Rapport de Projet TP – SSN (Shell sur Netcat)

Étudiants : Florentin GIRARDET & Paul GENIAUX

Systèmes UNIX

Date : 12/06/2025

Table des matières

1. Objectif du projet

L'objectif du projet est de concevoir un shell distant basé sur Netcat permettant à un client d'envoyer des commandes à un serveur, qui les exécute et retourne les résultats à partir de Bash. Le projet évolue en quatre étapes : écoute simple, exécution distante, authentification, et chiffrement monoalphabétique.

3. Détail des exercices

Pour les 3 premiers exercices, le client est vraiment basique. On lance seulement la commande : `nc localhost 12345`

3.1 Exercice 1 – Serveur en écoute

Pour réaliser cet exercice, nous envoyons à la commande Netcat date et le message "Bienvenue sur le serveur".

Nous exécutons Netcat avec `nc -l -s localhost -p 12345` ce qui indique que l'on est en mode écoute et que l'on se concentre sur la machine sur le port 12345.

3.2 Exercice 2 – Exécution distante de commandes

Cette fois ci, on passe par des tunels fifo. Pour ce faire on exécute la commande `nc -l -s localhost -p 12345 < ./fifo | interpret > ./fifo.` Ce qui permet de communiquer sans arrêt entre nc et notre fonction. Celle-ci va afficher la saisie du client, si la saisi est « exit », la fonction va s'arrêter. Sinon elle va exécuter et retourner le résultat au client.

3.3 Exercice 3 – Authentification par mot de passe

Le serveur va récupérer dans le fichier config le mot de passe utilisateur, il va ensuite le comparer avec la saisie client et si la saisi est bonne, il va réaliser l'exercice 2.

3.4 Exercice 4 – Chiffrement monoalphabétique

Dans cet exercice nous avons 2 fichiers, un pour le client et l'autre pour le serveur.

Coté Serveur :

Nous récupérons déjà l'input du client qui contient le mot de passe et la clé du client ainsi que le mot de passe contenu le fichier config.

Si les mots de passe sont identiques et que la clé est valide, nous passons à l'étape suivante : On récupère l'entrée client et on la déchiffre puis on l'exécute et on la renvoie chiffrés.

Pour chiffrer déchiffrer le message, on utilise le chiffrement monoalphabétique avec :
encode() { echo "$1" | tr 'abcdefghijklmnopqrstuvwxyz' "$cle"}

Et inversement pour decoder.

Cette fois si, on utilise encore des fifo mais on les as encoder en temps que sortie de bash pour rendre le code plus lisible :

```
exec 3<> fifo_client_to_server.fifo

exec 4<> fifo_server_to_client.fifo

nc -l -p 12345 >&3  <&4 | interpret
```

Coté Client :

Dans un premier temps, nous récupérons le mot de passe de l'utilisateur ainsi que la clé de chiffrement en nous assurant qu'elle soit valable. Dès que nous somme bien connecter au serveur on fait l'interface entre le serveur et le client en récupérant ses prompt et en les envoyant chiffré au serveur avec les mêmes méthodes que celui-ci. Puis on récupère les réponses du serveur, on les déchiffre et on les affiche.

4. Lancement et utilisation des scripts

Pour lancer les scripts, on commence par s'assurer qu'ils sont exécutables, si ce n'est pas le cas, on réalise : chmod +x [nom du fichier]. On le fait pour les fichiers ./Client.sh et/ou ./Serveur.sh. Il ne reste plus qu'as suivre les indications du script.

5. Annexes

Le code : TP-PROJET-SSN