SSH mit Key und 2Factor

Du möchtest dich gerne für unsere Hilfe erkenntlich zeigen 🥺. Gerne. Wir bedanken uns bei dir für deine Spende! 🤣

Hauseigenes Apt-Repo: https://apt.iteas.at

GITLAB Enterprise: 

Um SSH sicher zu gestalten bedienen sich die meisten eines SSH-Key's. Diesen kann man mit einem Passwort versehen, oder auch nicht. Hat den Vorteil das man ohne jeglicher Eingabe sich auf einen Host einloggen kann, wo der Publickey hinterlegt ist. Kommt der private Schlüssel in falsche Hände, kracht es.

Es gibt auch noch andere Möglichkeiten. Z.B. den Key mit einem zweiten Faktor zu kombinieren.

Ziel:

- SSH-Key ohne Passwort
- Login nur mit Key und zweiten Faktor möglich
- Kein Passwortlogin

Installation und Konfiguration

Das ganze konfigurierst du auf einem Debian Bullseye (Proxmox → Sollte auch auf Ubuntu gleich einrichtbar sein). Hierfür ist die Installation eines weitern Paketes notwendig.

```
apt install libpam-google-authenticator -y
```

Also root einloggen und den Befehl `google-authenticator` ausführen. Nun wirst du durch einen einfachen Einrichtungsprozess geführt. Am Ende kannst den 2Factor also QRcode abscannen oder auch den Code direkt raus kopieren. Bei der Einrichtung wird das File angelegt:

```
ls -l /root/.google_authenticator
-r-------- 1 root root 129 Nov  4 12:33 /root/.google_authenticator
```

Dieses kannst du auch auf andere Hosts weiter kopieren für gleichen 2Factor, oder du editierst die Datei und schreibst deinen eigenen 2Factor hinein. Änderungen sind sofort wirksam.

Konfiguration SSH

Kopiere nun zuerst deinen Public Key auf den Rootaccout mit `ssh-copy-id`

In der `nano /etc/ssh/sshd_config` nimmst du folgende Änderungen vor:

sshd_config

```
PermitRootLogin yes
PasswordAuthentication no
ChallengeResponseAuthentication yes
AuthenticationMethods publickey,password publickey,keyboard-interactive
```

Weiters musst du noch in der `nano /etc/pam.d/sshd` folgende Änderungen durchführen:

```
- @include common-auth
+ #@include common-auth

# Disallow non-root logins when /etc/nologin exists.
account    required     pam_nologin.so

+ auth required pam_google_authenticator.so
```

Nun noch SSHD neustarten und dein Server ist um einiges sicherer. `systemctl restart sshd`