Fragenausarbeitung: Betriebssysteme (VO Puschner)

Vorwort:

Hi, diese Fragenausarbeitung umfasst die Theoriefragen von vier Prüfungen aus der VO Betriebssysteme. Hab diese Ausarbeitung 2 Tage, bevor ich zur Prüfung angetreten bin, verfasst. Das soll heißen, ich hatte es ziemlich eilig und gib daher keine Garantie auf Vollständigkeit und absolute Korrektheit der Antworten. Ich stelle das ganze auch im .odt-Format online, damit Interessierte Fehler ausbessern oder die Fragenausarbeitung erweitern können.

Prüfung vom 17. November 2006:

Nennen und erklären Sie die Threats of Security. Geben Sie jeweils ein konkretes Beispiel an!

Interruption(Denial of Service): bedroht Availability, vorübergehende oder permanente Unterbrechung eines Services. Bsp.: Durchtrennen einer Netzwerkleitung, physikalische Zerstörung einer Festplatte, Überlastung eines Servers durch permanentes senden von Anfragen (DDoS-Attacke).

Interception: bedroht Confidentiality(Secrecy), unauthorisiertes Lesen von Daten. Bsp.: Abhören einer Leitung, unauthorisiertes kopieren von Dateien

Modification: bedroht Integrity, Abfangen von Daten und Weiterleiten von veränderten Daten. Bsp.: Abfangen von Netzwerknachrichten und weiterleiten von geänderten Daten. Ändern des Inhaltes einer Datei.

Fabrication: bedroht Authenticity, Einbringen von gefälschten Daten in den Informationsfluss. Bsp.: Hinzufügen von Dateieinträgen

Erklären Sie die Funktionsweise von Kerberos!

Kerberos ist ein verteilter Authentifizierungsdienst für offene und unsichere Netze (z.B.: Inernet).
Bei Kerberos sind drei Parteien beteiligt: Server, Client und Kerberos-Server
Der Kerberos-Dienst authentisiert sowohl den Server gegenüber dem Client, als auch den Client gegenüber dem Server. Auch der Kerberos-Server selbst authentisiert sich gegenüber dem Client und Server und verifiziert selbst deren Identität.
Kerberos verwendet sog. *Tickets* zur Authentifizierung. Um den Kerberos-Dienst nutzen zu können, muss sich ein Client zuerst beim Kerberos-Server anmelden. Er fordert vom Kerberos-Server ein sog. Ticket Granting Ticket (TGT) an. Hierzu muss der Nutzer des Clients entweder ein Passwort eingeben oder das TGT wird direkt bei der Benutzeranmeldung angefordert. Mit dem TGT ist der Client in der Lage, weitere Tickets für Dienste anzufordern, ohne nochmal ein Passwort eingeben zu müssen. Es wird auch ein Session Key für die Kommunikation zwischen Client und Kerberos-Server ausgehandelt. Er kann benutzt werden, um den Datenverkehr zu verschlüsseln. Um einen Dienst, der Kerberos unterstützt, benutzen zu können, fordert der Client ein weiteres Ticket an. Dieses Ticket sendet der Client dann an den Dienst, der überprüft, ob er dem Client den Zugriff gestatten soll. Auch hierbei wird ein Sitzungsschlüssel vereinbart und die Identität von Client, Server und Kerberos-Server überprüft.

Wie funktioniert Public Key Encription?

Public Key Encription ist eine asymmetrische Verschlüsselungsmethode. Es gibt zwei Schlüssel, einen Public Key und einen Private Key. Wird eine Nachricht mit einem der beiden Schlüssel

verschlüsselt, so kann sie jeweils nur mit dem anderen Schlüssel wieder entschlüsselt werden. Jeder Teilnehmer dieser Verschlüsselungsmethode generiert die zwei Schlüssel. Einen Schlüssel behält er für sich, der Private Key, den Anderen stellt er allen anderen Teilnehmern zur Verfügung – Public Key. Will ihm jemand eine Nachricht senden, so muss sie mit dem Public Key verschlüsselt sein. Damit ist sichergestellt, dass diese Nachricht nur der Besitzer des Private Key entschlüsseln kann. Andererseits, wenn ein Teilnehmer eine Nachricht mit seinem Private Key verschlüsselt und sie einem anderen Teilnehmer sendet, so kann dieser die Nachricht nur mit dem Public Key des Senders entschlüsseln und kann somit eindeutig feststellen, von wem die Nachricht stammt.

Beschreiben Sie das UNIX Passwort Scheme!

Hierbei wird in einer Liste zu jeder User-ID das entsprechende Passwort verschlüsselt gespeichert. Das maximal 8 Zeichen lange Passwort wird in 56 Bit ASCII Zeichen codiert. Diesen 56 Bit wird ein 12 Bit salt-Wert hinzugefügt, welcher beispielsweise von der Zeit der Passworterstellung abhängt. Diese Resultierende Bitfolge wird anschließend 25 mal durch einen Verschlüsselungsalgorithmus geschickt. Der 12-Bit salt-Wert dient dazu, die für das Knacken des Passwortes durch brude-force benötigte Zeit erheblich zu verlängern. In der Liste mit den User-IDs und den dazugehörigen verschlüsselten Passwörtern wird nun auch der salt-Wert plain gespeichert. Will sich ein Benutzer mit einer bestimmten ID authentifizieren, so wird das von ihm eingegebenen Passwort mit dem zur Benutzer-ID gespeicherten salt-Wert durch den Verschlüsselungsalgorithmus geschickt und das Ergebnis mit dem gespeicherten Passwort verglichen.

Erläutern Sie zwei Ansätze zum Angriff auf ein konventionelles Encryption Scheme!

1) Am lokalen Computer auf einem Gast-Account oder mit anderen Mitteln einloggen und versuchen Passwörter anderer User mittels einem sog. Passwort-cracker-Programm zu erraten (brude-force).

2) Gelangt man irgendwie an das Passwortfile, so kann man auf einem anderen Rechner mittels cracker-Programm versuchen an Passwörter zu kommen.

Verständnisfragen:

Welche Aussagen treffen beim Fixed Partitioning zu?

- ✔ Die Partitionen sind immer von gleicher Größe
- ✔ Der vorhandene Speicher wird uneffizient ausgenutzt
- ✔ Das Problem der Internal Fragmentation tritt auf

Zu welchen Effekten kann es beim Paging kommen?

- ✗ Unterschiedliche Längen der Pages
  - ✔ Internal Fragmentation
- ✗ External Fragmentation

Bei Speicherverwaltung mittels Segmentierung berechnet sich die physikalische Adresse aus der virtuellen Adresse durch:

- ✗ Addition der physikalischen Segmentstartadresse mit der virtuellen Adresse.
- ✗ Multiplikation der physikalischen Segmentstartadresse mit dem Offset-Teil der virtuellen Adresse
  - ✔ Addition der physikalischen Segementstartadresse mit dem Offset-Teil der virtuellen Adresse
- ✗ Ersetzen der Seitennummer in der virtuellen Adresse durch die physikalische

Segementstartadresse

Prüfung vom 24. Oktober 2007:

Verständnisfragen:

Bei folgender Speicherverwaltungstechnik kann das Problem der internen Fragmentierung auftreten:

- ✔ fixed partitioning
- ✗ simple segmentation
- ✔ virtual-memory paging
- ✗ dynamic patitioning
- ✔ simple paging
- ✔ virtual memory with combination of paging and segmentation

Beim Fixed Partioning sind die Partitionen immer von gleicher Größe.

- ✔ richtig
- ✗ falsch

Bei folgender Speicherverwaltungstechnik tritt sowohl interne als auch externe Fragmentierung auf.

- ✗ fixed partitioning
- ✗ simple segmentation
- ✗ virtual-memory paging
- ✗ dynamic partitioning
- ✗ simple paging
- ✗ virtual memory with combination of paging and segmentation

In einem fehlerfreien System können zwei oder mehr virtuelle Adressen auf eine physikalische Adresse abgebildet sein.

- ✗ richtig
- ✔ falsch (nicht zeitgleich)

Prüfung vom 28. Jänner 2008:

Welche Arten der Allokation kennen Sie zur Implementierung einer Datei im sekundären Speicher? Erklären Sie diese und erläutern Sie Vor- und Nachteile!

-*Contiguous Allocation:*

Eine Datei belegt eine einzige, aneinander grenzende Menge von Blöcken.

VT: gute Performance beim Lesen

NT: Platzprobleme beim Vergrößern einer Datei, externe Fragmentierung

Verwendung: CD-ROM, DVD-ROM
-*Chained Allocation:*
Belegung einzelner Blöcke, welche über Zeiger mit Folgeblock verkettet sind.
VT: keine externe Fragmentierung
NT: keine Lokalität der Blöcke => langsamer Zugriff bei Random Access: Man muss beim Suchen immer beim ersten Block beginnen und Link für Link folgen, bis man beim gesuchten Block ist
Nutzdaten pro Block < 2^n weil Zeiger auch Platz benötigt.
-*Indexed Allocation:*
Wie Chained Allocation aber Zeiger auf Folgeblock werden in einer Tabelle im Speicher (FAT File Allocation Table) gehalten.
VT: Sowohl direkter als auch sequentieller Zugriff gut unterstützt, Blöcke ganz für Nutzdaten verfügbar.
NT: großer Platzbedarf für FAT im Arbeitsspeicher
-*I-Nodes(Indexed Node):*
Datenstruktur für jedes File, enthält Fileattribute und Referenzen auf die Blöcke des Files.
VT: I-node wird nur im Memory gebraucht, wenn ein File verwendet wird.
NT: Anzahl der Blockreferenzen pro i-node ist begrenzt => bis zu dreifach indirekte Blöcke, d.h.: Eintrag in Block ist keine Referenz auf einen Teil des Files sondern auf einen Block, der wiederum eine Referenzen auf File-Blöcke enthält.

Nennen und erklären Sie vier typische Operationen beim File Management!
-*Create, Delete:* Anlegen eines neuen Files, Löschen eines bestehenden Files.
-*Open, Close:* Öffnen einer Datei zum Lesen oder Schreiben, Schließen einer geöffneten Datei.
-*Read, Write, Append:* Lesen einer Datei, Schreiben in eine Datei, Anfügen von Daten an eine bereits bestehende Datei.
-*Seek:* Springen an eine bestimmte Stelle in einer Datei.
-*Get* Attributes, Set Attributes: Datei-Attribute anfordern/neu setzen.
-*Rename:* Zuweisen eines neuen Namens einer Datei.
-*Lock:* Datei für Zugriffe sperren (z.B.: es darf nur ein Benutzer/Prozess gleichzeitig auf eine Datei zugreifen).

Nennen und erklären Sie fünf fundamentale Arten der Datei-Organisation!
-*unstructured sequence of bytes:*
-*pile:*
Records variabler Länge werden in der Reihenfolge in der sie ankommen aneinandergereiht.
Suche einzelner Records sehr aufwändig, speichereffizient und leicht erweiterbar.
-*sequential file:*
Jeder Record hat eine fest definierte Länge und besteht aus fest definierten Feldern die in einer definierten Reihenfolge auftreten. Am Anfang jedes Records gibt es ein Key-Feld. Die Reihenfolge der Dateien wird durch ordnen nach dem Wert im Key-Feld festgelegt.
Schlechte Performance bei Zugriff auf/Update von einzelnen Records. Einfügen eines neuen Eintrages, dessen Key-Wert nicht ganz am Schluss des Files angesiedelt ist, kann aufwändig sein.
-*indexed-sequential file:*
Sequenital file erweitert um einen Index und ein overflow-file. Ein Eintrag im Index besteht aus zwei Feldern: ein Key-Feld und ein Feld, dass einen Pointer auf einen entsprechenden Eintrag in der Datendatei beherbergt. Dieser Eintrag muss nicht der gesuchte Eintrag sein, aber ein Eintrag in der Nähe des gesuchten Eintrages => Suchzeit eines bestimmten Records wird verkürzt.
Diese Methode kann auch rekursiv in mehreren Schritten realisiert werden, d.h.:

Der Pointer eines Eintrags im Indexfile zeigt auf ein weiteres Indexfile in welchem sich alle Pointer auf Einträge in der Nähe des gesuchten Eintrages befinden.
NT: Es kann nur nach Einträgen mittels deren Key-Wert gesucht werden, nicht mittels Werten anderer Datenfelder der Einträge. => indexed file
-indexed file:
Es gibt index-files für alle in den Records, welche nun variabler länger sein dürfen, vorkommenden Felder. Die Records besitzen nun auch kein Key-Feld mehr, nach dem sie geordnet werden. Die Reihenfolge der Records ist beliebig, da nur mehr mittels der index-files auf sie zugegriffen wird.
NT: up-to-date halten der Index-files ist sehr aufwendig.
-direct (hashed) file:
Eine Hashfunktion wird auf ein Key-Feld angewendet und so ein direkter Zugriff auf einen gesuchten Eintrag ermöglicht. Keine sequentielle Reihenfolge der Einträge.

Welche Arten der Block-Verwaltung kennen Sie? Erklären Sie ein Verfahren der Block-Verwaltung anhand eines Beispiels!
fixed blocking:
Records fixer Länge, Records werden nicht geteilt => block-interne Fragmentierung
variable-lenght spanned blocking:
Records variabler Länge, Records können geteilt werden => keine block-interne Fragmentierung
variable-length unspanned blocking:
Records variabler Länge, Records werden nicht geteilt => block-interne Fragmentierung

Erklären Sie die Begriffe: Confidentiality, Integrity und Availability.
Confidentiality(Secrecy):
Geheimhaltung von Information: Nur authorisierte Personen dürfen Daten einsehen (lesender Zugriff), oder überhaupt von deren Existenz wissen.
Integrity:
Integrität der Daten: Nur authorisierte Personen dürfen Daten verändern (schreibender Zugriff), löschen, erzeugen.
Availability:
Verfügbarkeit der Daten: Für authorisierte Personen müssen die Daten verfügbar sein, wenn sie benötigt werden.

Ordnen Sie die vier Bedrohungsarten Interruption, Interception, Modification und Fabrication jeweils einem der Begriffe Confidentiality, Integrity und Availability zu.
Interruption ↔ Availability
Interception ↔ Confidentiality
Modification ↔ Integrity
Fabrication ↔ Integrity

Nennen Sie mindestens vier Design Prinzipien für Security und beschreiben Sie diese kurz.
Open Design: nicht „Security by Obscurity“
Default Einstellungen: keine Berechtigung
Least Privilege: Benutzer hat nur die Rechte, die er unbedingt benötigt.
Economy of Mechanisms: Fehlervermeidung durch Einfachheit der Sicherheitsmechanismen, Implementierung auf möglichst niedriger Ebene.
Acceptibility: akzeptierbare Mechanismen, Schränken die Mechanismen den Benutzer in seiner Standard-Tätigkeit zu sehr ein, so werden auf lange Sicht diese Mechanismen

höchstwahrscheinlich ausgestellt. z.B.: ständige Eingabe des Administrator Passwortes notwendig => Benutzer arbeitet defaultmäßig als Administrator
Überprüfung der gegenwärtigen Berechtigung
Complete Mediation: Kontrolle aller Zugriffe auf Resourcen (auch Ausnahmesituationen)

Nenne Sie mindestens 4 Maßnahmen gegen Passwortattacken.
-*One-time Password* – unpraktisch
-*Zeitablauf von Passwörtern* – unpraktisch
-*Challenge Response Protokolle:*
Herausforderung-Antwort: Bsp.: Client möchte sich bei Server authentifiziere: Server schickt Client Zahl, Client verschlüsselt Zahl mit seinem Passwort und schickt Ergebnis an Server, Server verschlüsselt Zahl ebenfalls mit dem Passwort des Clients und vergleicht die beiden Ergebnisse.
VT: Passwort selbst muss nicht übertragen werden
-*Call Back bei Login* über Wählleitung – Wählleitungen im Aussterben
-*Zeitverzögerung* nach gewisser Anzahl an fehlerhaften Loginversuchen.
-*Logging:* Anzeige der Daten des letzten Login → Benutzer kann selbst erkennen, ob jemand andere eingeloggt war.

Erklären Sie die für das Auftreten eines Deadlock notwendigen und hinreichenden Bedingungen.
-*Mutual Exclusion:* Resourcen können nur von einem Prozess gleichzeitig verwendet werden.
-*Hold and Wait:* Prozesse können Resourcen halten, während sie auf weitere Resourcen warten.
-*No Preemption:* Prozessen können einmal zugewiesene Resourcen nicht wieder entzogen werden.
-*Circular Wait:* Es kann bei einer Gruppe von Prozessen ein Szenario auftreten, bei dem jeder Prozess bereits Resourcen hält und auf das Freiwerden von Resourcen wartet, die bereits einem anderen Prozess zugeteilt sind.

Erklären Sie den Begriff der Deadlock Prevention. Führen Sie zu jedem der beiden Arten von Deadlock Prevention ein Beispiel für eine Strategie zur Behandlung von Deadlocks an.
Deadlock Prevention bedeutet, dass eine der 4 Deadlockbedingungen im Betriebssystemdesign nicht erfüllt wird. Dadurch können Deadlocks nie auftreten. 2 Arten der Deadlock Prevention:
-*Indirect Deadlock Prevention:*
Es wird eine der ersten drei Bedingungen (Mutual Exclusion, Hold and Wait, No Preemption) nicht erfüllt.
Mutual Exclsion: kann nicht verhindert werden
Hold and Wait: Prozesse fordern alle Resourcen auf einmal an oder blockieren bis genügend Resourcen frei sind. → Prozess muss wissen welche Resourcen er brauchen wird, Lange Verzögerung
No Preemption: Prozess gibt alle bereits allokierten Resourcen wieder frei, wenn er eine Resource nicht bekommt und versucht es später nochmal.
Oder Anforderung eines Prozesses führt dazu, dass ein anderer Prozess Resourcen freigeben muss.
-*Direct Deadlock Prevention:*
Bedingung 4 (Circular Wait) wird nicht erfüllt → Umsetzung mittels Einführung einer linearen Ordnung unter den Resourcen. Resourcen werden immer in der Reihenfolge angefordert, dass Resourcen höherer Ordnung nur nach der Anforderung aller Resourcen niedrigerer Ordnung angefordert werden können. Resourcen der gleichen Ordnung müssen alle gleichzeitig angefordert werden, d.h. ein Prozess erhält entweder alle angeforderten Resourcen dieser Ordnung oder gar

keine.

Worin liegt der Unterschied zwischen Deadlock Prevention und Deadlock Avoidance?
Bei der Deadlock Prevention wird ein Deadlock dadurch verhindert, dass eine der 4 Deadlockbedingungen durch entsprechendes Betriebssystemdesign nicht erfüllt wird.
Bei der Deadlock Avoidance können die 4 Deadlockbedingungen prinzipiell erfüllt sein.
Man unterscheidet zwei Arten der Deadlock Avoidance:
-Process Initiation Denial:
Ein Prozess wird nicht gestartet, falls seine Anforderungen zu einem Deadlock führen könnten
-Resource Allocation Denial:
Eine Resourcenanforderung eines Prozesses wird nicht gewährt, falls dadurch ein Deadlock entstehen könnte. (siehe Banker's Algorithm)
→ Organisatorischer Overhead, Wissen über Resourcenbedarf der Prozesse notwendig,

Prüfung vom 22. Oktober 2008:

Beschreiben Sie den Unterschied zwischen Security by Obscurity und Open Design. Geben Sie die Vor- und Nachteile an.
Security by Obscurity:
Der Geheimhaltungsmechanismus (z.B.: Verschlüsselung) funktioniert nur, solange niemand weiß wie dieser funktioniert. Bsp.: Haustürschlüssel unter der Fußmatte.
Open Design:
Der Geheimhaltungsmechanismus funktioniert auch wenn man dessen Funktionsweise preisgibt.
Bsp.: Public Key Encription, RSA-Verschlüsselung

Nennen Sie drei Abstraktionen die ein Betriebssystem zur Verfügung stellt. Geben Sie für jede dieser Abstraktionen einerseits die Benutzerwahrnehmung andererseits die zur Realisierung der Abstraktion notwendigen Betriebssystemmechanismen an.
-*Prozesse:*
Benutzerwahrnehmung: Prozesse werden parallel ausgeführt → mehrere Prozessoren
BS-Mechanismen: Scheduling, Signalisierung von Ereignissen, Mutual Exclusion, Synchronisation, Deadlockvermeidung/behandlung, IPC
-*Virtual Memory:*
Benutzerwahrnehmung: quasi unendlich großer Speicher, jeder Prozess hat eigenen Speicher (Adressbereich)
BS-Mechanismen: Speicherverwaltung/zuteilung an Prozesse, Aus/Einlagern von Speicher aus/in den Sekundärspeicher, Zugriffsschutz, Shared Memory, Paging/Segmentierung, Adressumsetzung(virtuelle → physikalische Adresse)
-*Virtual File System:*
Benutzerwahrnehmung: strukturierte zusammenhängende Dateien, Ordnerstrukture
BS-Mechanismen: File I/O Management, Datenorganisation am Sekundärspeicher, Directories

Welche Schritte muss ein Betriebssystem bei einem Process Switch (Context Switch) durchführen?
Sichern des Kontextes (Programm-Counter, Prozessor-Register, PSW) des alten Prozesses, ändern seines Zustandes je nach Ursache der Process Switches (in Ready, Blocked, Blocked-Suspend), Einfügen in die jeweilige Queue (Blocked-Queue, Ready-Queue,...).
Finden einer Scheduling Entscheidung, d.h. Auswählen eines neuen Prozesses aus der Ready-Queue

und seinen Zustand auf Running setzen.
Laden des Kontextes des neuen Prozesses in den Prozessor.

Nennen Sie die prozessspezifischen Informationen, die ein Betriebssystem im Process Control Block verwaltet.
Process Identification: Process-ID, User-ID, Parent-Process-ID,
Prozess State Information: Kontext(Registerinhalt), Programcounter, Program Status Word(PSW), Stack Pointer
Process Control Information: Prozesszustand, ev. Ereignis auf das gewartet wird,Scheduling-Info(Priorität), Verweise auf andere Prozesse (Realisierung von Prozessqueue mittels Linked List), IPC-Info, Info über verwendete Resourcen (Files, I/O Devices), Accounting-Info, Privileges, MM: Verweise auf Segment/Seiten-Tabellen

Erklären Sie den Unterschied zwischen den Begriffen Prozess und Thread. In welcher Beziehung stehen diese beiden Konzepte?
Ein Prozess bildet eine Dispatching(=kurzfristiges Scheduling) Einheit und hat einen virtuellen Adressraum und allokierte Resourcen, auf die kein anderer Prozess zugreifen kann.
Ein Prozess kann nun aus einem oder mehreren Threads bestehen. Ein Thread kann, je nach Implementierungsart, einen eigenen Ausführungszustand(Running, Ready,...)haben und hat einen eigenen Kontext und Stack.
Normalerweise ist es notwendig, damit zwei Prozesse kommunizieren können, dass sie IPC-Mechanismen des Betriebssystems verwenden.
Werden Threads verwendet, so können diese über den gemeinsamen Speicher(z.B. globale Variable) und die gemeinsamen Resourcen des übergeordneten Prozesses Daten austauschen.

In der Vorlesung wurden zwei Möglichkeiten der Implementierung von Threads genannt. Um welche Implementierungen handelt es sich dabei? Wie machen sie die Unterschiede der Implementierung für Applikationen bemerkbar?
User-Level Threads:
BS sieht keine Threads, Threads werden mittels eigener Library implementiert, Thread-Switch daher im User Mode, BS scheduled Prozesse → Prozess scheduled Threads,
NT: Blockieren einzelner Threads nicht möglich, nur ganzer Prozesse,
VT: Kein Mode-Switch notwendig für Thread-Switch innerhalb eines Prozesses → schneller

Kernel-Level Threads:
Thread-Management durch Kernel, Kernel stellt Thread-API zur Verfügung, Scheduling auf Thread-Basis,
VT: Threadweises blockieren möglich, gleichzeitiges scheduling mehrerer Threads eines Prozesses bei Mehrprozessorsystemen möglich,
NT: Mode-Switch notwendig für Thread-Switch auch innerhalb eines Prozesses

Erklären Sie, was man unter (a) einem monolithischen Kernel und (b) einem Microkernel versteht.
Monolithischer Kernel: alle Betriebssystemfunktionen(scheduling, file-system, networking, device-driver, mm, ...) im Kernel implementiert. D.h. als ein riesiger Prozess der im Kernel-Mode (also mit allen Rechten) läuft.
Microkernel: Implementiert nur Basisservices (Process Switching, Basic MM, Interrupts, Basic I/O,

IPC). Rest wird durch Server-Prozesse realisiert, die im User-Mode laufen.
VT: sehr klein, flexibel und leicht erweiterbar

Verständnisfragen:

Beurteilen Sie die folgenden Aussagen auf ihre Richtigkeit:

- ✔ Threshold detection ist ein statistisches Verfahren zur Intrusion Detection.
- ✗ Das Least Privilege Prinzip besagt, dass jeder Benutzer alle Rechter per default hat.
- ✔ Bei Masquerading wird der Inhalt einer Message verändert.
- ✗ Eine denial-of-service Attacke ist eine passive Attacke.