SSH (Secure Shell)

7 februari 2009, Johan Swenker
Linux Platform HCC Compusers

Inloggen en bestanden kopieren

Als eerste stap hebben we een commando regel nodig om opdrachten te kunnen geven. Bij Ubuntu zit die onder Applications, Accessories, Terminal, of de vertaling daarvan.

Mijn laptop heeft IP-adres 192.168.2.134[1], een gebruiker met naam `gast`[2] en wachtwoord `gast`. Je kunt inloggen met:

```
ssh -l gast 192.168.2.134
```

Kryptische melding: hoe weet je zeker dat je met het juiste systeem praat. Het systeem meldt zich met zijn cryptografische identiteit. Die moet je controleren.

Je kunt nu op mijn laptop opdrachten geven als de gebruiker gast.

```
gast@zot:~$[3] uptime ; date ; who ; cat /proc/cpuinfo
```

Uitloggen en een andere functionaliteit van ssh gebruiken.

```
gast@zot:~$ exit
```

Met SSH kun je prima files kopieren. Maak eerst een dummy document aan :

```
date > document_van_jan
```

en kopieer dat eens heen en weer

```
scp document_van_jan gast@192.168.2.134:
scp gast@192.168.2.134:document_van_jan document_cp
```

RSA authenticatie

Bij TCP/IP heb je niet alleen adressen zoals www.hcc.nl of 82.201.119.119 maar ook poorten. Met een poort adresseer je een specifiek programma op de server. Mijn laptop is de server voor SSH.

Ik heb nog een tweede SSH-server op mijn computer staan. Die is iets anders geconfigureerd. Om die te benaderen moet je expliciet aangeven welke poort gebruikt moet worden. Gebruik poort 2222 als volgt:

```
ssh -l gast 192.168.2.134 -p 2222
```

Waarom werkt het niet? Probeer eens de optie `-v` om het verbose (met uitgebreide informatie) te doen.

```
ssh -l gast 192.168.2.134 -p 2222 -v
```

De ssh-deamon op de default poort vindt het best als je een naam en wachtwoord opgeeft. De ssh-deamon op poort 2222 eist public key authenticatie. Dit heeft 2 voordelen:

- wachtwoord raden is niet mogelijk ;
- met een trucje heb je single sign on.

---

1 In dit document staat 192.168.2.134 als IP-adres van de SSH-server waarop je kunt inloggen. Het IP-adres wat beschikbaar is bij de workshop, was is tijdens het schrijven van dit document nog niet bekend.

2 `Tekst in dit lettertype bevat letterlijke opdrachten voor de computer.`

3 gast@zot:~$ is de prompt van de SSH-server. Commando's die hier achter staan moet je dus op de server geven.

Laten we dus maar zo'n public/private key paar gaan maken: `rsa-keygen`, accepteer de defaults en kies een wachtwoord, bij voorkeur **niet** gast.

De private key komt in het bestand `id_rsa` in de subdirectory voor ssh. Met dit bestand identificeer je jezelf. Iedereen die dit bestand kan gebruiken, kan zich als jou voordoen. Ik heb `id_rsa` alleen staan op computers waar ik als enige root op ben. Dus niet op servers op kantoor, en niet op servers van xs4all. Het wachtwoord is een bescherming van `id_rsa`, zonder dat wachtwoord is dit bestand niet te gebruiken.

De public key komt in bestand `id_rsa.pub`. Met dit bestand kan iemand anders verifieren dat jij over de bijbehorende `id_rsa` kunt beschikken, kortom dat jij 'jij' bent. Dit bestand moet je naar alle computers (laten) kopieren waarop je mag en wilt inloggen. Mijn `id_rsa.pub` staat dus onder andere bij xs4all.

Het bestand moet dus gekopieerd worden naar alle computers waarop je wilt inloggen. In dit geval naar mijn laptop. Maar, dat moet niet gewoon met `scp`. Op mijn laptop staat immers al mijn `id_rsa.pub` en bovendien wil ik nu al jullie public keys opnemen. De public key moet in `authorized_keys` terecht komen. Dat kan met het commando `ssh-copy-id`:

```
ssh-copy-id gast@192.168.2.134
```

En nu eens kijken wat ssh nu zegt. Laten we maar meteen de versie gebruiken op poort 2222. Misschien komt er nu al een agent te voorschijn die aanbiedt om het wachtwoord van de private key te onthouden.

```
ssh -l gast 192.168.2.134 -p 2222
gast@zot:~$ exit
```

Ik gebruik dit op alle SSH-servers die aan het internet hangen. Die kun je dus niet benaderen door wachtwoorden te proberen. Ik moet er dus wel voor zorgen dat de private key ook echt private blijft.

Maar het kan nog makkelijker. Daarvoor hebben we de `ssh-agent`. Moderne linux-distributies hebben allemaal al een agent draaien voor het bijhouden van wachtwoorden van private keys. (Of je zet natuurlijk geen wachtwoord op de private key, maar dan moet je nog secuurder zijn op het beschermen daarvan).

Met het commando `ssh-add` geef je de `ssh-agent` de beschikking over het wachtwoord van de private key. Je kunt nu inloggen zonder verder een wachtwoord te gebruiken. Feitelijk heb je nu single sign on gecreeerd: overal waar dezelfde public key staat, heb je met dezelfde private key toegang.

```
ssh-add
ssh -l gast 192.168.2.134 -p 2222
gast@zot:~$ exit
```

Tunneling

SSH kan verschillende soorten netwerkverkeer tunnelen door de gecrypte verbinding.

Een daarvan is X-windows.
Gebruik de optie `-X` om het tunnelen van X-windows aan te zetten. Uit security overwegingen staat

dit default uit.

```
ssh -l gast 192.168.2.134 -X
```

Probeer of het werkt door een eenvoudige X-applicatie zoals xclock op te starten op mijn laptop. De uitvoer moet op jouw scherm komen.

gast@zot:~$ `xclock`

afbreken met control-C

Ook de agent kan getunneld worden door de gecrypte verbinding. Probeer eerst of je nu zonder wachtwoord op te geven nog een keer kunt inloggen op mijn laptop met:

gast@zot:~$ `ssh localhost`

gast@zot:~$ `exit`

Een wachtwoord wordt gevraagd. Ook het tunnelen van de ssh-agent staat default uit. Dit is aan te zetten met de optie `-A`. Probeer het nog maar een keer door vanuit je eigen PC in te loggen op mijn laptop en daarna nog een keer naar localhost.

```
ssh -l gast 192.168.2.134 -X -A
```

gast@zot:~$ `ssh localhost`

gast@zot:~$ `exit`

gast@zot:~$ `exit`

Portforwarding (tunneling) met SSH

Met de opties `-L` en `-R` kun je TCP-poorten forwarden.

```
ssh -l gast 192.168.2.134 \
   -L 8888:localhost:80
```

door `\` negeert bash de regelovergang

- `-L` een Locale verbinding naar poort
- `8888` wordt doorgezet naar
- `localhost` (de server waarop je inlogt)
- op poort `80`

Dit commando kun je op twee regels typen, dan moet je een `\` typen vlak voor de regelovergang. Je mag het commando ook als een regel typen:

```
ssh -l gast 192.168.2.134 -L 8888:localhost:80
```

Dit betekent dat verbindingen die op jouw locale machine naar poort 8888 gaan, doorgestuurd worden naar 'localhost' op poort 80. Die localhost is de localhost nadat de verbinding gemaakt is. Dat is dus mijn laptop.

En probeer maar wat er gebeurt als je op jouw eigen PC de webpagina localhost:8888 opent.

```
firefox localhost:8888
```

gast@zot:~$ `exit`

Dit is heel geschikt voor het forwarden van:

- webverkeer naar je ADSL-modem, normaal kun je niet 'achter' je eigen router komen.
- inkomend mailverkeer, de wachtwoorden van pop3 zijn af te luisteren. Op het netwerk van je eigen provider is dat geen punt. Over het internet is dat ongewenst.

De optie `-R` gebruik je als je vanuit Verwegistan iets wilt gaan doen op servers in je eigen omgeving. Het enige voorbeeld wat ik heb is een patchserver in de eigen omgeving, die je vanuit Verwegistan wilt gebruiken.

Afronding

Lees ook eens de manual pagina's van `sshd_config` en `ssh_config`. Dan kun je in de configuratiefile opnemen dat je altijd X-forwarding wilt, of dat je altijd bepaalde TCP-poorten wilt forwarden.

```
man ssh
```

bladeren met pijltjes en page-up/down; zoeken met `/zoekstring` ; afsluiten met `q` .

```
man sshd
man ssh_config
man sshd_config
```

Onder Linux wordt openssh gebruikt. Deze wordt in vrijwel alle distributies meegeleverd. Onder Windows worden putty en winscp gebruikt.