zurück

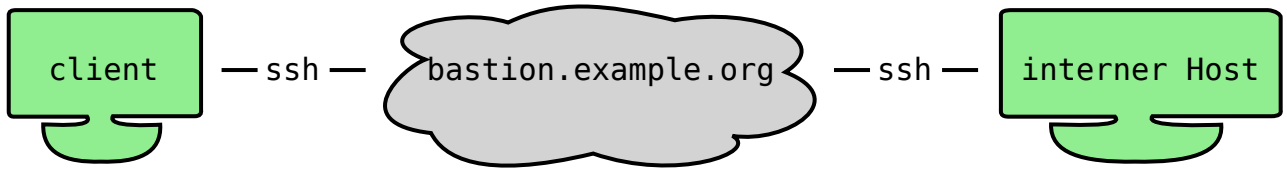
SSH Bastion ProxyJump

Eine Bastion-Host-Lösung erlaubt es, interne Systeme mit privaten IP-Adressen sicher von außen zu erreichen, ohne dass diese direkt im Internet exponiert werden. Der Zugriff erfolgt über einen einzigen öffentlich erreichbaren Server („Bastion“), der als Proxy für alle internen Verbindungen dient.

Architektur

- **Bastion-Host**: öffentlich über DNS/Domain erreichbar (z. B. bastion.example.org)
- **Interne Systeme**: haben private IP-Adressen (z. B. 192.168.x.x)
- **Client**: verbindet sich von außen über den Bastion-Host, ohne VPN notwendig
- Verbindung erfolgt über die SSH-Funktion `ProxyJump` oder per `-J` Parameter

`Diagram Beispielaufbau`



Einrichtung auf dem Bastion-Host

- SSH-Server installiert (z. B. OpenSSH)
- Nur Public-Key-Authentifizierung erlaubt
- TCP-Forwarding aktiviert
- Keine unnötigen Funktionen (z. B. TTY, X11) aktiv

Beispiel ‘’/etc/ssh/sshd_config’’:

```
Port 58222
PasswordAuthentication no
PermitRootLogin no
UseDNS no

AllowTcpForwarding yes
PermitOpen any
GatewayPorts no

PermitTTY no
X11Forwarding no
```

```
ClientAliveInterval 300
ClientAliveCountMax 2
```

Nach Änderungen:

```
systemctl restart ssh
```

Einrichtung auf dem Client

Datei: ''~/.ssh/config''

```
Host bastion
    HostName bastion.example.org
    Port 58222
    User proxyuser
    IdentityFile ~/.ssh/id_ed25519
    ServerAliveInterval 30
    TCPKeepAlive yes
    ControlMaster auto
    ControlPath ~/.ssh/cm-%r@%h:%p
    ControlPersist 10m

# Beispiel für internen Host via Bastion
Host server-intern
    HostName 192.168.100.10
    User admin
    ProxyJump bastion
    IdentityFile ~/.ssh/id_ed25519
```

Aufruf:

```
ssh server-intern
```

Alternative: Temporärer Tunnel

Falls ProxyJump nicht möglich ist:

```
ssh -i ~/.ssh/id_ed25519 -p 58222 -L 2222:192.168.100.10:22
proxyuser@bastion.example.org
ssh -p 2222 admin@localhost
```

Erweiterung: Alle internen Hosts automatisch via Bastion

Wildcard in ‘’~/.ssh/config’’:

```
Host 192.168.*
    User admin
    ProxyJump bastion
    IdentityFile ~/.ssh/id_ed25519
```

→ alle 192.168er-Hosts gehen automatisch über den Bastion-Host.

Typische Fehlerquellen und Behebung

- **Fehler: “Permission denied (publickey)”**
  - Public-Key fehlt oder falscher Benutzername
  - Lösung: Key korrekt in ‘’~/.ssh/authorized_keys’’ auf Bastion hinterlegen
- **Fehler: “Connection refused”**
  - Bastion lauscht nicht auf Port (z. B. 58222)
  - Lösung: Firewall prüfen, ‘’sshd’’ neu starten
- **Fehler: “channel 0: open failed: administratively prohibited”**
  - TCP-Forwarding auf Bastion nicht erlaubt
  - Lösung: in ‘’sshd_config’’ ‘’AllowTcpForwarding yes’’ setzen
- **Problem: Nach dem Login lande ich auf Bastion, nicht auf dem internen Host**
  - ProxyJump in ‘’~/.ssh/config’’ fehlt
  - Lösung: Host-Eintrag prüfen, ‘’ProxyJump bastion’’ hinzufügen
- **Problem: Verbindung funktioniert nur im LAN / über VPN**
  - Externe DNS oder Portweiterleitung nicht korrekt
  - Lösung: Prüfen, ob ‘’bastion.example.org:58222’’ von außen erreichbar ist

Sicherheitshinweise

- Bastion sollte gehärtet sein (keine Passwörter, keine Root-Logins).
- Fail2Ban oder CrowdSec sinnvoll.
- Logs regelmäßig prüfen.
- Schlüsselverwaltung (evtl. mit SSH-Agent oder Vaultwarden) nutzen.

Fazit

Mit einem SSH-Bastion-Host und ‘‘ProxyJump’’ lassen sich interne Systeme sicher und einfach von außen erreichen, ohne diese direkt ins Internet zu exponieren. Die Konfiguration ist robust, erweiterbar und erspart den Aufbau zusätzlicher VPNs, wenn nur SSH-Zugriff benötigt wird.

Lars Weiß 19.08.2025 14:36