

SSH einrichten

Über SSH ist es möglich eine sichere Verbindung zu einem entfernten System aufzubauen, um Befehle abzusetzen oder Daten auszutauschen (SFTP). Zugriff auf einen entfernten Server (<REMOTEHOST>) ist i.d.R. eingeschränkt auf den root und die user des Servers. Über die Datei **/etc/ssh/sshd_config** läßt sich der Zugang regeln.

Empfehlenswert:

- Direkter Login von root untersagen
- Zugang nur mit Schlüssel - login ohne Schlüssel (nur Passwort) unterbinden
- Optional: Eingeschränkter Zugriff über SFTP auf eingeschränkte Bereiche

Hinweise: <https://wiki.ubuntuusers.de/SSH/>

SSH-Schlüssel

Für den Zugriff mit Schlüssel müssen diese zunächst generiert werden.

SSH-Key unter Linux generieren

Auf einem lokalen System (Hier Debian 11) Programm open-ssh-client installieren und Schlüssel generieren.

```
$ sudo apt install openssh-client
```

```
$ ssh-keygen -t rsa -b 4096
```

t = Typ (hier rsa)

b = Schlüssellänge (hier 4096 Bit)

Schlüssel automatisch im (versteckten) Verzeichnis ~/.ssh/

Password nur freilassen, wenn Zugriff auf <REMOTEHOST> ausschließlich mit Schlüssel möglich.

Der private Schlüssel muss unbedingt vor fremdem Zugriff geschützt bleiben!

Es werden 2 Dateien angelegt:

- id_rsa (privater Schlüssel)
- id_rsa.pub (öffentlicher Schlüssel)

Der öffentliche Schlüssel wird auf das entfernte System übertragen, auf das zugegriffen werden soll. Der private Schlüssel bleibt auf dem lokal System, auf dem er generiert wurde. Für jedes weitere (lokale) System sollte jeweils ein eigener Schlüssel generiert werden.

Zur Übertragung auf einen Server muss der <USER> dort angelegt sein und der Zugriff ohne Schlüssel (i.d.R. mit Passwort) freigegeben sein.

Den öffentlichen Schlüssel(id_rsa.pub) auf den Server übertragen (kopieren):

```
$ ssh-copy-id <USER>@<REMOTEHOST>
```

Ersetzen: <REMOTEHOST>, <USER>

Das Passwort vom <REMOTEHOST> wird abgefragt.

Im <REMOTEHOST>-Home-Verzeichnis vom <USER> liegt die Datei **~/.ssh/authorized_keys**. In diese Datei werden die gültigen Public-Keys (automatisch) eingetragen. `.ssh` ist versteckt.

Parallel wird auf dem lokalen (Linux-)Rechner der neu zugefügte Host in der Datei **~/.ssh/known_hosts** aufgenommen.

SSH-Key mit Putty generieren

Es ist möglich, mit dem Programm **PuTTYgen** unter Windows oder anderen OS, einen Schlüssel zu erstellen. Dann kopiert man den angezeigten Block direkt aus PuTTYgen heraus in die entfernte **~/.ssh/authorized_keys** des Servers - ggf. die Datei neu erstellen.

Die Datei hat folgende Struktur (alles hintereinander):

- ssh-rsa « dieser Text und 1 Leerzeichen
- rsa-pub-key « der eigentliche Schlüssel aus Puttygen
- Key Kommentar « Im Textblock von puttygen bereits enthalten

Liegt bereits ein SSH-Key vor, dann kann dieser auch für den Zugriff mit Putty oder SFTP umgewandelt werden.

Programm PuTTYgen: Private-key importieren und als PuTTY-private-key speichern. Der kann von diversen Systemen (FileZilla, etc.) genutzt werden, sofern im entfernten Server hinterlegt. Da dieser Schlüssel kopier- und übertragbar ist, sollte er immer zusätzlich mit einem Passwort geschützt sein.

Für Konvertierung Programm **PuTTYgen** aufrufen.

1. Load an existing private key file
2. Save private key » jetzt mit Endung `.ppk`

Login

Login über Linux-Shell

Login mit Passwort oder wenn der key im <REMOTEHOST> hinterlegt ist:

```
$ ssh <USER>@<REMOTEHOST>
```

<USER>@ kann weggelassen werden, wenn <USER> mit dem lokalen User übereinstimmt.

Beim ersten Login, wenn der public-key noch nicht auf dem Server ist oder dieser sich geändert hat.

```
$ ssh -i ~/.ssh/id_rsa <USER>@<REMOTEHOST>
```

- `~/.ssh/id_rsa` = (relativer) Pfad und Name privater Schlüssel
- `<USER>@<REMOTEHOST>` = User und IP# des entfernten Systems

Beim ersten Login erfolgt eine Validierung mit dem Passwort des Systems. Bei Folgeaufrufen nur noch mit dem PW des SSH-Keys bzw. wenn keines vergeben wurde, ohne PW.

Beim ersten Login werden die dann bekannten Hosts lokal in `~/.ssh/known_hosts` gespeichert. Gibt es Änderungen an einem Host und ggf. damit verbundene Probleme, dann kann der Host daraus, oder die ganze Datei, gelöscht werden. Wird dann beim nächsten Aufruf neu generiert.

Wurde der Schlüssel am Server geändert, oder der Server neu eingerichtet, muss er aus der lokalen Datei `~/.ssh/known_hosts` ausgetragen werden. Händisch oder mit dem Befehl (IP des betroffenen Servers):

```
$ ssh-keygen -f "~/<USER>/.ssh/known_hosts" -R "<REMOTEHOST>"
```

Login mit PuTTY

Mit PuTTY die Verbindung wie folgt definieren:

- <REMOTEHOST>
- Port (weglassen, wenn 22 - Normalfall)
- SSH
- Name (Saved Session)
- /Connection/SSH/Auth/ » Laden: Private Key File¹
- Option: /Connection/Data/ » Auto-Login username = <USER>
- Option: /Connection/ » Secons between keepalives = 600 (verhindert das auto-lockout)
- Zurück auf „Session“ und **Save**

Login mit FileZilla

Für den Zugriff kann die mit Puttygen generierte .ppk-Datei genutzt werden.
Verbindungsart: **Schlüsseldatei**.

SSH-Zugriff am Server konfigurieren

```
$ nano /etc/ssh/sshd_config
```

```
PermitRootLogin no           #root-login von Außen verbieten
PasswordAuthentication no    #Login nur mit key-file (außer
s.u.)
Subsystem sftp internal-sftp
Match group ftpuser          #Sonderregelung für Gruppe
"ftpuser"
    X11Forwarding no
    AllowTcpForwarding no
    ChrootDirectory /var/www/ #FTP beschränkt auf dieses
Verzeichnis
    ForceCommand internal-sftp
    PasswordAuthentication yes #Passwort-Login für diese Gruppe
```

ok

Dateien kopieren über SSH

Kopieren der Datei "foobar.txt" von einem entfernten Rechner auf den lokalen Rechner.

```
$ scp <USER>@<REMOTEHOST>:foobar.txt /some/local/directory
```

Kopieren der Datei "foobar.txt" vom lokalen Rechner auf einen entfernten Rechner.

```
$ scp foobar.txt <USER>@<REMOTEHOST>:/some/remote/directory
```

Kopieren der Datei "foobar.txt" vom Remote-Host "<REMOTEHOST_1>," auf den Remote-Host "<REMOTEHOST_2>".

```
$ scp <USER>@<REMOTEHOST_1>.edu:/some/remote/directory/foobar.txt \
<USER>@<REMOTEHOST_2>:/some/remote/directory/
```

Einzelne Verzeichnisse kopieren Kopieren des Verzeichnisses "foo" vom lokalen Rechner in das Verzeichnis "bar" eines entfernten Rechners.

```
$ scp -r foo <USER>@<REMOTEHOST>.edu:/some/remote/directory/bar
```

Quelle: <https://www.davidkehr.com/linux-kopieren-von-und-zu-einem-computer-per-scp-ssh/>

¹⁾

Mit Puttygen generierte .ppk-Datei

From:

<https://wiki.bluegnu.de/> - **wiki**

Permanent link:

https://wiki.bluegnu.de/doku.php/open:it_ssh?rev=1669732110

Last update: **2024/06/22 10:15**

