

5.3 Übungsaufgabe: ACL vs. Capability

Betrachten Sie folgende Zugriffskontroll-Matrix:

Objekt \ Subjekt	O ₁	O ₂	O ₃
A	r	w	r,w
B	-	x	r
C	r,w	-	x

- Wie sehen die dazugehörigen ACLs aus? Geben Sie dazu eine formelle Definition einer ACL und dann von aller in der Matrix verwendeten ACLs an.
Tipp: Formalisieren Sie zuerst die Zugriffskontroll-Matrix und leiten daraus eine Definition der ACL an.
- Wie sehen die dazugehörigen Capabilities aus? Geben Sie dazu eine formelle Definition einer Capability und dann aller in der Matrix verwendeten Capabilities an.
- Subjekt A delegiert seine Rechte an Subjekt B. Wie sieht die neue Zugriffskontroll-Matrix aus?
- Führen Sie die Delegation aus c) durch die Transformation der ACLs und Capabilities durch. Was ist in diesem Fall aufwändiger: ACLs oder Capabilities?
- O₂ wird aus der Ausgangsmatrix gelöscht. Wie sieht die neue Matrix aus?
- Führen Sie die Löschung aus e) durch die Transformation der ACLs und Capabilities durch. Was ist in diesem Fall aufwändiger: ACLs oder Capabilities?

ACLs

Objekt \ Subjekt	O ₁	O ₂	O ₃
A	r	w	r,w
B	-	x	r
C	r,w	-	x

Cap.

R=read, w=write, x=execute

- ACL= access controll list
Listet die Rechte aller Subjekte zu einem Objekt
ACLs sind also die spalten der Zugriffskontroll-Matrix

O₁: A read , B----, C read write

O₂:A write ,B execute , C----

O₃:A read-write , B-read, C execute

B)

Zugriffsausweis / Capability: Listet die Rechte eines Subjekts auf alle Objekt (Zeile aus der Zugriffskontrollmatrix)

A: O1 read O2 write O3 read-write

B: O1 – O2 execute O3 read

C: O1 read-write O2 – O3 execute

c)

Jemanden rechte deligieren->jemanden seine rechte geben. Man selber behält die rechte. So verstehe ich das

Objekt \ Subjekt	O ₁	O ₂	O ₃
A	<i>r</i>	<i>w</i>	<i>r, w</i>
B	<i>r</i>	<i>w</i>	<i>r, w</i>
C	<i>r, w</i>	-	<i>x</i>

d)

A: O1 read O2 write O3 read-write

B: O1 read O2 write-execute O3 read-write

C: O1 read-write O2 – O3 execute

O1: A read , B read, C read write

O2:A write ,B write execute , C----

O3:A read-write , B read-write, C execute

ACL ist aufwendiger, weil wir in 3 Listen einträge ändern müssen. Bei cap. Lediglich in der Liste von B

Aufwand ist also entsprechen der Anzahl der Objekte bei ACL

e)Ich gehe von der Ursprungs Matrix aus, vor der delegation

Objekt \ Subjekt	O ₁	O ₃
A	<i>r</i>	<i>r, w</i>
B	-	<i>r</i>
C	<i>r, w</i>	<i>x</i>

f)

O1: A read , B----, C read write

O3:A read-write , B-read, C execute

A: O1 read O3 read-write

B: O1 – O3 read

C: O1 read-write O3 execute

Cap ist aufwendiger, da wir in jeder liste den eitnrag von O2 löschen müssen, bei ACL nur die Liste von O2. Der aufwand bei cap. Ist also gleich die Anzahl der Subjekte