Benutzerverwaltung
Rechte – Teil 2

Benutzerverwaltung

- Typische Aufgaben im Zusammenhang mit Benutzerverwaltung
  - das Anlegen und Löschen von Benutzerkonten,
  - die Prüfung der Qualität von Passwörtern (Password Policy festlegen),
  - Änderungen von Passwörtern, welche die Benutzer vergessen haben und
  - die Überwachung des von Anwendern belegten Speicherplatzes

Rechte - WICHTIG

Beim Löschen einer Datei ist nur Schreibrecht auf dem Verzeichnis, NICHT jedoch für die Datei erforderlich.

Ein Benutzer mit Schreibzugriff auf einem Verzeichnis kann also Dateien in diesen Verzeichnis löschen, auch wenn er an den Dateien selbst KEINE Rechte hat.

Berechtigungen neue VZ/Dateien

- mit umask kann man Voreinstellungen für die Rechte neu angelegter Dateien setzen
- *umask [Berechtigungsmaske]*
- Die Eingabe von umask ohne Parameter gibt die aktuell eingestellte Maske wieder
- maske ist eine 3-oder 4-stellige Oktalzahl (abhängig vom System)
- umask gibt nicht an, welche Rechte gegeben werden, sondern welche entzogen werden

777 - Berechtigungsmaske = Berechtigung für VZ

umask 022:	777-022=755	rwx	r-x	r-x
umask 027:	777-027=750	rwx	r-x	---
umask 177:	777-077=700	rwx	---	---

666 - Berechtigungsmaske = Berechtigung für Datei

umask 022:	666-022=644	rw-	r--	r--
umask 027:	666-026=640	rw-	r--	---
umask 177:	666-066=600	rw-	---	---

Achtung:
Es gibt nur **eine** umask, die auf Verzeichnisse und reguläre Dateien angewendet wird!!!

SUID und SGID (1)

- SUID (Set User ID) und SGID (Set Group ID) werden durch ein s anstelle eines x bei den Eigentümer- und bei den Gruppenrechten ausgedrückt

- Das s-Recht an einer ausführbaren Datei bedeutet, daß der Benutzer, der es startet, während des Programmlaufes die UID des Dateibesitzers bzw. die GID der Besitzergruppe erhält. Das hängt davon ab, ob das s beim Besitzer oder der Besitzergruppe steht. Im Falle des Programms */bin/passwd wird man z. B. während des* Programmlaufs zum Superuser!

- z.B. `ls -l /etc/passwd /etc/shadow /usr/bin/passwd`

```
-rw-r--r-- 1 root root 1377 2008-03-20 12:55 /etc/passwd
-rw-r----- 1 root shadow 911 2008-03-20 12:54 /etc/shadow
-rwsr-xr-x 1 root root 29104 2007-05-18 11:59 /usr/bin/passwd
```

SUID und SGID (2)

- wird das SGID-Recht auf ein Verzeichnis gegeben, so hat es eine andere Funktion. In diesem Fall erhalten alle Dateien, die in diesem Verzeichnis erstellt werden, automatisch die gleiche Gruppe, die auch dem Verzeichnis zugeordnet ist
- z.B. (angemeldet als root)

```
ls –l /home/susi
drwxr-sr-x 2 susi users 4096 2009-03-22 22:21 mydir

cd /home/susi/mydir
touch a b
ls –l
-rw-r--r-- 1 root users 0 2009-03-22 23:27 a
-rw-r--r-- 1 root users 0 2009-03-22 23:27 b
```

Sticky-Bit

- das Sticky-Bit wird heutzutage meistens auf Verzeichnissen gesetzt
- nur root oder der Eingentümer können Dateien löschen oder umbenennen
- ist das Sticky-Bit nicht gesetzt, kann jeder Benutzer mit Schreibrecht auf einem Verzeichnis Dateien löschen oder umbenennen
- auf Dateien ist dieses Bit mittlerweile unüblich und wird je nach Unix-Derivat sogar ignoriert
- mit *chmod +t dirname kann man das Sticky-Bit setzen*
- Beispiel:
  - /tmp ist Verzeichnis, in das viele Anwendungen temporär Files während der Ausführung schreiben
  - `ls -ld /tmp`

```
drwxrwxrwt 4 root sys 485 Nov 10 06:01 /tmp
```

Ändern der Zugriffsrechte

- Eigentümer kann Zugriffsrechte auf seine Datei ändern
- Superuser kann alle Rechte ändern

```
chmod [Bereich] Operand Berechtigung
```

Bereich	u	Eigentümer (user)
	g	Gruppe (group)
	o	Übrige Benutzer (others)
	a	alle
Operand	+	Recht hinzufügen
	-	Recht wegnehmen
	=	Recht absolut setzen
Berechtigung	r	Read
	w	Write
	x	execute
	s	UID/GID (abhängig von Bereich) [Zugriff mit User/Group-Rechten]
	t	Sticky-Bit [Dateien können nur vom Eigentümer gelöscht werden]

chmod numerisch

Recht	Oktalzahl
Read	4
Write	2
Execute	1

Recht	Oktalzahl
SUID	4
SGID	2
STICKY	1

Pro Berechtigungsklasse wird eine Oktalzahl angegeben.
Zugeteiltes Recht ergibt sich aus der Summe der Einzelrechte:

rwx =
read + write + execute ergibt 4+2+1 = 7

777 ergibt rwx rwx rwx (ugo)
754 ergibt rwx r-x r-- (ugo)
4711 ergibt rws --x --x (SUID,ugo)
2712 ergibt rwx --s -w- (SGID,ugo)
1712 ergibt rwx --x --t (Sticky, ugo)

Beispiele chmod (1)

Ändern der Berechtigungen auf Dateien mit chmod

Beispiele:

```
chmod 644 Makefile
chmod 755 /opt/myapp
chmod 700 privat.dat
chmod u=r,go=rwx privat.dat
chmod u+rwx,g-rwx privat.dat
chmod a=rwx privat.dat
```

(a=all, Recht wird für u,g,o gemeinsam gesetzt)

Beispiele chmod (2)

Ändern der Berechtigungen auf Verzeichnissen mit chmod

Beispiel:

```
chmod -R g=rwx /home/bsy2vzg1
```

setzt die Rechte für alle Dateien und Verzeichnisse in `/home/bsy2vzg1` und allen Unterverzeichnissen.

Ändern Eigentümer / Gruppe

- Eigentümer kann geändert werden
  - Benutzer wird neuer Eigentümer, alter Eigentümer verliert Rechte an Datei
  - `chown username dateiliste`
  - Befehl kann nur vom Superuser ausgeführt werden

    Kein „Verschenken“ von Dateien durch normale Benutzer möglich
- Gruppenzugehörigkeit einer Datei ändern
  - `chgrp gruppenname dateiliste`
  - Befehl kann nur vom Superuser oder vom Eigentümer ausgeführt werden, sofern dieser auch Mitglied der neuen Gruppe ist.
- Manchmal auch vorhanden:
  - `chown Benutzer:Gruppe Dateien`
- Rekursives Ändern über mehrere VZ-Ebenen
  - `chown -R root /home`