Le protocole SSH et son implémentation `openssh`

Travail demandé Vous déposerez sur Universitice votre compte-rendu de TP, au plus tard le 6 novembre 2024. Il doit répondre aux problématiques posées par les différents exercices, en explicitant votre démarche et les difficultés rencontrées. Vous pouvez y insérer des extraits de captures de trames, des scripts, des fichiers de configuration, etc.

Exercice 1 (Montage du réseau) Mettez en place un réseau contenant deux machines Linux : un serveur et un client. Vous pouvez utiliser `LXC`, `LXD`, ou des machines virtuelles. Décrivez rapidement la création du réseau, et donnez les caractéristiques de vos OS (les commandes et paquets proposées ci-après sont pour Ubuntu 22.04).

Vérifiez la connectivité de votre réseau. Vérifiez que les paquets `openssh-client` et `openssh-server` sont bien installés sur chaque machine ou installez-les (notez les versions). Créez un compte utilisateur à votre nom sur le serveur et le client.

Exercice 2 (Telnet) Installez et lancez le serveur telnet (paquet `telnetd`) sur la machine serveur (`service openbsd-inetd start`). Connectez-vous en telnet depuis le client, analysez les trames [1], montrez que l'on peut retrouver le mot de passe de l'utilisateur. Désactivez ensuite définitivement le serveur telnet.

Exercice 3 (Serveur SSH) Démarrez le serveur `sshd` : `service ssh start` et vérifiez que le port ssh (le 22) est bien ouvert avec `ss -atlp`. Quelle est votre version de OpenSSH ?

Regardez les fichiers de configuration dans le répertoire `/etc/ssh`, en particulier `sshd_config` (voir le man). Vérifiez la version du protocole ssh utilisée. Le serveur s'authentifie auprès du client à l'aide de cryptographie asymétrique. Quelles sont les clefs utilisées par votre serveur ? Affichez leur fingerprint avec la commande

```
ssh-keygen -f ssh_key.pub -l
```

Comparez ce fingerprint avec celui de vos voisins. Dans quel cas serait-il identique ? Que faudrait-il faire ?

Quels doivent être les permissions Linux sur la clef privée ?

Modifiez la ligne `PermitRootLogin` à `no`. Expliquez.

Modifiez la ligne `PrintLastLogin` à `Yes`. Expliquez.

Quelle est la différence entre les fichiers de configuration `sshd_config`, `ssh_config` et `~/.ssh/config` ?

Exercice 4 (Authentification Client par mot de passe) La suite `openssh` fournit le programme `ssh` ainsi que `scp, sftp, ssh-add, ssh-agent, ssh-copy-id, ssh-keygen, ssh-keysign, ssh-keyscan`.

Lancez une capture de trames. Connectez-vous en ssh sur le serveur. Que veut dire la phrase `The authenticity of host '...' can't be established.` ? Que doit-on faire ?

1. en utilisant par exemple `tcpdump -s0 -w` et `wireshark`

Analysez les paquets capturés, délimitez le protocole SSH-TRANS (Transport Layer Protocol, (RFC 4253 (2006)), qui assure la négociation des algorithmes, l'authentification du serveur, la mise en place d'une clef de session, puis l'intégrité et la confidentialité des données échangées. Où intervient l'authentification du serveur ?

Quels sont les algorithmes symétriques activés par défaut sur le serveur ? Vous pouvez vous aider du `man sshd_config` (fichier de configuration, donc section 5).

De même, quels sont les MAC (Message Authentication Control, rien à voir avec les adresses du même sigle) autorisés sur votre serveur ? Comment le configurer ?

Comme vu en cours, attachez la commande `strace` au processus `sshd`, connectez-vous au serveur avec mot de passe, et récupérez le mot de passe du client.

Exercice 5 (Utilisation d'une autorité SSH et d'un certificat serveur) À l'aide de l'utilitaire `ssh-keygen`, générez un couple de clefs publique/privée nommés `ca_key` et `ca_key.pub`. Générer à présent un certificat pour la clé publique du serveur (avec les options `-h`, `-n` et `-s`), ayant pour **principal** l'adresse IP ou le nom de domaine du serveur et **certifié** (ou **signé**) par la clé "ca". Renseigner ce certificat dans la configuration SSH du serveur (`HostCertificate`).

Inclure la clé publique SSH de l'autorité dans le fichier `known_hosts` du client, précédé de la mention `@cert-authority *` et supprimer d'autres éventuelles entrées. Activez alors le mode verbeux de SSH pour constater la vérification du certificat serveur. Que risque-t-on si la clé privée de l'autorité fuite ?

Exercice 6 (Redirection de ports) Installez Nginx sur le serveur, avec la configuration par défaut, à ceci près que le serveur doit être en écoute sur la boucle locale (interface lo). Celui-ci n'est donc pas accessible depuis le réseau.

Utilisez le système de redirection de port local de `ssh` (avec les options `-L` et `-N`) pour accéder au service Web (page par défaut) depuis le client.
En tant qu'administrateur, comment empêcher cette redirection (tunnelling) ? Mettre en place cette restriction et tester.

Exercice 7 (Authentification client par clef RSA) À l'aide de l'utilitaire `ssh-keygen`, générez pour le client un couple de clefs publique/privée que vous nommerez `id_test_ssh` et `id_test_ssh.pub`. Comment la clef (privée) est-elle stockée ? Copiez la clef publique sur le serveur (avec l'outil `ssh-copy-id`). Où la clef est-elle copiée par défaut ? Que se passe-t-il si l'authentification par clef RSA échoue ? Quelle est la sécurité apportée finalement ?

Précisez les configurations à effectuer pour n'autoriser pour votre utilisateur que l'authentification par clef RSA (a priori, avec uniquement les droits de l'utilisateur et non ceux de l'administrateur).

Étudiez et expliquez la configuration `StrictModes` du fichier `sshd_config` côté serveur.

Exercice 8 (Configuration détaillée) Lisez le document de l'ANSSI sur OpenSSH et affinez vos configurations. Détaillez.