Chapitre 3
Protocoles Sécurisés

Introduction

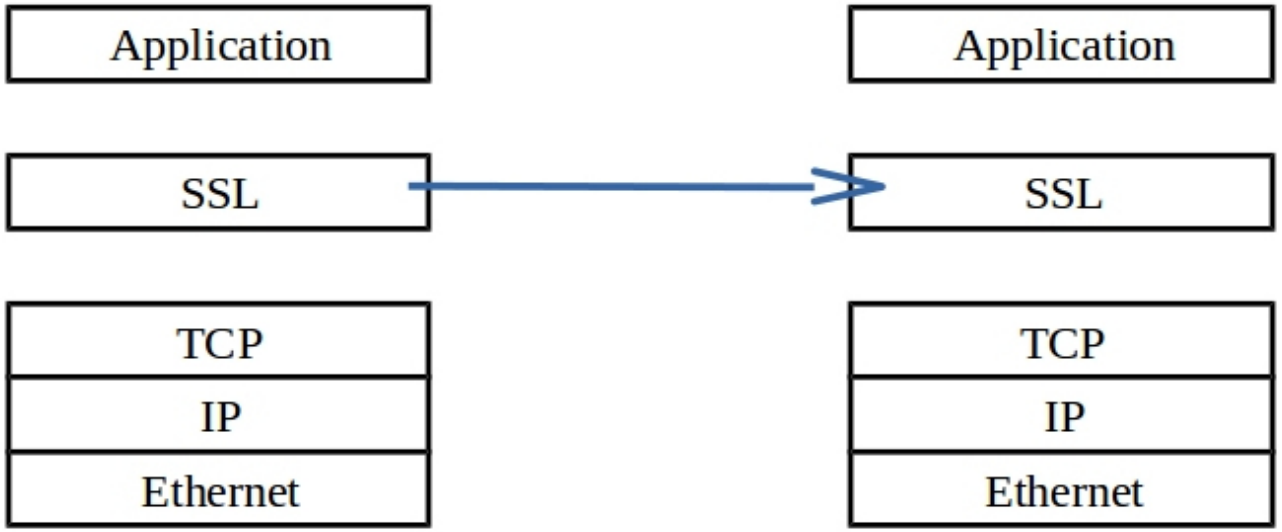
La plupart des protocoles TCP ne sont pas sécurisés. Ce qui signifie que les données transitent en clair sur le réseau.

Pour une sécurité des donnes qui circulent sur le réseau, des protocoles **sécurisés** ont été développés.

SSL (Secure Sockets Layer)

SSL est un logiciel permettant de sécuriser les communications sous HTTP ou FTP.

Le rôle de SSL est de crypter les messages entre un navigateur et un serveur Web. Le niveau d'architecture où se place SSL est illustré dans la figure suivante. Il s'agit d'un niveau compris entre TCP et les applications.

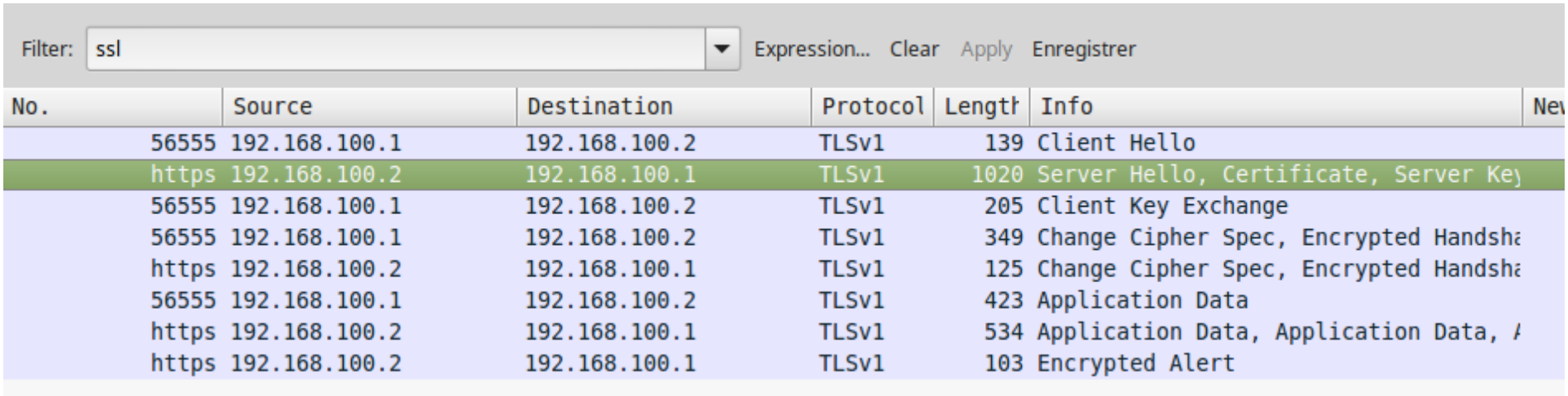


SSL (Secure Sockets Layer)

Un serveur web qui utilise SSL possède une URL (Uniform Ressource Locator) qui commence par **https ://** (s : secured - sécurisé).

SSL

L'initialisation d'une communication SSL commence par un handshake (poignée de main), qui permet l'authentification réciproque.



Messages du protocole Handshake

Les messages échangés pour réaliser le protocole Handshake sont les suivants :

- **Client Hello** : initialisation de la communication par l'envoi d'un hello du client vers le serveur.
- **Server Hello** : peut contenir un certificat et demander une authentification de la part du client.
- **Server Key Exchange** si les certificats ne sont pas pris en charge, ce message permet d'effectuer l'échange de clés publiques.
- **Server Hello Done** : permet d'indiquer que la partie serveur du message hello est achevée.

Messages du protocole Handshake

- **Certificate Request** : requête envoyée par le serveur au client lui demandant de s'authentifier. Le client répond soit avec un message envoyant le certificat, soit avec une alerte indiquant qu'il ne possède pas de certificat.
- **Certificate Message** : message qui envoie le certificat réclamé par le serveur.
- **No Certificate** : message d'alerte qui indique que le client ne possède aucun certificat susceptible de correspondre à la demande du serveur.
- **Client Key Exchange** : échange de la clé du client avec le serveur.
- **Finished** : message qui conclut le handshake pour indiquer la fin de la mise en place de la communication.

Établissement de la connexion SSL

L'établissement d'une connexion SSL se présente comme suit :

1. authentification du serveur auprès du client (chiffrement à clé publique) ;
2. choix d'un algorithme de chiffrement pour l'établissement de la connexion sécurisée ;
3. optionnellement, authentification du client auprès du serveur (techniques de chiffrement à clé publique) ;
4. échange des secrets partagés nécessaires à la génération d'une clé secrète (clé de session) pour le chiffrement symétrique ;
5. établissement d'une connexion SSL chiffrée à clé secrète.

TLS (Transport Layer Security) - Couche de Transport Sécurisée)

C’est le successeur de SSL. Il ne présente que des différences mineures par rapport à SSL.

Le protocole SSH (Secure shell- shell sécurisé)

SSH est un protocole réseau sécurisé qui permet :

- l'établissement de connexions interactives ;
- l'exécution de commandes distantes ;
- le transfert de fichiers.

SSH met en jeu des mécanismes de chiffrement pour la confidentialité des données mais présente également des mécanismes d'authentification similaires à ceux utilisés par SSL.

Fonctionnement de SSH

Le fonctionnement de SSH est basé sur le modèle client/serveur. Un programme serveur (sshd) tourne en permanence sur une machine offrant le service SSH. Un ensemble de commandes clientes permettent d'interagir avec ce serveur afin d'ouvrir des sessions interactives, d'exécuter des commandes distantes ou encore de transférer des données.

OpenSSH

Sous Linux, le serveur **ssh** disponible de façon libre et gratuite s'appelle **OpenSSH**.

Connexion à partir d'un client Linux

Pour se connecter à partir d'un client, tapez : `ssh login@adresse`.
Par exemple :

```
ssh smi@192.168.56.2
```

Utilisation de ssh comme ftp sécurisé

Pour utiliser le serveur **ssh** comme serveur **ftp** sécurisé, tapez la commande : `sftp login@adresse`. Par exemple :

```
sftp smi@192.168.56.2
```

Après saisi du mot de passe, vous obtiendrez l'invite de commandes :

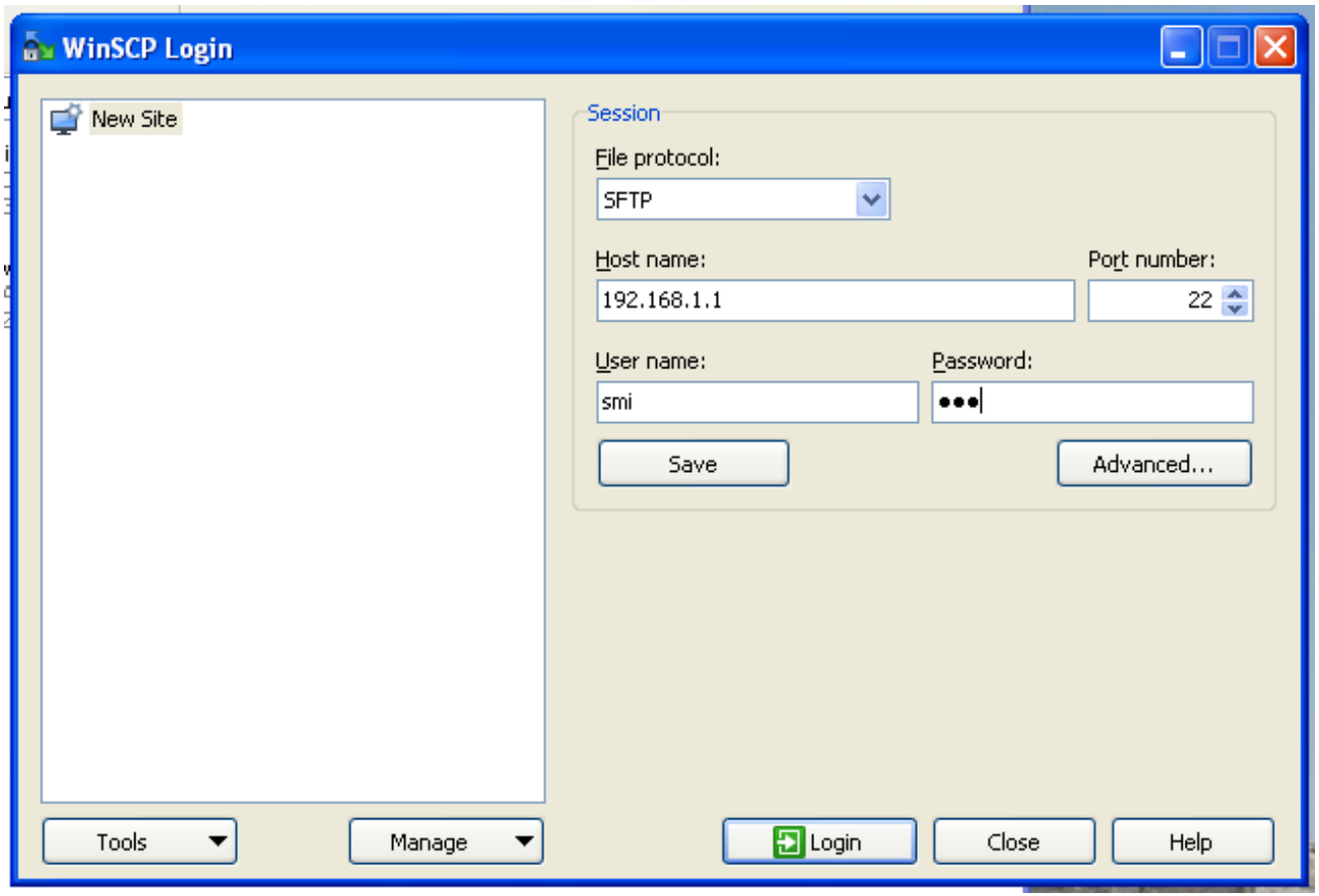
```
sftp>
```

Pour fermer la connexion, tapez **quit**, **bye** ou **exit** dans l'invite de commandes. Pour plus de commandes, tapez dans l'invite « help » ou « ? ». Vous pouvez aussi utiliser le manuel en ligne de sftp :

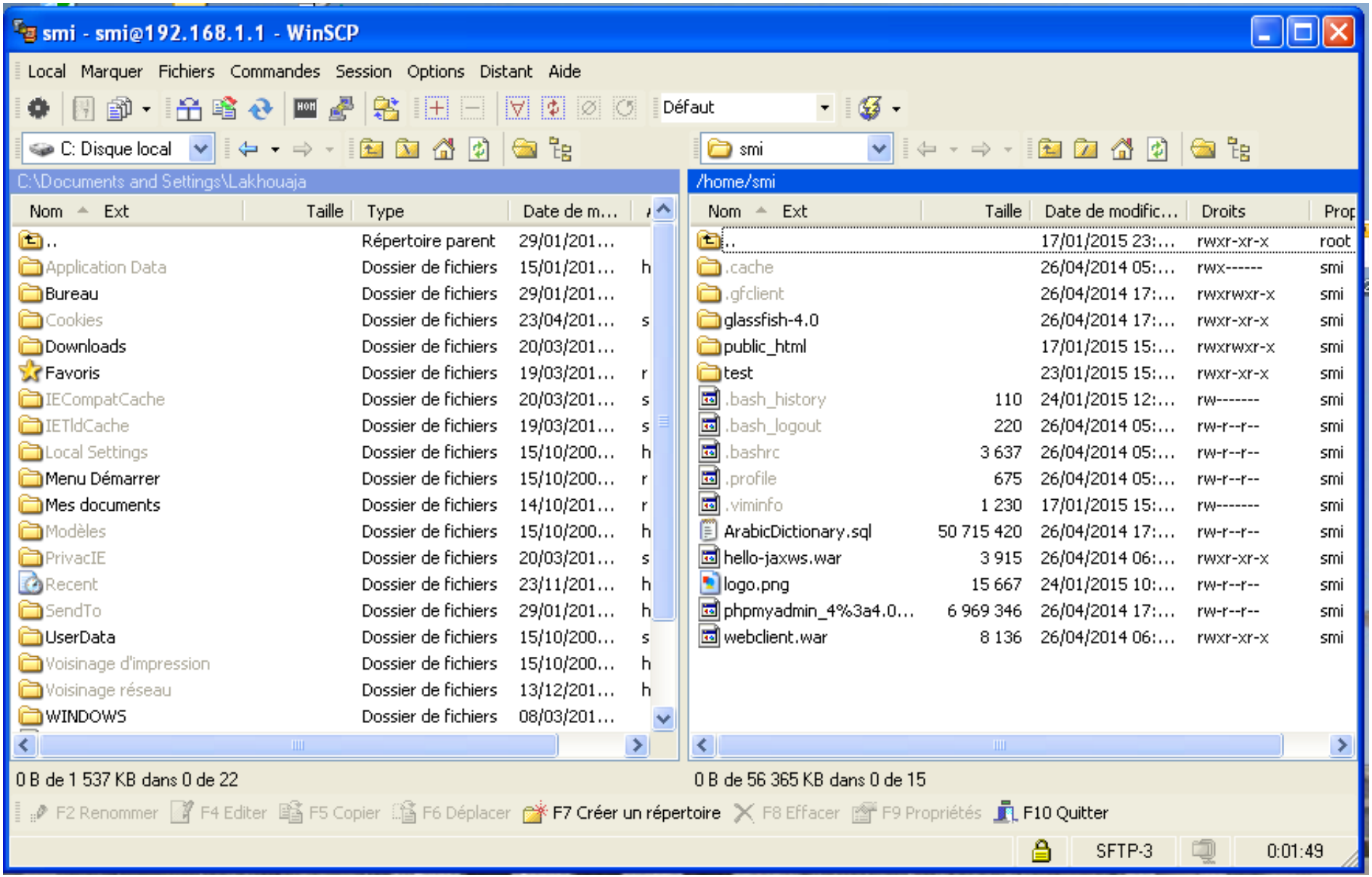
```
man sftp.
```

Connexion à partir d'un client Windows

Sous Windows il existe l'application **winscp** disponible en téléchargement à partir su site officiel `http://winscp.net`. Son interface graphique se présente comme suit :



Connexion à partir d'un client Windows



Copie vers le serveur

Pour copier un fichier ou un répertoire dans le serveur ssh, vous pouvez utiliser la commande **scp** (analogue à la commande **cp** de Linux). Son utilisation est comme suit :

```
scp fichier1 fichier2 ... smi@192.168.56.2:
```

Pour copier un répertoire, il faut simplement ajouter l'option `-r` :

```
scp -r Rep smi@192.168.56.2:
```

Remarque : il ne faut pas oublier **:**, sinon la copie se fera en local (utilisation de **cp**).