FACULTEIT INGENIEURSWETENSCHAPPEN
EN ARCHITECTUUR

Voorbereiding Labo SSH

Bert De Saffel

Master in de Industriële Wetenschappen: Informatica Academiejaar 2018–2019

UNIVERSITEIT
GENT

Inhoudsopgave

Hoofdstuk 1

Inleiding

SSH (Secure Shell) laat toe om te authenticeren op een ander toestel dan waarop je fysiek aanwezig bent. SSH vervangt de taak van vaak voorkomende protocols of programmas zoals *FTP*(File Transfer Protocol), *telnet*, *RSH*(Remote Shell) en *RCP*(Remote Copy). Het gebruik van SSH biedt twee **voordelen**: Het voorkomt pakket sniffing door gebruik te maken van encryptie en het biedt authenticatie aan om hijacking van een sessie te voorkomen. Tabel 1.1 toont de binaries die op de meeste linux systemen aanwezig zijn, met de belangrijkste aangeduid in het vet. De programmas **ssh2**, **sshd2**, ... (dus de 2 toevoegen) bestaan ook, maar deze zijn echter van een volgende release. Het is nog niet duidelijk welke versie op het labo zal gebruikt worden, maar we gaan ervan uit dat deze versie 1 is. Deze binaries zijn verkrijgbaar via het *SSH Secure Shell Server* pakket.

Binary	**Vervangt**	**Description**
ssh	telnet, rsh	Secure shell client (remote login program)
sshd	telnetd, rshd	Secure shell daemon
scp	rcp	Secury copy client
sftp	ftp	Secure ftp client
ssh-keygen	-	Authentication key pair generation
ssh-agent	-	Authentication agent
ssh-add	-	Toevoegen van identiteiten voor een authentication agent
ssh-chrootmgr	-	Opzetten van chroot-ready omgevingen voor gebruikers.
ssh-pubkeymgr	-	Opzetten van public key authenticatie

Tabel 1.1: Binaries van het SSH Secure Shell Server pakket.

Hoofdstuk 2

Configuratie

Dit hoofdstuk bespreekt de verschillende configuratiemogelijkheden die SSH aanbiedt.

2.1 Server Configuratie

Het programma **sshd** wordt uitgevoerd op een server. Deze zoekt zijn configuratie in het bestand **etc/ssh/sshd_config**. Elke regel in het bestand begint met een sleutel gevolgd door een spatie en dan de waarde voor die optie. Een lijst van de belangrijkste opties zijn:

- **PublicKeyAuthentication**: Geeft aan of dat public key authentication gebruikt wordt of niet (yes, no).
- **HostbasedAuthentication**: Geeft aan of dat Host based authentication gebruikt wordt of niet (yes, no).
- **Port**: Instellen van de poort waarop geluisterd moet worden.
- **UseDNS**: Is enkel nuttig indien clients reverse DNS kunnen toepassen.
- **Protocol**: Geeft aan welke versie van SSH gebruikt moet worden (1 of 2).
- **AuthorizedKeysFile**: Specificeert het bestand waarin de publieke sleutels staan. Default is dit **$HOME/.ssh/authorized_keys**.

2.2 Client Configuratie

De programmas die gebruikt worden op een clienttoestel zijn **ssh**, **sftp** en **scp**. De client configuratie kan enerzijds op **userniveau** gebeuren in het bestand **$HOME$/.ssh/config** of op **systeemniveau** in het bestand **/etc/ssh/ssh_config**. Volgende twee lijnen tonen het gebruik van ssh en scp.

```
ssh gebruiker@hilbert # connectie naar hilbert als 'gebruiker'
scp file gebruiker@hilbert:/home/user # kopie van het bestand 'file' naar
                                      # user directory van 'gebruiker'
```

Een lijst van de belangrijkste opties zijn:

- **HostName**: De default hostname instellen
- **User**: De default user instellen
- **PubkeyAuthentication** Geeft aan dat public key authentication gebruikt wordt of niet (yes, no).
- **LocalForward**: Deze optie heeft twee parameters, LPORT en RHOST:RPORT. Deze optie zal LPORT mappen op RHOST:RPORT.
- **RemoteForward**: Deze optie heeft twee parameters, RPORT en LHOST:LPORT.

2.3 Public Key Authentication

Public key authentication is de meest aangewezen methode om informatie te versturen. Deze configuratie moet gebeuren voor elke gebruiker op een toestel.

1. De eerste stap is het aanpassen van zowel het **sshd_config** bestand op de server als het **ssh_config** bestand op de client.
   - **sshd_config.** Hier moet de lijn **PublicKeyAuthentication yes** aanwezig zijn.
   - **ssh_config.** Hier moet de lijn **PubkeyAuthentication yes** aanwezig zijn.

   In beide gevallen zijn dit ook de default waarden.
2. Bij een ingelogde gebruiker moet er een sleutelpaar gegenereerd worden met `ssh-keygen -t rsa -f $HOME/.ssh/id_rsa`. bij het uitvoeren van dit commando zal er gevraagd worden om een **passphrase** te geven. laat deze passphrase leeg indien de testfase nog bezig is.
3. De inhoud van het bestand dat gegenereerd wordt (**id_rsa.pub**) moet toegevoegd worden in het bestand dat gespecificeert staat bij de optie **AuthorizedKeysFile** in **sshd_config**.
4. Er kan nu ingelogd worden met `ssh` *`servernaam`*. Hier wordt er gevraagd naar de **passphrase**. Om te vermijden dat deze passphrase telkens opnieuw moet ingegeven worden, kan gebruik gemaakt worden van volgende twee lijnen:

   ```
   eval 'ssh-agent'
   ssh-add ~/.ssh/id_isa
   ```

   Zolang de terminal actief is waarin deze twee lijnen werden uitgevoerd, moet de passphrase niet opnieuw ingegeven worden.

2.4 Host-based Authentication

Host-based authentication is nuttig wanneer er scripts moeten uitgevoerd worden op een remote toestel. Deze soort authenticatie vereist configuratie op zowel het servertoestel als het clienttoestel.

- **Server.**
  1. Eerst moet het bestand **sshd_config** aangepast worden met volgende lijnen:

```
HostbasedAuthentication yes
IgnoreRhosts            no
IgnoreUserKnownHosts    no  # optioneel
```

De daemon moet herstart worden om deze nieuwe configuratieregels in te lezen. Gebruik hiervoor `service sshd restart`.

2. De client zijn public key moet gekopieerd worden naar de SSH server in **/etc/ssh/ssh_known_hosts** op systeemniveau of **$HOME/.ssh/known_hosts** op userniveau. Deze public keys kunnen gevonden worden in **/etc/ssh/ssh_host_rsa_key.pub** op het clienttoestel.
3. De FQDN (Fully Qualified Domain Name) van een client moet in **$HOME/.shosts** komen. De permissies van dit bestand moet zo ingesteld zijn dat enkel de user eigenaar is en mag niet schrijfbaar zijn voor groepen/anderen.

- **Client.**

1. De inhoud van het bestand **ssh_config** moet **HostbasedAuthentication yes** bevatten.
2. Er moet een RSA key pair zijn. Indien deze bestaan zijn ze te vinden in

```
/etc/ssh/ssh_host_rsa_key
/etc/ssh/ssh_host_rsa_key.pub
```

Indien deze nog niet bestaan, gebruik dan

```
ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key -N ""
```

om deze te genereren.

2.5 TCP forwarding

SSH can TCP verkeer forwarden via een SSH verbinding om zo applicaties te beveiligen. Om forwarding toe te passen moet er eerst ingelogd worden op een SSH server (zie sectie 2.2)

2.5.1 Local Forwarding

Local Forwarding geeft aan dat een bepaalde poort op het lokale toestel geforward moet worden naar een bepaalde host en poort op een remote toesteL. Het meest eenvoudige commando ziet er zo uit:

```
ssh -L sourcePort:forwardToHost:onPort connectToHost
```

Dit betekent letterlijk: *Connecteer met ssh naar connectToHost, en forward alle connecties op het lokaale sourcePort naar onPort van het toestel forwardToHost.* De parameters connectToHost en forwardToHost kunnen dezelfde zijn.

2.5.2 Remote Forwarding

Dit doet het omgekeerde als Local Forwarding. Een poort van een remote host zal geforward worden naar het client toestel. Hier moet de optie -L vervangen worden door -R. De parameters blijven dezelfde.