Faculté des Sciences de Limoges
DESS Sécurité de l'Information
Cryptographie 2002–2003

Travaux pratiques, séance 1

OpenSSH

Le protocole SSH permet des connexions sécurisées sur comptes distants à travers un réseau ouvert. Ses ingrédients essentiels sont l'identification mutuelle des deux parties (client/serveur) et le chiffrement du flux de données une fois la communication négociée. Il y a en fait deux protocoles SSH, le deuxième étant ammené à remplacer le premier et intégre en plus un mécanisme assurant l'intégrité du flux de données. L'implémentation et spécification initiales de SSH sont dues à Tatu Ylonen. Cette implémentation était sous licence libre jusqu'à la version 1.2.12 (implémentant le prococole 1), mais les versions 2.x.x (implémentant le protocole 2) sont commerciales.

Le développement de SSH sous licence libre à été repris par les développeurs d'OpenBSD à partir des dernières sources libres de Tatu Ylonen, et porté sous Linux. Le résultat est OpenSSH, qui implémente les deux protocoles.

1. Installation

Option 1 : .tar.gz

Pour installer `OpenSSH`, il faut d'abord installer `OpenSSL`. Les serveurs officiels sont `www.openssh.com` et `www.openssl.org`.

1. — Télécharger OpenSSH et OpenSSL depuis l'intranet :

```
http://truffe/softs/openssh/openssh-3.5p1.tar.gz
http://truffe/softs/openssl-engine-0.9.6g.tar.gz
```

Vous aurez peut-être aussi besoin de `zlib`, disponible aussi sur truffe.

2. — Désarchiver, compiler et installer OpenSSL (lire le fichier INSTALL).

3. — Désarchiver, compiler et installer OpenSSH (lire le fichier INSTALL).

```
$ ./configure --with-md5-passwords
$ make
$ make install
```

Trois paires de clés identifiant votre machine sont automatiquement créées lors de l'installation, dans le répertoire `/usr/local/etc/`.

- Une paire de clés RSA pour le protocole 1, la clé privée dans `ssh_host_key` et la clé publique dans `ssh_host_key.pub`.
- Une paire de clés RSA pour le protocole 2, dans `ssh_host_rsa_key` et `ssh_host_rsa_key.pub`.
- Une paire de clés DSS (pour le protocole 2), dans `ssh_host_dsa_key` et `ssh_host_dsa_key.pub`.

Option 2 : RPMs

On peut aussi installer à partir des RPMs de Mandrake.

```
$ rpm -i --force openssh-3.4p1-4mdk.i586.rpm \
                 openssh-askpass-3.4p1-4mdk.i586.rpm \
                 openssh-clients-3.4p1-4mdk.i586.rpm \
                 openssh-server-3.4p1-4mdk.i586.rpm
```

Il faut alors créer les clés "manuellement" par la commande `make host-key`.

Option 3 : C'est déjà installé

Vérifier seulement que vous avez des clés hôte dans le répertoire `/etc/ssh` et remercier chaleureusement l'administrateur système.

2. Configuration Hôte

1. — Décommenter (si ce n'est déjà fait) la ligne `ForwardX11 yes` dans le fichier `/etc/ssh_config` et la ligne `X11Forwarding yes` dans le fichier `/etc/sshd_config`. Dans ce dernier fichier, remplacer `PermitRootLogin yes` par `PermitRootLogin no`.

2. — Ajouter les lignes suivantes dans le fichier `/etc/rc.d/rc.local` :

```
if [ -x /usr/local/sbin/sshd ]; then
        echo -n " sshd"
        /usr/local/sbin/sshd
fi
```

pour que le serveur SSH soit activé au démarrage de la machine. Ou bien, si jamais vous avez une Mandrake, utilisez l'outil de configuration système (menu configuration des services) pour sshd soit activé au démarrage.

3. — (Eventuellement). Interroger, avec la commande `ssh-keyscan`, des machines ayant un serveur SSH, pour obtenir leurs clés publiques. Mettre dans `/etc/ssh_known_hosts` les clés obtenues. Rechercher de préférence des clés correspondant au protocole 2 (avec `-t rsa` ou `-t dsa`).

3. Configuration utilisateur

Le problème des truffes

Votre répertoire utilisateur est partagé entre les postes individuels et le serveur par NFS. Cela a pour conséquence, si on laisse les choses se faire, que les clés secrètes des différents comptes sur vos différents postes seront physiquement toutes sur truffe, ce qui serait un non-sens pour la sécurité (et en fait impossible puisqu'un fichier ne peut contenir qu'une seule clé secrète).

Pour résoudre ce problème, je propose une manipulation sordide. 1) Créer un répertoire `/home/untel` (seul l'administrateur peut le faire) qui ne sera pas partagé (contrairement à `/users/untel`). 2) Créer un répertoire de même nom sur truffe (je l'ai déjà fait). 3) Créer des répertoires `/home/untel/.ssh` sur truffe et sur votre poste (`mkdir /home/untel/.ssh`). 4) Créer un lien symbolique `/users/untel/.ssh` pointant vers `/home/untel/.ssh` par `ln -s ~/.ssh /home/untel/.ssh` (ce dernier lien n'étant à créer qu'une seule fois, mais sera actif sur tous les postes).

Configuration

1. — Générer une paire de clés (publique/privée) par la commande `ssh-keygen`, avec l'option `-t rsa` ou `-t dss` pour le protocole 2. Le résultat se retrouve dans `.ssh/id_rsa` et `.ssh/id_rsa.pub`, ou dans `.ssh/id_dsa` et `.ssh/id_dsa.pub`.

2. — Sur une machine distante (mettez-vous en binôme), créer le répertoire `~/.ssh` (s'il n'existe pas encore) et recopier la clé publique locale dans le fichier `.ssh/authorized_keys` (attention : ne pas introduire de caractères de fin de ligne au milieu de la clé).

3. — Se connecter sur la machine distante à partir de la machine locale (la première fois, la clé hôte de la machine distante est recopiée dans le fichier `.ssh/known_hosts`, sauf si vous l'avez préalablement installée manuellement dans `/usr/local/etc/ssh_known_hosts`). La commande à utiliser est `slogin` en remplacement de `rlogin`. L'option -v permet de voir ce qui se passe au cours de l'établissement de la connexion.

4. — Tunnels. Tester l'utilisation de clients X11 sur la machine distante à travers le tunnel créé par SSH.

5. — Tester aussi les commandes `scp` (en remplacement de `rcp`), `ssh` (en remplacement de `rsh`, et `sftp` (en remplacement de `ftp`). Tester aussi l'utilisation de clients X distants à travers le tunnel créé par la connexion SSH.

4. Tunnel sécurisé pour POP3

On peut ouvrir un tunnel sécurisé vers `truffe` (serveur POP3 sur le port 110) par la commande :

```
$ ssh -f -N -L 10110:truffe:110 truffe
```

On peut alors relever son courrier en utilisant le port 10110 local au lieu de truffe:110.

5. Référence

• Bonne référence à lire : la page écrite par Joël Marchand :

```
http://www.medicis.polytechnique.fr/doc/medicis/ssh.html
```