Bewertung

- **Sicherheit ohne erhöhte Anforderungen an Speicher oder Prozessor**
- **Randaspekte: mehrere Threads, verteilte Systeme**
- **Eigene Sprachen (SPL@) oder Spracherweiterungen (`jif`)**
- **DRM-Implementierungen?**



Quellen

- **Untrusted Hosts and Confidentiality:**
  Secure Program Partitioning (Cornell University)
  `http://www.cs.cornell.edu/zdance/zznm01.ps`

- **Language-Based Information-Flow Security**
  `http://www.cs.cornell.edu/andru/papers/jsac/sm-jsac03.pdf`