SelfLinux-0.13.0

Benutzerverwaltung unter Linux

Autor: Heiko Degenhardt (*hede@pingos.org*)
Formatierung: Matthias Hagedorn (*matthias.hagedorn@selflinux.org*)

Inhaltsverzeichnis

1 Einleitung

Dieses Kapitel führt in das Benutzerkonzept von Linux ein. Die Benutzerverwaltung sowie alle daraus resultierenden Aufgaben werden in späteren Kapiteln ausführlich behandelt. Die Grundlagen zum Benutzer- und Berechtigungskonzepte unter Linux werden in diesem Kapitel erklärt.

2 Grundlagen

Zu den wichtigsten Eigenschaften von Linux zählt sicher die Fähigkeit zu echtem Multitasking und zum Multiuser-Betrieb. **"Multitasking"** bedeutet dabei, dass mehrere Prozesse (englisch **"tasks"**) parallel ablaufen, und sich dabei die vorhandenen Ressourcen teilen. **"Multiuser"** bedeutet, dass mehrere Benutzer gleichzeitig am System arbeiten und damit die Multitasking-Fähigkeiten des Systems nutzen können. Hierbei trennt das Betriebssystem die einzelnen Benutzer, die von diesen Benutzern gestarteten Prozesse/Tasks sowie die von ihnen angelegten Dateien und Verzeichnisse voneinander und kontrolliert restriktiv den Zugriff darauf.

In der Windows-Welt ist dieser Multiuser-Betrieb wenig bekannt. Zwar ist es auch unter Windows z. B. möglich, mehrere Benutzer auf demselben System anzulegen. Diese können im Normalfall aber nicht gleichzeitig an diesem System arbeiten.

3 Vorteile von Multiuser-Systemen

Für Privatanwender scheint ein Multiuser-System zunächst wenig vorteilhaft zu sein. In erster Linie denkt man bei solchen Systemen sicher an Server, auf die über Shell-Accounts, FTP, Internet usw. viele Kunden gleichzeitig zugreifen können. Trotzdem bringt dieses Konzept auch für **"normale"** Anwender entscheidende Vorteile:

* Ein Multiuser-System ist eine gute Basis für diszipliniertes Arbeiten. Man kann z. B. sehr genau beobachten, unter welchem Account man welche Aufgaben erledigt.
* Es bietet eine wesentlich höhere Sicherheit als ein "Single-User-System", da die Zugriffsrechte auf Daten, Prozesse etc. je nach Benutzer und Nutzergruppe kontrolliert und eingeschränkt werden können.
  Dadurch ist es z. B. möglich, dass ein Virus nur Zugriff auf die Daten desjenigen Benutzers erhält, über den er in das System gelangt ist. Voraussetzung dafür ist natürlich das bereits genannte diszipliniertes Arbeiten.

4 Das Benutzerkonzept von Linux

Unter Linux werden alle Benutzer, die zur Arbeit am System berechtigt sein sollen, zu Benutzergruppen zusammengefasst. Alle Ressourcen können nun für bestimmte Gruppen oder Benutzer freigegeben werden. Diese Rechtevergabe beruht auf:

* User-ID (uid)
  Dies ist die ID, mit der sich ein Benutzer am System anmeldet.
* effektive User-ID (euid)
  Dies ist die ID, mit der ein Benutzer auf eine Ressource zuzugreifen versucht.
* Group-ID (gid)
  Dies ist die ID, mit der sich eine Gruppe am System anmeldet.
* effektive Group-ID (egid)
  Dies ist die ID, mit der eine Gruppe auf eine Ressource zuzugreifen versucht.

Für die Zugriffskontrolle überprüft das System bei jedem Zugriff auf eine Ressource, unter welcher effektiven uid und guid der Zugriff erfolgt. Durch diese Unterscheidung von ID und effektiver ID ist es z. B. möglich, dass ein Benutzer zwar unter seiner eigenen uid arbeitet, zugleich aber (temporär) auf Dateien zugreift, die anderen Gruppen gehören.

Beispiel:

> Wird das Passwort eines Benutzers geändert, so muss das Programm `passwd` z. B. auf die Datei /etc/shadow zugreifen. Normalerweise ist kein Benutzer berechtigt, auf diese Datei zuzugreifen. Trotzdem kann das Programm `passwd` diese Datei lesen, da es während dieses Vorganges einmal von der `uid` des aufrufenden Benutzers zur `uid` von root wechselt und damit über alle notwendigen Rechte verfügt. Dies ist möglich, da bei der ausführbaren Datei `/usr/bin/passwd` das Bit **"set uid"** gesetzt ist (in der folgenden Codezeile wird dies durch das erste "s" angezeigt):

```
user@linux ~/temp/ $ ls -al /usr/bin/passwd
-rwsr-xr-x    1 root     root        24680 Apr  7 17:59
```

Weitere Informationen hierzu finden Sie mit `"info chmod"` und `"man chmod"`. Mehr Information zu man und info finden Sie im Kapitel Linux Hilfe.

5 Wichtige Befehle

5.1 ls -al - Anzeigen von Datei- und Verzeichniseigenschaften

Beispiel:

```
user@linux ~/temp/ $ ls -al

total 8
drwxr-xr-x    2 hede     hede         4096 Jul 10 07:25 .
drwxr-xr-x    8 hede     hede         4096 Jul 10 07:25 ..
-rw-r--r--    1 hede     hede            0 Jul 10 07:25 test.txt
```

Die Datei test.txt gehört dem Benutzer hede und der Gruppe hede. Der Eigentümer hat Lese- und Schreibzugriff auf die Datei, während die Gruppe und alle anderen Benutzer die Datei nur lesen dürfen.

5.2 chmod - Ändern der Dateizugriffsrechte

Beispiel:

> Das folgende Shellkommando bestimmt, dass alle Mitglieder der Gruppe auch Schreibzugriff auf die Datei erhalten und andere Benutzer sie nicht lesen dürfen:

```
user@linux ~/temp/ $ chmod 660 test.txt
```

Überprüfung:

```
user@linux ~/temp/ $ ls -al

total 8
drwxr-xr-x    2 hede     hede         4096 Jul 10 07:25 .
drwxr-xr-x    8 hede     hede         4096 Jul 10 07:25 ..
-rw-rw----    1 hede     hede            0 Jul 10 07:25 test.txt
```

5.3 chown - Ändern des Eigentümers und der Gruppenzugehörigkeit von Dateien und Verzeichnisse

Test:

```
user@linux ~/temp/ $ ls -al

total 8
drwxr-xr-x    2 hede     users        4096 Jul 10 07:25 .
drwxr-xr-x    8 hede     users        4096 Jul 10 07:25 ..
-rw-rw----      1 user      users              0 Jul 10 07:25 test.txt
```

Die folgende Codezeile ändert den Besitzer der Datei:

```
user@linux ~/temp/ $ chown hede.users test.txt
```

```
user@linux ~/temp/ $ ls -al

total 8
drwxr-xr-x    2 hede     users         4096 Jul 10 07:25 .
drwxr-xr-x    8 hede     users         4096 Jul 10 07:25 ..
-rw-rw----      1 hede     users               0 Jul 10 07:25 test.txt
```

Um den Eigentümer und die Gruppe zu ändern, braucht man ausreichende Rechte. Meist müssen derartige Änderungen also vom root-Benutzer vorgenommen werden. Grundsätzlich können sie nur von einem Angehörigen derjenigen Gruppe durchgeführt werden, die zum Eigentümer der Datei erklärt werden soll.

5.4 chgrp - Ändern der Gruppenzugehörigkeit von Dateien oder Verzeichnissen

Auch mit diesem Befehl kann man die Gruppenzugehörigkeit von Dateien und Verzeichnissen ändern. Soll nun die Gruppe users der Eigentümer der Datei test.txt werden, geht man folgendermaßen vor:

```
user@linux ~/temp/ $ chgrp users test.txt
```

Nachweis:

```
user@linux ~/temp/ $ ls -al

total 8
drwxr-xr-x     2 hede      hede          4096 Jul 10 07:25 .
drwxr-xr-x     8 hede      hede          4096 Jul 10 07:25 ..
-rw-rw----     1 hede      users            0 Jul 10 07:25 test.txt
```

Auch diese Änderung kann nur von einem Benutzer vorgenommen werden, der über die erforderlichen Rechte verfügt.

5.5 id - Ermitteln der effektiven UIDs und GIDs

Mit dem Befehl "id" kann man ermitteln, unter welcher UID man eingeloggt ist, zu welchen Gruppen man gehört, welche die primäre Gruppe ist usw.:

```
user@linux ~/temp/ $ id

uid=1000(hede) gid=1000(hede)
groups=1000(hede),25(floppy),100(users)
```

5.6 groups - die Gruppenzugehörigkeit ermitteln

Mit dem Befehl groups kann man anzeigen lassen, welchen Gruppen man angehört:

```
user@linux ~/temp/ $ groups

hede floppy users
```

Diese Informationen kann man auch für andere Benutzer anzeigen lassen:

```
user@linux ~/temp/ $ groups root

root : root
```

5.7 Angemeldete Benutzer ermitteln

Der folgende Befehl zeigt an, wer momentan beim System angemeldet ist:

```
user@linux ~/temp/ $ w

07:37:58 up 15 days, 49 min,  5 users, load average: 0.05, 0.08 0.03
 USER   TTY      FROM      LOGIN@   IDLE   JCPU   PCPU   WHAT
 root   tty1     -         25Jun02  6:20   0.82s  0.82s  -bash
 hede   tty2     -         02Jul02  7days  0.30s  0.30s  -bash
 hede   tty3     -         Thu16    5days  0.21s  0.21s  -bash
 hede   pts/0    www2      Mon11    3.00s  3.94s  3.45s  vim t.txt
 hede   pts/1    www2      07:32    0.00s  0.13s  0.06s  w
```

Damit kann man also auch erkennen, wer sich von wo aus angemeldet hat und was die einzelnen Benutzer gerade machen.

Eine verkürzte Information liefert der Befehl who:

```
user@linux ~/temp/ $ who

root     tty1         Jun 25 06:49
hede     tty2         Jul  2 07:47
hede     tty3         Jul  4 16:28
hede     pts/0        Jul  8 11:45 (www2.sentec-elektronik.de)
hede     pts/1        Jul 10 07:32 (www2.sentec-elektronik.de)
```

Um zu ermitteln, unter welcher ID man selbst angemeldet ist, gibt man den Befehl whoami ein. Dieser zeigt aktuelle uid an:

```
user@linux ~/temp/ $ hede@www:~/temp$ whoami

hede
```

5.8 Informationen über laufende Prozesse, offene Dateien etc. ermitteln

Mit `"lsof"` (**"list open files"**) kann man herausfinden, welche Dateien im Moment geöffnet sind, und wer sie geöffnet hat.

`"ps aux"` zeigt die laufenden Prozesse sowie zusätzliche Informationen über Eigentümer, Ressourcenverbrauch usw. an.

5.9 Benutzerverwaltung

Die folgenden Befehle dienen der Benutzerverwaltung, also dem Anlegen, Ändern und Löschen von Benutzern und Gruppen:

`useradd`	Neuen Benutzer anlegen
`usermod`	Existierenden Benutzer ändern
`groupadd`	Gruppe anlegen
`groupmod`	Existierende Gruppe ändern

Diese Befehle werden in einem eigenen Kapitel ausführlich behandelt.

Die einzelnen Distributionen bieten darüber hinaus noch weitere Möglichkeiten, diese Verwaltungsaufgaben zu erledigen. So stellt z. B. Debian die Tools `adduser`, `addgroup`, `deluser`, `delgroup` bereit. Unter SuSE Linux kann man diese Aufgaben auch bequem mit `yast` erledigen.

6 Hinweise

Hier folgen noch ein paar Hinweise zur Arbeit mit dem Benutzerkonzept:

* Man sollte grundsätzlich nur dann als root arbeiten, wenn es unbedingt notwendig ist.
  Dadurch wird gewährleistet, dass man nicht aus Versehen wichtige Dateien löscht. Alle Prozesse (also z.B. auch Mail-Tools), die unter dem root-Account laufen, können uneingeschränkt auf das ganze System zugreifen. Tritt dabei ein Fehler auf oder enthält das Programm z. B. potenziell schädlichen Code, so ist das gesamte System gefährdet!
* Alle Dienste und Server sollten mit möglichst wenigen Rechten betrieben werden.
  Dieses Verhalten ist bei den meisten Systemen als Standard voreingestellt.
* Als Benutzer sollte man sich gut überlegen, welche Zugriffsrechte man Dateien und Verzeichnissen gibt.
  Mit einer behutsamen Rechtevergabe schützt man nicht nur die eigene Privatssphäre, sondern auch die Sicherheit des Systems.