Grundlegende Absicherung

Externe Server bzw. generell alle Server, die am Internet hängen, sollten Anmeldungen auf der Konsole ausschließlich über einen SSH-Schlüssel zulassen. Zudem sollte sich der User root gar nicht von außen anmelden können. Außerdem sollte eine Firewall Zugriffe nur über explizit erlaubte Ports erlauben und eine Software installiert sein, die Angriffsversuche erkennt und blockiert. Am Beispiel eines Hetzner Cloud vServers unter Ubuntu 22.04 LTS wird hier beschrieben, wie ein solcher Server abgesichert werden kann.

Anlegen eines neuen Users mit root Rechten

Erfreulicherweise können bei Hetzner, wie auf der Seite Hetzner vServer bestellen & absichern beschrieben, in der Bestellung eines Servers bereits alle SSH-Schlüssel, die auf dem Server erlaubt sein sollen, mit angegeben werden. Voraussetzung dafür ist, dass die öffentlichen Schlüssel im Hetzner-Account abgelegt wurden. Diese Option sollte unbedingt genutzt werden, denn so ist der Server gleich von Beginn an mit einem Grundschutz gegen Attacken von außen versehen, denn eine Anmeldung als root ohne einen passenden privaten Schlüssel ist nicht mehr möglich.

Nachdem der Server das erstmals gestartet wurde, erfolgt die Anmeldung mit `ssh root@<ip_adresse>`. Dazu ist es unbedingt erforderlich, dass ein gültiger öffentlicher Schlüssel im .ssh Verzeichnis des aufrufenden Users liegt. Dafür hat Hetzner bei der Anlage des Servers gesorgt. Alle in der Bestellung angegebenen öffentlichen Schlüssel sind in der Datei ~/.ssh/authorized_keys eingetragen.

Nach einer erfolgreichen Anmeldung wird als User root auf dem Server gearbeitet. Aus diesem Grund enthalten die nachfolgend aufgeführten Kommandos auch kein vorangestelltes sudo.

Als erste Aktion nach einer Neuinstallation und der Erstanmeldung als root ist unbedingt das Betriebssystem zu aktualisieren. Dazu ist der Befehl `apt update && apt upgrade -y` auszuführen. Unter Umständen weist das Upgrade am Ende darauf hin, dass ein Reboot des Systems erforderlich ist. Das kann schnell mit dem Befehl `reboot` herbeigeführt werden.

Im nächsten Schritt ist die Firewall zu konfigurieren und aktivieren. Die Firewall ufw ist bereits installiert, aber noch nicht aktiviert. Bevor sie aktiviert werden kann, sind allerdings noch ein paar Konfigurationen erforderlich. Eingehender Verkehr ist nur auf den explizit zugelassenen Ports zulässig, ausgehender Verkehr wird generell erlaubt.

```
# Eingehenden Verkehr generell verbieten
sudo ufw default deny incoming


# Ausgehenden Verkehr generell eerlauben
```

```
sudo ufw default allow outgoing

# SSH Verbindungen zulassen
sudo ufw allow ssh

# Wenn auch HTTP/HTTPS erlaubt sein soll
sudo ufw allow http
sudo ufw allow https

# Firewall aktivieren
sudo ufw enable
```

Im nächsten Schritt ist ein User mit sudo Rechten einzurichten:

```
#
# Benutzer 'sysadmin' einrichten und zum sudoer machen.
# Dabei wird vorausgesetzt, dass es sich beim eingelogten
# user root und dem neuen user sysadmin um ein und dieselbe
# Person handelt.
#
adduser sysadmin
usermod -a -G sudo sysadmin
mkdir /home/sysadmin/.ssh
cp ~/.ssh/authorized_keys /home/sysadmin/.ssh/
chmod 700 /home/sysadmin/.ssh
chmod 400 /home/sysadmin/.ssh/authorized_keys
chown sysadmin:sysadmin /home/sysadmin/.ssh
chown sysadmin:sysadmin /home/sysadmin/.ssh/authorized_keys
```

In einem neuen Terminal sollte nun ein ssh Anmeldeversuch mit dem neuen User erfolgen. Die Anmeldung muss ohne Eingabe eines Passworts gelingen, es sei denn, das System fragt nach dem Passwort für den ssh Key, wenn dieser mit einem Passwort abgesichert ist.

Nach erfolgreicher Anmeldung muss abschließend geprüft werden, ob der neue User sysadmin mittels sudo mit root-Rechten arbeiten kann. Um dem User zukünftig die Passworteingabe nach einem sudo Befehl zu ersparen, führen folgende Schritte zum Ziel:

```
#
# Eine Datei im Verzeichnis /etc/sudoers.d anlegen
#
sudo vi /etc/sudoers.d/sysadmin-user
```

```
#
# In dieser Datei folgende Zeile einfügen:
#
sysadmin ALL=(ALL) NOPASSWD:ALL

#
# In einem neuen Terminalfenster testen:
#
sudo -i
```

> Der Verzicht auf das sudo Password ist eine Schwächung der Systemsicherheit!

> Die Nachfolgenden Schritte dürfen nur durchgeführt werden, wenn sich der neue User per ssh ohne Password mit seinem ssh Key anmelden und root Rechte einnehmen kann. Ansonsten droht der Ausschluss aus dem Server!.

Anmeldung des Users root verbieten

Ein nächster wichtiger Schritt hin zu einem sicheren System ist das Unterbinden des Login als User root und des Login mit Passwort. Ein Login soll nur noch mittels ssh Schlüssel möglich sein. Dazu sind in der Datei */etc/ssh/sshd_config*folgende Einträge wie dargestellt zu ändern:

```
#
# Anmelden mit root verbieten und nur publickey erlauben
#
PermitRootLogin no
AuthenticationMethods publickey
PasswordAuthentication no
```

Nach der Änderung muss der ssh Service mit dem Befehl `sudo systemctl restart sshd` neu gestartet werden.

Revision #1
Created 2026-09-26 13:21:21 UTC by ralf
Updated 2026-09-26 13:21:21 UTC by ralf