





Systèmes UNIX. Projet TP.

SSN - Shell sur Netcat

Mode du travail préférée : binômes.

L'objectif de ce projet est de créer un système d'exécution de commandes à distance. La partie mise en réseau peut être faite avec netcat vu en cours et en TD. Voici un exemple d'utilisation de netcat : <https://kirgizov.link/teaching/esirem/unix/examples/nc-server.sh>

Chaque groupe présentera au fur et à mesure ses réalisations pendant les TP. À la fin de la dernière séance de TP vous devez donner le code à l'intervenant de TP. Votre code doit être accompagné d'un court rapport (2-4 pages) expliquant comment exécuter et utiliser votre logiciel et décrivant l'architecture et principes de son fonctionnement. La note finale évaluera votre travail en classe, votre code et votre rapport, elle sera sur une échelle de 0 à 20.

-  **EXERCICE 1.** (3 points). Concevoir un script en bash qui écoutera le port TCP numéro 12345, et transmettra au client la date de connexion et un message de bienvenue. Une fois que le client a mis fin à la connexion, le serveur ne doit pas s'arrêter complètement, il doit écouter à nouveau le port 12345.
-  **EXERCICE 2.** (5 points). Chaque ligne saisie côté client doit être transmise au serveur, exécuté là-bas, et le résultat doit être renvoyé au client.
-  **EXERCICE 3.** (5 points). Le serveur doit demander un mot de passe et vérifier qu'il est correct avant d'autoriser l'exécution des commandes. Côté serveur, le mot de passe doit être stocké dans un fichier de configuration.
-  **EXERCICE 4.** (7 points). La communication client-serveur doit être chiffrée par la substitution monoalphabétique. En plus du mot de passe, le système demande à l'utilisateur du côté client la clé de chiffrement. Côté serveur, la clé doit être stockée dans le fichier de configuration.

Voici le schéma de transfert de données pour cet exercice :

1. L'utilisateur rentre la commande côté client.
2. Le client chiffre la commande et transmet la commande chiffrée au serveur.
3. Le serveur déchiffre la commande, exécute la commande, chiffre le résultat, et renvoi le résultat chiffré au client.
4. Le client déchiffre le résultat et l'affiche à l'utilisateur.