Tunneling SSH

Le protocol SSH nous permet de créer des tunnels pour accéder à des appareil distant, à condition d'avoir une connexion SSH sur un appareil au sein du réseau.

SSH Local Port Forwarding (-L)

- Concept : Relie un port spécifique de votre machine à un port spécifique d'une machine distante via le serveur SSH.
- Commande :

```
ssh -L [PORT_LOCAL]:[IP_DESTINATION]:[PORT_DESTINATION] utilisateur@serveur_ssh
```

- Exemple : `ssh -L 8080:192.168.1.1:80 user@remote`
  - **Accès** : Tapez http://localhost:8080 pour voir l'interface du routeur 192.168.1.1

SSH Dynamic Port Forwarding (-D)

- Concept : Transforme votre connexion SSH en un Proxy SOCKS. Au lieu de viser une seule IP, votre navigateur peut atteindre n'importe quelle IP du réseau distant.
- Commande:

```
ssh -D [PORT_PROXY] utilisateur@serveur_ssh
```

- Exemple : `ssh -D 1080 user@remote` Vous pouvez aussi le lancer en arrière plan avec la commande `ssh -f -N -D 1080 utilisateur@serveur_ssh`
  - `-f` : Met SSH en arrière plan
  - `-N` : N'ouvre pas de termanl distant (uniquement le tunnel)

Sur macOS, une fois le tunnel ouvert vous pouvez ouvrir un navigateur avec :

```
open -a "Google Chrome" --args --proxy-server="socks5://localhost:1080"
```

- Ça fonctionne avec tous les navigateurs basé sur chronium comme Arc par exemple.
  - Exemple avec Arc `open -a "Arc" --args --proxy-server="socks5://localhost:1080"`

Une fois le nagivateur ouvert, tapez juste l'ip voulue pour y accéder. Si vous voulez changer le port ne le changer pas sur la commande faites juste `adresse_ip:port`.

Le port 1080 sert uniquement de porte d'entrée.

- Pour fermer le tunnel vous pouvez utiliser la commande `pkill -f "ssh -D 1080"` qui stoppera tous les commandes contenant `ssh -D 1080`

Revision #3
Created 16 February 2026 16:08:13 by Bibi
Updated 16 February 2026 16:25:27 by Bibi