OpenSSH Cheat Sheet

Client-Parameter

SSH-Verbindung zu Server „srv.example.com“ unter **Port 2222** als Benutzer „user“
SSH-Standardport ist 22
`ssh -p 2222 user@srv.example.com`
Betroffene Server-Optionen:
`Listen 2222`,
`AddressFamily` für IPv4/IPv6,
`ListenAddress` kann auch Port beinhalten

Verbindungsaufbau über einen **Jumphost** „jumphost“
mit Benutzer „jumpuser“
auf Zielhost „target“ mit Benutzer „user“
`ssh -J jumpuser@jumphost user@target`

Betroffene Server-Optionen auf Jumphost:
`DisableForwarding no`, `AllowTcpForwarding yes`,
`PermitOpen target:22`, `MaxSessions 0`

Öffnen einer **Local Forwarding** Verbindung, um Anfragen auf den Client-Port 80 über den Server „srv“ nach „www“ auf Port 80 weiterzuleiten
`ssh -L 80:www:80 srv`

Betroffene Server-Optionen auf Jumphost:
`DisableForwarding no`, `AllowTcpForwarding yes`,
`PermitOpen target:22`, `MaxSessions 0`

Öffnen einer **Remote Forwarding** Verbindung, um Anfragen auf den Server-Port 8080 über den Server „srv“ nach „localhost“ auf Port 80 weiterzuleiten
`ssh -R 8080:localhost:80 srv`

Betroffene Server-Optionen auf Jumphost:
`DisableForwarding no`, `AllowTcpForwarding yes`,
`PermitOpen target:22`, `MaxSessions 0`

Prüfen eines SSHFP-DNS-Records mit dem zugehörigen Server-Fingerprint beim Verbindungsaufbau zu Server „srv“
als Benutzer „user“
`ssh -o "VerifyHostKeyDNS yes" user@srv`
Achtung: Verbindung wird bei unzulässigem SSHFP-Record nicht blockiert
SSHFP-Records mittels `ssh-keygen -r hostname` ausgelesen und via DNS publiziert

Server-Konfiguration `/etc/ssh/sshd_config`

Einrichten, dass **Public-Key- und Passwort-Authentisierung zusammen** erfüllt sein müssen
`AuthenticationMethods publickey,password`

Einrichten der **Dateiübertragung** für Benutzer „file“, welcher **nur innerhalb dem Pfad** „/data/sftp“ operieren darf

```
Match User file
    ForceCommand internal-sftp
    ChrootDirectory /data/sftp
```

`Match`-Blöcke am Ende der Konfigurationsdatei anfügen
Pfad muss Benutzer „root“ gehören, Unterordner darf „file“ gehören („file“ kann somit nichts direkt im Pfad schreiben)

Nur Benutzer der Gruppe „sshaccess“ den SSH-Zugriff erlauben
`AllowGroups sshaccess`
Hierbei geht es um den Zielbenutzer auf dem Server

Ausführung des Befehls „echo hello“ für Benutzer mit Kommandozeilenzugriff **forcieren**
`ForceCommand echo hello`
SSH-Sitzung wird nach Ausführung des Befehls geschlossen
Alternativ kann z.B. ein Skript mit einer Auswahl an Befehlen angegeben werden

Authentisierungs-Agent

Agent in der aktuellen Shell-Sitzung **starten**
`eval $(ssh-agent -s)`
Eventuell könnte dieser bereits gestartet sein, prüfen mit z.B. `pgrep -l ssh-agent`

Schlüssel „~/.ssh/id_ed25519“ **zum Agent hinzufügen mit der Bedingung**, dass dieser nur für die Verbindungen „alice@srv1“ und bei aktivem Agent-Forwarding von „srv1“ nach „bob@srv2“ verwendet werden darf

```
ssh-add -h 'alice@srv1' \
 -h 'srv1>bob@srv2' ~/.ssh/id_ed25519
```

Die Server-Namen müssen hierzu bereits in der Datei `~/.ssh/known_hosts` vorhanden sein
Betroffene Server-Optionen:
`AllowAgentForwarding yes`
Agent-Forwarding mit SSH-Client-Option `-A` aktivieren

OpenSSH Cheat Sheet

Es folgt eine Ausgabe der **SSH-Server-Konfigurationsdatei**, wie sie auf den Servern dieser Arbeit implementiert wurde, mit sämtlichen angewandten Variationen (farblich markiert). Die Algorithmen-Wahl wurde aus Platzgründen im Cheat Sheet entfernt, stattdessen wird die OpenSSH-Standardauswahl genommen. Folgendes ist zudem zu bemerken:

- Beim ausschliesslichen Einsatz von YubiKeys mit PIN-Abfrage (`PubkeyAuthOptions verify-required` ist bereits hinterlegt) könnte die Passwort-Authentisierung in der Option `AuthenticationMethods` entfernt werden
- Die Option `AllowGroups` hat hinterlegt, dass nur Benutzer der Gruppe „sshaccess“ (hier „cmd“, „file“, „jump“ und „agent“) Zugriff erhalten
- Konfigurierte Zertifikate und Schlüssel sind entsprechend zu erstellen
- Public-Keys und/oder erlaubte Principals (bei Zertifikats-Authentisierung) sind bei den Zielbenutzern in der zugehörigen Datei (`.ssh/authorized_keys` für Public-Keys, `.ssh/authorized_principals` für Principals) zu hinterlegen
- Für Dateiübertragungen mit Benutzer „jump“ ist ein entsprechender Pfad zu erstellen und zu wählen

```
# Listen ----------------------------
Port                    22
AddressFamily           any
ListenAddress           0.0.0.0
ListenAddress           ::
# Private keys of server
HostKey /etc/ssh/ssh_host_rsa_key
HostCertificate /etc/ssh/
    ssh_host_rsa_key-cert.pub
    # RSA host certificate
HostKey /etc/ssh/ssh_host_ecdsa_key
HostCertificate /etc/ssh/
    ssh_host_ecdsa_key-cert.pub
    # ECDSA host certificate
HostKey
    /etc/ssh/ssh_host_ed25519_key
HostCertificate /etc/ssh/
    ssh_host_ed25519_key-cert.pub
    # Ed25519 host certificate

# Algorithms ------------------------
# Ignore to trust OpenSSH selection
#CASignatureAlgorithms
#Ciphers
#HostKeyAlgorithms
#KexAlgorithms
#MACs
#PubkeyAcceptedAlgorithms
RequiredRSASize         3072

# Ciphers and keying ----------------
RekeyLimit              default none
    # Rekey after ciphers default
    # amount, no timebased rekeying

# Logging ---------------------------
SyslogFacility          AUTH
LogLevel                INFO

# Authentication --------------------
LoginGraceTime          1m
StrictModes             yes
AuthenticationMethods
    publickey,password
    # Force PIN when using
    # FIDO auth algo
    # (i.e. ecdsa-sk or ed25519-sk)
    Only allow pubkey +
    # pwauth combined
PubkeyAuthentication    yes
AuthorizedKeysFile
    .ssh/authorized_keys
    # change to "none" to enforce
    # certificate authentication
PubkeyAuthOptions   verify-required
    # Force PIN when using
    # FIDO auth algo
    # (i.e. ecdsa-sk or ed25519-sk)
HostbasedAuthentication no
PasswordAuthentication  yes
PermitEmptyPasswords    no
KbdInteractiveAuthentication no
# Certificate Authentication -------
TrustedUserCAKeys   /etc/ssh/ca.pub
    # Trusted CA
AuthorizedPrincipalsFile
    .ssh/authorized_principals
    # defined principals

# User / Group Filter ---------------
AllowGroups             sshaccess
PermitRootLogin         no

# Forwarding / Tunnel ---------------
DisableForwarding       yes
AllowAgentForwarding    no
AllowStreamLocalForwarding  no
AllowTcpForwarding      no
PermitListen            none
PermitOpen              none
GatewayPorts            no
X11Forwarding           no
PermitTunnel            no

# Other settings --------------------
PermitTTY               no
    # Don't give user a terminal
ForceCommand            exit
    # exit session
PrintMotd               no
PrintLastLog            no
TCPKeepAlive            yes
PermitUserEnvironment   no
PermitUserRC            no
Compression             no
UseDNS                  no
    # activate if using DNS,
    #not using DNS atm
PidFile             /var/run/sshd.pid
MaxStartups             10:30:100
ChrootDirectory         none
VersionAddendum         none
Banner                  none

# sftp subsystem            sftp
Subsystem               /usr/libexec/sftp-server

# Use cases -------------------------
# --- commandline -------------------
# insert to configure commandline
# access for user "cmd"
# User cmd ----------------------------
Match User cmd
    PermitTTY           yes
    ForceCommand        none

# --- filetransfer ------------------
# insert to configure filetransfer
# access over sftp for user "file"
# in a defined directory
# User file ---------------------------
Match User file
    ForceCommand        internal-sftp
    ChrootDirectory     /data/sftp

# --- jumphost ----------------------
# insert to configure jumphost
# access for user "jump"
# and to open a TCP forwarding
# sessions to defined destinations
# User jump ---------------------------
Match User jump
    DisableForwarding   no
    AllowTcpForwarding  yes
    PermitOpen 192.168.2.1:22
    MaxSessions         0

# --- agent forwarding --------------
# insert to enable agent-forwarding
# and configure commandline access
# for user "agent"
# User agent --------------------------
Match User agent
    AllowAgentForwarding yes
    PermitTTY           yes
    ForceCommand        none
```