

Benutzer und Rechte Teil 2, Paketverwaltung

Linux-Kurs der Unix-AG

Benjamin Eberle

7. Dezember 2015



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
 - ▶ Dienste, die unter einem normalen Benutzer laufen, können so nicht direkt das ganze System übernehmen falls sie kompromittiert werden
- ▶ Nachvollziehbarkeit von Benutzeraktionen
 - ▶ wer hat sich wann und von wo aus eingeloggt
 - ▶ welche Systemänderungen wurden wann und von wem durchgeführt

w/who

- ▶ `w` und `who`: listet alle derzeit eingeloggten Benutzer auf (lokale und remote Logins, sowohl grafische als auch Shell)
- ▶ `w` zusätzlich: Uptime des Systems, aktuelle Systemauslastung und 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
 - ▶ `-a` : zeigt bei Remote-Logins den vollständigen Hostname an
- ▶ Argument:
 - ▶ optional: Benutzername - es werden dann nur die Einträge vom angegebenen Benutzer angezeigt
 - ▶ statt Benutzername `reboot` : es werden die Systemstarts angezeigt
 - ▶ ohne Argument: es werden alle verfügbaren Einträge angezeigt

lastlog

- ▶ zeigt an, wann und von wo sich jeder Benutzer zum letzten Mal eingeloggt hat
- ▶ wichtige Optionen:
 - ▶ `-u benutzer`: zeigt nur den letzten Login dieses Benutzers an

Benutzertypen

- ▶ „echte“ Benutzer
 - ▶ werden vom Administrator angelegt
 - ▶ werden für das normale Arbeiten verwendet
 - ▶ können sich auf dem System einloggen
- ▶ System-Benutzer
 - ▶ werden für Server-Dienste verwendet
 - ▶ haben eine UID aus einem reservierten Bereich
 - ▶ haben kein Passwort
 - ▶ sollten sich auf dem System nicht einloggen
 - ▶ Ausnahme: root

Benutzerkennung (UID)

- ▶ jeder Benutzer hat eine eindeutige UID
 - ▶ Benutzer werden systemintern anhand der UID unterschieden
 - ▶ Benutzernamen sind für Menschen leichter zu handhaben als UIDs
 - ▶ Benutzerrechte hängen von der UID ab
 - ▶ mehrere Benutzer können auch dieselbe UID haben, sind für das System dann der gleiche Benutzer
- ▶ Benutzer root
 - ▶ spezieller Systembenutzer mit der UID 0
 - ▶ für administrative Aufgaben gedacht
 - ▶ hat vollen Zugriff auf alles

- ▶ 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 (Login-Shell) wird bereitgestellt, d.h. man befindet sich im home-Verzeichnis des neu eingeloggt Benutzers
 - ▶ `-c`: führt einen Befehl in der Shell als anderen Benutzer aus (vgl. `sudo`)
- ▶ Argument:
 - ▶ optional: Benutzername: man loggt sich als angegebenen Benutzer ein (Passwort erforderlich)
 - ▶ ohne Argument: man loggt sich als root ein (root-Passwort erforderlich)

sudo

- ▶ führt einen Befehl als anderer Benutzer aus (Standard: root)
- ▶ setzt voraus, dass man in der `/etc/sudoers` steht, d.h. die Berechtigung hat, den Befehl `sudo` aufzurufen
- ▶ Authentifikation mit **eigenem** Passwort
- ▶ wichtige Optionen:
 - ▶ `-u`: gibt den Benutzer an, mit dessen Rechten das Programm ausgeführt werden soll
 - ▶ `-i`: startet eine Login-Shell; optional kann zusätzlich ein Programm, das darin ausgeführt werden soll, angegeben werden
- ▶ Argument:
 - ▶ Programm, das mit anderen Benutzer-Rechten ausgeführt werden soll – 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)
 - ▶ Installieren von Software ohne Verwendung des Paketsystems auch möglich, jedoch nicht zu empfehlen

Paketverwaltungssysteme

Debian

- ▶ Debian-basierte Systeme (Debian, Ubuntu)
- ▶ `dpkg -i <Paket.deb>`
- ▶ `apt-get install <Paketname>`

RPM

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

Installation und Deinstallation von Paketen

- ▶ Installation:
 - ▶ `apt-get install <Paketname>`
 - ▶ `apt-get` lädt die Pakete automatisch herunter und installiert ggf. Abhängigkeiten
- ▶ Deinstallation:
 - ▶ `apt-get remove <Paketname>`
 - ▶ `purge` statt `remove` löscht auch Konfigurationsdateien
- ▶ Verfügbare Updates für die installierten Pakete einspielen:
 - ▶ `apt-get upgrade`

Paketdatenbank (1)

Zweck der Paketdatenbank

- ▶ enthält Informationen über alle Pakete, die über die Paketverwaltung installiert werden können

Paketdatenbank aktualisieren

- ▶ `apt-get update`

Paketdatenbank (2)

Details über Pakete

- ▶ `apt-cache show <Paketname>`

Pakete suchen

- ▶ `apt-cache search <Paketname>`

Spickzettel

Alle Befehle

Befehl	Optionen	Argument
w/who		[Benutzername]
last	-n Anzahl, -a	[Benutzername]
lastlog	-u Benutzername	
su	-, -l, -c Befehl	[Benutzername]
sudo	-i, -u Benutzername	Befehl
apt-get	update, install, remove	Paket(e)
apt-cache	search, show	Paket