

Rapport de projet



Module	administration système
Nom du projet:	SSH applications
Formateur :	Mme Soukaina Mihi
Date :	11 avril 2025

GROUPE

Mohsine Adam

Mermarh Aymane

Bachar Youssef

Abdellatif harakat

Yahya Fahmi

Introduction

Ce projet a pour objectif de présenter et de mettre en pratique les différentes fonctionnalités avancées du protocole SSH (*Secure Shell*), un outil essentiel dans le monde de l'administration système et du réseau.

L'utilisation de SSH ne se limite pas uniquement à la connexion sécurisée à distance entre un client et un serveur, mais propose également plusieurs fonctionnalités très puissantes comme :

- le tunneling SSH,
- le transfert graphique via X11 Forwarding,
- l'authentification par clés SSH,
- la configuration avancée de SSH,
- et les bonnes pratiques de sécurisation d'un serveur SSH.

Module	administration système
Nom du projet :	SSH applications
Formateur :	Mme Soukaina Mihi
Date :	11 avril 2025

GROUPE
Mohsine Adam
Mermarh Aymane
Bachar Youssef
Abdellatif harakat
Yahya Fahmi

Objectifs du projet

L'objectif principal de ce projet est de découvrir, comprendre et maîtriser les différentes fonctionnalités avancées du protocole SSH (*Secure Shell*), largement utilisé dans le monde de l'administration système.

À travers des démonstrations pratiques et des cas réels d'utilisation, ce projet a pour but de :

Objectifs généraux :

- Apprendre à utiliser SSH pour se connecter à distance de manière sécurisée.
- Explorer les différentes applications avancées de SSH.
- Mettre en pratique les fonctionnalités utiles d'un administrateur système.
- Comprendre les risques liés à SSH et les bonnes pratiques de sécurisation.

Objectifs techniques :

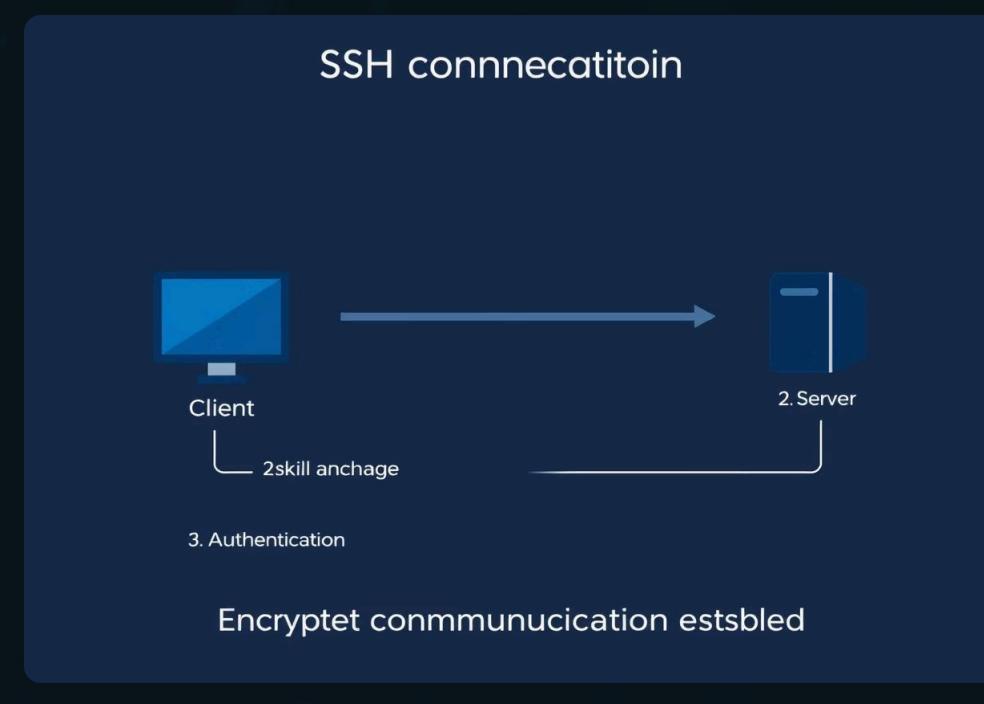
- Mettre en place des connexions SSH simples.
- Réaliser un tunneling SSH (local, distant et dynamique).
- Utiliser le X11 Forwarding pour afficher des interfaces graphiques d'un serveur distant.
- Gérer l'authentification par clés SSH.
- Configurer et personnaliser les fichiers de configuration SSH.
- Sécuriser un serveur SSH contre les attaques (fail2ban, changement de port, désactivation du root, etc.).

SSH : Définition et architecture

Définition: SSH (Secure Shell) est un protocole réseau sécurisé qui permet d'établir une connexion chiffrée entre un client et un serveur (établir une connexion avec une machine dans le même réseau ou une machine distante).

Côtes Sécurité de SSH : SSH offre une authentification sécurisée et protège l'intégrité et la confidentialité des données échangées grâce au chiffrement.

Protocoles avant SSH: il existent des Protocoles non sécurisé, exemple: Talent, Rsh.



Processus de connexion SSH

Architecture du protocole :

1. Protocole de transport (SSH-TRANS) :

- Assure l'authentification du serveur.
- Établit le chiffrement et la protection d'intégrité.
- Fonctionne généralement sur le port TCP 22.

2. Protocole d'authentification (SSH-AUTH) :

- Vérifie l'identité de l'utilisateur auprès du serveur.
- Supporte plusieurs méthodes d'authentification :
 - Par mot de passe.
 - Par clé publique/privée.
 - Par hôte.

3. Protocole de connexion (SSH-CONN) :

- Multiplex le tunnel chiffré en plusieurs canaux logiques.
- Gère les requêtes de service comme les sessions de terminal, le transfert de port et l'exécution de commandes à distance.

Protocole	Port par défaut	Chiffrement	Authentification	Cas d'usage typique
SSH	22	Oui	Multiple	Administration système, transfert de fichiers sécurisé
Telnet	23	Non	Mot de passe	Accès terminal (obsolète)
FTP	21	Non	Mot de passe	Transfert de fichiers (non sécurisé)
SFTP	22 (via SSH)	Oui	Multiple via SSH	Transfert de fichiers sécurisé
HTTPS	443	Oui	Certificats	Navigation web sécurisée

OpenSSH et ses outils

Définition: OpenSSH est une suite logicielle libre qui implémente le protocole SSH pour établir des connexions sécurisées. Elle permet, entre autres, l'administration à distance de systèmes, le transfert de fichiers sécurisé et la mise en place de tunnels de communication chiffrés.

Outils SSH:

SSH

Client permettant la connexion à distance.

sshd

Démon serveur qui accepte les connexions des clients.

ssh-keygen

Utilitaire de génération et de gestion de clés.

sftp et sftp-server

Client et serveur pour le transfert de fichiers sécurisé.

Caractéristiques principales:

- **Portabilité:** Disponible sur différents systèmes (Linux, MacOs, Windows)
- **Mécanismes de chiffrement:** Supporte plusieurs algorithmes de chiffrement, clés et de méthodes d'authentification
- **flexibilité:** Permet l'ajout de clés publiques, la configuration fine du chiffrement et le tunnelage de connexions.

Configuration d'OpenSSH

La configuration d'OpenSSH s'effectue via deux fichiers principaux :

- `/etc/ssh/sshd_config` : Configuration du serveur SSH
- `/etc/ssh/ssh_config` : Configuration par défaut du client SSH

Les paramètres de configuration importants incluent :

- Ports d'écoute
- Méthodes d'authentification autorisées
- Algorithmes de chiffrement et d'échange de clés
- Restrictions d'accès par utilisateur ou groupe.

Bonnes pratiques pour l'utilisation de OpenSSH

OpenSSH est un outil incontournable pour établir des connexions sécurisées entre machines distantes. Afin de garantir une sécurité optimale, il est essentiel de suivre certaines bonnes pratiques. Cette partie présente les recommandations clés pour utiliser OpenSSH de manière sécurisée sur Ubuntu.

Cryptographie

La sécurité d'OpenSSL repose principalement sur la cryptographie. Afin de garantir des connexions sécurisées, il est impératif d'utiliser des algorithmes modernes et robustes. Sur Ubuntu, on peut configurer les algorithmes de chiffrement utilisés en modifiant le fichier `/etc/ssh/sshd_config`.

Il est fortement recommandé de **désactiver les anciens algorithmes** comme **arcfour**, **3des** ou **aes128-cbc**, qui présentent aujourd'hui des vulnérabilités connues.

Authentification

L'une des meilleures pratiques est **d'éviter l'authentification par mot de passe**, car elle est vulnérable aux attaques par force brute. On recommande plutôt d'utiliser l'authentification par clé publique, beaucoup plus sécurisée.

Pour un niveau de sécurité encore plus élevé, l'intégration d'une **authentification à deux facteurs** (2FA) est possible à l'aide de Google Authenticator.

Génération de Clés

Les clés SSH doivent être générées avec des algorithmes modernes et des tailles adaptées. Le plus recommandé actuellement est ED25519, qui est à la fois rapide, sécurisé et génère des clés courtes.

Si on préfère RSA pour des raisons de compatibilité, il est conseillé d'utiliser une taille de **4096 bits**.

Qualité du Générateur de Nombres Aléatoires

La sécurité d'une clé dépend fortement de la qualité du générateur de nombres aléatoires utilisé. Sur des environnements virtuels ou légers, il est recommandé d'ajouter un générateur d'entropie comme haveged.

Renforcement du système et contrôle d'accès

Renforcement à la compilation

Le service **SSHD** s'exécute souvent avec les privilèges de l'utilisateur **root**, car il nécessite des droits élevés pour changer d'utilisateur après l'authentification.

Il est donc **indispensable de renforcer son code** afin de **retarder ou empêcher toute compromission potentielle**.

Une première étape dans le renforcement du service sshd consiste à utiliser des options de compilation appropriées.

Séparation des privilèges

La séparation des privilèges est une technique de sécurité qui consiste à limiter l'impact potentiel d'une faille en appliquant le principe du moindre privilège.

Depuis la version 5.9 d'**OpenSSH**, cette séparation des privilèges a été renforcée grâce à l'intégration de mécanismes d'isolation plus avancés, tels que **SECCOMP** sur **Linux** et **systrace** sur **OpenBSD**.

Chroot et SFTP

Le service **sshd** fournit également un serveur **SFTP** utilisé pour le téléchargement ou l'envoi de fichiers. Ce service ne nécessite aucun accès **shell**.

Les utilisateurs ou groupes dédiés au service SFTP doivent être "**chrootés**" dans une partition dédiée du système via l'option **ChrootDirectory**.

Authentification et contrôle d'accès

Plusieurs mécanismes peuvent être utilisés pour authentifier les différents utilisateurs. L'authentification doit être réalisée en utilisant, par ordre de préférence :

- **Cryptographie asymétrique avec ECDSA**
- **Cryptographie asymétrique avec RSA**
- **Cryptographie symétrique (tickets Kerberos via GSSAPI)**
- **Modules d'authentification** qui n'exposent ni le mot de passe de l'utilisateur ni son empreinte
- **Vérification du mot de passe** à l'aide d'une base de données ou d'un annuaire

Note : Dans tous les cas, l'authentification par mot de passe ne doit pas être utilisée pour les utilisateurs disposant de privilèges élevés.

Tests pratiques et conclusion

Cette section présente une série de tests pratiques effectués dans un environnement de virtualisation à l'aide de **deux machines Ubuntu**, l'une installée sur **VirtualBox** et l'autre sur **VMware**.

Installation et activation du serveur SSH

Objectif : Installer le serveur SSH sur la machine cible VM2 (celle qui recevra les connexions SSH).

Commandes :

```
sudo apt install  
openssh-server  
sudo systemctl enable  
ssh  
sudo systemctl start ssh
```

Ce projet nous a permis d'explorer en profondeur le protocole SSH et ses nombreuses applications dans l'administration système. Nous avons pu mettre en pratique les différentes fonctionnalités avancées de SSH et comprendre l'importance des bonnes pratiques de sécurité pour protéger les connexions à distance.

- Installation et activation du serveur SSH

Objectif : Installer le serveur SSH sur la machine cible VM2 (celle qui recevra les connexions SSH).

```
youssef@youssef-VirtualBox:~$ sudo apt install openssh-server  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :  
liblomm17t64 python3-netifaces  
Veuillez utiliser « sudo apt autoremove » pour les supprimer.  
Les paquets supplémentaires suivants seront installés :  
openssh-client openssh-sftp-server  
Paquets suggérés :  
keychain libpam-ssh monkeysphere ssh-askpass molly-guard  
Les paquets suivants seront mis à jour :  
  openssh-client openssh-server openssh-sftp-server  
3 mis à jour, 0 nouvellement installés, 0 à enlever et 115 non mis à jour.  
Il est nécessaire de prendre 1 452 ko dans les archives.  
Après cette opération, 7 168 o d'espace disque seront libérés.  
Souhaitez-vous continuer ? [O/n]  
Réception de :1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-sftp-server amd64 1:9.6p1-3ubuntu13.9 [37,3 kB]  
Réception de :2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-server amd64 1:9.6p1-3ubuntu13.9 [509 kB]  
Réception de :3 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-client amd64 1:9.6p1-3ubuntu13.9 [905 kB]  
1 452 ko réceptionnés en 3s (491 ko/s)  
Préconfiguration des paquets...  
(Lecture de la base de données... 200297 fichiers et répertoires déjà installés.)  
Préparation du dépaquetage de .../openssh-sftp-server_1%3a9.6p1-3ubuntu13.9_amd64.deb .  
...  
Dépaquetage de openssh-sftp-server (1:9.6p1-3ubuntu13.9) sur (1:9.6p1-3ubuntu13.8) ...  
Préparation du dépaquetage de .../openssh-server_1%3a9.6p1-3ubuntu13.9_amd64.deb ...  
Dépaquetage de openssh-server (1:9.6p1-3ubuntu13.9) sur (1:9.6p1-3ubuntu13.8) ...  
Préparation du dépaquetage de .../openssh-client_1%3a9.6p1-3ubuntu13.9_amd64.deb ...  
Dépaquetage de openssh-client (1:9.6p1-3ubuntu13.9) sur (1:9.6p1-3ubuntu13.8) ...  
Paramétrage de openssh-client (1:9.6p1-3ubuntu13.9) ...  
Paramétrage de openssh-sftp-server (1:9.6p1-3ubuntu13.9) ...  
Paramétrage de openssh-server (1:9.6p1-3ubuntu13.9) ...  
Traitement des actions différences (« triggers ») pour man-db (2.12.0-4build2) ...  
Traitement des actions différences (« triggers ») pour ufw (0.36.2-6) ...  
Règles mises à jour pour le profil « OpenSSH »  
youssef@youssef-VirtualBox:~$
```

```
youssef@youssef-VirtualBox:~$ sudo systemctl enable ssh  
Synchronizing state of ssh.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.  
Executing: /usr/lib/systemd/systemd-sysv-install enable ssh  
Created symlink /etc/systemd/system/sshd.service → /usr/lib/systemd/system/ssh.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/ssh.service → /usr/lib/systemd/system/ssh.service.  
youssef@youssef-VirtualBox:~$
```

- Test de la connexion SSH

Objectif : Vérifier que la machine cliente (VM1) peut se connecter via SSH à la machine cible(VM2).

```
youssef@srvubunto:~$ ssh youssef@192.168.1.7  
youssef@192.168.1.7's password:  
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.11.0-19-generic x86_64)  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/pro  
  
La maintenance de sécurité étendue pour Applications n'est pas activée.  
115 mises à jour peuvent être appliquées immédiatement.  
64 de ces mises à jour sont des mises à jour de sécurité.  
Pour afficher ces mises à jour supplémentaires, exécuter : apt list --upgradable  
  
Activez ESM Apps pour recevoir des futures mises à jour de sécurité supplémentaires.  
Visitez https://ubuntu.com/esm ou exécutez : sudo pro status  
Last login: Mon Mar 10 13:25:37 2025 from fe80::419:8c1:d36:6bfb%enp0s3  
youssef@youssef-VirtualBox:~$
```

- Test de l'authentification par clé

Objectif : Mettre en place l'authentification SSH sans mot de passe via des clés RSA.

a. Sur machine cliente

```
youssef@srvubunto:~$ ssh-keygen  
Generating public/private ed25519 key pair.  
Enter file in which to save the key (/home/youssef/.ssh/id_ed25519):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/youssef/.ssh/id_ed25519  
Your public key has been saved in /home/youssef/.ssh/id_ed25519.pub  
The key's randomart image is:  
+--[ED25519 256]--+  
|... o.  
|=+ o+.  
|&o+ +E  
%@.*..=  
B=B.+. oS  
=o.= ..  
|o . *.  
| ...+  
| ... .  
+---[SHA256]-----+  
youssef@srvubunto:~$
```