TCP

```
MARK RODERICK
```

Le protocole TCP (*Transmission Control Protocol*) est le pilier de la couche Transport (Couche 4 du modèle OSI). Si IP s'occupe de trouver le chemin pour les paquets, TCP s'assure que les données arrivent entières, dans l'ordre et sans erreurs.

---

1. Les 4 Piliers de TCP

Contrairement à son cousin UDP (qui envoie les paquets en espérant qu'ils arrivent), TCP est défini par quatre caractéristiques majeures :

- **Orienté Connexion :** Avant d'envoyer la moindre donnée, le client et le serveur doivent "se mettre d'accord" (le fameux *Handshake*).
- **Fiabilité (Reliability) :** Chaque paquet envoyé doit être acquitté (confirmé) par le destinataire. Si un paquet est perdu, TCP le renvoie automatiquement.
- **Séquencement :** TCP numérote les paquets. Si les paquets arrivent dans le désordre, le destinataire les remet à l'endroit grâce aux numéros de séquence.
- **Contrôle de Flux et de Congestion :** TCP ralentit la cadence si le destinataire est saturé ou si le réseau est encombré.

---

2. L'Établissement de Connexion (3-Way Handshake)

C'est la base absolue. Pour ouvrir une session, TCP utilise un échange en trois étapes :

1. **SYN (Synchronize) :** Le client dit "Je veux établir une connexion, voici mon numéro de séquence initial".
2. **SYN-ACK :** Le serveur répond "J'ai reçu ta demande, voici mon numéro, et j'accepte la tienne".
3. **ACK (Acknowledge) :** Le client répond "C'est noté, on commence !".

---

3. Anatomie du Segment TCP (L'En-tête)

Un segment TCP ne contient pas que des données, il contient des méta-données cruciales pour la gestion du transport :

Champ	Rôle
Ports (Source & Dest)	Identifient l'application (ex: 80 pour HTTP, 443 pour HTTPS).
Sequence Number	Position du paquet dans le flux de données.
Acknowledgment Number	Prochain numéro de séquence attendu par le récepteur.
Flags (Drapeaux)	Indiquent l'état du paquet (SYN, ACK, FIN, RST, PSH).
Window Size	Quantité de données que le récepteur peut accepter sans renvoyer d'ACK.
Checksum	Vérifie que les données n'ont pas été corrompues pendant le voyage.

---

4. La Fin de Connexion (4-Way Handshake)

Pour fermer proprement une connexion, il faut un échange en quatre étapes car TCP est "Full-Duplex" (chaque côté doit fermer sa direction) :

1. Client → **FIN**
2. Serveur → **ACK**
3. Serveur → **FIN**
4. Client → **ACK**

---

5. TCP vs UDP : Lequel choisir ?

Caractéristique	TCP	UDP
Mode	Fiable / Lourd	Rapide / Léger
Vérification	Oui (Acquittements)	Non (Best effort)
Ordre	Garanti	Aléatoire
Usage type	Web (HTTP), Email (SMTP), Fichiers (FTP)	Streaming Vidéo, Gaming, DNS, VoIP

🚀 Les concepts "Système" à connaître

En tant qu'administrateur système, tu seras souvent confronté à ces états TCP via la commande `netstat` ou `ss` :

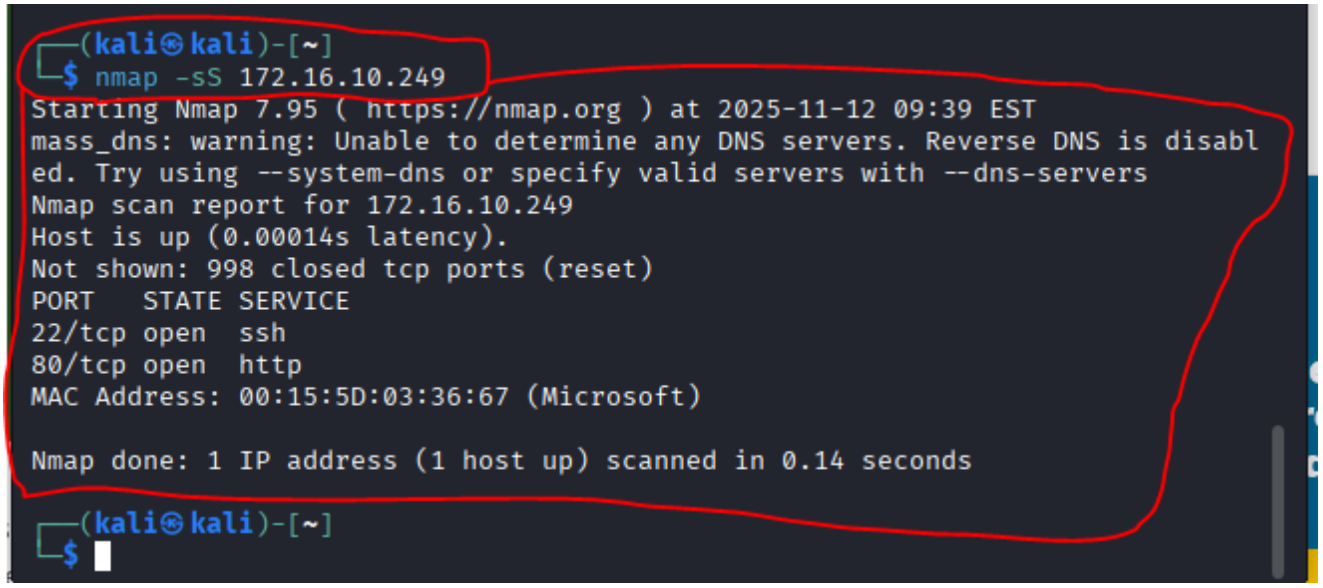
- **LISTEN :** Le serveur attend une connexion (ex: ton serveur web attend un client).
- **ESTABLISHED :** La connexion est active, les données circulent.
- **TIME_WAIT :** La connexion est fermée, mais le système garde le port réservé quelques secondes pour s'assurer qu'aucun paquet retardataire ne traîne sur le réseau.
- **CLOSE_WAIT :** Le correspondant a fermé la connexion, mais ton application locale ne l'a pas encore fait (souvent signe d'un bug applicatif).

> **Le saviez-vous ?** On appelle TCP le protocole "polis" du web. Il demande la permission avant de parler, s'excuse s'il va trop vite, et répète s'il n'est pas sûr d'avoir été entendu.

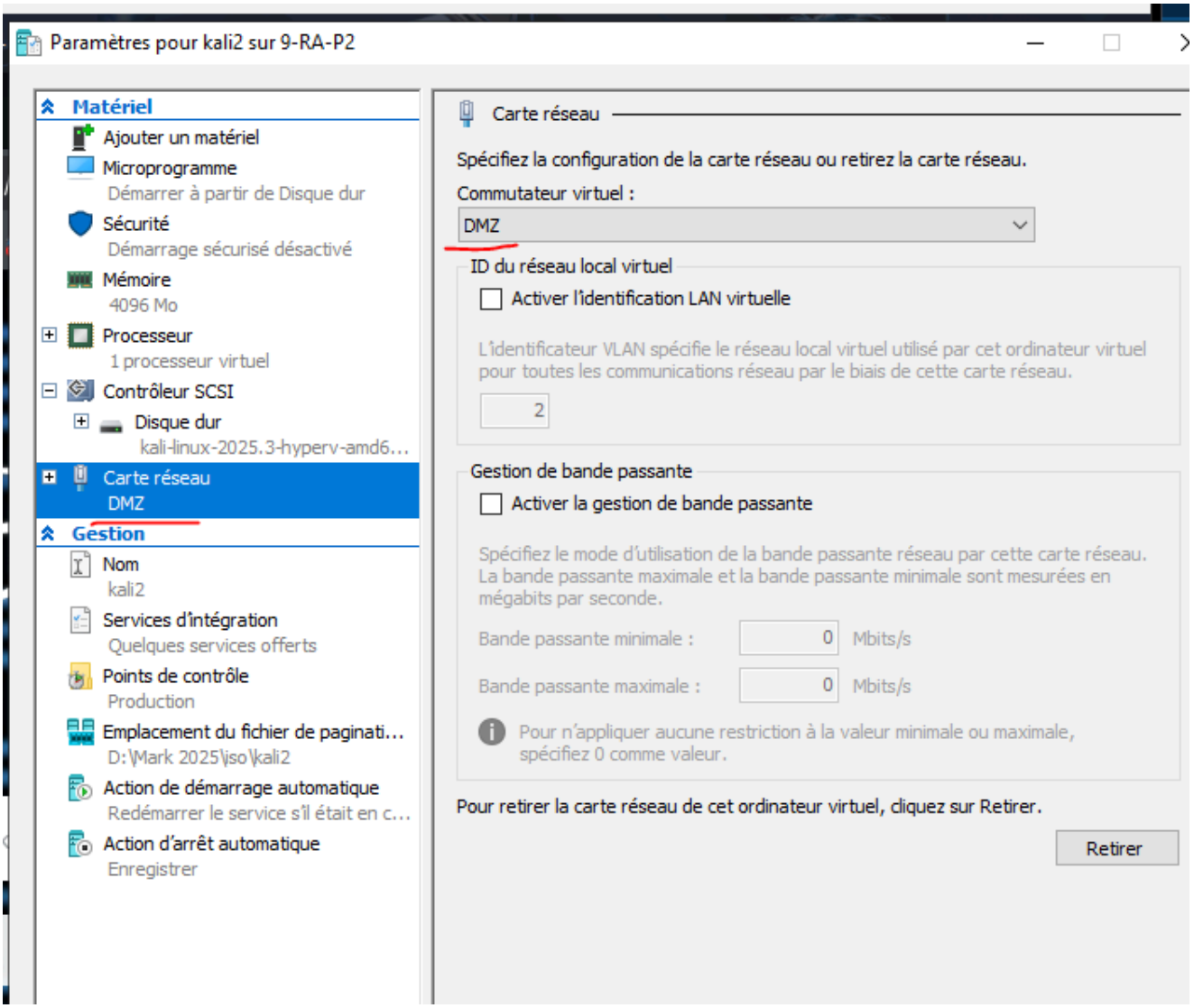
Souhaites-tu approfondir un mécanisme spécifique, comme la **Fenêtre Glissante** (Sliding Window) ou le calcul du **RTT** (Round Trip Time) ?

UDP = messages
TCP = circuits
mais les deux font des paquetsl

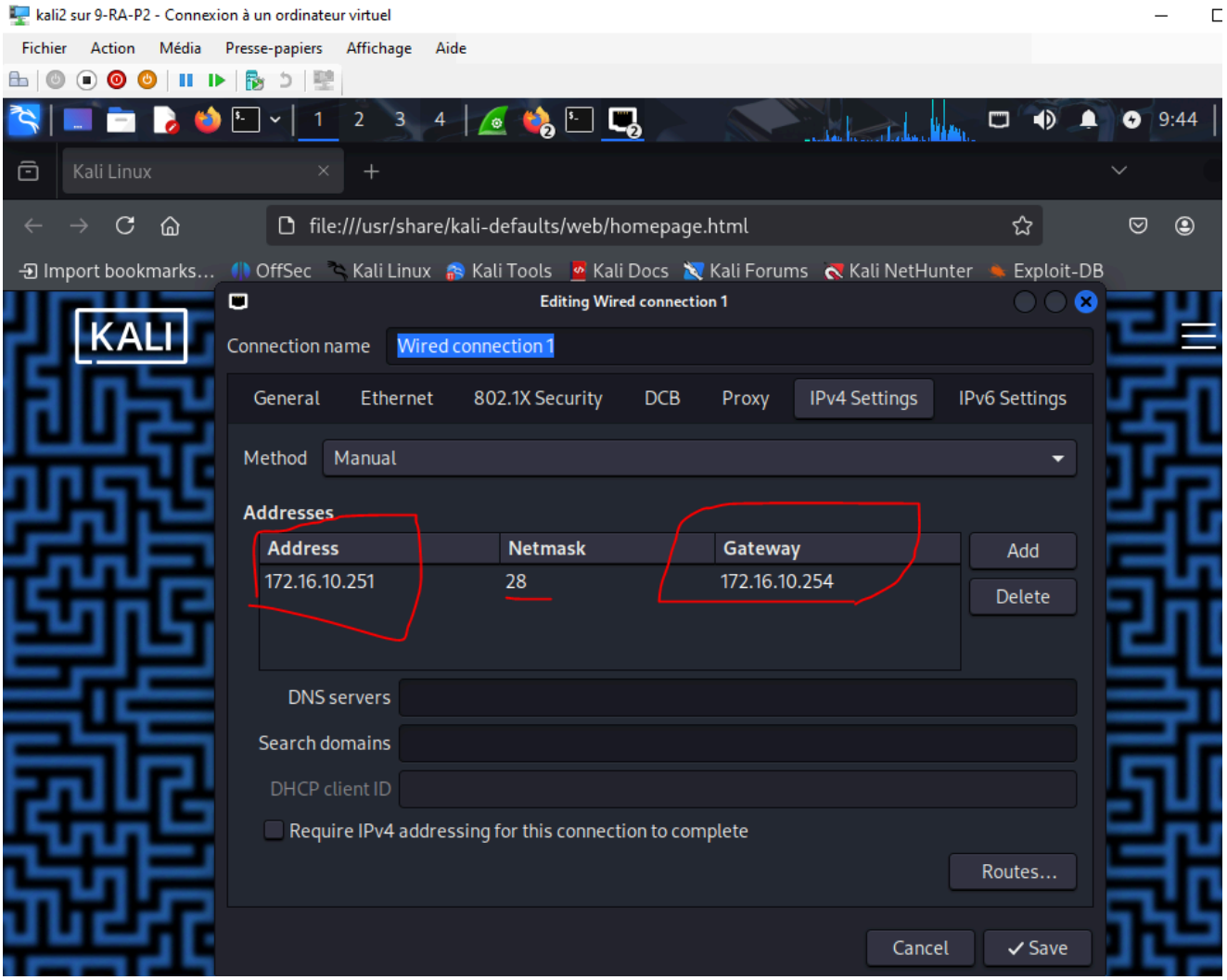
nmap



parametrage du kali

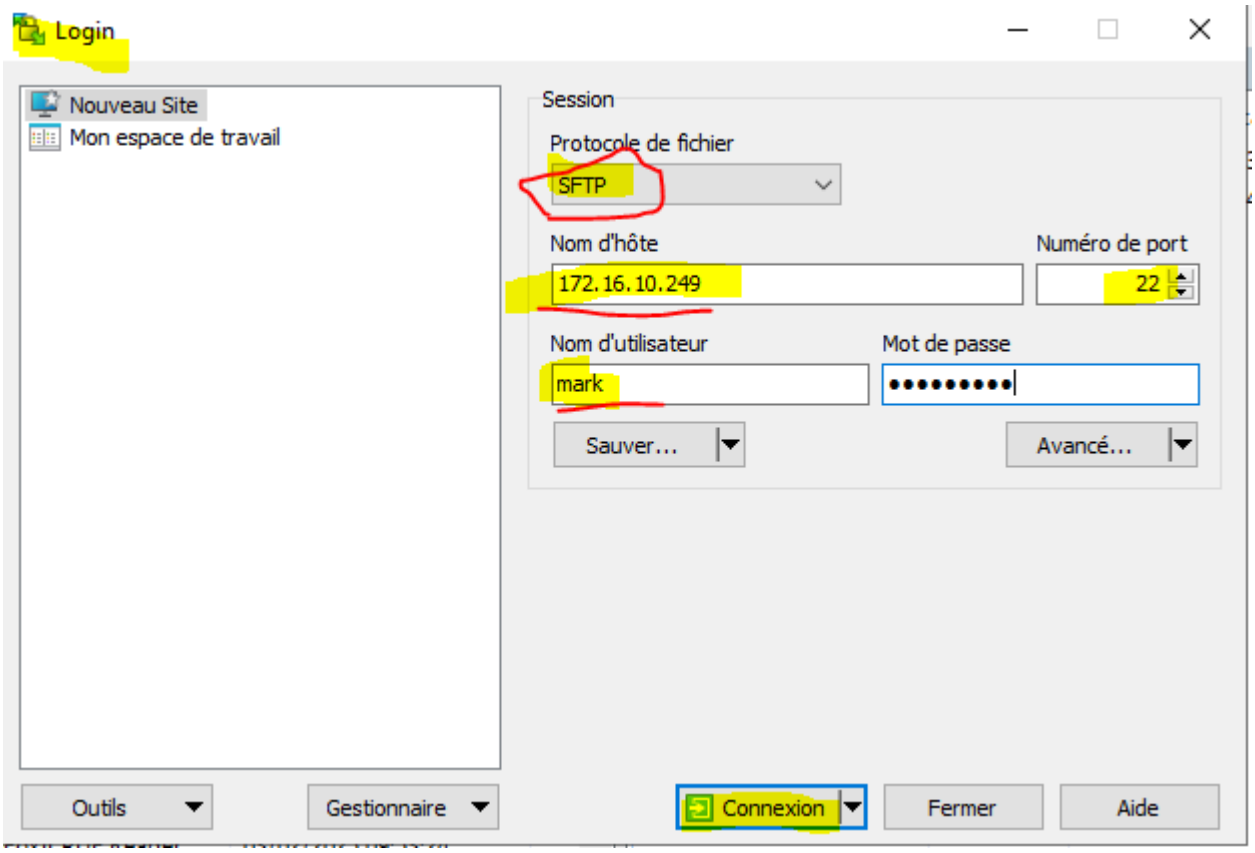


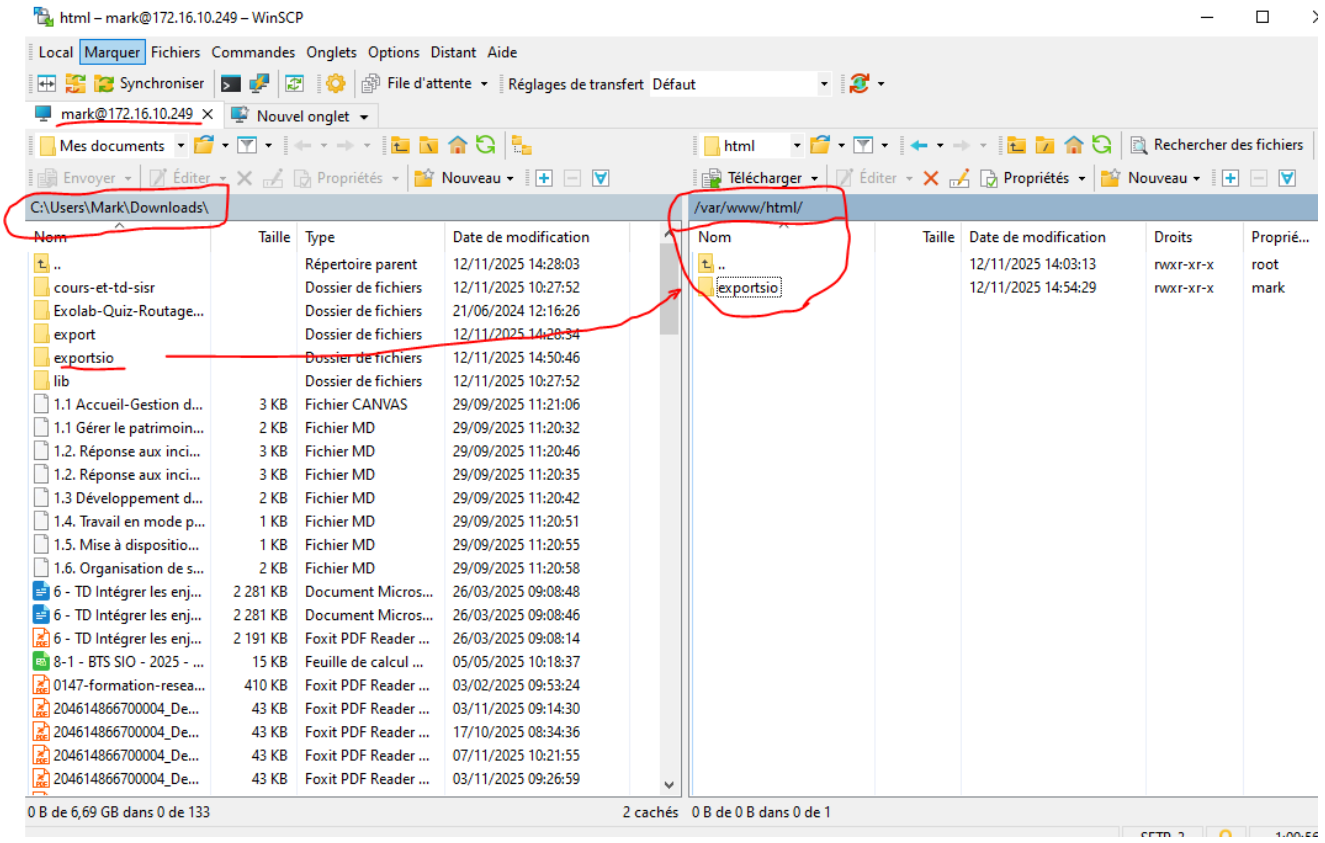
parametrage ip du kali



portfolio
avec winscp

sur la page
nom d'utilisateur: mark (nom du debian)
hote: 172.16.10.249
protocole: SFTP





Pour accorder les permissions , il faut activer sur debian avec les commandes ; su pour entreer entant que admin

```
mark@mark:~$ su
Mot de passe :
```

puis chown et le chemin du dossier

```
root@mark:/home/mark#
root@mark:/home/mark# chown mark /var/www/html
root@mark:/home/mark#
```