zurück

SSH-Zugang über Bastion Host unter Windows ohne Passwort

Dieser Eintrag beschreibt, wie unter Windows ein SSH-Zugang über einen Bastion-Host (Jump Host) eingerichtet wird, sodass Verbindungen zu internen Systemen **ohne Passwort**, ausschließlich über **SSH-Key-Authentifizierung**, möglich sind.

Die Anleitung basiert auf folgenden Zielen:

- Anmeldung am Bastion Host **ohne Passwort**
- Weiterleitung zu internen Servern automatisch über `ProxyJump`
- Nutzung einer sauberen `~/.ssh/config` für komfortable Befehle wie

```
ssh bastion
ssh server1
ssh pihole
```

---

1. Voraussetzungen

- Windows 10/11 mit installiertem **OpenSSH-Client**
- Ein erzeugtes SSH-Schlüsselpaar (z. B. ED25519)
- Zugriff auf den Bastion Host (öffentliche IP + SSH-Port)
- Interne Systeme sind vom Bastion aus erreichbar

---

2. SSH-Key unter Windows erzeugen

Falls noch kein Key vorhanden ist:

```
ssh-keygen -t ed25519 -C "windows-client"
```

Die Dateien werden automatisch erstellt unter:

```
C:\Users\<Benutzer>\.ssh\id_ed25519
C:\Users\<Benutzer>\.ssh\id_ed25519.pub
```

Die **Passphrase kann leer bleiben**, wenn keine Eingabe gewünscht ist.

3. Public Key auf dem Bastion Host hinterlegen

Der öffentliche Schlüssel muss auf dem Bastion in:

```
~/.ssh/authorized_keys
```

hinterlegt werden.

Beispielübertragung aus PowerShell:

```
type $env:USERPROFILE\.ssh\id_ed25519.pub | ssh benutzer@bastion.fqdn -p <PORT> "mkdir -p ~/.ssh && cat >> ~/.ssh/authorized_keys"
```

Wichtig:

- Die Datei `authorized_keys` darf **nur dem Benutzer selbst gehören**
- Rechte setzen:

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

4. Verbindung zum Bastion Host testen

```
ssh bastion
```

Erwartetes Verhalten:

- Keine Passwortabfrage
- Anmeldung über Public-Key
- Stabiler SSH-Login

Falls der Key nicht akzeptiert wird: prüfen, ob der **richtige Public Key** auf dem Bastion eingetragen wurde.

5. Konfiguration der Datei `~/.ssh/config` unter Windows

Unter Windows befindet sich die Datei hier:

```
C:\Users\<Benutzer>\.ssh\config
```

Beispielkonfiguration:

snippet.sshconfig

```
# Bastion Host
Host bastion
    HostName bastion.example.com
    Port 58222
    User jumpuser
    IdentityFile C:/Users/<Benutzer>/.ssh/id_ed25519
    ServerAliveInterval 30
    ServerAliveCountMax 3

# Interne Hosts werden automatisch über Bastion geroutet
Host server1
    HostName 192.168.100.10
    User admin
    ProxyJump bastion
    IdentityFile C:/Users/<Benutzer>/.ssh/id_ed25519

Host pihole
    HostName 192.168.100.20
    User admin
    ProxyJump bastion
    IdentityFile C:/Users/<Benutzer>/.ssh/id_ed25519
```

Hinweise:

- Windows verwendet **Schrägstriche** `/` in Pfaden
- Kein `ControlMaster` unter Windows verwenden (nicht kompatibel)

---

6. Verbindung zu internen Systemen nutzen

Durch die Config reichen nun einfache Befehle:

```
ssh server1
ssh pihole
```

Ablauf:

1. Windows verbindet automatisch zum Bastion Host
2. Authentifizierung erfolgt per SSH-Key
3. Der Zielserver wird über einen Tunnel erreicht
4. Es erfolgt keine Passwortabfrage

7. Fehlersuche (Troubleshooting)

Problem	Ursache	Lösung
——————————————-	————————————–	——————————————-
Permission denied (publickey)	Public Key fehlt oder falscher User	Key erneut prüfen, in `authorized_keys` eintragen
Verbindung nutzt Port 22 statt Bastion-Port	Host-Name stimmt nicht überein	In `Host`-Block identischen Namen verwenden
„Not a socket“ / ProxyJump bricht ab	Fehlerhafte config oder Troll-Proxy	Config minimieren, Multiplexing entfernen
Passwort wird verlangt	Passwortauth aktiviert	Auf Bastion: `PasswordAuthentication no` setzen

8. Sicherheitshinweise

- Jeder Client sollte **einen eigenen SSH-Key** besitzen
- Alte Keys regelmäßig aus `authorized_keys` entfernen
- Datei-Rechte streng setzen (600 bzw. 700)
- Optional: SSH-Zertifikate (CA-basiert) einsetzen, wenn viele Nutzer verwaltet werden

9. Zusammenfassung

Mit der oben beschriebenen Einrichtung ermöglicht Windows eine vollständig passwortlose SSH-Anmeldung über einen Bastion Host. Die Nutzung der `~/.ssh/config` vereinfacht die tägliche Arbeit erheblich und sorgt für sichere, wiederholbare Verbindungen — egal, ob zu extern erreichbaren Systemen oder internen Servern, die ausschließlich über den Bastion Host zugänglich sind.