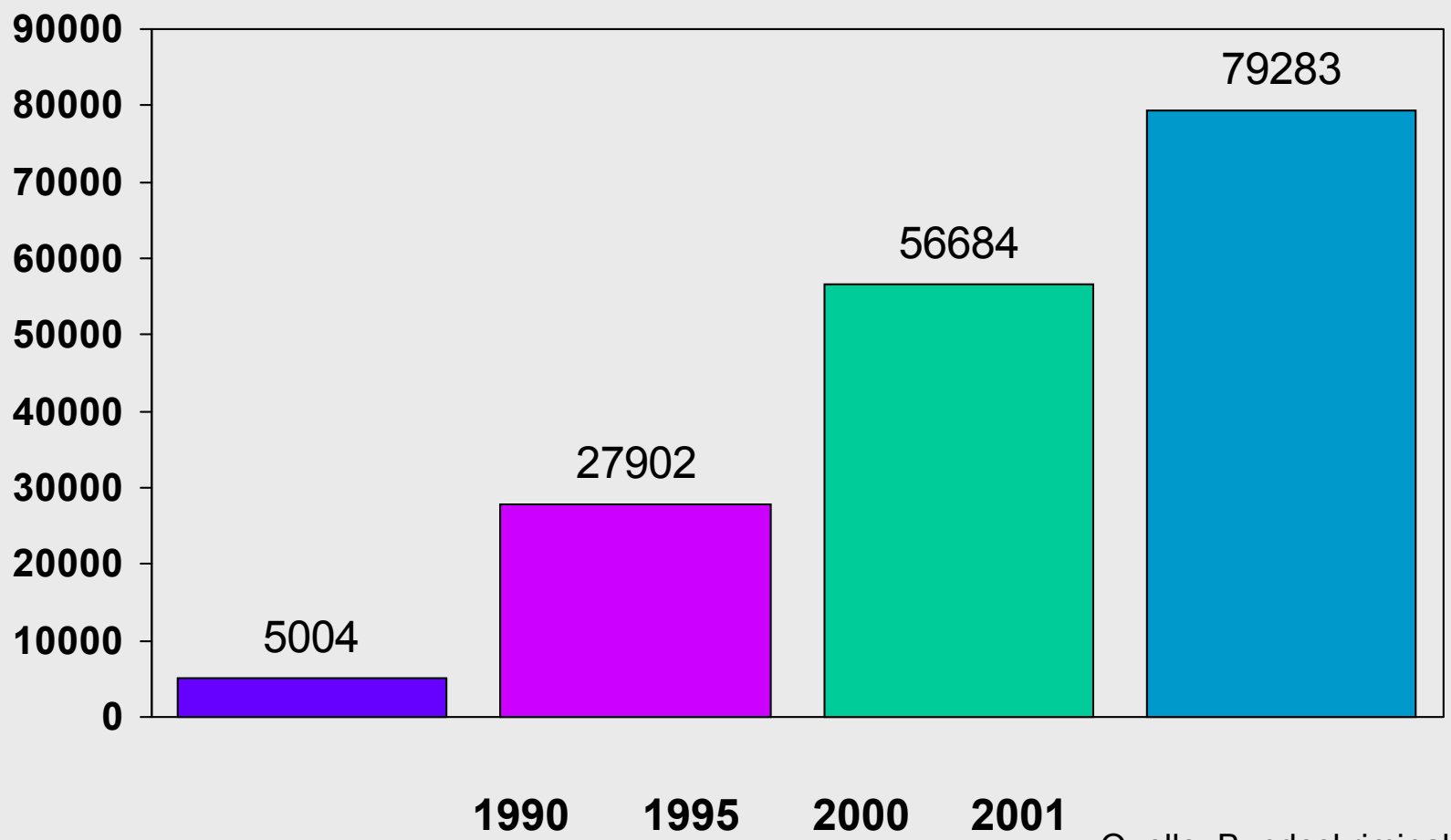
Computerkriminalität

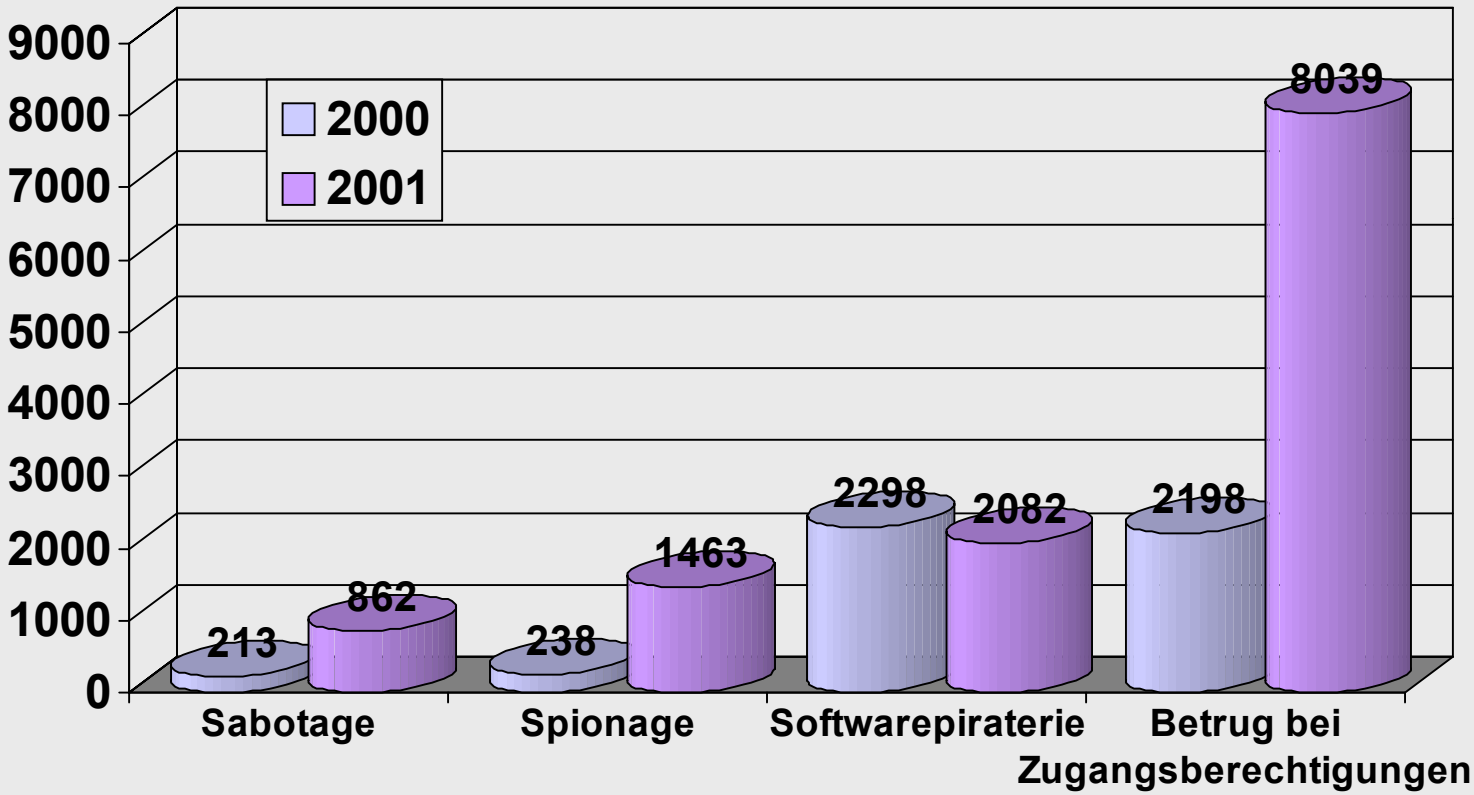
(Optional)



Quelle: Bundeskriminalamt

Ausgewählte Bereiche

(Optional)



Quelle: Bundeskriminalamt

Sicherheit der Objekte im Unix-Dateibaum

Motivation für Angriffe

- Bandbreite
  (Trittbrett, Gruppenangriff)
- CPU
- Festplattenspeicher
  (Warez, Mpgs)
- Daten
  Geschäftsgeheimnisse
- Chaos verursachen

Teil 1

- Dateisystem
- Inode
- Dateiblöcke
- Dateibaum
- Dateieinträge
- Dateimodus
- read / write / execute
- Grundeinstellungen und “umask”

Teil 2

- UserID
- GroupID
- SVTX (Sticky-Bit)
- SetuserID und SetgroupID
- Symbolic Links (im folgenden: Symlink)
- Hardlinks
- Einschränken der Ressoucen für bestimmte Gruppen und Benutzer `ulimit`

Zusammenhänge

Ziele der Angreifer

- Zugreifen auf fremde Dateien (Informationen)
- Programme zu Aktionen verleiten, die nicht vorgesehen sind. (Manipulationen)
- Vollständige Kontrolle über den Rechner (ROOT-Berechtigung)

Layout eines Dateisystems

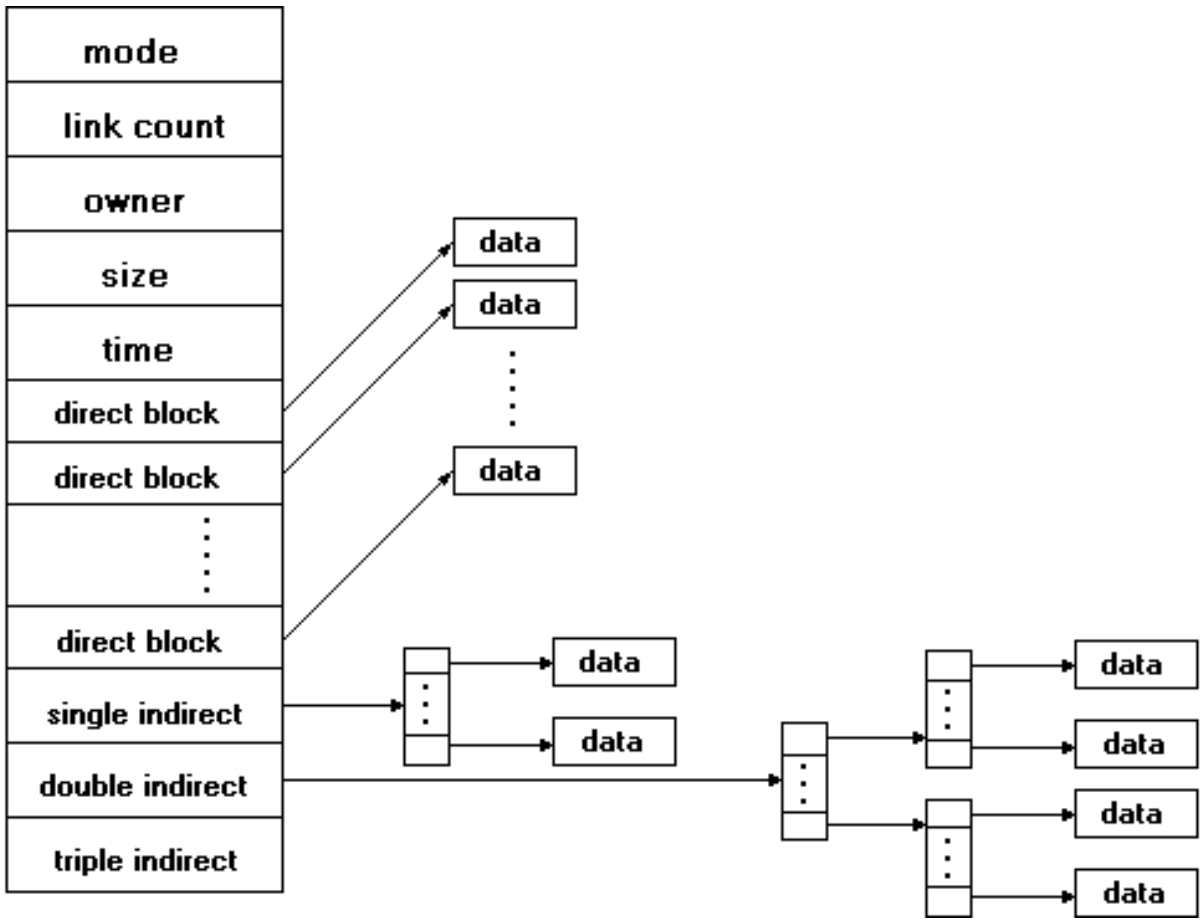
Physikalische Struktur

boot-block | super-block | inodes | data-blocks

Der Superblock enthält Informationen über die Anordnung und den Entwurf des Dateisystems, wie die Zahl der Inodes, der Blöcke...
Durch die Zerstörung des Superblocks wird das das Datei-system unbenutzbar.

Inode (64 Byte groß)

ls -i listet Inode Nr.



Datei-Blöcke

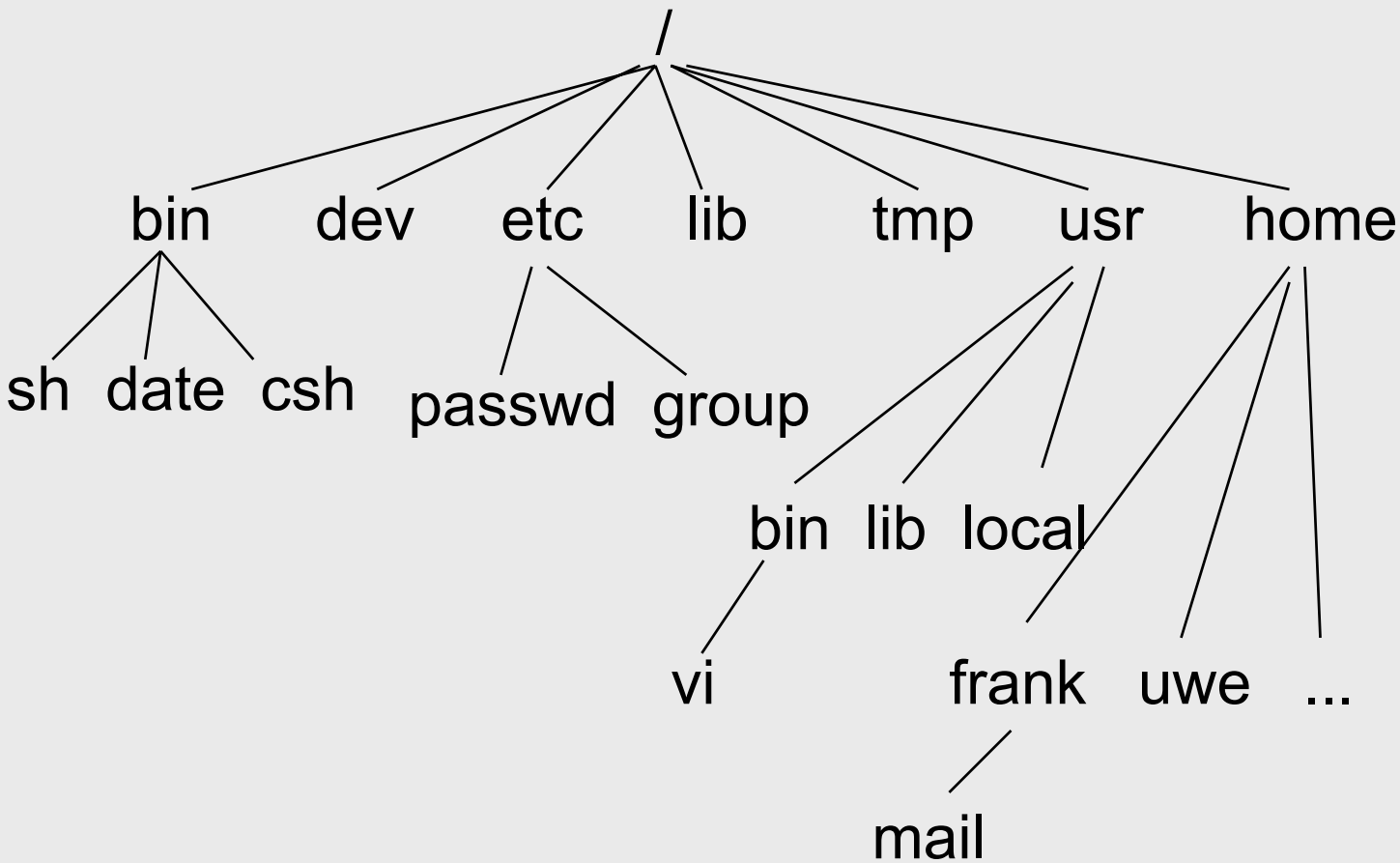
(Optional)

- Inode enthält (logische) Block Adressen
- 10 direct blocks (System V, in SunOS: 12)
- Je 1 single/double/triple indirect block
- Zugriff auf die ersten Datenblöcke sehr schnell
- ca. 85% der Dateien sind kleiner als 8 Kbyte

Bei 1 KByte pro Block erhält man 256 Adressen pro **indirektem** Block.

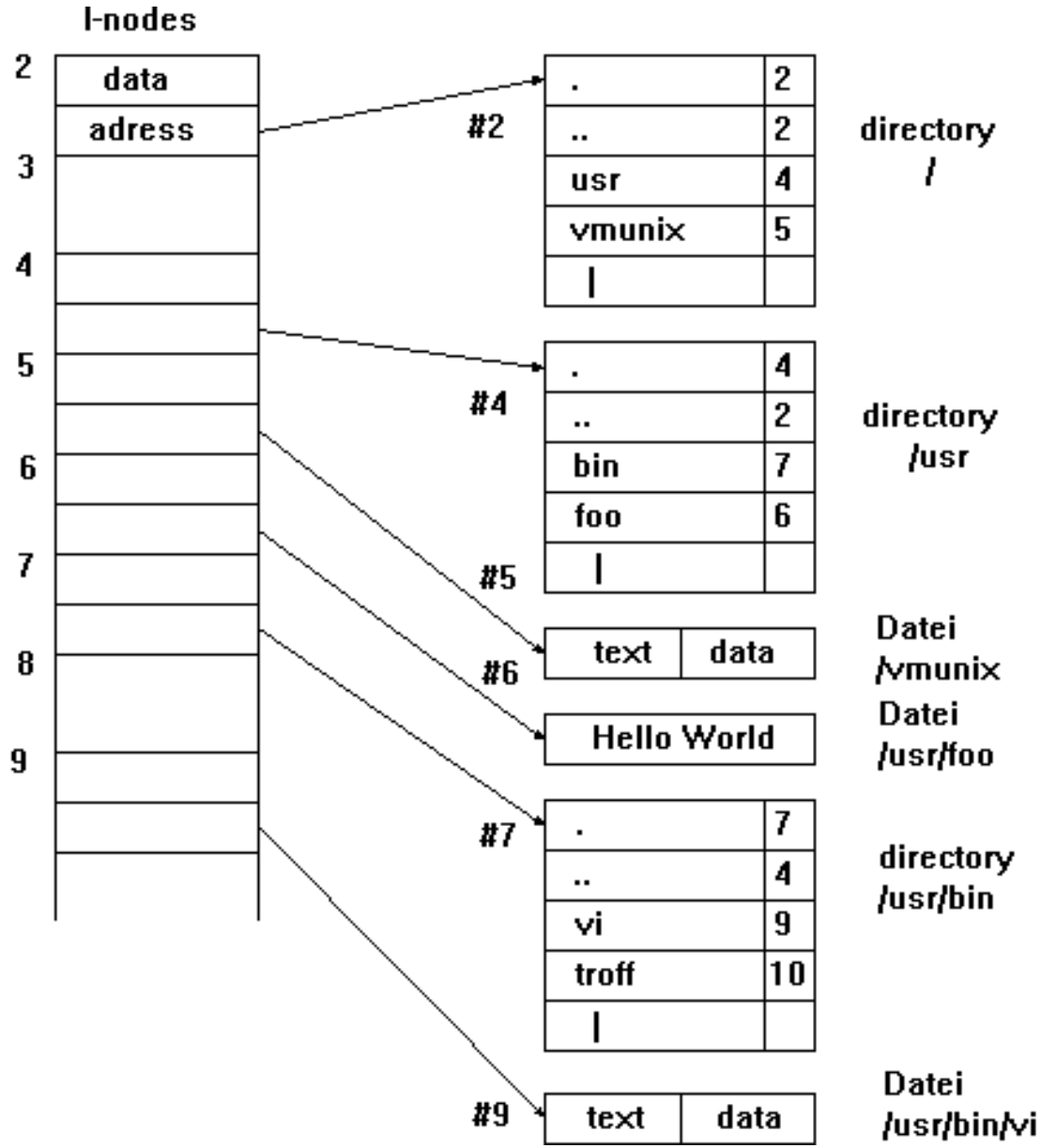
direkte Blöcke --> Größen bis 10 KByte,
indirekte Blöcke --> Größen bis 256 KByte,
zweifach indirekte Blöcke --> Größen bis 64 MByte
dreifach indirekte Blöcke --> Größen von 16GByte.

Der Dateibaum



Beispiel:
Zugriff auf

/usr/bin/vi



Dateieinträge

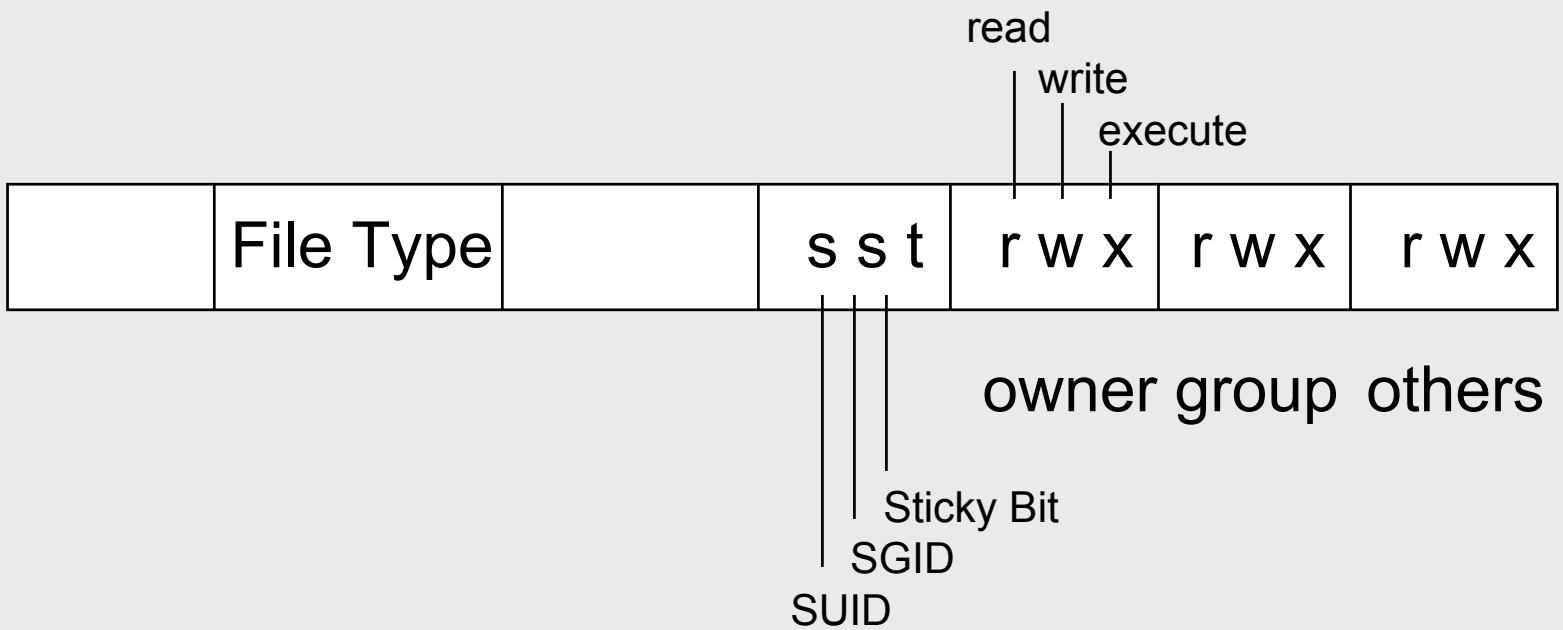
Dateien bestehen aus Ihrer Inode und dem Dateiinhalt

Dateieinträge sind charakterisiert durch den Name und die Inode.

- Gewöhnliche Files (Programme, Texte, Daten)
- Verzeichnisse
- Gerätefiles (Drucker, CD-Rom, Bildschirm, Netzwerk)
- Sockets (TCP/IP-Verbindungen)
- FIFO´s (named pipes)
- Links (hard und symbolic)

Dateimodus

Inode enthält Zugriffsrechte



read (r), write (w), execute (x)

- Dateien kennen unterschiedliche Rechte für den Besitzer, die Gruppe und die übrigen Benutzer.

- ```
  user@linxrechner$ ls -l a.txt
  -rw-rw-r—   1 user   group  24043 Mai 20 08:40  a.txt
  ```

- geändert werden diese Rechte mit “chmod”
- Darstellungsweise: Binär, Oktal oder symbolisch
- Symbole werden in der Reihenfolge owner / group / others interpretiert.
  ```

Bedeutung der Zugriffsrechte

Zugriffsrecht	Datei	Verzeichnis
r	Dateiinhalt lesen	Verzeichnisinhalt lesen
w	Dateiinhalt modifizieren	Anlegen, Umbenennen und Löschen von Dateien
x	Datei Ausführen	In Verzeichnis wechseln

Grundeinstellung “umask”

- Standardmässig kennt das System 777 als Grundeinstellung. Bei Dateien wird zusätzlich noch das x abgezogen.
- `umask` schränkt die allgemeinen Rechte um einen individuell einstellbaren Wert ein.
- Mit einem `umask` Wert = 027 erhalten automatisch also alle Dateien zunächst (- rw- r-- ---)
- Man kann nachträglich mit `chmod` keine Rechte setzen, die über die Voreinstellungen von `umask` hinaus gehen.

ACL (access-control-lists)

- Problem: Es ist nicht möglich zwei Benutzern Rechte an einer Datei zu geben, ohne eine neue Gruppe anzulegen.
- Mit ACL´s kann man Dateien mehreren Benutzern zuordnen

```
[root]# setfacl -m user:claudia:rw- urlaub.txt
[root]# getfacl urlaub.txt
# file: urlaub.txt
# owner: angela
# group: buch
user::rw-
user:claudia:rw-
group::r--
mask:rw-
other:r--
```

- Probleme: Nur lowlevel Backup-Systeme unterstützen ACLs

Attribute bei EXT2

(Optional)

- a - (append): Nur Anhängen möglich.
- d - (dump): Von der inkrementellen Sicherung durch dump ausgenommen.
- i - (immutable): Lässt sich in keiner Weise verändern. (Auch keine Links) - Rootprivilegien sind erforderlich.
- s - (secure): Die Datenblöcke werden physikalisch gelöscht.
- S - (Sync): Jede Veränderung der I-Node wird synchron durchgeführt.
- u - (undelete): Datei wird nach dem Löschen noch intakt gehalten werden, damit das Retten der Daten evtl. möglich ist.
- c - (compressed): Die Datei wird komprimiert gespeichert

UserID

- Jeder Benutzer erhält eine eindeutige Nummer.
- wird in der etc/passwd gespeichert
- Die UID = 0 gehört root. Wer mit “UID = 0” arbeitet, wird von keiner Sicherheitsschranke des Systems aufgehalten.
- Normale Benutzer erhalten in der Regel UID ab einen bestimmten Wert, z.B. 100 oder 500.
- Alle unter 100 liegenden UID's gehören gewöhnlich Systemprozessen.

GroupID

- Unix-Systeme haben mehrere Gruppen.
- Die Administratoren können beispielsweise der Gruppe root oder wheel zugeordnet sein.
- Benutzer sind meistens in der Gruppe user oder werden Gruppen für ihren Tätigkeitsbereich zugeordnet, z.B. wwadmin, students, accounting etc.
- Ein Benutzer kann 1...NGROUPS_MAX (32) Gruppen angehören.
- Jeder Benutzer kann seine Datei jeder Gruppe zuordnen, der er angehört.

SetUID und SetGID

- Das **SetUID-Flag** (Nur für ausführbare Programme) gibt dem Programm beim Ausführen die (effektive) UID des Besitzers und nicht die UID des Ausführenden (reale UID).
- Bsp: Programme zum Ändern des eigenen Passwortes
- Skripte können unter Linux nicht als SetUID-Programme laufen!
- Das gleiche gilt für das **SetGID-Flag**. (Nur bei Verzeichnissen) Eine abgelegte Datei wird nicht unter der eigenen Gruppen-zugehörigkeit, sondern unter der, des Verzeichnisses abgelegt. (Für gemeinsame Projekte sinnvoll)
- Finden aller SetUID Programme mit
  Machine# find / \(-perm –2000 –o –perm –4000\) -ls

SVTX (Sticky-Bit)

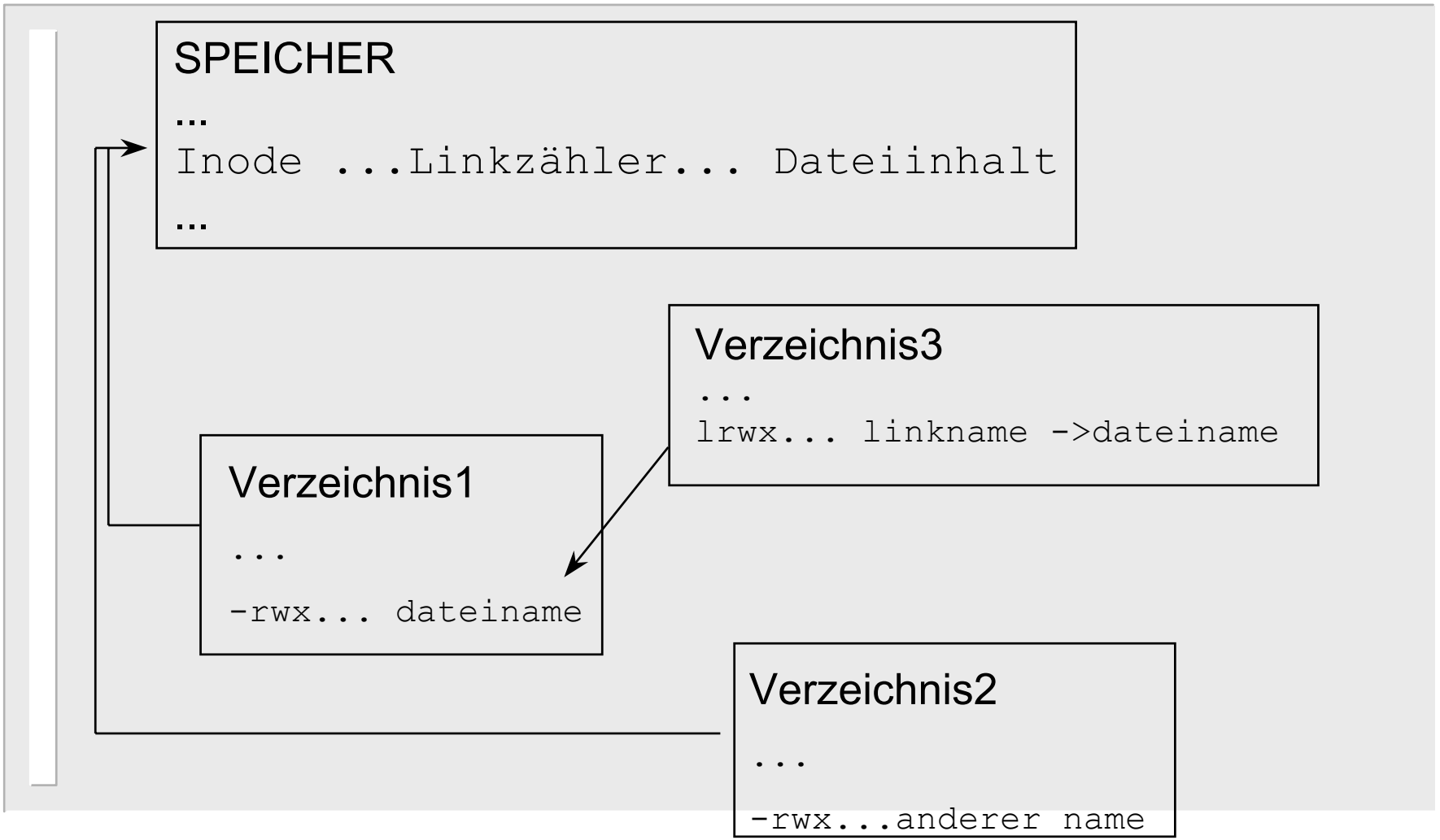
- SVTX (Sticky-Bit) bei Verzeichnissen: Nur der Besitzer eines Eintrages kann diesen Löschen.
- Bsp: Temp-Verzeichnisse

- FRÜHER:
  Bei Programmen kann wiederholtes Ein- und Auslesen von der Festplatte durch das Setzen des **Savetext (SVTX)**-Bits vermieden werden.

Probleme

- Zeitfenster sog. races, die zwischen Prüfen der Rechte und Auführen bestimmter Aktionen entstehen
- Wird bei einem priviligierten Programm, das Dateien im TMP-Verzeichnis ablegt, nicht getestet, ob die zu beschreibende Datei bereits existiert, ist es möglich durch erstellen eines Symlinks mit identischem Namen die Ausgabe in eine andere Datei umzuleiten.

Hardlinks und Symlinks



Einen Hardlink anlegen

Befehl: ln [quelle] [ziel]

```
Machine$ ls - li
876193 -rw- --- --- 1 george twinlks  07 May 08:15 file1
578283 -rw- --- --- 1 bonnie twinlks  19 Dec 10:39 file2

Machine$ ln file2 link1
Machine$ ls - li
876193 -rw- --- --- 1 george twinlks  07 May 08:15 file1
578283 -rw- --- --- 2 bonnie twinlks  19 Dec 10:39 file2
578283 -rw- --- --- 2 bonnie twinlks  19 Dec 10:39 link1
```

Hardlinks

- Verschiedene Namen für gleiche Datei (gzip, gunzip)
- Hard verlinken ist nur innerhalb eines Dateisystems möglich. (Die Inode beginnt in jedem Dateisystem wieder bei 1)
- Das File ist unabhängig vom Dateinamen
- Link hat identische Daten/Rechte/Eigenschaften!
- Wenn Linkzähler = 0 kann File überschrieben werden.
- Nur auf Dateien können Hardlinks gezeigt werden (Ausnahme: Superuser! Genau genommen ist z.B. (..) ein Hardlink in das übergeordnete Verzeichnis.)

Symlinks

- Inhalt der Verzeichnisdatei enthält einen Pfad
- Pfad kann auf alles zeigen
- Über Dateisystemgrenzen möglich (Bei absoluten Pfadangaben)
- Symlinks können auch auf Verzeichnisse zeigen
- Wird ein Symlink gelöscht bleibt die Datei unberührt
- Wird die Datei gelöscht zeigt der Link ins Leere
- ln -s [quelle] [ziel]

Einschränkungen von Ressourcen “ulimit”

- Gleichzeitige Prozesse
- CPU
- Speicherplatz
- Soft- und Hard-limits
  (Warnung bzw. Sperren)

- Bsp. Core-Dateien: Tritt in einem Pro-gramm ein kritischer Fehler auf (vulgo: Wenn es abstürzt), dann schreibt dieses Programm einen Speicherabzug von sich auf die Platte, bevor es terminiert.
- `ulimit -c unlimited`

Schlussbemerkung

- fehlerhaft erteilte Dateiberechtigungen sind die Türen für potentielle Angreifer
- Sorgfältiger Umgang mit Zugangsberechtigungen erhöhen die Wahrscheinlichkeit einer erfolgreichen Gegenwehr
- Wenn root erobert ist helfen keine Sicherheitsvorkehrungen mehr.
- Eine Möglichkeit jedoch die Allmächtigkeit des Root-users einzuschränken bieten Programme, wie etwa LIDS

Literatur

- **(Hatch, Lee, Kurtz) Das Anti-Hacker Buch, mitp-Verlag Bonn, 2001**
- **Vortrag zu Computerkriminalität des BKA**
  http://www.bka.de/aktuell/agenda98/vtr99/vtr_moeller.html
- **Statistik des Bundeskriminalamtes**
  http://www.bka.de/text/pks.html
- **Vom Menschen zum Unix-Hacker**
  http://www.computec.ch/dokumente/unix/vom_menschen_zum_unix-hacker/vom_menschen_zum_unix-hacker.html
- **PC-Magazin Go!Linux**
  http://www.pc-magazin.de
- **Vorlesungsskripte der Unis und Hochschulen**
- (Erlangen, Tübingen, Hamburg, München, Stuttgart, Karlsruhe)

Sonstige Internetseiten:

- **Einführung in Linux:**
http://www.linuxinfo.de/de/grundlagen/linux/index.html
- **Die Linuxfibel**
http://www.linuxfibel.de/
- **Einführung in Unix der Firma Siemens**
http://w4.siemens.com/schule/unix/unix.html
- **www.susse.de**
- **www.heisse.de**
- **linux.org**
- **linux.de**
- **www.oreilly.de**