SSH-Server-Konfiguration

ITS-Net-Lin

Sebastian Meisel

7. Mai 2025

1 Einführung

Der SSH-Server wird in der Datei `/etc/ssh/sshd_config` konfiguriert. Dies ist eine einfache Textdatei. Zeilen, die mit einer Raute (#) beginnen, sind auskommentiert, d. h., sie werden nicht eingelesen.
Hier stelle ich eine einfache Konfiguration vor, die einen sicheren SSH-Server einrichtet:

1.1 Weitere Einstellungen in `/etc/ssh/sshd_config.d/` erlauben

Mit dieser Zeile wird ermöglicht, dass zusätzliche Dateien im Unterverzeichnis `/etc/ssh/sshd_config.d/` mit der Endung `.conf` weitere Konfigurationsoptionen enthalten können. So bleibt die Konfiguration übersichtlich:

```
1 Include /etc/ssh/sshd_config.d/*.conf
```

1.2 Port ändern

SSH lauscht standardmäßig auf Port 22. Dies kann geändert werden, jedoch bringt dies nur geringen Sicherheitsgewinn:

```
1 Port 2222
```

Wenn diese Einstellung vorgenommen wird, muss zudem die Datei `~/.ssh/config` auf dem Client angepasst werden:

```
Host Debian debian deb
    HostName       debian
    Port           2222              # Angepasster Port!
    UserName       <benutzername>
    IdentityFile   ~/.ssh/<key-file>  # Pfad zur Schlüsseldatei
```

<benutername> und <key-file> müssen hier natürlich angepasst werden.

2 Authentifizierung

Es gibt eine Reihe von Optionen, die bestimmen, wie Benutzer sich auf einem SSH-Server anmelden können. Die Authentifizierung mittels öffentlichem Schlüssel (public key) ist sicher. Daher sollte diese erlaubt und die Anmeldung per Passwort verboten werden:

```
1 PubkeyAuthentication yes
2 PasswordAuthentication no
3 PermitEmptyPasswords no
```

Zudem sollte die interaktive Anmeldung per Tastatur deaktiviert werden:

```
KbdInteractiveAuthentication no
```

Auf Linux-Systemen sollte immer der Standard-Anmeldemechanismus PAM genutzt werden:

```
UsePAM yes
```

2.1 Weitere sinnvolle Einstellungen

Die Spracheinstellungen des Clients sollten durch Umgebungsvariablen an den Server übermittelt werden können:

```
AcceptEnv LANG LC_*
```

Das SFTP-Subsystem sollte aktiviert sein, um Dateiübertragungen wie mit `scp` zu ermöglichen:

```
Subsystem       sftp    /usr/lib/openssh/sftp-server
```

3 Hashing und Verschlüsselungsalgorithmen

SSH unterstützt verschiedene Verschlüsselungsalgorithmen. Die folgende Konfiguration erlaubt nur die sichersten Algorithmen:

```
# KEX-(Key Exchange)-Algorithmen zum Schlüsselaustausch
KexAlgorithms curve25519-sha256@libssh.org,ecdh-sha2-nistp521,ecdh-sha2-nistp384,
    ecdh-sha2-nistp256,diffie-hellman-group-exchange-sha256

# Verschlüsselungsalgorithmen
Ciphers chacha20-poly1305@openssh.com,aes256-gcm@openssh.com,aes128-gcm@openssh.
    com,aes256-ctr,aes192-ctr,aes128-ctr

# MAC-(message authentication code)-Algorithmen
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-
    etm@openssh.com,hmac-sha2-512,hmac-sha2-256,umac-128@openssh.com
```

4 Fail2ban

Fail2ban dient der weiteren Absicherung des Server. Installieren Sie dazu das Paket mit:

```
sudo apt update # Wenn nicht in den letzen 24h ausgeführt
sudo apt upgrade -y # optional, um das System auf den neuesten Stand zu bringen
sudp apt install fail2ban # installiert das eigentlich Paket
```

4.1 Konfiguration

Damit der Dienst und Debian funktioniert, muss die Datei `/etc/fail2ban/jail.d/defaults-debian.conf` angepasst werden:

```
[sshd]
mode=aggressive
enabled = true
backend = systemd
maxretry = 2
```

Die wichitgste Zeile ist `backend = systemd`, da hier auf das von Debian verwendete System zum Logging umgeschaltet wird. Fail2ban greift auf Logging-Informationen von `sshd` zurück, um verdächtige Loggingversuche zu sperren.
Mit `mode=aggressive` werden neben mehrmals fehlgeschlagenen Passwortanmeldeversuchen auch Anmeldeversuche mit einem falschen Publickey gebannt.
Die Zeile `maxentry = 2` legt fest nach wie vielen fehlgeschlagenen Anmeldeversuchen, eine IP gebannt werden soll.