Schutzkonzepte in Multics

Benedict Herzog

Seminar Ausgewählte Kapitel der Systemsoftwaretechnik

Lehrstuhl für Informatik 4
(Verteilte Systeme und Betriebssysteme)

21. Dezember 2015

Motivation

- Mehrbenutzer- & Mehrprogrammbetrieb
  - Schutz von Subsystemen und Supervisor
  - Isolation und Vertraulichkeit von Daten/Prozeduren
  - Kontrolliertes Teilen von Daten/Prozeduren
  - Authentifizierung von Benutzer
- Fehler
  - Finden von Programmierfehler
  - Eindämmen von menschlichen/technischen Versagen

Motivation

Wir haben (noch immer) diesselben Probleme

- (Kontrolliertes) Teilen von Informationen
- Schutz der Privatssphäre
- Authentifizierung von Benutzer
- Fehlersuche und Fehlereindämmung

Motivation

> *"In trying to identify the ideas related to protection which were introduced by Multics, a certain amount of confusion occurs."*
>
> Jerome Saltzer (1974) [2]

Agenda

Motivation

Adressraumschutz

Dateisystemschutz

Authentifizierung

B2 Zertifizierung

Fazit

Agenda

Motivation

Adressraumschutz
- Entwurfsprinzipien
- Virtueller Speicher
- Schutzmodell

Dateisystemschutz

Authentifizierung

B2 Zertifizierung

Fazit

Entwurfsprinzipien

- Sicherheit braucht systematische Vorgehensweise
- Während der Entwicklung entstandene Prinzipien
  - Grundsatz: Alles verbieten, explizit erlauben
  - Jeden Zugriff überprüfen
  - Kein *Security through Obscurity*
  - Prinzip des geringsten Rechts
  - einfache/natürliche Schnittstelle
- Müssen Benutzern und Entwicklern kommuniziert werden

$\Longrightarrow$ Ziel: Flexible Schutzmechanismen

$\Longrightarrow$ Ziel: Dezentrale Konfiguration und Selbstverwaltung

Adressraumschutz

Adressräume vor Multics

- *Memory-Bounds-Register*
  $\longrightarrow$ Aufteilen des Adressraums in zwei Domänen[1]
- Alles-oder-Nichts Schutz
  $\longrightarrow$ unflexibel

Adressräume in Multics

- Zusammenfassen logisch zusammenhängender Informationen
  $\longrightarrow$ Segment
- Adressraum besteht aus Segmenten

[1] z.B. GE-635

Voraussetzungen des Schutzmodells

- Bei jeder Anmeldung wird ein neuer Prozess erzeugt
  - Prozessen wird der Benutzer zugeordnet
  - Benutzer können nur durch Prozesse mit dem System interagieren
- Alle zugreifbaren Informationen sind in Segmenten organisiert
  - Segment muss einem Adressraum zugeordnet sein, um zugreifbar zu sein
  - Prozess muss passende Zugriffsrechte auf Segment haben
- Segmenten sind Zugriffsrechte zugeordnet
  - Zugriffskontrolllisten enthalten autorisierte Benutzer und Zugriffsrechte

Eigenschaften des Adressraum

- Jeder Prozess komplett isolierbar

  $\longrightarrow$ Kein anderer Benutzerprozess hat Zugriff auf ein Segment

- Jeder Prozess kann kontrolliert Informationen teilen

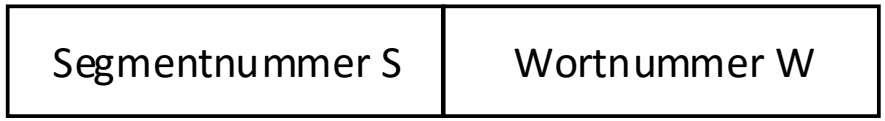
  $\longrightarrow$ Bestimmte Segmente können freigegeben werden

- Übergang zwischen Schutzdomänen transparent

  $\longrightarrow$ Schutz von Segmenten vor Missbrauch

Segmentierung

- Segmentadresse besteht aus zwei Teilen
  - Segmentnummer S $\longrightarrow$ bestimmt Segment
  - Wortnummer W $\longrightarrow$ bestimmt Wort im Segment



- Beschrieben durch Segmentdeskriptor
  - Basisadresse
  - Länge
  - Zugriffsrechte
- (Seitenverwaltung ignoriert)

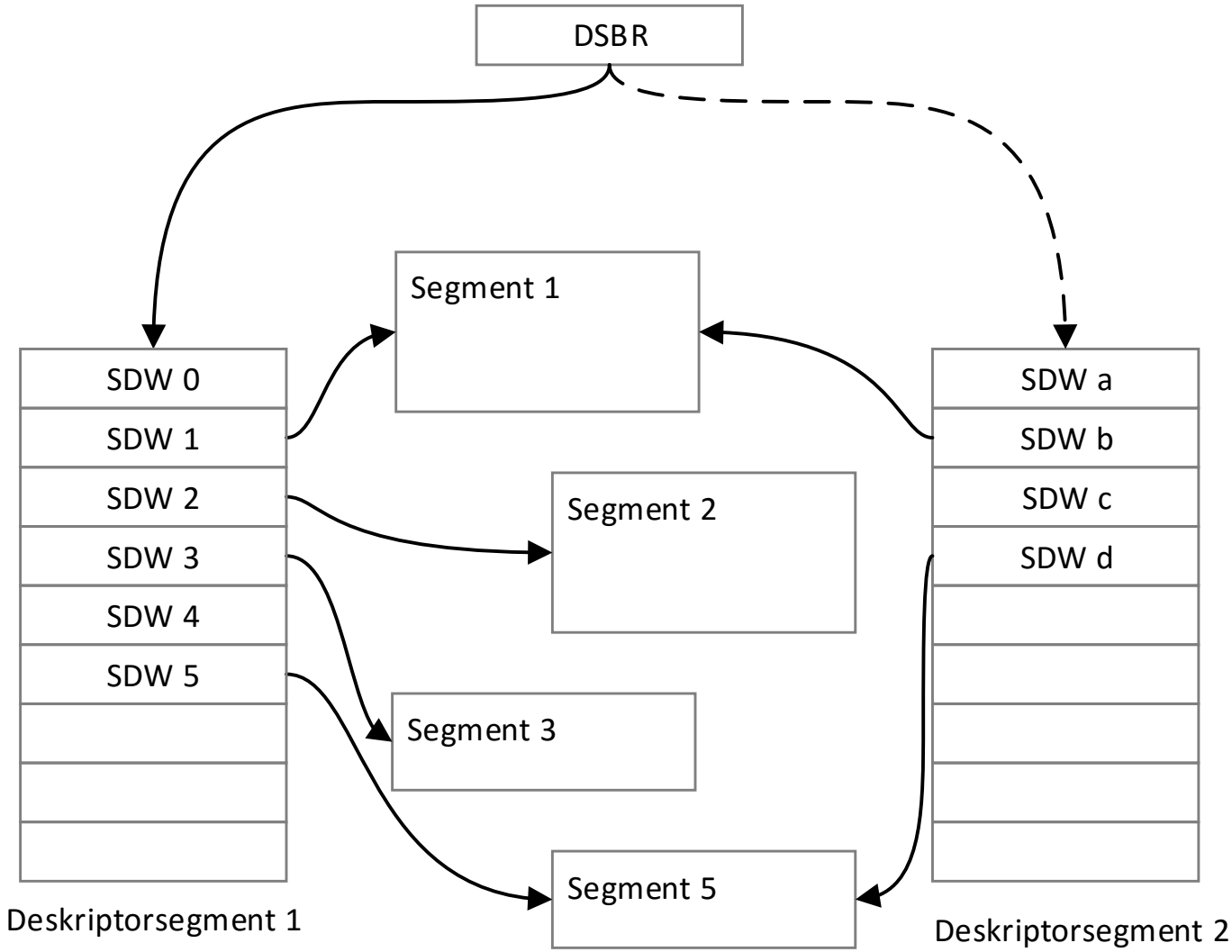
Segmentierung

- Pro Prozess ein Deskriptorsegment
  - enthält eingeblendete Segmente
  - besteht aus *Segment Descriptor Words* (SDWs)
  - Zugriffsrechte
    - Schreiben
    - Lesen
    - Ausführen
    - (. . .)
- *Descriptor Segment Base Register* (DSBR)[2]
  - Basisregister für Deskriptorsegment
  - Ändern des DSBR implementiert Kontextwechsel

[2] auch: *Descriptor Base Register* (DBR)

Segmentierung

Abbildung basierend auf [1]



Schutzdomänen

Abstraktes Schutzkonzept

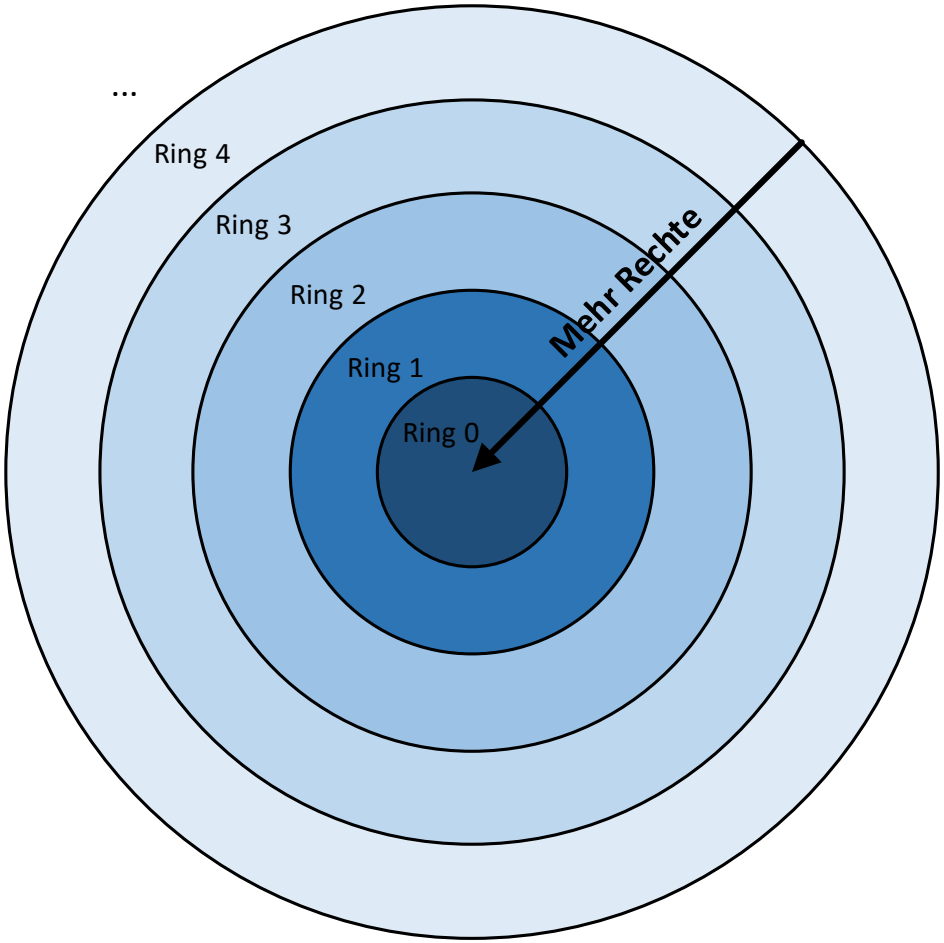
- Benötigte Rechte können sich im Laufe der Ausführung ändern
  - Zugriff auf bestimmte Daten
  - Supervisoraufruf
  - ...
- Zusammenfassen einer Menge von Rechte zu Schutzdomänen
- Theoretisch unbegrenzte Zahl von Schutzdomänen pro Prozess

Schutzringe

Implementierung in Multics

- Schutzringe sind Schutzdomänen mit Einschränkungen
- Begrenzte Zahl von Schutzringen pro Prozess
  - Jeder Schutzring hat Zahl zwischen 0 und $r-1$ (r Schutzringe)
  - Anfangs bei 64 und mehr
  - Später auf 8 festgelegt
- Schutzringe bilden Rechtehierarchie
  - Ring 0 hat die meisten, Ring $r-1$ die wenigsten Rechte
  - Menge aller Rechte von Ring m muss Teilmenge der Rechte aller Schutzringe n sein, mit $n < m$

Schutzringe



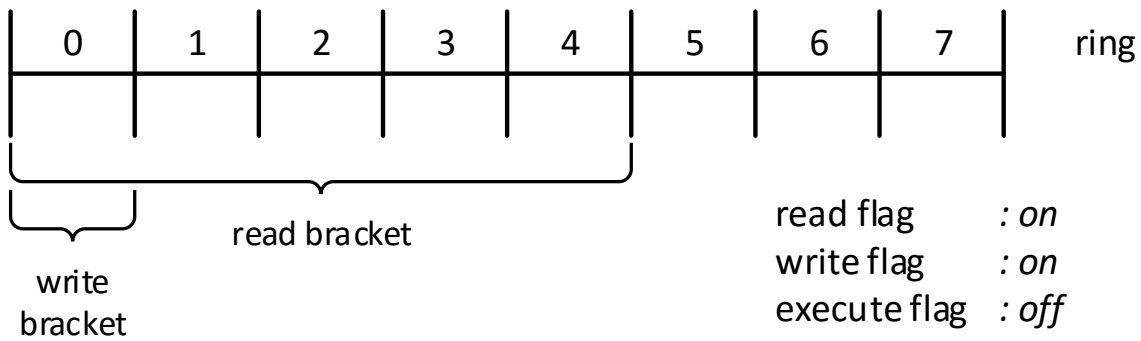
Schutzringe

- Jedem Recht muss zugewiesen werden, für welche Ringe es gilt
- Wegen Hierarchievoraussetzung nur obere Grenze x nötig
  - Recht geht von Ring 0 bis Ring x (Bracket)
- Pro Recht ein Flag, dass das Recht komplett an-/ausschalten kann

$\Longrightarrow$ Es entstehen Rechtebereiche

Schutzringe

- Beispiel Datensegment:
  - 8 Ringe
  - Schreiben: Ring 0, Flag an
  - Lesen: Ring 4, Flag an
  - Ausführen: Ring 0, Flag aus



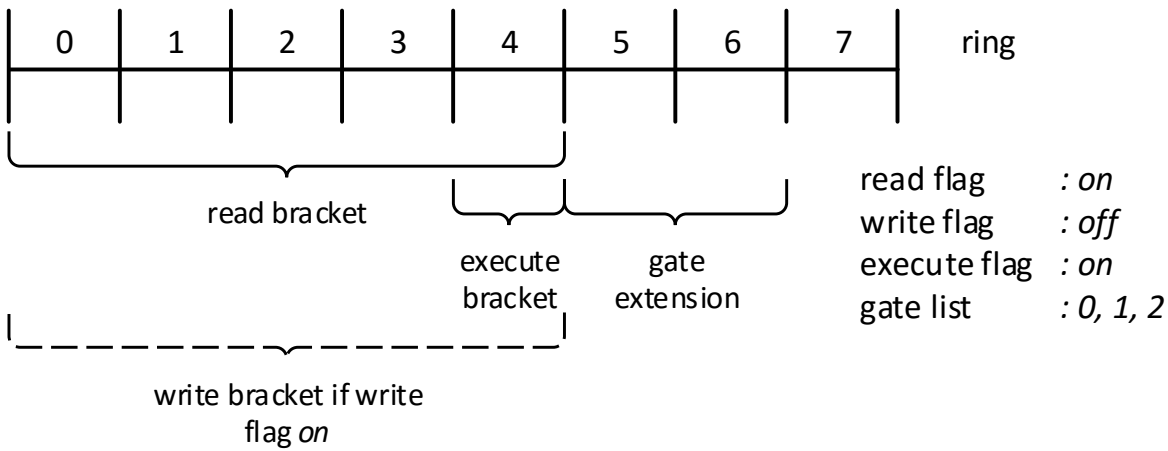
Erweiterte Schutzringe

- Recht für Ringwechsel (abwärts) nötig
  - Zum Beispiel von Ring 3 in Ring 0
  - Wechsel in niedrigeren Ring über Gate möglich
  - Gate ist eine bestimmte Stelle im Speicher (=Programm)
  - Liste aller möglichen Gates wird an Segment angehängt
- *Gate Extension* gibt an aus welchen Ringen Ringwechsel möglich ist
  - schließt sich an Ausführenrechtebereich an
- Programme sollen normalerweise in einem bestimmten Ring laufen
  - Zusätzliche untere Grenze für Ausführenbereich

Erweiterte Schutzringe

Abbildung basierend auf [3]

- Beispiel Codesegment:
  - 8 Ringe
  - Schreiben: Ring 4, Flag aus
  - Lesen: Ring 4, Flag an
  - Ausführen: Ring 4, Flag aus
  - Gate Extension: Ring 5-6
  - Gate Liste: 0, 1, 2



Schutzringaufteilung (1)

- Ring 0: Sicherheitskernel (*hard core*)
  - Ein-/Ausgabe
  - Prozessverwaltung
  - Speicherverwaltung
  - ...
- Ring 1: Kernelsubsysteme
  - Streamingsubsystem
  - Dateisystemoperationen
  - ...
- Ring 2-3: Benutzersubsysteme
  - Logging
  - Authentifizierung
  - geschützte Datenbanken
  - ...

Schutzringaufteilung (2)

- Ring 4: Benutzerprozesse
  - Benutzerapplikationen
- Ring 5: Benutzerprozesse (eingeschränkt)
  - debuggte Benutzerapplikationen
  - fremde Programme
  - ...
- Ring 6-7: Nicht vertrauenswürdige Prozesse
  - Kein Zugriff auf Ring 0/1
  - unbekannte Programme
  - ???

Agenda

Motivation

Adressraumschutz

Dateisystemschutz

Authentifizierung

B2 Zertifizierung

Fazit

Dateisystem

- Hierarchisches Dateisystem
- Datei besteht aus geordnete Menge von Elementen
- Dateien können nur durch das Dateisystem angelegt, verändert oder gelöscht werden
- Es bestehen dieselben Schutzbedürfnisse wie bei Segmenten
  - Isolation
  - Kontrolliertes Teilen

Dateisystem und Segmentierung

- Jede Datei bzw. Verzeichnis besitzt eine Zugriffskontrollliste
  - jeder Datei bzw. Verzeichnis können unabhängig voneinander Zugriffsrechte vergeben werden
- Ein Eintrag in der Zugriffskontrollliste enthält
  - Kennung zur Authentifizierung
  - Zugriffsrechte
  - Ringe in denen die Rechte gelten (Rechtebereiche)
- Verknüpfungen erben die Zugriffsrechte der referenzierten Datei
- Dateien werden als Segmente in den Adressraum eingebunden

$\Longrightarrow$ Adressraum- und Dateisystemschutz greifen nathlos ineinander

Zugriffsrechte

- Ausnahmeattribut
  - ist eine Ausnahmeliste mit Ausnahmeprozeduren zugeordnet
  - Ausnahmeprozeduren können Standardzugriffsattribute überschreiben
  - gedacht für benutzerspezifischen Authentifizierungsschemata
- Zugriffsattribute
  - Lesen
  - Schreiben
  - Ausführen
  - Anhängen

Zugriffsrechte

- Bedeutung der Attribute abhängig vom Typ (Datei, Verzeichnis)
- Datei: intuitiv (Lesen für Lesen, . . .)
  - Schreiben erlaubt kein Vergößern
  - Anhängen erlaubt kein Verändern vorhandener Daten
- Verzeichnis: nicht intuitiv
  - Lesen: Auflistung aller Einträge eines Verzeichnis
  - Ausführen: Suchen nach Eintrag
  - Schreiben: Verändern vorhandener Einträge
  - Anhängen: Erstellen neuer Einträge
- Zugriffsrechte gelten nur für Einträge des Verzeichnisses selbst
  - nicht für darunterliegende Einträge
- zwei initiale Zugriffskontrolllisten um Aufwand zu reduzieren

Agenda

Authentifizierung

- Alle Prozesse authentifiziert
  - interaktiv mit Passwort
  - aus bereits authentifizierten Prozess entstanden
- Benutzer müssen sich authentifizieren
  - „Eindeutige“ Benutzerkennung (Nachname + 2 Initialen)
  - Benutzerkennungen ewig gültig
  - achtstelliges Passwort (ASCII)
- Passwörter sollten niemals Dritten zugänglich werden
  - verschlüsselt mit „Hash“-Verfahren
  - Drucker druckt keine Passwörter
  - Proxy Logins

Prozesskennungen

- Jedem Prozess wird eine Kennung zugeordnet:

  Benutzerkennung : **Gruppe** : **Abteil**[3]

- Benutzerkennung
  - Kennung des Benutzers
- Gruppe
  - enstpricht in etwa Unix Gruppen
  - Benutzer können in beliebig vielen Gruppen sein
  - Gruppe für Prozess wird bei Prozesserzeugung festgelegt
- Abteil
  - Heute keine Entsprechung mehr
  - Selbsteinschränkung
  - Abteil wird bei Prozesserzeugung festgelegt

[3] engl.: Compartment

Zugriffskontrolllisteneinträge

- Selber Aufbau wie Prozesskennungen
- Jeder Teil kann auch ein Asterisk (*) enthalten
  - entspricht: Keine Einschränkung

Benutzerkennung : **Gruppe** : **Abteil**

Beispiele

Prozess	**HerzogBe** :	**Student** :	**Normal**	
Einträge	**HerzogBe** :	* :	*	✔
	HerzogBe :	* :	**StrengGeheim**	✖
	* :	**Student** :	*	✔

Agenda

1970er

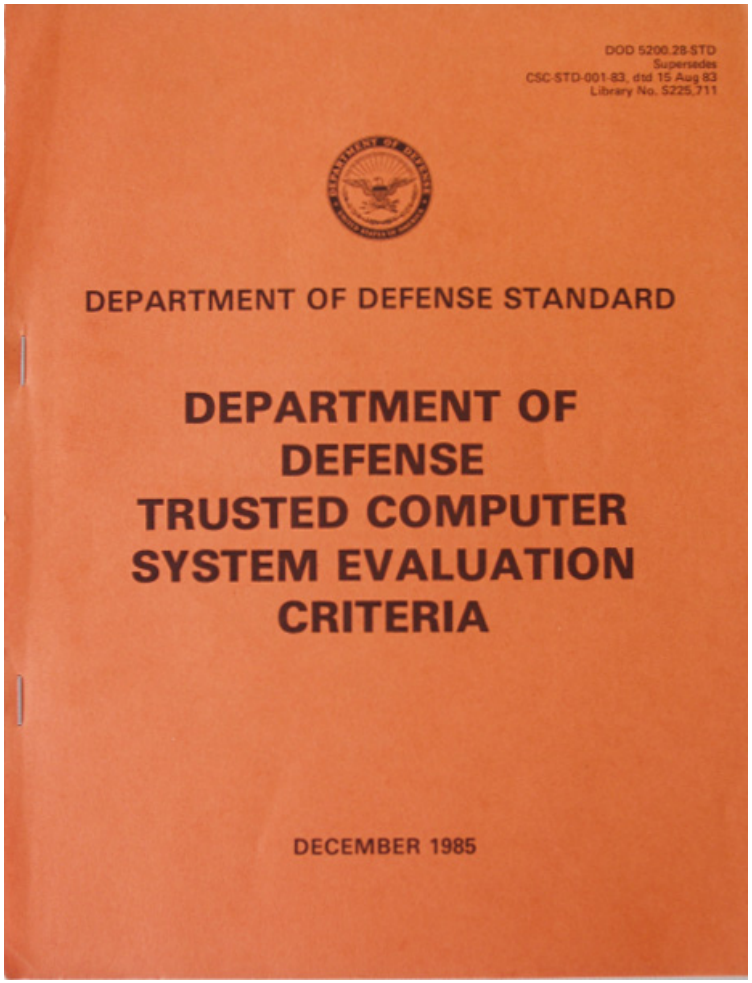
- Militär unzufrieden mit vorhandenen Computersystemen
  - Kein sicherer Mehrbenutzerbetrieb
  - Ein System pro Anwendung
  - Komplettes Löschen des Systems vor Anwendungswechsel
- Multics hatte schon Sicherheitsfeatures implementiert
- Startete mehrere Projekte über Sicherheit in Computersystemen
- Auch im Multics Kontext Sicherheitsprojekte
  - Project Guardian, ...

Sicherheitsprobleme in Multics

- Meinung Anfang der 1970er: Multics kann sicher betrieben werden
- Aber: Große Codebasis in Ring 0
  - über 55.000 Zeilen
- *Project Guardian* versuchte 1975 Codebasis zu verringen
  - führte zu großen Leistungsverlusten
  - wurde 1976 wieder eingestellt, ohne das Ergebnisse in Multics flossen
- 1977 schaffte es die US-Luftwaffe in die Entwicklungsinstallation einzubrechen
  - Fehler in einem nicht verwendeten Ring 0 Programm
  - Konnten alle Speicherstellen/Dateien lesen und patchen
  - Schafften es die „Hash“-funktion der Passwörter zu entschlüsseln

1980er

- Ergebnisse münden 1983 in *Department of Defense Trusted Computer System Evaluation Criteria* (TCSEC)
- "Orange Book"
- Einheitliche und systematische Bewertung von Sicherheit in Computersystemen



Orange Book

- definiert Sicherheitslevel
  - D, C1, . . . , A1
  - D $\longrightarrow$ minimale Sicherheit
  - C $\longrightarrow$ teilweise Sicherheit (Ermessensentscheidung)
  - B $\longrightarrow$ verbindliche Sicherheit
  - A $\longrightarrow$ verifizierte Sicherheit
- Multics erreichte 1985 die B2 Zertifizierung nach zweijähriger Evaluation
  - definiertes und dokumentiertes Sicherheitsmodell
  - verbindliche Zugriffskontrolle auf alle Objekte
  - . . .
- U.a. zu große Codebasis in Ring 0 verhindert B3 Zertifizierung

Common Criteria (1)

- Nachfolgedokument des Orange Book 1996: *Common Criteria for Information Technology Security Evaluation* (CC)
- Harmonisierung der us-amerikanischen, kanadischen und europäischen Kriterien
- Vorgehen zur Evaluation von IT-Produkten
  - Anwendungen, Betriebssysteme, Hardware, ...
- Besteht grob aus zwei Teile
  - produktunabhängiges Sicherheitsprofil basierend auf Funktionalität
  - Ableitung und Evaluation von Sicherheitsanforderungen aus dem Sicherheitsprofil

Common Criteria (2)

- *Evaluation Assurance Level* (EAL) gibt Prüfungstiefe an
  - entspricht in etwa den Sicherheitsleveln des Orange Book
  - gehen von EAL1 (etwa D bis C1) bis EAL7 (etwa A1)
- Kombination aus Sicherheitsprofil und EAL ergibt Vertrauenswürdigkeit
- Kritik wegen formaler Vorgehensweise
- Heute akzeptierter Standard

Agenda

Und heute?

Technische Details haben sich geändert

- Kaum Segmentierung
- Bessere Kryptographie
- Schwaches Schutzringkonzept

(Viele) Konzepte haben überlebt

- Mehrbenutzerbetrieb (Clouds, . . .)
- Zugriffskontrolllisten
- Trennung Schutzmechanismen und -merkmal
- Kein *Security through Obscurity*
- Prinzip des geringsten Rechts
- . . .

Danke für die Aufmerksamkeit

Fragen?

Common Criteria (3)

Anzahl der Zertifizierungen pro Jahr (1999-2015)

1999	2000	2001	2002	2003	2004	2005	2006	2007
1	2	1	4	3	6	11	8	334

2008	2009	2010	2011	2012	2013	2014	2015	Total
124	153	161	207	255	232	256	233	1991

Anzahl der Zertifizierungen pro Land (1999-2015)

Australien	Deutschland	Frankreich	Japan	USA
66	591	518	173	121

Großbritannien	Indien	Malaysia	Andere	Total
36	3	26	457	1991

Common Criteria (4)

- Nur ein Produkt hat die höchste Zertifizierung EAL7+
  - Ein-Wege Kommunikationssystem von Fox IT: DataDiode
- Im Bereich Betriebssysteme haben einige Produkte EAL5+
  - entspricht etwa B2-B3 des Orange Book
  - aber nur Spezialsysteme, keine Universalbetriebssysteme
  - v.a. PR/SM Hypervisor für IBM Z Systeme
- Red Hat und SUSE Enterprise Linux Versionen mit EAL4+
  - etwa B1 des Orange Book

Quellen I

Multicians.
Execution environment.
http://www.multicians.org/exec-env.html, 2015.
[Online; abgerufen am 01. Dezember 2015].

J. H. Saltzer.
Protection and the control of information sharing in Multics.
Communications of the ACM, 17(7):388–402, 1974.

M. D. Schroeder and J. H. Saltzer.
A hardware architecture for implementing protection rings.
Communications of the ACM, 15(3):157–170, 1972.