Grundlegende Absicherung eines neuen Servers

Updaten und Upgraden des Systems

Aktualisieren der Paketquellen mit dem Befehl:

```
sudo apt update
```

Ausführen von Upgrades der installierten Pakete mit dem Befehl:

```
sudo apt upgrade
```

Beide Befehle lassen sich auch mit folgendem Befehl kombinieren:

```
sudo apt update && sudo apt upgrade -y
```

Optional - Durchführen eines Distribution-Upgrades: Falls verfügbar, kann auch ein Upgrade der kompletten Ubuntu-Installation auf eine neuere Version durchgeführt werden:

```
sudo apt dist-upgrade
```

Installation von automatischen Sicherheitsupdates

Mit dem Tool `unattended-upgrades` können automatisch Sicherheits- sowie andere essenzielle Updates vom System installiert werden. In der Regel ist dies automatisch installiert, wenn nicht, kann es mit folgendem Befehl installiert werden:

```
sudo apt update
sudo apt install unattended-upgrades
```

In der Grundkonfiguration werden Bugfixes und Sicherheitsupdates automatisch installiert, allerdings das System nicht neu gestartet, wenn dies notwendig ist. Für eine ausführlichere Konfiguration empfehle ich den Artikel von DigitalOcean zum Thema.

Hinzufügen eines neuen Benutzers

Standardmäßig kommt ein Cloud-Server mit einem Root-Login. Dieser wurde je nach Auswahl bei der Einrichtung mit einem Passwort versehen oder bereits mit einem SSH-Key für den Login ausgestattet. Egal was hier zutrifft, es empfiehlt sich den Root-User nicht zu benutzen. Dafür muss mit folgendem Befehl erst mal ein neuer User erstellt werden:

```
adduser <username>
```

Dabei werden nach grundlegenden Informationen wie Passwort, voller Name und anderen Informationen gefragt. Um am Ende auch sudo-Befehle ausführen zu können, muss dieser Benutzer noch der sudo-Gruppe hinzugefügt werden. Das kann mit folgendem Befehl erreicht werden:

```
usermod -aG sudo <username>
```

Dieser Benutzer hat allerdings noch keinen SSH-Key für den Login hinterlegt. Um diesen hinzuzufügen, gibt es mehrere Optionen.

Existierender SSH-Key auf dem Server

Wenn der Root-Benutzer bereits einen hat und dieser auch für den neuen Benutzer genutzt werden soll, kann das Key mit folgenden Befehlen kopiert und mit den richtigen Berechtigungen versehen werden. Das muss mit dem neuen Benutzer ausgeführt werden.

Erstellen des `.ssh-Ordners` für den Key

```
mkdir /home/$USER/.ssh
```

Das Verzeichnis nur für den Benutzer ausführbar machen

```
chmod 700 /home/$USER/.ssh
```

Kopieren der `authorized_keys`-Datei welche den öffentlichen Schlüssel enthält

```
sudo cp /root/.ssh/authorized_keys /home/$USER/.ssh/authorized_keys
```

Alles im `.ssh`-Verzeichnis zum Eigentum des Benutzers machen

```
sudo chown -R $USER:$USER /home/$USER/.ssh
```

Die Datei nur für den Benutzer lesbar machen

```
sudo chmod 600 /home/$USER/.ssh/authorized_keys
```

Existierender SSH-Key auf dem eigenen Computer

Eine andere Möglichkeit für das Übertragen des SSH-Keys ist das Kopieren dieses von einem System, welches bereits einen generiert hat. Das kann mit folgendem Befehl erreicht werden. Voraussetzung ist allerdings, dass der SSH-Login mit dem Passwort für den neuen Benutzer noch nicht deaktiviert wurde:

```
ssh-copy-id <username>@<hostname.example.com>
```

Hier sollte `<username>` der neu angelegte User sein und `<hostname.example.com>` die IP-Adresse oder Hostname des Servers. Ab jetzt sollte ein Login mit dem SSH-Key möglich sein.

Abhärten vom SSH-Server

Um den Login für den Root-Benutzer sowie das Anmelden mit einem Passwort zu deaktiveren, müssen folgende Zeilen in der Datei `/etc/ssh/sshd_config` angepasst werden.

Mit einem Editor die Datei `sshd_config` öffnen

```
sudo nano /etc/ssh/sshd_config
```

Wenn nicht schon auf `no` gesetzt, dies bei `ChallengeResponseAuthentication` machen

```
ChallengeResponseAuthentication no
```

Dann den Punkt `PermitRootLogin` auf `no` setzen

```
PermitRootLogin no
```

Und den Punkt `PasswordAuthentication` sowie `UsePAM` auf `no` setzen

```
PasswordAuthentication no
UserPAM no
```

Zu guter Letzt den Editor verlassen und den SSH-Dienst mit folgendem Befehl neu starten:

```
sudo systemctl reload ssh
```

Ab jetzt ist ein Root- sowie Passwort basierter Login nicht mehr möglich.

Fail2ban für SSH einrichten

Fail2ban kann Brute-Force-Angriffe erheblich eindämmen, indem es Regeln erstellt, die die Konfiguration der Firewall so ändert, dass bestimmte IPs nach einer bestimmten Anzahl von erfolglosen Anmeldeversuchen gesperrt werden. So kann sich der Server gegen diese Zugriffsversuche abhärten, ohne dass man selbst eingreifen muss.

Zu aller erst muss die Software installiert werden. Dies kann mit folgenden Befehlen gemacht werden:

```
sudo apt update
sudo apt install fail2ban
```

Nach der Installation ist noch keine Konfiguration aktiv. Hierfür muss die Beispielkonfiguration `jail.conf` kopiert und ggf. angepasst werden:

```
sudo cp /etc/fail2ban/jail.{conf,local}
```

Out of the box sind die Einstellungen schon für den normalen Gebrauch gut gesetzt. Diese können aber auch beliebig angepasst werden. Ein guter Einstieg in das Thema bietet der Artikel von linuxize.com.

Zu guter Letzt muss auch der Dienst neu gestartet werden, um die Konfiguration zu aktiveren. Dies kann mit folgendem Befehl gemacht werden:

```
sudo systemctl restart fail2ban
```

Revision #7
Created 30 April 2023 08:48:24 by Joshua Lieder
Updated 11 October 2023 14:03:46 by Joshua Lieder