Debian SSH Key Login

Inhaltsverzeichnis

Vorwort:

Um dieses Tutorial zu meistern solltest du bereits wissen was Linux ist und wie du dich zu deinem Linux Server verbinden kannst. Bitte beachte das ich alle Schritte genau erkäre und du alle Befehle sicher und gefahrenlos per Copy & Paste anwenden kannst. Solltest du dir nicht sicher sein oder dieser Anleitung nicht vertrauen liegt es an dir dich über die einzelnen Befehle weiter zu erkundigen.

Ziel des Tutorials:

Wenn ihr das Tutorial Schritt für Schritt befolgt und abgeschlossen habt wird sich der Benutzer root nicht mehr zum Server verbinden können. Stattdessen werden wir uns mit einem neu angelegten Benutzer mit einem SSH Key an Stelle eines Passwortes an unserem Server anmelden. Damit habt ihr ein gutes für die Sicherheit eurer Server beitragen und könnt Nachts etwas besser schlafen :)

1 Vorwort

Um dieses Tutorial zu meistern solltest du bereits wissen was Linux ist und wie du dich zu deinem Linux Server verbinden kannst. Bitte beachte das ich alle Schritte genau erkäre und du alle Befehle sicher und gefahrenlos per Copy & Paste anwenden kannst. Solltest du dir nicht sicher sein oder dieser Anleitung nicht vertrauen liegt es an dir dich über die einzelnen Befehle weiter zu erkundigen.

2 Ziel des Tutorials

Wenn ihr das Tutorial Schritt für Schritt befolgt und abgeschlossen habt wird sich der Benutzer root nicht mehr zum Server verbinden können. Stattdessen werden wir uns mit einem neu angelegten Benutzer mit einem SSH Key an Stelle eines Passwortes an unserem Server anmelden. Damit habt ihr ein gutes für die Sicherheit eurer Server beitragen und könnt Nachts etwas besser schlafen :)

Image not found or type unknown

3 Wir starten!

Zu Beginn bauen wir eine neue Sitzung zu unserem Server auf und loggen uns als Benutzer "root" ein und führen ein Update & Upgrade der aktuellen Packages durch:

Code

```
apt update && apt upgrade
```

Danach überprüfen ob wir bereits "sudo" installiert haben. Sollte das der Fall sein kannst du mit SCHRITT 3 weitermachen.

Sudo ist also noch kein Bestandteil deiner Server Package Liste und das solltest du schnell ändern.

Hier ein Auschnitt aus Wikipedia zu Sudo:

"sudo ([?su?du?],[4] Akronym für su “do”[5]) ist ein Befehl unter Unix und unixartigen Betriebssystemen wie Linux oder macOS, der dazu benutzt wird, Prozesse mit den Rechten eines anderen Benutzers (z. B. des Superusers root) zu starten. Im Gegensatz zu dem nicht zu sudo gehörenden su ist einstellbar, welche Befehle ausgeführt werden dürfen."

Quelle: https://de.wikipedia.org/wiki/Sudo

Kurz: Wir loggen uns später als Benutzer XY ein und wechseln per sudo -u root zb. zum Root Benutzer.

Wir installieren sudo mit:

Code

```
apt install sudo
```

3.) Jetzt erstellen wir den neuen Benutzer mit dem wir uns zukunftig am Server anmelden wollen.

Code

```
adduser [NEUERNAME]
```

Beispiel: adduser evarioo

und fügen diesen Benutzer der sudo Gruppe hinzu. Somit kannst du später per "sudo su root" zum root User wechseln.

Code

```
usermod -aG sudo [NEUERNAME]
```

Beispiel: usermod -aG sudo evarioo

Tipp: Setze statt evarioo den Usernamen ein denn du bei "adduser" gewählt hast!

4.) Aktuell sollten wir noch als Benutzer root angemeldet sein und wechseln daher mit

Code

```
su [NEUERNAME]
```

Tipp: Nutze statt NEUERNAME einfach den Loginnamen den du zuvor bei "adduser" gewählt hast!

zu unserem neuen Benutzer. Wir wechseln mit

Code

```
cd
```

in das Benutzer Verzeichniss. Dort erstellen wir wenn nicht vorhanden (Wenn vorhanden springe bitte zu Schritt 5) den Ordner ".ssh" (Bitte achte unbedingt auf den Punkt zu Beginn des Dateinamens!) und wechseln anschliessend direkt mit cd in das neu erstellte Verzeichniss.

Code

```
mkdir .ssh & cd .ssh
```

5.) Nun können wir den SSH Key für unseren neuen Benutzer erstellen. Dazu nutzen wir gerne bordeigene Mittel. Du kannst natürlich deinen SSH Key auch gerne auf andere Weise zb. online generieren. Vor oder

Nachteile wirst du denke ich dadurch keine haben.

Code

```
ssh-keygen -t rsa -b 2048
```

Was macht dieser Befehl: Wir generieren nach Eingabe des Befehls ein Schlüsselpaar mit einer Grösse von 2048 Bits.

Gefolgt von der Eingabe werden euch ein paar Fragen gestellt.

Die wichtigsten Fragen sind:

Nummer 1: Möchtest du deinen Schlüssel durch eine Passphrase sichern?

Frage: Was ist eine Passphrase?

Antwort: Eine Passphrase besteht im Vergleich zu einem Passwort aus einer größeren Anzahl an Zeichen. Aufgrund längerer und schwerer zu erratender Zeichenketten lässt sich eine größere Sicherheit durch die Verwendung von Passphrasen erzielen. Eine Passphrase kann für Verschlüsselungen, Signaturen oder für den Zugangsschutz von IT-Systemen eingesetzt werden.

Quelle: https://www.security-insider.de/was-ist-eine-passphrase-a-752388/

Nummer 2: An welchem Ort möchtest du deine neuen Schlüssel speichern?

Eigentlich total einfach: Da wir uns aktuell im Verzeichniss ".ssh" des richtigen Benutzers befinden wäre der perfekte Speicherort auch genau dort. Als neuen Speicherort solltest du also auf Nachfragen genau das angeben.

Code

```
~/.ssh/USERNAME
```

6.) Soweit so gut: Von dem eben erstellen Key benötigen wir nun den "Public" Teil des Schlüsselpaares. Diesen speichern wir in der "authorized_keys" Datei welche wir im Verzeichniss ".ssh" des neuen Benutzers erstellen.

Code

```
nano authorized_keys
```

und fügen dort den Teils des Schlüssels ein der sich in der Datei mit der Endung .pub befindet.

Wir haben jetzt soweit alles vorbereitet und kommen zu dem Teil an dem wir den SSH Server nun so einstellen das sich der Benutzer root nicht mehr verbinden kann und sich generell alle Benutzer mit einem SSH Key anmelden müssen. Wir setzen daher folgende Einstellungen in der sshd_config unseres Servers. Wir finden diese Datei im Ordner etc/. Wir verlassen den aktuellen User mit

Code

```
exit
```

und sind nun wieder als Root Benutzer angemeldet. Mit

Code

```
cd /etc/ssh & nano sshd_config
```

öffnen wir die SSH Server Konfiguration und ändern dort nun folgende Teile:

Code

```
PermitRootLogin no ## Verbietet den Login als Benutzer root
MaxAuthTries 6 ## Maximal 6 Authentifizierungs Versuche pro Benutzer
MaxSessions 10 ## Maximaler gleichzeitiger Verbindungen per SSH
PubkeyAuthentication yes ## Authentifizierung per Public Schlüssel
AuthorizedKeysFile .ssh/authorized_keys ## Datei in dem Public Schlüssel hinterlegt sind
PasswordAuthentication no ## Verbietet Benutzern Passwort Authentifizierung und erlaubt nur gültige Schlüssel
PermitEmptyPasswords no ## Leere Passwörter .. nahh!
```

7.) Wir können nun den SSH Server neustarten:

Code

```
systemctl restart sshd
```

```
(Danke für den Hinweis bzw den Vorschlag :)
```

und können anschließend testen ob unser neuer Login funktioniert. Öffnet dazu einfach euer SSH Programm / Tool .. oder was auch immer ihr nutzt und legt den neuen Benutzer inklusive des SSH Schlüssel an. Achtet hierbei darauf bitte den Teil des Schlüssel zu benutzen der keine Datei Endung hat.

Jaaa .. ich denke das war es .. ich konnte bei mir alle Schritte nachvollziehen. Sollte es an einer Stelle zu Fehlern kommen bitte ich um eine kurze Nachricht damit ich das überprüfen und bearbeiten kann.

Liebe Grüsse

4 Nachtrag

Einen Server egal welcher Art zu administrieren bedarf Köpfchen und Verstand. Ihr habt eine gewisse Verantwortung euren Mitmenschen gegenüber. Solltet ihr weitere Fragen dazu haben steht euch das Forum jederzeit offen für Probleme und Anregungen.