Benutzer und Rechte Teil 1

Linux-Kurs der Unix-AG

Zinching Dang

19. November 2012

UNIX AG
TU Kaiserslautern

RH Regionales Hochschul-
Rechenzentrum Kaiserslautern RK

Wozu verschiedene Benutzer? (1)

Datenschutz

- mehrere Benutzer pro Rechner, insbesondere auf Server-Systemen
- unterschiedliche Zugriffsrechte für einzelne Benutzer und Gruppen
- Benutzer können ihre eigenen Daten vor dem Zugriff anderer Benutzer schützen

Wozu verschiedene Benutzer? (2)

Sicherheit

- nicht jeder Benutzer kann Systemänderungen vornehmen
- nur `root` darf solche Änderungen durchführen
- normale Benutzer müssen zunächst `root`-Rechte erlangen
- Nachvollziehbarkeit von Benutzeraktionen bei (Sicherheits-)Problemen
  - wer hat sich wann und von wo aus eingeloggt
  - welche Systemänderungen wurden wann und von wem durchgeführt

Benutzer und Gruppen unter Linux

- Unterscheidung zwischen Benutzern und Gruppen
- jeder Benutzer hat eine Benutzerkennung (UID) und eine primäre Gruppenkennung (GID)
- jeder Benutzer kann beliebig viele sekundäre Gruppen haben
- Gruppen können mehrere Benutzer enthalten und Benutzer können Mitglied mehrerer sekundärer Gruppen sein

id

- listet die UID, GID und Gruppenzugehörigkeiten auf
- wichtige Optionen:
  - `-u`: zeigt die UID an
  - `-g`: zeigt die GID der primären Gruppe an
  - `-G`: listet die GIDs aller Gruppen auf (primäre und sekundäre)
  - `-n`: nur in Kombination mit `-u`, `-g` und/oder `-G`: zeigt den zugehörigen Namen anstatt der ID an
- Argument:
  - optional: Benutzername eines anderen Benutzers
  - ohne Argument: wie Aufruf mit eigenem Benutzernamen

groups

- listet die Gruppenzugehörigkeiten auf
- hat keine Optionen
- Argument:
  - optional: Benutzername eines anderen Benutzers
  - ohne Argument: wie Aufruf mit eigenem Benutzernamen

w/who

- `w` und `who`: listet alle eingeloggten Benutzer auf (lokale und remote Logins, sowohl grafische als auch Shell)
- `w` zusätzlich: Uptime des Systems und aktuelle Systemauslastung, in der Shell laufende Prozesse
- Argument (nur bei `w`):
  - optional: Benutzername - es werden dann nur die Einträge vom angegebenen Benutzer angezeigt
  - ohne Argument: es werden alle eingeloggten Benutzer angezeigt

last

- listet auf, wer sich wann und von wo aus eingeloggt hat
- wichtige Optionen:
  - `-n anzahl`: listet die letzten `n` Logins auf
  - `-F`: zeigt an, wann sich der jeweilige Benutzer ein- und ausgeloggt hat
  - `-a`: zeigt bei Remote-Logins den vollständigen Hostname an
- Argument:
  - optional: Benutzername - es werden dann nur die Einträge vom angegebenen Benutzer angezeigt
  - ohne Argument: es werden alle verfügbaren Einträge angezeigt

Benutzertypen

- „echte“ Benutzer
  - werden vom Administrator angelegt
  - können sich auf dem System einloggen
- System-Benutzer
  - werden für Server-Dienste verwendet
  - haben UID aus anderem Bereich
  - haben **kein** Passwort
  - sollten sich auf dem System **nicht** einloggen
  - Ausnahme: `root`

su

- wechselt den Benutzer in der Shell, d.h. man loggt sich mit den Zugangsdaten eines anderen Benutzers ein
- wichtige Optionen:
  - `- oder -l`: Umgebung wie bei „echtem“ Login wird bereitgestellt, d.h. man befindet sich im `home`-Verzeichnis des neu eingeloggten Benutzers
  - `-c`: führt einen Befehl in der Shell als anderen Benutzer aus (vgl. `sudo`)
- Argument:
  - optional: Benutzername - Zugangsdaten des Benutzers erforderlich (Benutzername und Passwort)
  - ohne Argument: man loggt sich als `root` ein (`root`-Passwort erforderlich)

sudo

- setzt voraus, dass man in der `/etc/sudoers` steht, d.h. die Berechtigung hat, den Befehl `sudo` aufzurufen
- führt einen Befehl als einen anderen Benutzer aus (standard: `root`)
- wichtige Optionen:
  - `-u`: gibt den Benutzer an, mit dessen Rechten das Programm ausgeführt werden soll
  - `-i`: wie `su -l`, optional: Programm, das direkt ausgeführt werden soll
- Argument:
  - Programm, das mit anderen Benutzer-Rechten ausgeführt werden soll - zwingend erforderlich, wenn nicht mit der Option `-i` aufgerufen

Unterschiede zu Microsoft Windows

- Windows:
  - Installation einzelner Pakete über Datei-Download oder CD/DVD
  - keine zentrale Verwaltung für Installation neuer Pakete
- Linux-Distributionen:
  - zentrale Tools für Installation und Verwaltung der Pakete
  - Datenbank mit verfügbaren Paketen aus den Paketquellen (Repository)

Paketverwaltungssysteme

Debian

- Debian-basierte Systeme (Debian, Ubuntu)
- `dpkg -i <Paket.deb>`
- `apt-get install <Paketname>`
- `dpkg`, `apt-get` und `aptitude` als Konsolen-basierte Installationstools

RPM

- RPM-basierte Systeme (Red Hat, SUSE)
- `rpm -i <Paket.rpm>`
- Red Hat: `yum install <Paketname>`
- SUSE: `zypper in <Paketname>`

Paketverwaltungssysteme

Installation und Deinstallation von Paketen

- Installation:
  - `apt-get install <Paketname>`
- Deinstallation:
  - `apt-get remove <Paketname>`
  - `purge` statt `remove` löscht auch Konfigurationsdateien
- Alternativen:
  - deb-Paket downloaden und mit `dpkg -i <Paketname.deb>` installieren bzw. `dpkg -r <Paketname>` deinstallieren
- Aber:
  - `apt-get` lädt die Pakete automatisch herunter und installiert ggf. Abhängigkeiten

Paketdatenbank

Zweck der Paketdatenbank

- enthält Informationen über alle Pakete, die über die Paketverwaltung installiert werden können
  - Paketname
  - Beschreibung
  - Version
  - Abhängigkeiten

Details über Pakete

- `apt-cache show <Paketname>`

Pakete suchen

- `apt-cache search <Paketname>`

Spickzettel

Alle Befehle

Befehl	Optionen	Argument
`id`	`-u, -g, -G, -n`	[Benutzername]
`groups`		[Benutzername]
`w/who`		[Benutzername]
`last`	`-n ANZAHL, -F, -a`	[Benutzername]
`su`	`-, -l, -c BEFEHL`	[Benutzername]
`sudo`	`-i, -u BENUTZERNAME`	Befehl
`apt-get install`	Paket(e)	
`apt-get remove`	Paket(e)	
`apt-get update`		
`apt-cache search`	Paket	
`apt-cache show`	Paket	