

SSH

- SSH Public Key Authenticatie
- SSH shortcuts maken in Linux

SSH Public Key Authenticatie

public-private-300x127-1.png

SSH wordt gebruikt om op afstand in te kunnen loggen op een Linux systeem. Je kunt inloggen door middel van een gebruikersnaam en wachtwoord, maar veiliger is het om dit te doen met behulp van public key authenticatie. Hiervoor dienen we een public en private key te genereren op de PC waar vandaan we willen inloggen op het remote systeem. Hieronder volgen voorbeelden voor zowel een Linux als Windows PC.

SSH key's genereren op een Linux PC

Open een terminal (Ctrl-Alt-T) en voer het volgende commando uit om een private en public key te genereren:

```
ssh-keygen
```

Er wordt nu gevraagd een bestandsnaam op te geven. Wanneer je dit niet doet, wordt standaard de naam **id_rsa** (private key) en **id_rsa.pub** (public key) gegeven aan de bestanden. Standaard worden de private en public key opgeslagen in de verborgen directory genaamd **.ssh** van de gebruiker waarmee ingelogd is. Dit is dus de directory **~/.ssh**. Er wordt ook gevraagd om een "passphrase" tijdens het genereren. Dit is een wachtwoord waarmee de private key beveiligd wordt. Je kunt dit eventueel overslaan door bij de vraag niets op te geven. Het wachtwoord wordt gevraagd tijdens de inlog sessie op de remote machine. Indien geen wachtwoord is opgegeven wordt je direct ingelogd.

ssh-keygen-e1520435536763.png

Wanneer je een bestandsnaam opgeeft tijdens het genereren, dan dien je deze bestandsnaam op te geven in het **ssh** commando door middel van de optie **-i (identity file)**. Bij de standaard naam (**id_rsa**) is dit niet nodig. Hieronder staat een voorbeeld:

```
ssh -i ~/.ssh/linuxfun.key username@linuxfun.nl
```

Als je gebruik maakt van een SSH config bestand in de **.ssh** directory, dan kan je dit opgeven met de parameter **IdentityFile**. Zie het voorbeeld hieronder:

```
Host linuxfun
  HostName linuxfun.nl
  User username
  IdentityFile ~/.ssh/linuxfun.key
```

Meer informatie over het SSH config bestand vind je [**hier**](#). Nu we de private en public key gegenereerd hebben, moeten we de public key op de remote machine zien te krijgen. Dit kan o.a. door middel van het commando **scp**. Een voorbeeld staat hier onder:

```
scp ~/.ssh/id_rsa.pub username@linuxfun.nl:
```

De public key **id_rsa.pub** wordt hiermee gekopieerd naar de home directory van de gebruiker **<username>**. Wanneer de public key op de remote machine staat, dien je nu daarop in te loggen. We moeten de inhoud van de public key in het bestand **~/.ssh/authorized_keys** zien te krijgen. Wanneer de directory **.ssh** en het bestand **authorized_keys** nog niet bestaan op de remote machine, dan dienen we deze eerst aan te maken. Zie de commando's hier onder:

```
mkdir ~/.ssh  
touch ~/.ssh/authorized_keys
```

Wanneer het bestand **authorized_keys** al bestaat, dan wordt door bovenstaande commando's het bestand niet overschreven. We kunnen nu de inhoud van de public key toevoegen aan het bestand **authorized_keys** met het commando:

```
cat ~/id_rsa.pub >> ~/.ssh/authorized_keys
```

Je kunt controleren of de public key goed in het bestand **authorized_keys** staat met het commando:

```
more ~/.ssh/authorized_keys
```

Je kunt nu veilig de public key verwijderen van het remote systeem met:

```
rm ~/id_rsa.pub
```

Wanneer je een kopie wilt bewaren van de public key, dan kan je deze het beste verplaatsen naar de **.ssh** directory:

```
mv ~/id_rsa.pub ~/.ssh/
```

Het is nu mogelijk in te loggen op het remote systeem door middel van de ssh key's. Dit gebeurt nu automatisch zonder een inlognaam en wachtwoord op te hoeven geven. Wanneer men een private key heeft gegenereerd met een wachtwoord (passphrase), dan dient men dit wachtwoord op te geven tijdens het inloggen.

SSH key's genereren op een Windows PC

Onder Microsoft Windows kan je de public en private key's genereren met behulp van de tool **PuTTYgen**. Dit is een tool die geïnstalleerd wordt samen met de SSH client **PuTTY**. Je kunt dit [**hier**](#) downloaden. Na installatie vind je de tools terug in het Windows start menu onder **PuTTY**. Start nu het programma **PuTTYgen** op en klik op **Generate**.

puttygen01.jpg

Je kunt nu de key's genereren door de muis te bewegen in het vak **Key**.

puttygen02.jpg

Wanneer dit klaar is, zie je het volgende resultaat:

puttygen03.jpg

Door nu op de knop **Save private key** te klikken, kan je de private key opslaan op je Windows PC. Er wordt dan gevraagd of je de private key wilt opslaan zonder wachtwoord (passphrase). Wil je de private key beveiligen met een wachtwoord, vul dan de regel **Key passphrase** in en bevestig dit in de regel **Confirm passphrase**. De private key wordt opgeslagen met de extensie **.ppk** en kan direct gebruikt worden met de SSH client **PuTTY**. De public key, voor de remote machine, kunnen we naar het klembord kopiëren door met de rechter muisknop in het veld **public key** te klikken. Hier kiezen we vervolgens **Alles selecteren** (als dit nog niet geselecteerd is) en daarna voor **Kopiëren**. Door nu in te loggen op de remote machine, kan je vervolgens de public key plakken in het bestand **~/.ssh/authorized_keys** met behulp van je favoriete tekst editor.

puttygen04.jpg

Om de private key te gebruiken in **PuTTY** open je dit programma en maak je een nieuwe connectie aan of je opent een bestaande connectie. Vervolgens ga je in de linker boom structuur naar de categorie **Connection** en selecteer je hier onder **Data**. In het veld **Auto-login username** kan je de gebruiker invullen waarmee je automatisch wilt inloggen.

putty01.jpg

Vervolgens kies je onder **Connection** de optie **SSH** en hier onder optie **Auth**. In het veld **Private key file for authentication** blader je naar het private key bestand met de .ppk extensie.

putty02.jpg

Om de configuratie op te slaan selecteer je boven in de boom structuur voor **Session** en vervolgens **Save**.

putty03.jpg

Wanneer dit is opgeslagen, kan je de verbinding maken met de remote machine en log je automatisch in. Mocht je de private key beveiligd hebben met een wachtwoord, voer dit dan in wanneer je inlogt en hierom gevraagd wordt.

Uitschakelen password login in SSH

Nu we gebruik maken van public en private keys, is het verstandig om het inloggen via wachtwoorden uit te schakelen. Hier onder volgen de instructies voor zowel **OpenSSH** als **Dropbear** voor Debian Linux:

OpenSSH

Open het volgende bestand in een tekst editor:

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

Pas daar de volgende opties aan:

```
ChallengeResponseAuthentication no
PasswordAuthentication no
UsePAM no
```

Wanneer je niet wilt dat je als root kan inloggen, pas dan onderstaande optie aan:

```
class="lang:default decode:true " title="Disable root login">PermitRootLogin no
```

Sla het bestand op en herstart de ssh daemon met:

```
/etc/init.d/ssh reload
OF
sudo systemctl reload ssh
```

Dropbear

Open het volgende bestand in een tekst editor:

```
nano /etc/default/dropbear
```

En pas de volgende optie aan:

```
DROPBEAR_EXTRA_ARGS="-g -s"
```

Wanneer je niet wilt dat je als root kan inloggen, dan kan je de parameter **-w** nog toevoegen.
Herstart de Dropbear daemon met:

```
/etc/init.d/dropbear restart
```

Bronnen:

[Set up SSH public-key authentication to connect to a remote system](#)

[Dropbear, SFTP and passwordless logins in Debian](#)

[How to disable ssh password login on Linux to increase security](#)

SSH shortcuts maken in Linux

ssh-300x110-1.jpg

SSH wordt veelvuldig gebruikt in Linux om via een terminal in te loggen op een remote machine. Dit gaat dan via de command line en ziet er bijvoorbeeld als volgt uit: **ssh -p 2222**

root@192.168.7.6 (Hierin is -p de poort waar ssh op luistert, standaard poort 22). Nu kan je in Linux zogenaamde ssh shortcuts maken, zodat je niet iedere keer die lange commando's hoeft in te tikken. Je krijgt dan bijvoorbeeld een commando als: **ssh thuis** Dit kan je vastleggen in het bestand **config**. Dit bestand hoort dan te staan in de verborgen directory **.ssh** die op zijn beurt weer staat onder de Home directory van een Linux gebruiker. We gaan dus eerst naar de juiste directory met behulp van onderstaand commando:

```
cd ~/.ssh
```

Hierna kunnen we het bestand **config** gaan maken met:

```
nano config
```

Hier kunnen we de regels plaatsen die we nodig hebben. **Voorbeeld:**

```
Host thuis
  HostName 192.168.7.6
  User root
  Port 2222
Host example2
  HostName example.com
  User root
Host example3
  HostName 64.233.160.0
  User userxyz123
  Port 56000
Host amazon1
  HostName ec2.amazon.com
  User ec2-user
  IdentityFile /path/to/special/privatekey/amazon.pem
```

Sla het bestand op. De ssh shortcuts zijn direct actief. Uitgaande van bovenstaand voorbeeld kan je nu verbinding maken met de commando's: **ssh thuis, ssh example2, ssh example3, ssh amazon1**

Bron:

How to Create an SSH Shortcut