6.2 Server/SSH

6.2.1 Was ist ssh?

Mit dem Protokoll SSH[1] kann man eine Shell auf einem entfernten System ausführen, dort Befehle ausführen und Dateien transportieren. Standardmäßig benutzen SSH-Dienste den Port 22, alternativ oft 2222.

6.2.2 Der SSH-Client

Der SSH-Client ist unter dem Namen `ssh` verfügbar. Es gibt auch graphische SSH-Programme wie `asbru`, `easyssh` und `putty`. Die Syntax lautet: `ssh uwe@host`.

Beim ersten Öffnen der Verbindung mit einem bestimmten Server wird der Fingerabdruck des Servers angezeigt und gefragt, ob dieser Server vertrauenswürdig ist. Falls man dies bejaht, wird er in spezieller Form an die Datei `.ssh/known_hosts` angefügt:

Terminal

```
schueler@debian964:~$ ssh uwe@pc123
The authenticity of host 'pc123 (1.2.3.4)' can't be
established.
ECDSA key fingerprint is SHA256:Fv1Z......
Are you sure you want to continue connecting (yes/no/[fingerprint])?
yes
Warning: Permanently added 'pc123,1.2.3.4' (ECDSA) to the
list of known hosts.
uwe@pc123's password: ...
...
```

Bei allen weiteren Verbindungen mit diese Server entfällt diese Frage; der Fingerabdruck wird mit den Einträgen in `.ssh/known_hosts` verglichen. Beim Zusammenpassen von Servername und Name im Eintrag wird nachgesehen, ob die Fingerabdrücke übereinstimmen; falls ja, geht es ohne Unterbrechung weiter; falls nein, gibt es eine Warnmeldung.

Der SSH-Client erlaubt es, einen Befehl abzusetzen und anschließend die Verbindung zu schließen; der Befehl wird einfach als zweiter Parameter eingegeben:

Terminal

```
schueler@debian964:~$ ssh uwe@pc123 touch x
```

Hier wurde auf dem entfernten Rechner die Datei `x` angelegt. Der SSH-Client erlaubt viele Optionen. Hier eine kleine Auswahl:

- `-X` – Durchleiten der Verbindung zur GUI
- `-C` – Daten komprimiert übertragen
- `-p 1234` – Port 1234 verwenden

6.2.3 Dateiübertragung per SSH

Der Befehl `scp` erlaubt es, per SSH Dateien verschlüsselt zu übertragen und ist damit eine Alternative zu `FTPS` und `SFTP`. Der Aufruf von `scp` ist vergleichbar mit `cp`, nur, dass man zusätzlich einen oder zwei Rechnernamen angeben kann:

Terminal

```
schueler@debian964:~$ scp -P 1234 uwe@pc123/home/uwe/.bashrc .bashrc
```

Die Option `-P` für die Portnummer muss hier übrigens groß geschrieben werden.

[1] SSH ist der Nachfolger von RLOGIN und RSH, bei denen man sich ohne Authentifizierung auf entfernten Rechnern einloggen konnte. Die entfernte UID und die lokale waren dabei gleich. Später gab es TELNET, bei denen man sich mit Loginnamen und Passwort einloggen musste. Dort wurden der Loginname und das Passwort allerdings im Klartext übertragen.

6.2.4 Der SSH-Server `sshd`

Der SSH-Server hat den Namen `sshd`. Er wird in der Datei `/etc/ssh/sshd_config` konfiguriert. In diesem Beispiel soll er auf den Ports 1234 und 22 hören; die Konfigurationsdatei entält dann unter anderem die Zeilen:

```
1 ...
2 Port 22
3 Port 1234
4 ...
```

Weitere Information enthält die Manualpage zu `sshd_config`.

6.2.5 Schlüssel-Übertragung für SSH

Wenn man sich auf einem entfernten System ohne Eingabe eines Passwortes einloggen möchte, bietet sich die Authentifizierung mittels öffentlichem Schlüssel an.

Zur Erzeugung von Schlüsseln dient das Programm `ssh-keygen`. Mit dem Aufruf wird ein Schlüsselpaar erlaubt:

Terminal

```
schueler@debian964:~$ ssh-keygen -b8192
Generating public/private rsa key pair.
Enter file in which to save the key (/home/uwe/.ssh/id_rsa):
Created directory '/home/uwe/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/uwe/.ssh/id_rsa.
Your public key has been saved in /home/uwe/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:mcFm ..... uwe@pc123
+---[RSA 8192]----+
|     ...  . oo   |
+----[SHA256]-----+
```

Der Aufruf erzeugte zwei Dateien: `id_rsa` (privater Schlüssel) und `id_rsa.pub` (öffentlicher Schlüssel) im Verzeichnis `.ssh`.

Den privaten Schlüssel behält man bei sich. Der öffentliche Schlüssel wird nun auf das entfernte System kopiert und dort an die Datei `~/.ssh/authorized_keys` angehängt:

Terminal

```
schueler@debian964:~$ scp -P567 .ssh/id_rsa.pub uwe@pc123:/home/uwe/
schueler@debian964:~$ ssh -p567 uwe@pc123 'cat id_rsa.pub \
   >> .ssh/authorized_keys'
```

Alternativ kann man auch den Befehl `ssh-copy-id` verwenden:

Terminal

```
schueler@debian964:~$ ssh-copy-id -i .ssh/id_rsa.pub uwe@pc123
```

Die Option `-i` heißt hierbei: Nimm die folgende ID-Datei. Ab jetzt kann man sich als `uwe` einloggen auf `pc123`:

Terminal

```
schueler@debian964:~$ ssh -p 567 uwe@pc123
```

Wenn man will, kann man nun die Passwort-Authentifizierung am Server ausschalten, indem man in der Datei `/etc/ssh/sshd_config` in der folgenden Zeile `yes` auf `no` ändert:

```
1 PasswordAuthentication yes
```