SSH

Secure Socket Shell

Votre instructeur

Thierry DECKER
mail@thierry-decker.com

Secure Socket Shell

- Telnet est utilisé pour communiquer avec un système distant
- Mais Telnet n'est pas sécurisé
- Ne possède pas de mécanisme de chiffrement
- Les données transitent en clair (mot de passe compris)
- N'importe qui peut capturer les paquets transmis entre client et serveur
- Des informations critiques peuvent être exposées

- Pour parer à ce problème, SSH (Secure Socket Shell) est apparu

Secure Socket Shell

- Qu'est-ce que SSH ?
  - SSH est aussi connu sous le nom de Secure Socket Shell
  - Un protocole qui fournit un moyen sûr d'accéder à un système distant
  - SSH établit une connexion sécurisée (chiffrée) entre un client et un serveur
  - Les deux composants s’identifient mutuellement
  - Les données sont échangées au travers d’une communication chiffrée

Secure Socket Shell

- Comment fonctionne SSH ?
  - Le protocole SSH utilise
    - Le chiffrement symétrique
    - Le chiffrement asymétrique
    - Le hachage
  - Pour sécuriser le transport des informations entre client et serveur

Secure Socket Shell

- Comment fonctionne SSH ?
    - Trois étapes pour établir une connexion SSH
        - Vérification du serveur par le client
        - Création d'une clé de session pour chiffrer toute la communication
        - Authentification du client par le serveur

Secure Socket Shell

- Vérification du serveur par le client
    - Si le client accède pour la première fois au serveur, il lui est demandé de vérifier manuellement la clé publique du serveur
    - La clé vérifiée est ensuite ajoutée dans les fichiers de configuration du client (fichier ***known_hosts*** dans le répertoire ***~/.ssh***)
    - Si le client à déjà accédé au serveur, l'identité du serveur est vérifiée avec les informations précédemment enregistrées dans ***known_hosts***

Secure Socket Shell

- Création d'une clé de session
  - Après avoir vérifié le serveur, les deux participants négocient une clé de session en utilisant l'algorithme Diffie-Hellman
  - L'algorithme est conçu de façon à ce que les deux participants contribuent également à la création de la clé de session
  - La clé de session est une clé symétrique (utilisée pour chiffrer ET déchiffrer)

Secure Socket Shell

- Authentification du client par le serveur

  - L'authentification est faite en utilisant une paire de clés SSH
  - L'une est réputée publique (pour chiffrer et peut être diffusée)
  - L'autre est réputée privée (pour déchiffrer)
  - Après établissement d'un chiffrement asymétrique (première phase)
    - Le client commence par envoyer un ID (hash) pour la paire de clés avec laquelle il souhaite s'authentifier auprès du serveur
    - Le serveur vérifie cet ID avec son fichier ***authorized_keys***
    - Si une clé publique avec cet ID est trouvée, le serveur génère un nombre aléatoire et utilise la clé publique pour chiffrer ce nombre et l'envoyer
    - Si le client à la bonne clé privée, il peut déchiffrer ce message et obtenir le nombre aléatoire créé par le serveur

Secure Socket Shell

- Authentification du client par le serveur
    - Le client combine le nombre aléatoire obtenu du serveur avec la clé partagée de session et calcul le hash de cette valeur
    - Le client envoie le hash au serveur en réponse au message reçu précédemment du serveur
    - Le serveur utilise la même clé partagée et le nombre aléatoire initialement généré et calcul son propre hash
    - Si les deux hash sont identiques, cela prouve que le client est bien en possession de la clé privée et est donc authentifié

Secure Socket Shell

Merci de votre attention !