Configurare un server SSH

In questo articolo tratterò di come configurare un server SSH su una distribuzione GNU/Linux usando il famoso programma open-source OpenSSH.
Nota: questo procedimento è stato realizzato su una Debian Lenny usando OpenSSH v5.1

Download
Il solo pacchetto da scaricare è openssh-server (se state usando APT), altrimenti potrete trovarlo su http://openssh.com

Avvio/riavvio del server
Per far partire il server digitate sul terminale, da superutente
`# /etc/init.d/sshd start`
Per fermarlo o riavviarlo sostituire start rispettivamente con stop e restart.

Configurazione
Il file di configurazione si trova in `/etc/ssh/sshd_config`.
Ecco le opzioni più importanti che potreste voler cambiare:

`Port N` # qui specificherete la porta su cui ascoltare, di default la 22
`AllowUsers user1 user2 ...` # gli utenti che potranno connettersi al vostro server
`AllowGroups group1 group2 ...` # i gruppi di cui devono far parte per potersi connettere
`PermitRootLogin no` # decide se permettere ad un utente con privilegi di root di accedere direttamente. È consigliato scegliere no per la sicurezza.

```
RSAAuthentication yes
PubkeyAuthentication yes
PasswordAuthentication no
AuthorizedKeysFile %h/.ssh/authorized_keys
```

Qui dovrete decidere quale tipo di autenticazione usare, se via password o via chiave pubblica. Nel caso della seconda dovete specificare il path del file con le chiavi pubbliche degli utenti (il suo uso sarà spiegato dopo). Il carattere `%h` indica la home dell'utente che si sta connettendo.

`PermitEmptyPasswords no` # possibilità di eseguire login anonimi
`Banner /home/path/banner` # un banner da visualizzare appena ci si connette (file di testo)

Dopo aver aggiustato il file siete praticamente pronti per far girare il vostro server.

Aggiunta nuovi utenti
Bene, ora che avete il server è ora di "aprirlo" ai vostri utenti. Ecco il procedimento per permettere ad un utente di connettersi.

1. Aggiungere l'utente

Questo è possibile usando l'utility `adduser`. L'opzione `–ingroup` specifica il gruppo in cui va inserito (se mancante, il sistema creerà un nuovo gruppo con lo stesso nome utente). Altrimenti potrete crearlo a parte con `addgroup`.
Allo stesso modo esistono `deluser` e `delgroup` per l'operazione inversa.
Nota: se avete scelto l'autenticazione via password, questa che sceglierete voi sarà la password di accesso. Altrimenti servirà solo per eventuali usi di `sudo` o `su`.

2. Permesso

Nel file di configurazione, aggiungere l'utente e il gruppo di cui fa parte ad `AllowUsers` e

`AllowGroups` separandoli con uno spazio bianco. Notate che i valori di queste due entry sono necessari entrambi: se l'utente è presente in `AllowUsers` ma il gruppo di cui fa parte non è fra gli `AllowGroups`, l'accesso fallirà.

3. Key pair

A questo punto l'utente è pronto per autenticarsi via password. Per le chiavi pubbliche invece, l'utente deve disporre di una coppia di chiavi RSA, del numero di bit specificati dall'opzione `ServerKeyBits` in `sshd_config` (in genere 1024). Una nuova coppia si può creare su Linux con

```
$ ssh-keygen -t rsa -b 1024
```

Il file `id_rsa.pub` conterrà la chiave pubblica.
Se i vostri utenti usano Windows possono usare l'utility `puttygen` per generare le chiavi.

4. Authorized_keys

Andate nella home dell'utente, alla directory dove avete posto `authorized_keys`. Create il file ed inserite le chiavi pubbliche delle persone che volete far connettere come quell'utente. Ad esempio, mettiamo di avere i tre utenti fire foo e bar, corrispondenti a Lucia Anna e Stefano. Voglio che sia Lucia e Anna possano connettersi come foo, mentre Stefano deve potersi connettere come fire e bar. Quindi nell'`authorized_keys` di foo metterò le chiavi pubbliche di Lucia e Anna, mentre nel file di bar e fire metterò la chiave pubblica di Stefano.

Conclusione

Bene è tutto pronto! Adesso non vi rimane che settare i permessi come volete, e se non disponete di un IP statico potete usare i vari servizi che permettono di associare un nome host al vostro computer.
Un file di log particolarmente importante sarà a questo punto `/var/log/auth.log` che segnala tutti gli accessi (falliti o meno) al vostro sistema.
Enjoy :)

by shainer
<syn.shainer@gmail.com>