3.7 Grundkonfiguration/Konfiguration von sudo

3.7.1 sudo

Mit dem Programm `sudo` kann man einzelne Programme unter fremden Rechten ausführen:

Terminal
```
schueler@debian964:~$ sudo -u willi xeyes
```

Hier wird das Programm `xeyes` mit den Rechten des Benutzers `willi` ausgeführt. Eventuell wird man nach seinem Passwort gefragt.

Terminal
```
schueler@debian964:~$ sudo -g ifs1a xeyes
```

Hier wird das gleiche Programm mit den Rechten der Gruppe `ifs1a` ausgeführt.

Terminal
```
schueler@debian964:~$ sudo bash
```

Und hiermit bekommt man eine Root-Shell – wenn man das darf. Mit der Option `-l` findet man heraus, was man darf:

Terminal
```
schueler@debian964:~$ sudo -l
[sudo] password for schueler:
User schueler may run the following commands on this host:
    (root) /sbin/ifconfig
```

Natürlich muss diese Möglichkeit sehr stark und sicher eingeschränkt werden können. Dazu dient die Konfiguration des Programmes in der Datei `/etc/sudoers`. Aus Sicherheitsgründen ist das, was in dieser Datei stehen darf, eindeutig durch eine so genannte *Grammatik* festgelegt[1] – wie bei einer (einfachen) Programmiersprache. Enthält die Datei Unsinn, wird `sudo` einfach nicht ausgeführt. Daher muss man die Konfiguration von `sudo` immer mit dem Programm `visudo` vornehmen. Es macht nichts weiter, als einen Editor auszurufen und damit die Datei `/etc/sudoers` zu öffnen[2]. Wenn man den Editor verlassen will, prüft `visudo` die neue Datei. Ist sie fehlerhaft (passiert am Anfang häufig), kann man sie immer wieder verbessern, bis die Prüfung ein OK gibt.

3.7.2 /etc/sudoers

Das Format dieser Datei ist nicht schwierig zu verstehen; die Manual-Seite jedoch ist zutiefst abschreckend: Die Grammatik wird in Backus-Naur-Form angegeben. Hier ein Beispiel für einen Eintrag in `/etc/sudoers`:

```
1 werner host1=(root) /sbin/ifconfig
```

Das heißt: Der Nutzer `werner` darf (nur vom Rechner `host1` aus) das Programm `ifconfig` als Nutzer `root` ausführen. Er muss dazu sein eigenes (`werners`) Passwort eingeben. Am Anfang hat man evtl. Probleme mit Hostnamen, weil DNS gefragt ist. Dann schreibt man besser:

```
1 werner ALL=(root) /sbin/ifconfig
```

Der Benutzer `werner` darf dann von *irgendeinem* Rechner aus eingeloggt sein.

Durch Listen kann man mehrere Nutzernamen, Befehle usw. zusammenfassen:

a) Namensliste

```
1 werner,ida host1=(root) /sbin/ifconfig
```

[1] Ebenso ist festgelegt, wie der Inhalt interpretiert werden muss. Das ist die zugehörige `Semantik`.

[2] Welcher Editor geöffnet wird, steht in der Umgebungsvariablen `EDITOR`. Es muss nicht unbedingt `vi` sein.

b) Mehrere Host-Befehls-Kombinationen

```
1 werner host1=(root)/sbin/ifconfig : host2=(fritz)/usr/bin/xeyes
```

c) Hostliste

```
1 werner host1,host2=(root)/sbin/ifconfig
```

d) Userliste

```
1 werner host1=(ernie,bert)/sbin/ifconfig
```

e) Befehlsliste

```
1 werner host1=(root)/sbin/ifconfig, (root)/sbin/route
```

f) alles zusammen

```
1 werner,ida host1,host2=(ernie,bert)/sbin/ifconfig,\
2                        (ernie,bert)/sbin/route\
3           :host3,host4=(ratz,ruebe)/bin/ip,\
4                        (ratz,ruebe)/bin/tc
```

Mit einzelnen Tags (zwischen runden Klammern und dem Befehlsnamen) kann man Optionen eingeben. Mögliche Optionen sind:

```
NOPASSWD:, PASSWD:, NOEXEC:, EXEC:, SETENV:, NOSETENV:,
LOG_INPUT:, NOLOG_INPUT:, LOG_OUTPUT:, NOLOG_OUTPUT:
```

Im folgenden Beispiel kann `werner` den Editor `vi` *nicht* zu einer Root-Shell hin verlassen[3].

```
1 werner host1=(root)NOEXEC:/usr/bin/vi
```

[3]Achtung, diese Option ist bei manchen Betriebssystemen nicht aktiv. Bei Linux ist sie aber aktiviert.