OpenSSH – SSH, SFTP und Schlüssel

Installation (Debian)

Auf dem Client werden die OpenSSH-Programme benötigt:

```
sudo apt update
```

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

```
ssh -V
```

Ein System, das eingehende SSH-Verbindungen anbieten soll, benötigt zusätzlich einen SSH-Server:

```
sudo apt install openssh-server
```

```
sudo systemctl status ssh
```

Anmeldung und Abmeldung

```
ssh -i ~/.ssh/id_ed25519 <USER>@<REMOTEHOST>
```

Sollte nur ein Schlüssel im Verzeichnis ~/.ssh/ liegen und dieser den Konventionen der Dateibezeichnung für Schlüssel entsprechen, kann die Definition entfallen.
Ist der Benutzername von <USER> auf lokalem System und externem System identisch, kann auch die Angabe des Namens entfallen.

```
ssh <REMOTEHOST>
```

Zum Beenden der interaktiven Sitzung:

```
exit
```

Für SFTP gilt entsprechend:

```
sftp -i ~/.ssh/id_ed25519 <USER>@<REMOTEHOST>
```

In der SFTP-Eingabeaufforderung beendet bye die Sitzung. SSH entschlüsselt die Transportdaten automatisch; ein separater Entschlüsselungsbefehl ist nicht erforderlich.

Aufgaben und Schlüsselarten

OpenSSH schützt Remote-Sitzungen und Dateiübertragungen. Zu unterscheiden sind

Benutzerschlüssel zur Anmeldung und **Hostschlüssel** zur Erkennung des Servers. Die eigentlichen Verbindungsdaten werden mit ausgehandelten symmetrischen Sitzungsschlüsseln geschützt.

Benutzerschlüssel erzeugen

```
ssh-keygen -t ed25519 -a 100 -f ~/.ssh/id_ed25519 -C "beispiel-client"
```

Parameter	Bedeutung
-t ed25519	Erzeugt einen Ed25519-Schlüssel
-a 100	Verwendet 100 KDF-Runden zum Schutz des privaten Schlüssels durch die Passphrase
-f ~/.ssh/id_ed25519	Legt den Dateinamen fest
-C „beispiel-client“	Hinterlegt einen Kommentar zur Identifikation

Es entstehen ein privater Schlüssel und die öffentliche Datei mit Endung `.pub`. Für interaktive Schlüssel ist eine Passphrase sinnvoll.

Der öffentliche Schlüssel hat die folgende, durch Leerzeichen getrennte, Struktur:

- ssh-xxx = Schlüsseltyp (ssh-rsa, ssh-ed25519, ...)
- pub-key = Schlüsselinhalt
- Kommentar = Beschreibung

SSH-Key mit Putty generieren

Alternativ ist es möglich, mit dem Programm PuTTYgen, z.B. unter Windows, einen Schlüssel zu erstellen. Es ist möglich, den angezeigten Block direkt aus PuTTYgen heraus in die entfernte **~/.ssh/authorized_keys** des Servers zu kopieren - ggf. die Datei neu erstellen.

Liegt bereits ein SSH-Key vor (z.B. erstellt wie oben beschrieben), kann dieser auch für den Zugriff mit Putty oder SFTP umgewandelt werden.
Programm PuTTYgen: Private-key importieren und als PuTTY-private-key speichern.

Schlüssel von PuTTY können von diversen Systemen (FileZilla, etc.) genutzt werden, sofern der Public-Key im entfernten Server hinterlegt ist. 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

Öffentlichen Schlüssel übertragen

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub <USER>@<REMOTEHOST>
```

Das Zielsystem muss den SSH-Dienst anbieten. Der Befehl hinterlegt den öffentlichen Schlüssel

normalerweise in `~/.ssh/authorized_keys` des Zielbenutzers.

Verbindung und Dateiübertragung

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

```
sftp <USER>@<REMOTEHOST>
```

```
scp beispiel.txt <USER>@<REMOTEHOST>:/tmp/
```

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

Fingerprints und Hostschlüssel

```
ssh-keygen -lf ~/.ssh/id_ed25519.pub
```

```
ssh-keygen -F <REMOTEHOST>
```

Bei einer erstmaligen Verbindung muss der Host-Fingerprint über einen vertrauenswürdigen Weg geprüft werden. Unerwartete Änderungen können auf einen Serverwechsel oder einen Angriff hinweisen.

SSH-Agent

Ein SSH-Agent kann entsperrte private Schlüssel für eine Sitzung bereitstellen:

```
ssh-add ~/.ssh/id_ed25519
```

```
ssh-add -l
```

Der Agent ersetzt keine sorgfältige Zugriffskontrolle. Agent-Forwarding nur verwenden, wenn es erforderlich ist.

Berechtigungen

```
chmod 700 ~/.ssh
```

```
chmod 600 ~/.ssh/id_ed25519
```

```
chmod 644 ~/.ssh/id_ed25519.pub
```

Die privaten Schlüssel gehören nicht in öffentliche Repositories oder ungeschützte Backups. Für Geräte und automatisierte Aufgaben sind getrennte Schlüssel zweckmäßig.

Schlüsselwechsel

Zuerst den neuen öffentlichen Schlüssel auf dem Zielserver hinterlegen und den Zugang testen. Erst danach den alten Eintrag aus `authorized_keys` entfernen. Bei kompromittierten Schlüsseln außerdem Logdateien und weitere Berechtigungen prüfen.

Verbindungs-Protokolle

Quelle: https://www.strongdm.com/blog/view-ssh-logs

If you want to view ssh logs from a specific time range, you can use the since and until flags. Some examples:

```
sudo journalctl -u ssh --since yesterday
```

```
sudo journalctl -u ssh --since -3d --until -2d # logs from three days ago
```

```
sudo journalctl -u ssh --since -1h # logs from the last hour
```

```
sudo journalctl -u ssh --until "2024-12-20 07:00:00"
```

To watch the ssh logs in realtime, use the follow flag:

```
sudo journalctl -fu ssh
```

Use Ctrl-C to exit out of the log monitor.

TMUX - lang laufende Befehle

Mit tmux auf dem Server eine Terminalsitzung starten, die unabhängig von einer SSH-Verbindung weiterläuft.
Wenn die SSH-Verbindung abbricht oder die lokale Konsole geschlossen wird, bleibt die Sitzung erhalten. Später erneut verbinden und den Fortschritt verfolgen.

Einmalig auf dem Server installieren:

```
sudo apt install tmux
```

Sitzung starten:

```
tmux new -s migration
```

tmux	Startet den Terminal-Multiplexer
new	Erstellt eine neue Sitzung (new session)
-s	Legt den Namen der Sitzung fest (session name)

migration	Frei wählbarer Sitzungsname
tmux ls	Alle vorhandenen Sitzungen anzeigen
tmux attach -t migration	Sitzung migration wieder öffnen
tmux kill-session -t migration	Sitzung beenden
exit	Aktuelle Sitzung beenden, sofern keine weiteren Fenster offen sind

Mehrere unabhängige Sitzungen sind parallel möglich.

Anschließend Befehle/Scripte über Konsole starten.
- - -
Falls die SSH-Verbindung abbricht, erneut anmelden und eingeben:

```
tmux attach -t migration
```

Die laufende Sitzung einschließlich ihrer bisherigen Terminalausgabe wird angezeigt.

Bewusst von einer Sitzung lösen, ohne den laufenden Befehl zu beenden:

Strg + B danach D

Der Befehl läuft weiter.

Besonderheit bei TMUX

Scrollen mit der Maus aktivieren!
Für die dauerhafte Nutzung die Mausunterstützung aktivieren.

Innerhalb von tmux: - Strg+B drücken und loslassen. - : drücken. - Folgenden Befehl eingeben:

```
set -g mouse on
```

Mit Enter bestätigen.

Danach sollte das Scrollen mit dem Mausrad direkt funktionieren.

Damit die Einstellung auch bei zukünftigen Sitzungen gilt

```
sudo nano ~/.tmux.conf
```

```
echo 'set -g mouse on' >> ~/.tmux.conf
```

Zurück zur Übersicht

From:
https://wiki.bluegnu.de/ - **wiki**

Permanent link:
https://wiki.bluegnu.de/doku.php/open:it:secure:ssh

Last update: **2026/11/15 21:31**

