OpenSSH

OpenSSH (auch bekannt als OpenBSD Secure Shell) ist eine Sammlung von sicheren Netzwerkprogrammen, die auf dem Secure Shell (SSH)-Protokoll basieren, das einen sicheren Kanal über ein ungesichertes Netzwerk in einer Client-Server-Architektur bereitstellt.

OpenSSH begann als ein Fork des freien SSH-Programms, das von Tatu Ylönen entwickelt wurde; spätere Versionen von Ylönens SSH waren proprietäre Software, die von SSH Communications Security angeboten wurde. OpenSSH wurde erstmals 1999 veröffentlicht und wird derzeit als Teil des OpenBSD-Betriebssystems entwickelt.

Bei OpenSSH handelt es sich nicht um ein einzelnes Computerprogramm, sondern vielmehr um eine Reihe von Programmen, die als Alternative zu unverschlüsselten Protokollen wie Telnet und FTP dienen. OpenSSH ist in mehrere Betriebssysteme integriert, nämlich Microsoft Windows, macOS und die meisten Linux-Betriebssysteme, während die portable Version als Paket in anderen Systemen verfügbar ist.

Paket

```
pacman -S openssh
```

Start/Neustart

```
systemctl enable --now sshd.service
```

Denk dran, dass jede Änderung an `/etc/ssh/sshd_config` einen Neustart des Dienstes erfordert.

```
systemctl restart sshd.service
```

Root- und Passwort-Authentifizierung zulassen

Wenn man einen schnellen Zugang benötigt, zum Beispiel um den Server einzurichten.

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

```
Port 22
PermitRootLogin yes
PasswordAuthentication yes
```

SSH Schlüssel

Dies ist nicht nur sicherer, sondern erleichtert auch die Verbindung zum Server, ohne dass man jedes

mal das Passwort eingeben muss.

Konfiguration - Server

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

```
Port 22
HostKey /etc/ssh/ssh_host_ed25519_key
PermitRootLogin no
PubkeyAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
```

Schlüssel erstellen - Desktop

Ob man eine Passphrase verwendet oder nicht, hängt davon ab, wie man die Infrastruktur nutzen will und ob eine zusätzliche Sicherheitsebene gewünscht/benötigt wird. Zum Beispiel kann keepassxc Passphrasen verarbeiten und den Schlüssel dem SSH-Agenten für automatische Verbindungen hinzufügen.

```
ssh-keygen -t ed25519
```

Kopiere den Inhalt von `id_ed25519.pub` für den Server.

```
cat ~/.ssh/id_ed25519.pub
```

Host hinzufügen - Desktop

Ändere `$USER` und `$SERVERIP`.

```
nano .ssh/config
```

```
Host server
    HostName $SERVERIP
    Port 22
    User $USER
    IdentitiesOnly yes
    IdentityFile "~/.ssh/id_ed25519"
```

Pub-Schlüssel hinzufügen - Server

Füge den Inhalt von `id_ed25519.pub` zu `authorized_keys` hinzu.

```
cd
mkdir .ssh
```

```
chmod 700 .ssh
touch .ssh/authorized_keys
chmod 600 .ssh/authorized_keys
nano .ssh/authorized_keys
```

SSH-Agent - Desktop

```
mkdir -p ~/.config/systemd/user/
nano ~/.config/systemd/user/ssh-agent.service
```

```
[Unit]
Description=SSH key agent

[Service]
Type=simple
Environment=SSH_AUTH_SOCK=%t/ssh-agent.socket
ExecStart=/usr/bin/ssh-agent -D -a $SSH_AUTH_SOCK

[Install]
WantedBy=default.target
```

```
systemctl --user enable ~/.config/systemd/user/ssh-agent.service
systemctl --user start ssh-agent.service
```

Ein Neustart kann notwendig sein, wenn keepassxc eine Fehlermeldung wie “No agent running, cannot add identity” ausgibt.