REPUBLIQUE DU BENIN

******

MINISTERE DE L'EDUCATION SUPERIEURE ET DE LA

RECHERCHE SUPERIEURE

******

UNIVERSITE D'ABOMEY-CALAVI

******

ECOLE POLYTEHNIQUE D'ABOMEY-CALAVI

******

DEPARTMENT DE GENIE INFORMATIQUE ET TELECOMMUNICATIONS

******

***COURS:* Système d'exploitation (OS)**

Connexion à distance (réseau local) via SSH

***MEMBRES DU GROUPE 2*:**

AHOKOU Melvine

GBABLI MATHIEU Nahine

GNACADJA Jean de Dieu

***SOUS LA SUPERVISION DU*:**

Dr SOGBOHOSSOU Médésu

***GIT3 ANNEE ACADEMIQUE*: 2025-2026**

PLAN DE REDACTION

I. **INTRODUCTION**
II. **PREREQUIS**
III. **PRESENTATION ET FONCTIONNEMENT DE SSH**
    1.Définition
    2.Objectifs
    3.Fonctionnement
    4.Avantages
IV. **INSTALLATION DU SERVEUR SSH**
V. **CONNEXION LINUX VERS LINUX**
VI. **CONNEXION WINDOWS VERS LINUX**
VII. **TRANSFERT DE FICHIERS AVEC SCP**
VIII. **DEPANNAGE**
IX. **CONCLUSION**

I-INTRODUCTION

Cet exposé présente la mise en œuvre de connexions SSH

(Secure Shell) dans un réseau local.

SSH est un protocole réseau qui permet de:

- Se connecter à un ordinateur distant de manière sécurisée ;
- Exécuter des commandes à distance ;
- Transférer des fichiers de façon cryptée.

Objectifs de ce document:

- Installer un serveur SSH sous Linux ;
- Établir des connexions depuis Linux et Windows ;
- Transférer des fichiers via SCP.

Contexte:

Dans un environnement professionnel, SSH est essentiel pour

administrer des serveurs à distance sans compromettre la sécurité.

II-PREREQUIS

Pour y parvenir, nous aurons besoin d'un materiel adéquat et s'assurer que ***tous les postes soient connectés au même réseau.***

Matériel nécessaire:

- 3 ordinateurs en réseau local ;
- 2 postes Linux (Ubuntu ou Debian recommandés) ;
- 1 poste Windows (10 ou 11).

Logiciels requis:

- OpenSSH Server (sur le serveur Linux) ;
- OpenSSH Client (préinstallé sur Linux) ;
- PuTTY (sur Windows) **:** ***https://www.putty.org/***

Configuration réseau:

- Tous les postes sur le même sous-réseau (ex : 192.168.1.x) ;
- Pare-feu configuré pour autoriser le port 22.

III-PRESENTATION ET FONCTIONNEMENT DE SSH

1.Définition

SSH (Secure Shell) est un protocole de communication sécurisé permettant d'accéder à distance à un autre ordinateur (serveur ou poste). Il remplace les anciens protocoles non sécurisés comme ***Telnet ou rlogin***. Il fonctionne sur le port 22 (par défaut).

2.Objectifs

- Authentifier l'utilisateur ;
- Chiffrer la communication ;
- Garantir l'intégrité des données échangées.

3.Fonctionnement

Le client (ordinateur qui se connecte) établit une connexion SSH avec le serveur. Un échange de clés (clé publique et clé privée) est réalisé pour sécuriser la session. L'utilisateur s'authentifie (mot de passe ou clé privée). Une session distante est ouverte, permettant d'exécuter des commandes comme si on était sur le poste distant.

4.Avantages

- Sécurité (chiffrement fort).
- Simplicité d'utilisation (ligne de commande).
- Possibilité de transfert de fichiers (SCP, SFTP).
- Administration distante efficace.

IV-INSTALLATION DU SERVEUR SSH

Sur le poste Linux qui servira de serveur:

✓ ***Étape 1*** : ***Mettre à jour le système***

Commande: *sudo apt update*

Résultat attendu: Liste des paquets mise à jour.

✓ ***Étape 2 : Installer OpenSSH Server***

Commande: *sudo apt install openssh-server*

Résultat attendu: Installation réussie.

✓ ***Étape 3 : Vérifier l'installation***

Commande: *ssh –V*

Résultat attendu: OpenSSH_8.x, OpenSSL 1.x.

✓ ***Étape 4 : Démarrer le service SSH***

Commande: *sudo systemctl start ssh*

Résultat attendu: Service démarré.

✓ ***Étape 5 : Activer SSH au démarrage***

Commande: *sudo systemctl enable ssh*

Résultat attendu: Service activé automatiquement.

✓ ***Étape 6 : Vérifier le statut du service***

Commande: *sudo systemctl status ssh*

Résultat attendu: Active: active (running).

✓ ***Étape 7 : Configurer le pare-feu (si actif)***

Commande: *sudo ufw allow ssh*

Commande: *sudo ufw enable*

✓ ***Étape 8 : Obtenir l'adresse IP du serveur***

Commande: *ip a ou hostname –I*

Résultat: Notez l'adresse IP (exemple: 192.168.1.100).

V-CONNEXION LINUX VERS LINUX

Sur le poste Linux client:

✓ ***Étape 1 : Vérifier que le client SSH est installé***

Commande: *ssh -V*

Résultat attendu: *Version d'OpenSSH affichée*

✓ ***Étape 2 : Se connecter au serveur***

Commande: *ssh utilisateur@adresse_ip_serveur*

Exemple: *ssh john@192.168.1.100*

✓ ***Étape 3 : Accepter l'empreinte du serveur (première fois)***

Question affichée:

"Are you sure you want to continue connecting (yes/no)?"

Réponse: yes

✓ ***Étape 4 : Saisir le mot de passe***

Le système demande *le mot de passe de l'utilisateur* distant

✓ ***Étape 5 : Vérifier la connexion***

Une fois connecté, vous voyez l'invite du serveur distant

Commande test: *pwd*

Commande test: *ls*

- ✓ ***Étape 6 : Se déconnecter***

Commande: *exit* ou *appuyez sur Ctrl+D*

VI-CONNEXION WINDOWS VERS LINUX

Sur le poste Windows:

- ✓ ***Étape 1 : Télécharger PuTTY***

Site: *https://www.putty.org/*

Télécharger: *putty.exe (version 64-bit)*

- ✓ ***Étape 2 : Installer PuTTY***

Double-cliquer sur le *fichier .msi*

Suivre l'assistant d'installation

- ✓ ***Étape 3 : Lancer PuTTY***

Ouvrir le programme PuTTY

- ✓ ***Étape 4 : Configurer la connexion***

Dans *"Host Name (or IP address)":* 192.168.1.100

Port: 22

*Connection type****:*** SSH

✓ ***Étape 5 : Enregistrer la session (optionnel)***

Saved Sessions: "Serveur Linux"

Cliquer sur *"Save"*

✓ ***Étape 6 : Ouvrir la connexion***

Cliquer sur *"Open"*

✓ ***Étape 7 : Accepter la clé du serveur (première fois)***

Cliquer sur *"Accept"*

✓ ***Étape 8 : Se connecter***

login as: [nom_utilisateur]

password: [mot_de_passe]

✓ ***Étape 9 : Utiliser le terminal distant***

Vous pouvez maintenant exécuter des commandes

✓ ***Étape 10 : Fermer la session***

Taper: *exit* ou *fermer la fenêtre PuTTY*

VII-TRANSFERT DE FICHIERS AVEC SCP

SCP (Secure Copy Protocol) permet de copier des fichiers via SSH de manière sécurisée.

- ***Copier un fichier local vers le serveur distant***

Syntaxe: *scp fichier_local utilisateur@serveur:/chemin/*

Exemple pratique:

✓ ***Étape 1 : Créer un fichier de test***

Commande: *echo "Test SSH" > test.txt*

✓ ***Étape 2 : Copier vers le serveur***

Commande: *scp test.txt john@192.168.1.100:/home/john/*

Résultat: test.txt 100% |*| 9 bytes

✓ ***Étape 3 : Vérifier sur le serveur***

Commande: ssh john@192.168.1.100

Commande: *ls -l test.txt*

- ***Copier un fichier du serveur vers la machine locale***

Syntaxe: *scp utilisateur@serveur:/chemin/fichier ./*

Exemple:

✓ ***Étape 1 : Copier depuis le serveur***

Commande: *scp john@192.168.1.100:/home/john/test.txt ./test_retour.txt*

✓ ***Étape 2 : Vérifier localement***

Commande: *ls -l test_retour.txt*

- ***Copier un dossier complet (récursif)***

Syntaxe: *scp -r dossier/ utilisateur@serveur:/chemin/*

Exemple:

✓ ***Étape 1 : Créer un dossier de test***

Commande: *mkdir mon dossier*

Commande: *touch mon dossier/fichier1.txt*

Commande: *touch mon dossier/fichier2.txt*

- ✓ ***Étape 2 : Copier le dossier***

Commande: *scp -r mon_dossier/ john@192.168.1.100:/home/john/*

- ✓ ***Étape 3 : Vérifier***

Commande: *ssh john@192.168.1.100*

Commande: *ls -l mon_dossier/*

VII-DEPANNAGE

Problèmes courants et solutions:

PROBLEMES	***SOLUTIONS***
Connection refused	Vérifier que SSH est démarrer: *sudo systemctl start ssh*
Permission denied	-Vérifier le nom d'utilisateur - Vérifier le mot de passe - Vérifier les droits utilisateur

Host not found (Unable to connect)	- Vérifier l'adresse IP: *ping IP* - Vérifier la connexion réseau
Timeout	- Vérifier le pare-feu: *sudo ufw allow 22/tcp* - Vérifier que les postes sont sur le même réseau
Port 22 déjà utilisé	- Vérifier les processus: *sudo netstat -tulpn | grep :22*

Commandes de diagnostic utiles:

- Vérifier le statut SSH: *sudo systemctl status ssh;*

- Tester la connexion réseau: *ping adresse_ip;*

- Afficher les logs SSH: *sudo tail -f /var/log/auth.log;*

- Redémarrer le service SSH: *sudo systemctl restart ssh.*

CONCLUSION

Ce document a présenté la mise en œuvre de SSH pour la connexion à distance dans un réseau local.

✓ ***Récapitulatif des étapes réalisées :***

- Installation du serveur OpenSSH sur Linux;
- Connexion SSH depuis un client Linux;
- Connexion SSH depuis Windows avec PuTTY;
- Transfert de fichiers sécurisé avec SCP.

✓ ***Avantages de SSH :***

- Sécurité: Toutes les communications sont cryptées;
- Universalité: Fonctionne sur tous les systèmes (Linux, Windows, Mac);
- Polyvalence: Connexion, transfert de fichiers, tunneling;
- Performance: Consommation réseau minimale.

✓ ***Limites de SSH :***

- Nécessite une configuration initiale;
- Dépend de la qualité du réseau;

- Authentification par mot de passe peut être vulnérable (solution: utiliser des clés SSH).

✓ ***Applications pratiques :***

- Administration de serveurs à distance;
- Développement sur machines distantes;
- Transfert sécurisé de fichiers sensibles;
- Automatisation de tâches via scripts.

✓ ***Pour aller plus loin :***

- Authentification par clés SSH (ssh-keygen);
- Tunneling SSH (port forwarding);
- Configuration avancée (/etc/ssh/sshd_config);
- Utilisation de SSH avec des scripts Bash.