REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI DE TIZI OUZOU

FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT D'ELECTRONIQUE

Mémoire De Fin D'étude

En vue d'obtention D'un Diplôme De Master En Electronique

Option

Réseaux Et Télécommunications

thème

Conception D'une Interface De Configuration D'un Serveur SSH Sous Windows XP

Proposé Par :

M^r^ : LAHDIR Mourad

Réalisé Par :

MEKSEM Kaissa

MENAD lyla

Promotion

2010-2011

REMERCIEMENTS

Nous remercions tout d'abord, Allah qui nous a donné la force et le courage pour terminer nos études et élaborer ce modeste travail.

Nous tenons à remercier vivement notre promoteur « Mr.LAHDIR MOURAD» Pour ses conseils précieux et pour toutes les commodités et aisances qu'il nous a apportées durant notre étude et réalisation de ce projet.

Nos remerciements s'adressent également à monsieur le président de jury et les membres du jury pour l'honneur d'avoir assister à notre soutenance et juger ce travail.

Ainsi nous remercions toutes les personnes qui nous ont aidés dans la recherche de la documentation.

Dédicaces

Je dédie ce travail

- *A ma très chère mère, pour leur sacrifie et leur dévouement pour mon bonheur. Que Dieu la garde.*
- *A mes sœurs Karima et Rabiha*
- *A mon frère hamid et sa fille halla*
- *A tous mes cousins surtout le petit Massi*
- *A tous mes amis*
- *A toute ma promotion Master II*
- *A ceux qui m'aiment*

kaissa

Dédicaces

Je dédie ce travail

- *A mes très chère parents, pour leurs sacrifie et leurs dévouement pour mon bonheur. Que Dieu les garde.*
- *A mes frères hakim, farid, abdeslam, arezki et ma sœur houria et son marie*
- *A ma belle sœur ouardia et ses enfants*
- *A tous mes cousins surtout meriem*
- *A tous mes amis*
- *A toute ma promotion Master II*
- *A mon future marie halim et sa famille*

lyla

SOMMAIRE

INTRODUCTION GENERALE

Les systèmes informatiques sur lesquels reposent de plus en plus nos activités deviennent extrêmement complexes tant dans leur conception que dans leur réalisation. La pression sécuritaire due aux évolutions internationales et aux conflits politiques, économiques, militaires et terroristes qui en résultent, et la nouveauté de la problématique sécuritaire due à l'informatisation générale de nos activités, font de la recherche en sécurité informatique et de sa mise en œuvre technologique une priorité importante au niveau nationale et international.

Que ce soit pour un accès à des réseaux locaux ou étendus, filaire ou sans fils, que ces réseaux soient en architecture client-serveur ou répartie**,** l'authentification des équipements (routeur, point d'accès) et des services (web, IRC) est nécessaire. Tout ce qui concerne l'accès privé, c'est-à-dire le contrôle de la délivrance de l'information et de la fourniture des ressources réservées à certaines entités passe par l'authentification.Or,les procédures d'authentification classiques par identifiant et mot de passe ne suffisent plus. Sur les réseaux locaux comme sur internet, l'écoute de ligne est l'attaque numéro un. L'écoute de ligne permet de récupérer, facilement et pratiquement sans risque de détection, l'identifiant et le mot de passe que l'utilisateur envoie au serveur ou bien ses codes d'accès lors d'une connexion légitime. Rien de plus simple pour l'attaquant que de se connecter à son tour en rejouant les mêmes valeurs et ainsi de se faire passer pour un utilisateur autorisé.

La liste des solutions de sécurité est longue mais nous nous limiterons dans notre projet à l'étude du protocole SSH, ainsi que leurs mécanismes d'authentification et d'échange des clés.

Le protocole SSH est utilisé pour établir un accès sécurisé permettant d'effectuer des opérations sensibles sur des machines distantes et d'effectuer des

transferts de fichiers à travers un réseau public tout en garantissant l'authentification, la confidentialité et l'intégrité de données.

Le principal objectif de SSH était de résoudre le problème de transmission en clair de toutes les informations sur le réseau (LAN ou internet) ouvrant la porte à toutes les attaques du type homme de lieu (man in the middle).

Depuis son apparition, le rôle du protocole SSH a évolué pour ne pas se limiter à une simple fonctionnalité de connectivité à distance pour le Shell.

La version 2 de ce protocole, normalisée en janvier 2006 à l'IETF (Internet Engineering Task Force), propose la sécurisation de n'importe quel protocole applicatif et ceci grâce à ses mécanisme de « port forwarding » et de « tunneling ».

L'objetif de ce projet est de tester une solution pour sécuriser les interfaces entre ordinateurs, qui sont équipés d'un système d'exploitation Windows XP. Notre mémoire est divisé en plusieurs chapitres.

Le chapitre 1 présente des généralités sur les réseaux informatiques, les critères de la sécurité des réseaux (les protocoles de sécurité ainsi que les différentes méthodes cryptographique).

Dans le chapitre 2, nous exposerons le principe de fonctionnement du protocole SSH.

Le troisième chapitre sera consacré à l'application sur la configuration d'un serveur SSH.

Enfin, nous terminerons par une conclusion générale ainsi que par des perspectives ouvertes par ce travail.

I-Introduction :

Les réseaux ont pour fonction de transporter des données d'une machine terminale vers une autre. Pour ce faire, une série d'équipements et de processus sont nécessaires, allant de l'environnement matériel utilisant des câbles terrestres ou des ondes radio jusqu'à l'environnement logiciel, constitué de protocoles, c'est-à-dire de règles permettant de décider de la façon de traiter les données transportées.

La sécurité est une fonction incontournable des réseaux. Puisqu'on ne voit pas son correspondant directement, il faut l'authentifier. Puisqu'on ne sait pas par où passent les données, il faut les chiffrer. Puisqu'on ne sait pas si quelqu'un ne va pas modifier les informations émises, il faut vérifier leur intégrité.

Nous étudierons dans ce chapitre le terme de réseau et les différents types de protocoles les plus courants, les critères de sécurité des réseaux et comment se protéger contre les attaques utilisées.

II-Généralités sur les réseaux :

II-1-Définition :

Un réseau est un ensemble des équipements informatiques interconnectés les uns des autres pour le but d'assurer un service de communication.

II-2-Intérêt d'un réseau :

Un ordinateur est une machine permettant de manipuler des données. L'homme, en tant qu'être communiquant, a rapidement compris l'intérêt qu'il pouvait y avoir à relier ces ordinateurs entre-eux afin de pouvoir échanger des informations.

Un réseau informatique peut servir plusieurs buts distincts :

- Le partage de ressources (fichiers, applications ou matériels, connexion à internet, etc).

- La communication entre personnes (courrier électronique, discussion en direct, etc).
- La communication entre processus (entre des ordinateurs industriels par exemple).
- La garantie de l'unicité et de l'universalité de l'accès à l'information (bases de données en réseau).
- Le jeu vidéo multi-joueurs.

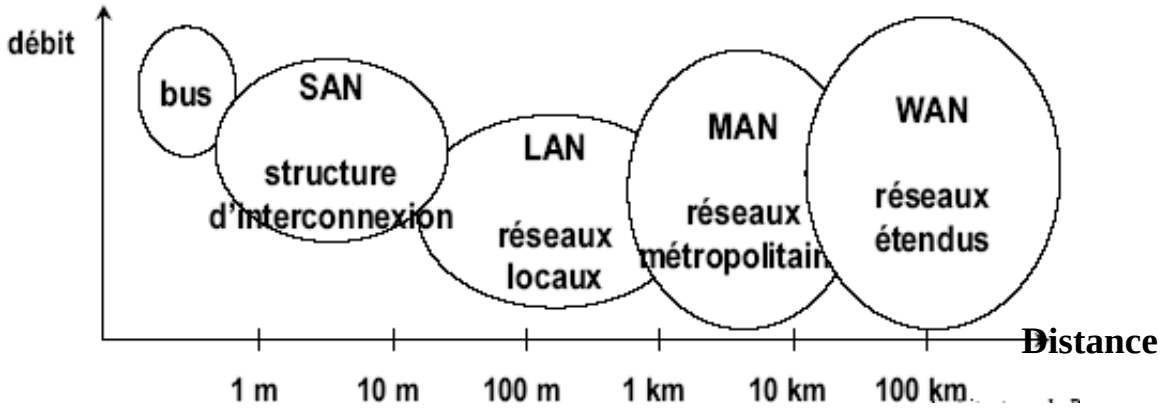
Le type de réseau à installer dépend des critères suivants :

- ✓ Taille de l'entreprise.
- ✓ Niveau de sécurité nécessaire.
- ✓ Type d'activité.
- ✓ Niveau de compétence d'administration disponible.
- ✓ Volume du trafic sur le réseau.
- ✓ Besoins des utilisateurs du réseau.
- ✓ Budget alloué au fonctionnement du réseau (pas seulement l'achat mais aussi l'entretien et la maintenance).

II-3-Les différents types de réseaux :

II-3-1-Classification selon leur taille :

On trouve des réseaux limités à des très courtes distances déterminées par des fils électriques spéciales à l'intérieur d'un même ordinateur, ces fils électriques sont appelés des bus. Cette approche peut être étendue pour atteindre un environnement local, on parle de LAN qui correspond à un réseau d'entreprise. Si la distance est plus grande, nous parlons de MAN qui correspond à un réseau de ville. Enfin, si la distance est très grande nous parlons de WAN qui est de réseau destiné à transporter les données à l'échelle d'un pays ou à l'échelle mondiale.



«Figure.I.1. Classification des réseaux informatiques selon leur taille »

II-3-2-Classification selon leur architecture :

On distingue généralement les deux types de réseaux suivants :

- Les réseaux poste à poste (Peer to Peer)
- Réseaux organisés autour du serveur (Client/serveur)

➢ Similitudes entre types de réseaux :

Les différents types de réseaux ont généralement les points suivant en commun :

- **Serveurs** : ordinateurs qui fournissent des ressources partagées aux utilisateurs par un serveur de réseau.
- **Clients** : ordinateurs qui accèdent aux ressources partagées fournies par un serveur de réseau.
- **Support de connexion** : conditionne la façon dont les ordinateurs sont reliés entre eux.
- **Données partagées** : fichiers accessibles sur les serveurs du réseau.
- **Imprimantes et autres périphériques partagés** : fichiers, imprimantes ou autres éléments utilisés par les usagers du réseau.

II-3-2-a- modèle Poste à poste :

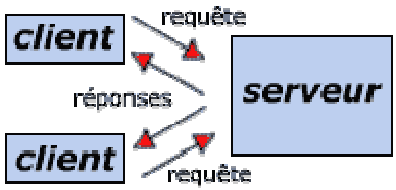
Chaque poste est à la fois serveur et client .ce modèle est le plus simple et moins onéreux pour les petits réseaux d'entreprise et particuliers.

- ✓ Inconvénient :-Moins de performance.
  -Plus sensible aux malveillances (virus…).

II-3-2-b- Modèle client /serveur :

Le modèle qui consiste à déterminer qu'une application attend passivement qu'une autre application initialise la communication est tellement répandu dans le domaine du traitement distribué qu'on lui a donné un nom : il s'agit du modèle d'interaction client-serveur. L'application qui établit le contact (de manière active) s'appelle le client, alors que celle qui attend (passivement) un contact s'appelle le serveur.

Un système client/serveur fonctionne selon La figure suivante :



«Figure.I.2. Fonctionnement du modèle client/serveur»

- Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur.
- Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port.

➢ **Requêtes, réponses et sens du flux de données :**

L'information peut circuler dans les deux sens, entre un client et un serveur. Habituellement, un client envoie une requête à un serveur et ce dernier lui fournit une réponse en retour.

Dans certains cas, un client peut envoyer une série de requêtes et le serveur retourne une série de réponses (par exemple, un client de base de données peut autoriser à un utilisateur l'accès à plus d'un élément à la fois).

Dans d'autre cas, le serveur fournit de l'information en continu, sans requête, dés que le client prend contact avec lui (à titre d'exemple, un serveur de météorologie locale peut émettre en permanence des rapports sur le temps, en mettant à jour la température et la pression barométrique).

Il est important de comprendre que les serveurs peuvent aussi bien accepter de l'information(entrante)qui fournit une information(sortante).A titre d'exemple, la plupart des serveurs de fichiers sont configurés de manière à exporter un ensemble de fichiers aux clients :cela signifie qu'un client envoie une requête qui contient un nom de fichier et que le serveur répond en envoyant une copie de ce fichier.

On peut aussi configurer un serveur de fichier pour qu'il importe des fichiers (le serveur permet alors à un client d'envoyer la copie d'un fichier, l'accepte et la stocke sur disque).

II-3-2-b-1-Caractéristiques des clients et des serveurs :

Bien qu'il existe des variantes mineures, la plupart des instances d'interaction client-serveur possèdent les mêmes caractéristiques générales.
Celles d'un logiciel client sont habituellement les suivantes :

- C'est un programme d'application conventionnel, qui devient temporairement client lorsqu'un accès distant est nécessaire, mais qui effectue aussi localement d'autres fonctions.
- Il est invoqué directement par un utilisateur et il ne fonctionne que pour une seule session.
- Il s'exécute localement sur l'ordinateur personnel de l'utilisateur.
- Il établit, de manière active, le contact avec un serveur.

- Il peut accéder à plusieurs services, si nécessaire, mais il n'établit de contact qu'avec un seul serveur distant à la fois.
- Il ne nécessite pas de matériel spécial ni de système d'exploitation complexe.

Au contraire, un serveur est caractérisé comme suit :

- C'est un programme spécialisé, conçu pour fournir un service, mais qui peut traiter plusieurs clients distants au même moment.
- Il est invoqué automatiquement lorsqu'un système se met en marche et il continue de s'exécuter à travers de multiples sessions.
- Il s'exécute sur un ordinateur partagé (il ne s'agit donc pas de l'ordinateur personnel d'un utilisateur).
- Il attend, de manière passive, un contact depuis des clients distants.
- Il peut être contacté par des clients quelconques, mais il n'offre qu'un seul service.
- Il requiert du matériel puissant et un système d'exploitation évolué.

II-3-2-b-2-Les avantages et les inconvénients de l'architecture client/serveur :

- **Les avantages :**

- Toutes les données sont centralisées sur un seul serveur, ce qui simplifie les contrôles de sécurité et la mise à jour des données et des logiciels.
- Les technologies supportant l'architecture client/serveur sont plus matures que les autres.
- Une administration au niveau du serveur, les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.

- Toute la complexité/puissance peut être déportée sur le serveur(s), les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.
- Un réseau évolutif : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.

➢ **Les inconvénients :**

- Si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge (alors que les réseaux pair à pair fonctionnent mieux en ajoutant de nouveaux participants).
- Si le serveur n'est plus disponible, plus aucun des clients ne fonctionnent (le réseau pair à pair continue à fonctionner, même si plusieurs participants quittent le réseau).
- Les coûts de mise en place et de maintenance sont élevés.
- un maillon faible : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui.

III-Les protocoles :

Un protocole est un ensemble de règle de communication et de messages assurant un service de communication.

Protocole propriétaire : Pour chaque marque d'ordinateur correspond un protocole différent ; Problèmes de compatibilité ; Nécessité d'utiliser des interfaces…

Protocole ouvert : Le concept OSI (Open System Interconnection) de l'ISO (International Standart Organisation). Toutes communications passent par un seul réseau ouvert, qui assure le transfert des informations, ce qui rend tout le monde compatible.

III-1-Le modèle OSI :

Le modèle OSI est un modèle qui comporte 7 couches, tandis que le modèle TCP/IP n'en comporte que 4. En réalité, le modèle TCP/IP a été développé à peu près au même moment que le modèle OSI, c'est la raison pour laquelle il s'en inspire mais n'est pas totalement conforme aux spécifications du modèle OSI. Les couches du modèle OSI sont les suivantes:

Une couche assurant la transmission de l'application demandée avec envoi de messages.	7	couche application	Gère les applications de types réseaux : courrier électronique, transfert de fichier, appel de procédure distantes…
	6	couche présentation	Assure une transparence en terme de codage (ex. ASCII).
	5	couche session	S'occupe de fiabiliser la communication des utilisateurs, gère des tours de parole, synchronisation.
Une couche de communication de base permettant de transmettre physiquement en respectant un certain nombre de règles.	4	couche transport	Optimise l'utilisation de la couche réseau et assure des travaux de type fragmentation de message (ex. TCP).
	3	couche réseau	Offre un nombre de services dont un service d'adressage (IP) permettant d'atteindre son destinataire, un service de routage déterminant un chemin à l'intérieur du réseau maillé et un contrôle de flux pour ne pas saturer le réseau.
	2	couche liaison de données	Permet d'assurer une liaison fiable par une bonne synchronisation et une détection d'erreur.
	1	couche physique	Emet des signaux assurant la bonne transmission.

«Tableau.I.1.Les differentes couches du modèle OSI»

III-2-Le modèle TCP/IP :

Le protocole TCP est défini dans le but de fournir un service de transfert de données de haute fiabilité entre deux ordinateurs "maîtres" raccordés sur un réseau de type "paquets commutés", et sur tout système résultant de l'interconnexion de ce type de réseau.

Le modèle TCP/IP, inspiré du modèle OSI, reprend l'approche modulaire (utilisation de modules en couches) mais contient uniquement quatre:

Modèle TCP/IP	Modèle OSI
Couche Application	Couche Application
	Couche Présentation
	Couche Session
Couche Transport (TCP)	Couche Transport
Couche Internet (IP)	Couche Réseau
Couche Accès réseau	Couche Liaison de données
	Couche Physique

«Tableau.I.2. La comparaison entre les différentes couches d'OSI et TCP/IP»

Comme on peut le remarquer, les couches du modèle TCP/IP ont des tâches beaucoup plus diverses que les couches du modèle OSI, étant donné que certaines couches du modèle TCP/IP correspondent à plusieurs couches du modèle OSI.

➢ **Les rôles des différentes couches de TCP/IP sont les suivants:**

- **Couche accès réseau :** spécifie la forme sous laquelle les données doivent être acheminées quelque soit le type de réseau utilisé.
- **Couche Internet :** elle est chargée de fournir les paquets de données (datagramme).
- **Couche Transport :** elle assure l'acheminement des données, ainsi que les mécanismes permettant de connaitre l'état de la transmission.
- **Couche Application :** elle englobe les applications standard du réseau (Telnet, SMTP, FTP, ...).

Voici les principaux protocoles faisant partie de la suite TCP/IP:

Modèle TCP/IP
Couche application Applications réseaux
Couche Transport TCP ou UDP
Couche Internet IP, ARP, RARP
Couche Accès réseau FDDI, PPP, Ethernet, Anneau à jeton

«Tableau.I.3. Les protocoles de TCP/IP »

III-3-Les différents types de protocoles :

III-3-1-Le protocole TCP :

Le protocole TCP (Transmission Control Protocol) est crée dans le but d'établir une communication à haute fiabilité entre deux tâche exécutées sur deux ordinateurs raccordés à un réseau (protocole orienté connexion).

III-3-2-Le protocole IP :

Le rôle du protocole internet est la transmission de bloc de données, appelé datagramme, d'une source vers une destination, la source et la destination étant des ordinateurs hôtes identifier par une adresse de longueur fixe.

Le protocole internet dispose des mécanismes permettant la fragmentation de long datagramme et leur réassemblage lors, de leur transmission à travers des réseaux.

III-3-3-Le protocole UDP :

Le protocole UDP (User Datagram Protocol) est comme TCP, un protocole de transport de données.

Cependant, contrairement à TCP, en qualifie l'UDP de transmission « en mode non connecté et non fiable » ou encore des protocoles « non orienté connexion ».

Ceci signifie simplement que la machine émettrice envoie des données sans prévenir la machine réceptrice, et la machine réceptrice reçoit les données sans envoyer d'avis de réception à la première.

Les données sont ainsi envoyées sous forme de bloc (datagramme).

Il n'ya pas de contrôle d'erreur, C'est un mécanisme simple d'échange de données entre les applications.

III-3-4-Le protocole FTP :

Le protocole FTP (File Transfer Protocol) définit la façon selon laquelle des données doivent être transférées sur TCP/IP.

Le protocole FTP a pour objectifs de :

- Permettre un partage de fichiers entre deux machines distantes.

- Permettre une indépendance aux systèmes de fichiers des machines clientes et serveur.
- Permettre de transférer des données de manière efficace.

III-3-5-Le protocole ICMP :

Le protocole ICMP (Internet Control Message Protocol) est un protocole qui permet de gérer les informations relatives aux erreurs aux machines connectées. Étant donné le peu de contrôles que le protocole IP réalisé, il permet non pas de corriger ces erreurs mais de faire part de ces erreurs aux protocoles des couches voisines. Ainsi, le protocole ICMP est utilisé par tous les routeurs pour signaler une erreur (appelé delivery problem).

Les messages d'erreur ICMP sont transportés sur le réseau sous forme de datagramme, comme n'importe quelle donnée. Ainsi, les messages d'erreur peuvent eux-mêmes être sujet d'erreur.

Toutefois en cas d'erreur sur un datagramme transportant un message ICMP, aucun message d'erreur n'est délivré pour éviter un effet « boule de neige » en cas d'incident sur le réseau.

III-3-6-Le protocole ARP :

Le protocole ARP (Address Resolution Protocol) permet de trouver l'adresse Ethernet à partir d'une adresse IP.

ARP est mis en place pour la transmission des trames de données .si des machines sur le même réseau veulent envoyer des datagrammes entre elles, il faut qu'elles sachent tout d'abord l'adresse physique de la machine pour pouvoir construire la trame Ethernet.

Pour trouver l'adresse Ethernet (adresse physique) à partir d'une adresse IP (adresse logique). Soit deux machines A et B sur le même réseau, A veut envoyer un datagramme IP à la machine B :

1. Au début, les machines sur le même réseau ne connaissent que l'adresse IP après son démarrage et sa connexion au réseau.

2. La machine A vérifie si elle a l'adresse physique de B dans sa mémoire cache.
3. Si A ne connait pas encore l'adresse physique de B, elle envoie une trame de diffusion qui demande l'adresse Ethernet de B en indiquant l'adresse IP de B.
4. Toutes les machines du réseau reçoivent la trame, mais seule la machine B répond à A en lui donnant son adresse Ethernet.
5. La machine A la sauvegarde dans sa mémoire cache et l'utilise pour construire les trames Ethernet

III-3-7-Le protocole SMTP (Simple Mail Transfert Protocol) :

Protocole de transfert de courrier électronique d'un ordinateur vers un autre à l'internet. Il occupe le port 25 d'un serveur ; est appelé serveur de courrier sortant, c'est le serveur utilisé pour l'envoie d'email à destination d'internet.

III-3-8-Le protocole POP3 (Port Office Protocol version 3) :

Occupe le port 110 d'un serveur, il est appelé serveur de courrier entrant ; c'est le serveur qui utilise la réception d'email issue d'internet.

III-3-9-Le protocole TELNET :

Le protocole TELNET est très utilisé sur internet, il utilise un modèle client/serveur qui permet d'exécuter des commandes à distance. Il est souvent utiliser pour exécuter des commandes sur un serveur à partir d'un terminal.

TELNET est employé non seulement pour connecter au serveur mais il peut également à n'importe quelle machine qui dispose d'un service de TELNET.

Le TELNET est utile non seulement pour récupérer des emails, l'information et les programmes mais également pour l'entretient de site web, et la configuration de routeur à distance. Le serveur TELNET n'est pas sécurisé, toutes les informations (y compris le compte d'utilisateur et le mot de passe) circulent en clair sur le réseau.

Ce protocole utilise le port 23/TCP, et pour pouvoir connecter au serveur TELNET il faut :

Le serveur (Telnetd : Telnet deamon) écoute sur le port 23. Le client initie une connexion depuis un port N non privilégié (>=1024) tiré aléatoirement vers le port 23 du serveur et le serveur répond au client vers le port choisi.

IV-Les critères de sécurité :

IV-1-La disponibilité :

Pour un utilisateur la disponibilité d'une source est la probabilité de pouvoir menu correctement à terme une session de travail.

La disponibilité d'une ressource est indissociable de son accessibilité. Cette disponibilité est mesuré sur la période de temps pendant laquelle le service offert est opérationnel ; le volume potentiel du travail susceptible d'être près en charge durant la période disponibilité d'un service déterminé par la capacité d'une ressource (serveur au réseau).

La disponibilité des services, des systèmes et des données est obtenue par :

-Un dimensionnement approprié une certaine redondance.

-Par une gestion opérationnelle efficace des infrastructures, ressources et services.

IV-2-L'intégrité :

Le critère d'intégrité est relatif au faite que des ressources, données, traitement, transaction ou service non pas été modifié, altéré ou détruite de façon intentionnelle qu'accidentelle.

IV-3-La confidentialité :

La confidentialité est le maintien des secrets des informations transposé dans le contexte de l'informatique et les réseaux.

La confidentialité peut être vue comme la protection des données, comme une divulgation non autorisée.

Il existe deux actions complémentaires permettent d'assurer la confidentialité des données :

-Limiter leur accès par un mécanisme de contrôle d'accès.

-Transformer des données par des procédures de chiffrement afin qu'il devient inutilisé aux personnes qui ne possèdent pas les moyens de les déchiffrer.

IV-4-L'identification :

L'identification est assurée par l'utilisation de mots de passe. Ceux-ci, pour offrir une bonne garantie de sécurité, doivent être échangé fréquemment et ne doivent pas être trop facile à trouver (ils peuvent notamment comporter des chiffres et des caractères spéciaux).

Dans les systèmes à haut niveau de sécurité (comme l'armement), on commence à utiliser des techniques fondées sur la biométrie (empreintes digitales, forme du visage ou de la pupille).

IV-5-L'authentification :

L'authentification est un problème plus complexe. Lorsque l'identification se fait sans contact directe entre l'ordinateur et l'utilisateur (ou le programme) qui se connecte, il faut échanger de l'information supplémentaire pour s'authentifier mutuellement.

On utilise des techniques issues de la cryptographie (échange de clés).

IV-6-La non répudiation :

La non répudiation est le fait de ne pouvoir rien ou rejeter un événement (action, transaction) c'est-à-dire quelle permet d'assurer que l'expéditeur a bien envoyé le message (autrement dit elle empêche l'expéditeur de nier avoir expédié le message).

V- la sécurité par le chiffrement :

V-1-La cryptographie :

La cryptographie est la science de rendre les données secrètes, elle est essentiellement basé sur l'arithmétique : il s'agit de transformer les lettres qui

compose le message en une succession de chiffre sous forme de bits puis faire des calculs sur ces chiffres pour :

-Les modifier.

-Faire en sorte que le destinataire sera les décryptés.

Le cryptage est le fait de coder le message de façon à le rendre secret.

V-1-a-Cryptage symétrique :

Dans le cryptage symétrique, encore appelé cryptage à clé secrète, la clé (ou algorithme) utilisée pour crypter les données est la même que celle utilisée pour les décrypter. Son principal avantage est d'être très rapide et facile à utiliser.

Cependant, ce cryptage à aussi un inconvénient, puisqu'il faut que les deux parties possèdent la clé secrète, il faut donc la transmettre d'un bout à l'autre, ce qui est risqué sur un réseau non fiable comme internet car la clé peut ainsi être interceptée.

Exemples d'algorithme de cryptage symétrique :

		DES	3DES	IDEA	RC4	RC5 et RC6	Blowfish	AES
Nom réel		Data Encryption Standard	Triple Data Encryption Standard	International Data Encryption Algorythm	Rivest Cipher 4	Rivest Cipher 5/6	Blowfish	Advanced Encryption Standard
Date		1973	1978	1992	1987	1994	1993	1998
Longueur	Clé	64 bits (56 effectifs)	192 bits (168 effectifs)	128 bits	jusqu'a 256 bits	Jusqu'à 2040 bits	entre 40 et 448 bits	128, 192, 256 bits
	Bloc	64 bits	64 bits	64 bits	Flux	32, 64, 128 bits	64 bits	128 bits

«Tableau.I.4. Les algorithmes de cryptage symétrique»

V-1-b-Cryptage asymétrique :

Ce système de cryptage utilise deux clés différentes pour chaque utilisateur : une est privée et n'est connue que par l'utilisateur ; l'autre est publique et donc accessible par tout le monde.

Les clés publique et privée sont mathématiquement liées par un algorithme de cryptage de telle manière qu'un message crypté avec une clé publique ne puisse être décrypté qu'avec la clé privée correspondante.

Une clé (la clé publique) est donc utilisée pour le cryptage et l'autre (la clé privée) pour le décryptage.

Exemple : Si Alice veut envoyer un message crypté à Bob, il faut que :

1. Alice encrypte le message avec la clé publique de Bob (que Bob lui aura préalablement envoyé).
2. Alice envoie le message à Bob.
3. Bob décrypte le message avec sa propre clé privée.

Le principal avantage du cryptage asymétrique est de résoudre le problème de l'envoi de clé privée sur un réseau non sécurisé. Bien qu'il soit plus lent que les cryptages à clé privée, il reste préférable car il permet plus facilement que le cryptage symétrique, l'authentification et les signatures numériques.

Exemples d'algorithmes de cryptage asymétriques :

RSA	(Rivest, Shamir, Adelman)	Crée en 1977, le RSA a été le premier algorithme à clé publique. Sa force réside dans la difficulté à factoriser de grands nombres. RSA utilise des longueurs de clés variables (512, 1024, 2048 bits...), le RSA est considéré comme fiable.
DH	Diffie-Hellman	Plus qu'un simple algorithme de cryptage, Diffie-Hellman est également un protocole qui permet d'échanger des clés secrètes. Il a été rendu public. Il est plus vulnérable que RSA à certains types d'attaques (notamment à l'attaque Man-In-The-Middle), mais reste toujours utilisé aujourd'hui.

«Tableau.I.5. Les algorithmes de cryptage asymétrique»

V-1-c-Cryptage à clé mixte (hybride) :

Les systèmes hybrides sont les plus utilisés : ils combinent la sûreté du cryptage asymétrique et la rapidité du cryptage symétrique.

Le cryptage asymétrique est trop lent, et le cryptage symétrique n'est pas sûr car il faut que l'expéditeur envoie sa clé au destinataire, ce qui est beaucoup trop risqué sur certains réseaux (comme Internet). Les systèmes hybrides consistent en fait à crypter un message à l'aide d'une clé privée qu'on fait parvenir à l'expéditeur en l'encapsulant dans un système asymétrique.

Exemple de système à clé mixte : PGP (Pretty Good Privacy)

PGP est un logiciel de cryptage qui utilise simultanément les systèmes de cryptage symétrique IDEA et asymétrique RSA :

Chiffrement	- PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé. - PGP chiffre la clé secrète IDEA précédemment créée au moyen de la clé RSA publique du destinataire.
Déchiffrement	- PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée. - PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

«Tableau.I.6. Le fonctionnement du protocole PGP»

VI- Les protocoles de sécurité :

VI-1-Le protocole SSL (Secure socket layer) :

SSL est un procédé de sécurisation des transactions effectuées via internet. Il repose sur un procédé de cryptographie par la clé publique afin de garantir la sécurité de la transmission des données sur internet.

Le système SSL est indépendant des protocoles utilisés, ce qui signifie qu'il peut aussi bien sécuriser des transactions faite sur le web par le protocole http(Hyper Text Transfer Protocole) que des connexions via le protocole FTP.

En effet, ce système agit sur une couche supplémentaire, il permet d'assurer la sécurité des données situées entre la couche d'application et la couche de transport (protocole TCP par exemple).

La sécurisation des transactions par SSL2.0 est basée sur un échange de clé entre le client et le serveur.

La transaction sécurisée par SSL se fait selon le schéma suivant :

-Dans un premier temps le client se connecte au site marchand sécurisé par SSL et lui demande de s'authentifier.

-Quand le serveur reçoit la requête ; il envoie un certificat au client, contenant la clé publique du serveur, signé par une autorité de certification (CA).

Le client vérifie la validité du certificat, crée une clé secrète, chiffre cette clé à l'aide de la clé publique du serveur et lui envoie le résultat (la clé de session).

-Le serveur est en mesure de déchiffrer la clé de session avec sa clé privée.

Ainsi, les deux entités sont en possession d'une clé commune dont elles sont seules connaisseuses.

VI-2-Le protocole SET (Secure Electronic Transaction) :

SET est la convergence des deux procédures de sécurisation STT (Secure Transaction Technology) de vista et Microsoft et SEPP (Secure Electronic Payement Protocol) de Mastercard, IBM et Netscape. Il permet de sécuriser les

transactions par carte bancaire (chiffrement par clé publique/privée et authentification des parties).

VI-3-Le protocole PGP (Pretty Good Privacy):

Le cryptage de toute l'information par une clé publique nécessitant un temps de calcule élevé, PGP utilise une technique plus rapide :

Le document est compressé (éviter les redondances) puis crypté avec une clé de session aléatoire (cryptage rapide), seul la clé de session est crypté par la clé publique du destinataire et ajouter au document.

Le destinataire utilise sa clé privée pour décrypter la clé de session et peut ainsi décrypter le document et le décompresser.

VI-4-Le protocole Secure HTTP (Secure Hyper Text Transfer Protocol) :

S-HTTP (Traduisible par le protocole http sécurisé) est un procédé de sécurisation des transactions HTTP. Il permet de fournir une sécurisation des échanges lors de transaction de commerce électronique en cryptant les messages afin de garantir aux clients la confidentialité de leur numéro de carte bancaire ou de toute autre information personnelle.

Contrairement à SSL au niveau de la couche de transport, s-http procure une sécurité basé sur des messages au dessus du protocole http, en marquant individuellement les documents html à l'aide de certificat.

Ainsi, alors que SSL est indépendant de l'application utilisée et crypte l'intégralité de la communication, S-HTTP est très fortement lié au protocole http et crypte individuellement chaque message.

Les messages s-http sont basés sur trois composantes :

- Le message HTTP.
- Préférences cryptographique de l'envoyeur.
-Les préférences du destinataire.

Ainsi, pour décrypter un message S-HTTP, le destinataire du message analyse les entités du message afin de déterminer le type de méthode qui a été utilisé pour crypter le message.

Puis, grâce à ses préférences cryptographiques actuelles, ainsi que des préférences cryptographiques précédentes de l'expéditeur capable de décrypter le message.

SSL permet de sécuriser la connexion internet tandis que s-http permet de fournir des échanges HTTP sécurisé.

VII-Conclusion :

Après l'étude des différents protocoles de communication, on constate que beaucoup d'utilisateurs de Telnet et ftp ne se rendent pas compte que leur mot de passe est transmis en clair à travers le réseau. L'utilisation des protocoles sécurisés ont pût permettre une transmission sécurisée à travers des réseaux non-sûrs. Parmi eux, nous proposons le protocole SSH que nous décrivons dans le deuxième chapitre.

I-Introduction :

Internet permet de réaliser un grand nombre d'opérations à distance, notamment l'administration de serveurs ou bien le transfert de fichiers. Le protocole Telnet et les r-commandes BSD (rsh, rlogin) permettant d'effectuer ces tâches distantes mais ils possèdent l'inconvénient majeur de faire circuler en clair sur le réseau les informations échangées, notamment l'identifiant (login) et le mot de passe pour l'accès à la machine distante.

Ainsi, un pirate situé sur un réseau entre l'utilisateur et la machine distante a la possibilité d'écouter le trafic, c'est-à-dire d'utiliser un outil appelé sniffer capable de capturer les trames circulant sur le réseau et ainsi d'obtenir l'identifiant et le mot de passe d'accès à la machine distante.

Même si les informations échangées ne possèdent pas un grand niveau de sécurité, le pirate obtient un accès à un compte sur la machine distante et peut éventuellement étendre ses privilèges sur la machine afin d'obtenir un accès administrateur.

Etant donné qu'il est impossible de maîtriser l'ensemble des infrastructures physiques situées entre l'utilisateur et la machine distante (internet étant par définition un réseau ouvert), la seule solution est de recourir à une sécurité au niveau logique (au niveau des données).

Le protocole **SSH** (Secure Shell) répond à cette problématique en permettant à des utilisateurs (ou bien des services TCP/IP) d'accéder à une machine à travers une communication chiffrée (appelée tunnel).

II-Evolution du protocole SSH :

Le protocole SSH (Secure Shell) a été mis au point en 1995 par le professeur Finlandais Tatu Ylönen.

Il s'agit d'un protocole permettant à un client (un utilisateur ou bien même une machine) d'ouvrir une session interactive sur une machine distante (serveur) afin d'envoyer des commandes ou des fichiers de manière sécurisée :

- Les données circulant entre le client et le serveur sont chiffrées, ce qui garantit leur confidentialité (personne d'autre que le serveur ou le client ne peut lire les informations transitant sur le réseau). Il n'est donc pas possible d'écouter le réseau à l'aide d'un analyseur de trames.
- Le client et le serveur s'authentifient mutuellement afin d'assurer que les deux machines qui communiquent sont bien celles que chacune des parties croit être. Il n'est donc plus possible pour un pirate d'usurper l'identité du client ou du serveur (spoofing).

La version 1 du protocole (SSH1) proposée dès 1995 avait pour but de servir d'alternative aux sessions interactives (Shells) telles que Telnet, rsh, rlogin . Ce protocole possède toutefois une faille permettant à un pirate d'insérer des données dans le flux chiffré. C'est la raison pour laquelle en 1997, la version 2 du protocole (SSH2) a été proposée en tant que document de travail à l'IETF.

Secure Shell Version 2 propose également une solution de transfert de fichiers sécurisé (SFTP, Secure File Transfer Protocol).

SSH est un protocole, c'est-à-dire une méthode standard permettant à des machines d'établir une communication sécurisée.

III-Définition du protocole SSH:

SSH est une version sécurisée de ces outils ; il permet de se connecter à distance sur une machine donnée suivant une architecture client/serveur. Il se compose d’un client qui sera invoqué sur la machine initiatrice de la communication et d’un serveur qui doit tourner sur la machine destinataire.il va

créer une communication sécurisée, en authentifiant les deux parties et en garantissant le secret de la communication et son intégrité.

SSH est classiquement utilisé pour, une fois la communication sécurisée établie, exécuter un interpréteur de commandes, un Shell (d'où le nom de cet outil), Mais il est possible de faire passer à travers un canal sécurisé n'importe quel trafic TCP(X11, SMTP, HTTP, etc.), ce qui offre une grande flexibilité ; On appelle cela créer un tunnel SSH.

IV-Architecture et fonctionnement de base :

Applications (ssh, sshd, scp, sftp) etc.)	
Protocole de connexion	Protocole d'authentification
Protocole de transport	
TCP	

«Tableau.II.1. Architecture de SSH-2 »

IV-1-La couche de transport SSH (SSH-TRANS):

C'est sur cette couche que reposent les deux autres couches ssh. SSH-TRANS fournit l'authentification du serveur, la confidentialité et l'intégrité des données au moyen d'un chiffrement symétrique, elle fournit en option la compression des données de session.

Cette couche doit être utilisée pour permettre au client de vérifier qu'il communique bien avec le serveur attendu. Ce protocole fonctionne au dessus de TCP/IP mais peut être utilisé au dessus de toute couche de transport fiable.

Il effectue : l'authentification du serveur, la négociation des algorithmes de protection des données, la mise en place d'une clef de session, la création

d'un secret partagé, l'intégrité, la confidentialité des données et l'identification de la session.

Ainsi, pour que la couche de transport de SSH puisse créer correctement une session sécurisée entre les deux entités communicantes, de nombreux élément sont négociés.

En effet, durant l'échange des clefs, le serveur s'authentifie auprès du client au moyen d'une clef publique RSA ou DSA, la clef du serveur est inconnue pour le client durant sa première communication.

SSH propose de contourner ce problème en permettant au client d'accepter la clef du serveur lors de leur première connexion SSH.

Ensuite, lors des connexions suivantes, la clef du serveur peut être vérifiée au moyen d'une version enregistrée au niveau du client, ce qui permet au client de s'assurer qu'il communique bien avec le serveur désiré. Ce mécanisme est l'un des inconvénients majeurs de ce protocole qui le rend vulnérable à des attaques de type homme du lieu.

A partir des valeurs publique DH (Diffie-Hellman) échangées dans chaque session, les deux entités génèrent automatiquement une clef de session utilisée avec des clefs dérivées pour le chiffrement des données. Dès qu'un certain volume de données est transmit, protégé de la même façon à l'aide d'une clef et d'un algorithme, une nouvelle clef de session doit être mise en place, ce qui nécessite de nouveaux échanges.

Le volume exacte des données au bout duquel un renouvellement de clefs a lieu est définit dans la norme à 1giga octet d'échange ou correspond à 1heure d'échanges sur la session ; tout dépend de la mise en application du protocole ssh.

Une fois que la couche transport a crée un tunnel sécurisé pour envoyer des informations entre les deux systèmes, le serveur indique au client les différentes méthodes d'authentification prise en charge pour l'authentification comme l'utilisation d'une clef asymétrique, ou l'entrée d'un mot de passe.

IV-2-La couche d'authentification SSH (SSH-AUTH) :

Après l'échange des méthodes d'authentification supportées à travers la couche SSH-TRANS, la couche d'authentification permet de certifier l'identité du client auprès du serveur.

En effet, étant donné que les serveurs peuvent être configurés de façon à permettre différent type d'authentification, cette couche donne aux deux parties un niveau de contrôle optimal.

Le serveur peut décider quelles méthodes d'authentification prendre en charge en fonction de son modèle de sécurité et le client peut choisir la méthode d'authentification a utilisée parmi celles qui sont disponibles.

Les messages échangés par cette couche sont sécurisés par la clef de chiffrement créée dans la couche SSH-TRANS.

Après l'authentification du client auprès du serveur, de nombreux services peuvent être utilisé de façon sécurisée, tels qu'une session Shell interactive, des applications X11 et des ports TCP/IP tunnels.

IV-3-La couche de connexion SSH (SSH-CONN) :

Cette couche s'appuie sur la couche d'authentification. Elle offre une variété riche de service aux clients en se servant de l'unique tunnel fournit par SSH-TRANS. Ces services comprennent tout ce qu'il faut pour gérer plusieurs sessions interactives : exécution du programme à distance (Shell, application

commandes systèmes, etc.), multiplexage de plusieurs flux (ou canaux), gestion des transferts X11, de port et d'agent.

Le transfert de port TCP permet d'encapsuler des protocoles applicatifs non sécurisé dans SSH de manière à apporter aux applications des services de chiffrement et d'intégrités de manière transparente.

Les fonctionnalités des trois couches de sshv2 sont résumées dans le tableau suivant :

Fonctionnalités des couches sshv2	
Couche	Description
Transport	-authentification du serveur, négociation des algorithmes, mise en place d'une clef de session, intégrité et confidentialité des données, compression, identification de session.
Authentification	-authentification du client (clé publique, mot de passe, clé d'hôte), chargement de mot de passe.
Connexion	-transfert de port TCP et transfert X, transfert d'agent d'authentification ; -gestion des sessions interactives, exécution de programmes distants ; -contrôle de flux, gestion des terminaux (modes et tailles des fenêtres) ; -compression des données ;

«Tableau.II.2. les fonctionnalités des couches SSH-2»

V- Le fonctionnement du protocole SSH :

V-1-Les méthodes d'authentification avec SSH :

Une fois que la connexion sécurisée est mise en place entre le client et le serveur, le client doit s'identifier sur le serveur afin d'obtenir un droit d'accès. Il existe plusieurs méthodes :

V-1-a-Authentification avec mot de passe :

La méthode la plus connue est le traditionnel mot de passe. Le client envoie un nom d'utilisateur (login) et un mot de passe au serveur à travers la communication sécurisée et le serveur vérifie si l'utilisateur concerné a accès à la machine et si le mot de passe fourni est valide.

V-1-b-Authentification par clés :

En plus de l'authentification traditionnelle par login/mot de passe, SSH peut authentifier l'utilisateur grâce à des algorithmes de cryptographie à clés publiques (RSA, DSA). L'utilisateur place sa clé publique sur les serveurs SSH sur lesquels il souhaite se connecter et garde sa clé privée sur sa station de travail. Lors d'une connexion vers un serveur, le client SSH utilise la clé privée du poste local pour prouver au serveur l'identité de l'utilisateur.

V-1-c-L'authentification par hôte (hostbased) :

Il s'agit d'une authentification similaire à celle utilisée par les commandes et les fichiers tels que /etc/rhosts et /.rhosts, qui certifient les sites client en ayant préalablement enregistré leur adresse dans le serveur. En effet, avec cette méthode d'authentification, quand le client demande une connexion à un serveur SSH, ce dernier va chercher dans le fichier rhosts un nom d'hôte qui correspond à l'adresse source de la connexion réseau du client. S'il trouve un nom d'hôte

qui correspond, le serveur vérifie que le programme demandé par le client est autorisée. Ceci tout simplement en vérifiant si le port en écoute est entre 1 et 1023.Si tout passe bien, l'authentification se poursuit, sinon, elle échoue.

V-1-d-L'authentification par certificat X.509 :

La distribution des clés publiques ou asymétriques n'est pas toujours sûre, pour être certain de leur provenance il faut utiliser un canal de transmission fiable.

Supposons qu'Alice veuille transmettre sa clé publique à BOB, elle l'envoie en claire sur le réseau.

Charlie, sur le même réseau, intercepte la clé sans la transmettre, et envoi à la place sa clé à BOB.

Les messages d'Alice seront alors refusés car la signature sera mauvaise et les messages envoyés par Charlie seront identifiés comme provenant d'Alice.

La seul méthode pour remédier à ce problème est l'utilisation des certificats X.509 fondés sur une infrastructure de gestion de clefs (ou PKI). Un certificat X.509, fourni par une autorité de confiance permet à la fois d'authentifier un individu, un serveur, une entreprise, ou toute autre entités et d'associer l'identité de cette entité avec sa clef publique.

Un certificat X.509 fourni principalement à son émetteur deux services de sécurité :

L'authentification forte et la non répudiation des transactions ou des données.

V-2-Mise en place d'un canal sécurisé :

V-2-1-Utiliser un proxy :

Un proxy, ou « serveur mandataire » ou « bastion » en français, est un serveur informatique dont le rôle est de servir de relais entre un client et un serveur. les entreprises utilisent très souvent un proxy, afin de pouvoir contrôler les sorties de leurs employés sur internet. Quand vous vous connectez à internet à partir du poste de travail, il se peut qu'une boite de dialogue s'ouvre et vous demande un identifiant et un mot de passe pour surfer sur internet : c'est le proxy qui demande cette authentification pour vous autorisez ou non l'accès au site désiré.

Un proxy n'assure pas un anonymat complet. C'est pour cela qu'il ne faut pas s'appuyer sur certains serveurs quant à leur anonymat.

SSH Proxy est un projet libre de serveur mandataire permettant de centraliser les connexions SSH et d'y appliquer des règles de contrôle d'accès.

Une fois installé sur un réseau, plus souvent en zone démilitarisée (DMZ), derrière le pare-feu, le SSH Proxy permet d'authentifier les utilisateurs qui accèdent au réseau en SSH, pour leur ouvrir l'accès aux équipements autorisés sans avoir à communiquer les mots de passe de ces équipements cibles.

Les utilisateurs n'ont plus qu'un seul mot de passe pour se connecter au SSH Proxy et accéder à leurs machines à distance.

V-2-2-Utiliser un tunnel SSH :

Il est aujourd'hui impossible d'ouvrir certains services directement sur Internet. Les attaques sont trop courantes. Nous utilisons donc des procédés de filtrage interdisant l'accès à nos serveurs depuis l'extérieur.

Cependant, cela peut empêcher les utilisateurs d'utiliser ces mêmes services. Pour contourner ce problème, sans pour autant compromettre la sécurité de l'infrastructure, nous allons utiliser le principe de tunnel SSH. La mise en œuvre d'un tunnel est un peu plus complexe que l'utilisation standard d'une application.

Un tunnel représente une connexion traversant plusieurs interfaces de manière transparente pour le client et le serveur. L'utilisation de tunnel SSH peut servir à différents buts tels que la sécurisation d'un protocole non crypté.

- **La création d'un tunnel sécurisé avec SSH :**

SSH combine cryptage asymétrique et cryptage symétrique.
SSH utilise les deux cryptages : asymétrique et symétrique. Cela fonctionne dans cet ordre :

- On utilise d'abord le cryptage asymétrique pour s'échanger discrètement une clé secrète de cryptage symétrique.
- Puis ensuite on utilise tout le temps la clé de cryptage symétrique pour crypter les échanges.

Le cryptage asymétrique demande beaucoup trop de ressources au processeur.il est 100 à 1000 fois plus lent que le cryptage symétrique.

Les ordinateurs s'échangent donc la clé de cryptage symétrique de manière sécurisée (grâce au cryptage asymétrique) et ils peuvent ensuite communiquer plus rapidement en utilisant tout le temps du cryptage symétrique.

Le cryptage asymétrique est donc utilisé seulement au début de la communication, afin que les ordinateurs s'échangent la clé de cryptage symétrique de manière sécurisée.

Ensuite, ils ne communiquent que par cryptage symétrique.

VI- Les fonctionnalités offertes par SSH :

Le protocole SSH implémente relativement un certain nombre de fonction tel que le Shell sur le système distant, le transfert de fichier et le port forwarding.

VI-1-l'accès à distance par Shell SSH :

Le Shell SSH (la commande SSH) est une version sécurisé de rsh et rlogin. SSH veut dire Secure Shell à l'image de rsh qui veut dire remote Shell. Quand rsh permet d'obtenir un Shell distant aisément ; mais sans mécanisme d'authentification satisfaisant (du point de vue de la sécurité), SSH procure le même service de façon sécurisé. Ainsi, pour utiliser SSH, il suffit d'utiliser la commande SSH à la place des commandes Telnet, rsh et rlogin.

VI-2-le transfert de fichier par SFTP :

SFTP (Secure File Transfer Protocol) est un sous protocole séparé qui se situe au dessus du protocole SSH. Il est utilisé dans le transfert sécurisé des fichiers. SFTP a plusieurs avantages par rapport au protocole non sécurisé FTP.

D'abord, SFTP chiffre le couple user-name/password ainsi que les données transférées en se basant sur des algorithmes cryptographiques. ce qui élimine la nécessité d'ouvrir un autre port sur le pare-feu. Ainsi l'utilisation de SFTP résout également le problème connu dans le protocole FTP.

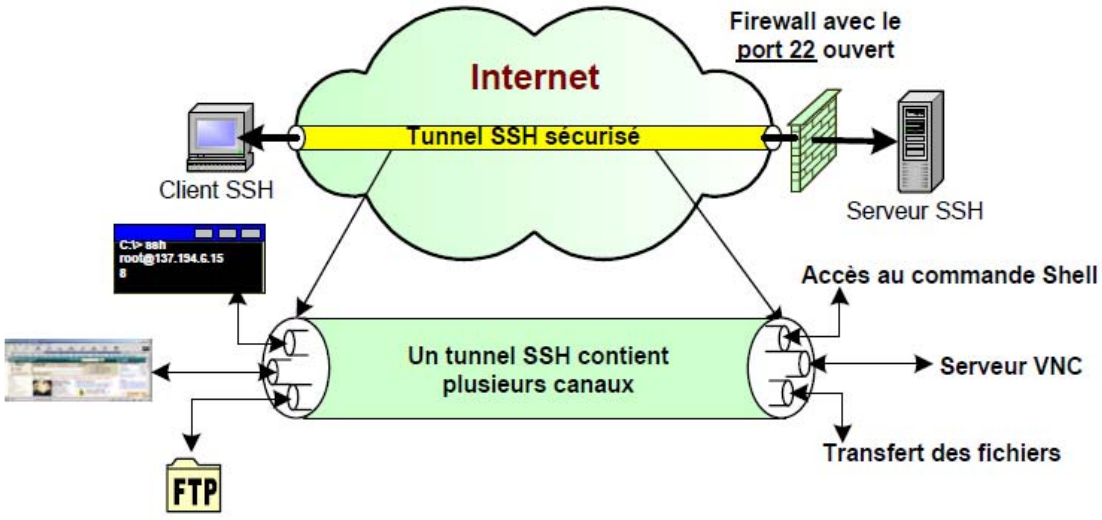
VI-3-le tunneling :

Le transfert, ou tunneling, consiste à encapsuler un autre service TCP/IP comme Telnet, dans une session SSH afin de lui apporter les bénéfices de la sécurité de SSH (Confidentialité, intégrité, authentification, autorisation). En transférant Telnet par exemple, via SSH, toutes les données seront chiffrées et

leur intégrité sera contrôlée.

En plus, l'authentification des clients sera assurée d'une manière sécurisée par SSH.

SSH reconnaît trois types de transfert : le transfert de port TCP, le transfert des sessions interactives de type X-Window et le transfert des agents SSH qui permet aussi d'utiliser des clés privées SSH sur des machines distantes.



« figure.II.1. Tunnel et canaux SSH »

VI-4-la redirection de port (port forwarding) :

Le SSH permet de rediriger n'importe quel flux TCP dans le tunnel de la session SSH.

Cela veut dire que le flux de n'importe quelle application circulant entre les ports client et serveur habituels, pourra être encapsulé à l'intérieur du tunnel créer par la session SSH (figure-2-).

Cette fonctionnalité permet d'établir entre deux point un canal sécurisé par lequel peut transiter n'importe quelle type de données IP.

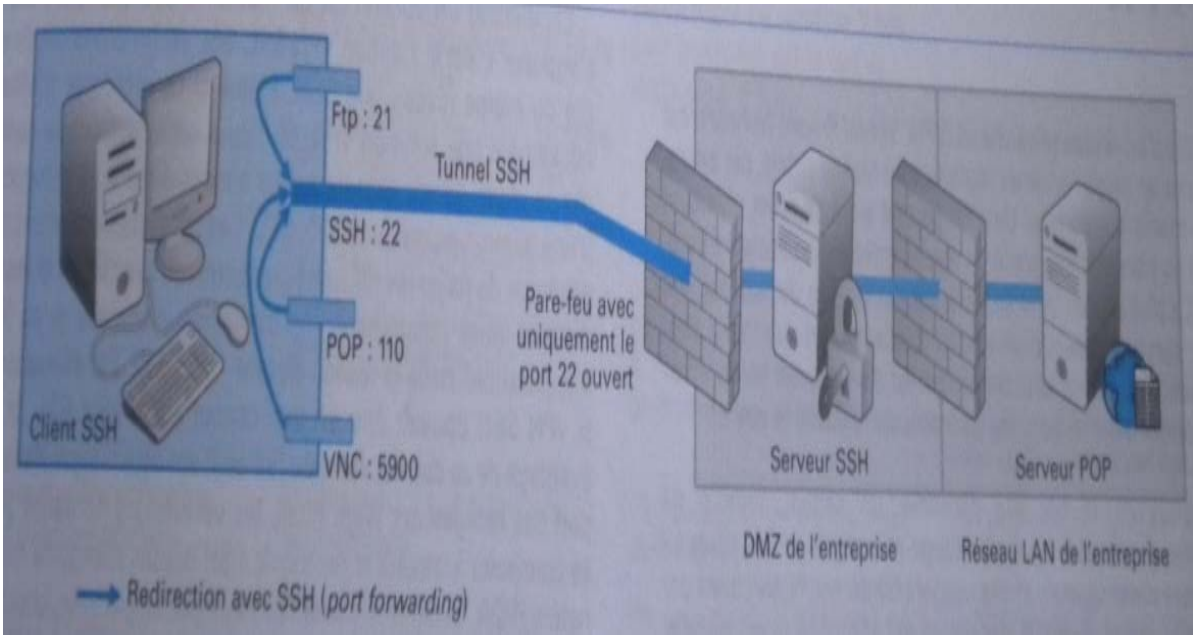
Le principe est simple. Prenant l'exemple de la mise en place d'un tunnel IP afin de sécuriser les connexions entre un client et un serveur POP situé dans l'intranet d'une entreprise et derrière un serveur SSH.

Les étapes sont les suivantes :

-Etablissement d'une connexion SSH entre le client et le serveur SSH.

-Sur le client : faire pointer le client POP sur le système SSH.

-Sur le serveur : transmettre les données arrivant depuis la connexion SSH au serveur POP.



« Figure.II.2. Port forwarding avec SSH »

VI-5-la redirection de l'authentification (agent forwarding) :

L'agent SSH est un mécanisme d'authentification auprès de multiples serveur SSH qui reconnaisse la clef privée d'un client sans devoir retaper à chaque fois sur sa machine la passphrase.

En effet un agent SSH est un programme, qui s'appel SSH-agent, qui garde les

clefs privées en mémoire et qui fournie les services d'authentification au client SSH.

Cette méthode permet à un utilisateur d'introduire sa passphrase lors de la première connexion à un serveur SSH.

L'ouverture d'une session sécurisée avec un nouveau serveur SSH se fait de manière transparente pour l'utilisateur. Si l'utilisateur souhaite, par exemple, faire une copie de fichier (SCP) entre deux serveurs distants (figure-3-), SSH offre une fonctionnalité qui s'appel agent forwarding qui permet à des machines distantes d'accéder à l'agent local de l'utilisateur pour pouvoir récupérer ses droit d'accès et d'exécuter ces programmes distants (dans ce cas c'est le programme SCP).

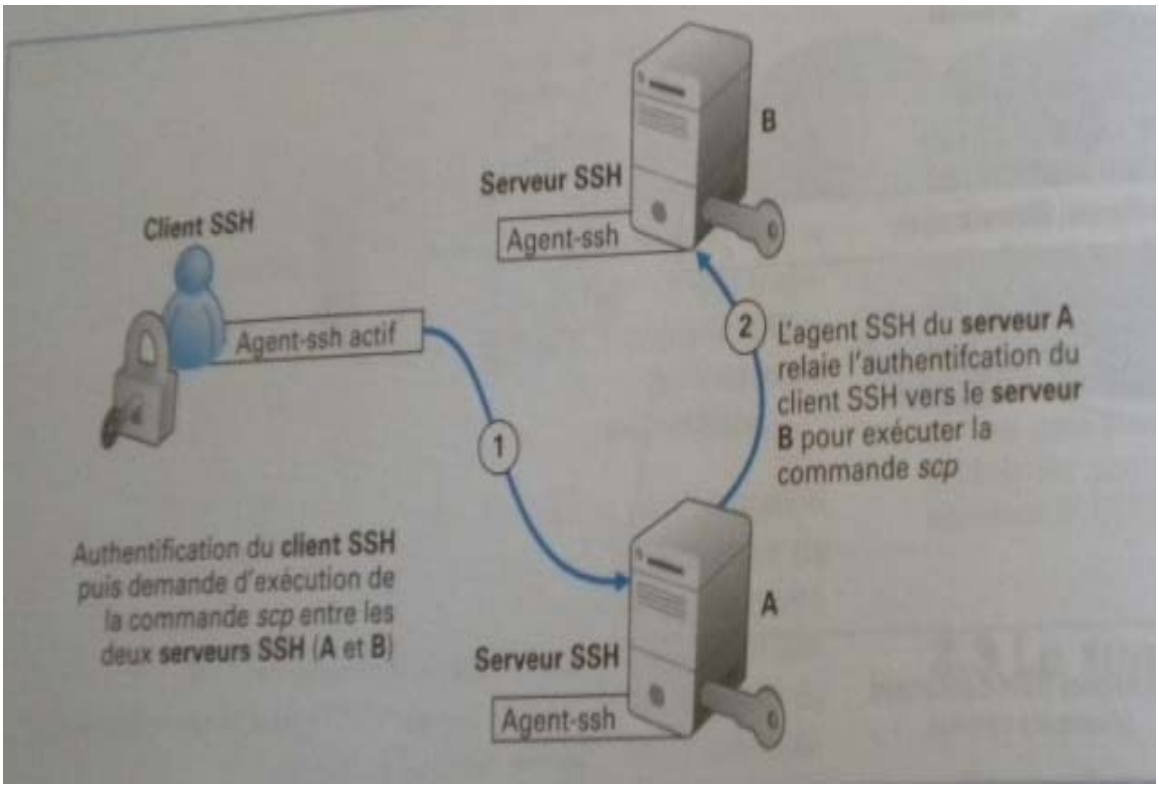
La clef privée de l'utilisateur reste toujours en local. L'agent SSH ne fait que relayer les requêtes d'authentification vers l'hôte distant pour identifier l'utilisateur. Le serveur distant se comporte comme un deuxième programme SSH-agent.

Ainsi le client ne retape pas son mot de passe une deuxième fois puisque les serveurs vont communiquer directement avec l'agent du client.

Notons qu'en activant cette fonctionnalité, l'utilisateur peut se connecter à un réseau SSH, éliminant les problèmes de compromission de sa clef privée.

L'utilisation de l'agent SSH est plus sûre que la mise en œuvre des clefs sans protection par « passphrase », mais cela ne semble pourtant par suffisant.

Par mesure de sécurité et en fonction de l'endroit d'où l'utilisateur se connecte, il est préférable de prendre la peine de saisir manuellement les phrases à chaque authentification.



« Figure.II.3. Authentification par agent et exécution de la commande scp entre deux serveurs SSH »

VIII-Conclusion :

Le protocole SSH constitue une approche puissante et pratique pour protéger les communications sur un réseau d'ordinateur. A travers son mécanisme d'authentification, SSH permet d'effectuer dans un tunnel sécurisée des connexions à distance, des transferts de fichiers et d'autre fonctionnalité importante telle que le tunneling et la redirection de port.ces deux derniers ont fait de SSH le protocole de sécurité le plus transparent pour les applications.

En effet une application non sécurisé peut bénéficier de l'ensemble des services de sécurité de SSH sans aucun changement/ajout dans son code ou noyau. C'est SSH tout seul qui va maintenir la sécurité des données applicative, l'authentification des entités communicantes, le chiffrement et le contrôle d'intégrité de donnée.

I-Introduction :

Jusqu'à présent, la meilleure utilisation du protocole SSH se faisait dans le cadre de la sécurisation des connexions à distance vers les serveurs d'application ou les machine réseau tel que les routeurs et les points d'accès. Cependant la plupart de ces équipements réseau proposent actuellement un mécanisme d'accès simple basé sur une interface web non sécurisé. D'où la nécessité de trouver très rapidement une solution de sécurité simple à déployer pour des utilisateurs non forcément expert dans la configuration de ces équipements réseau. La meilleure solution c'est l'accès à distance par SSH, et c'est ce que nous allons étudier dans ce chapitre.

II-Principe de fonctionnement :

- ✓ Utiliser deux ordinateurs équipés d'un système d'exploitation Windows XP, un pour le client, l'autre pour le serveur.
- ✓ Installer putty dans le poste client, cygwin dans le serveur.
- ✓ Configurer le serveur afin qu'il démarre.
- ✓ Pinguer à partir du poste client pour voir si le serveur est en ligne.
- ✓ Essayer les différentes méthodes d'authentification.
- ✓ Créer un tunnel SSH avec putty.
- ✓ Téléchargement de fichiers avec cygwin.

II-1-Le client SSH:

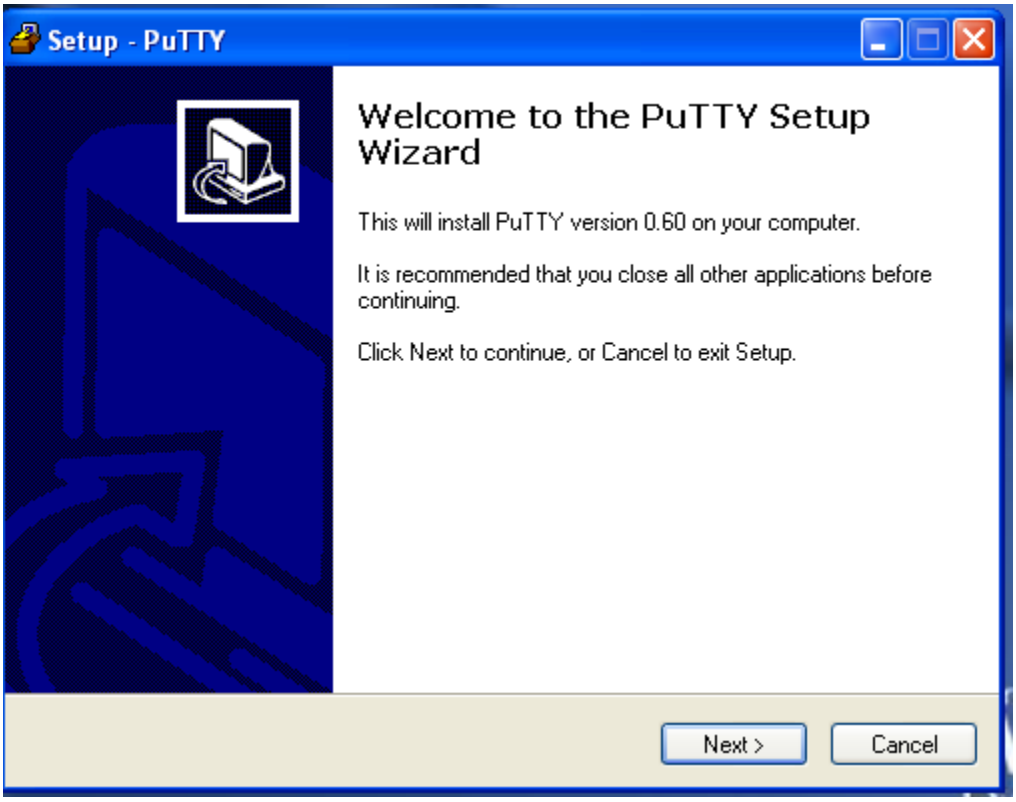
II-1-a-Définition :

PuTTY est le client SSH libre le plus avancé sous Windows, il offre quelques possibilités intéressantes telles que : enregistrement de sessions (adresse du serveur, protocole, login, commande à exécuter à la connexion...), établissement de tunnels (redirection de ports).

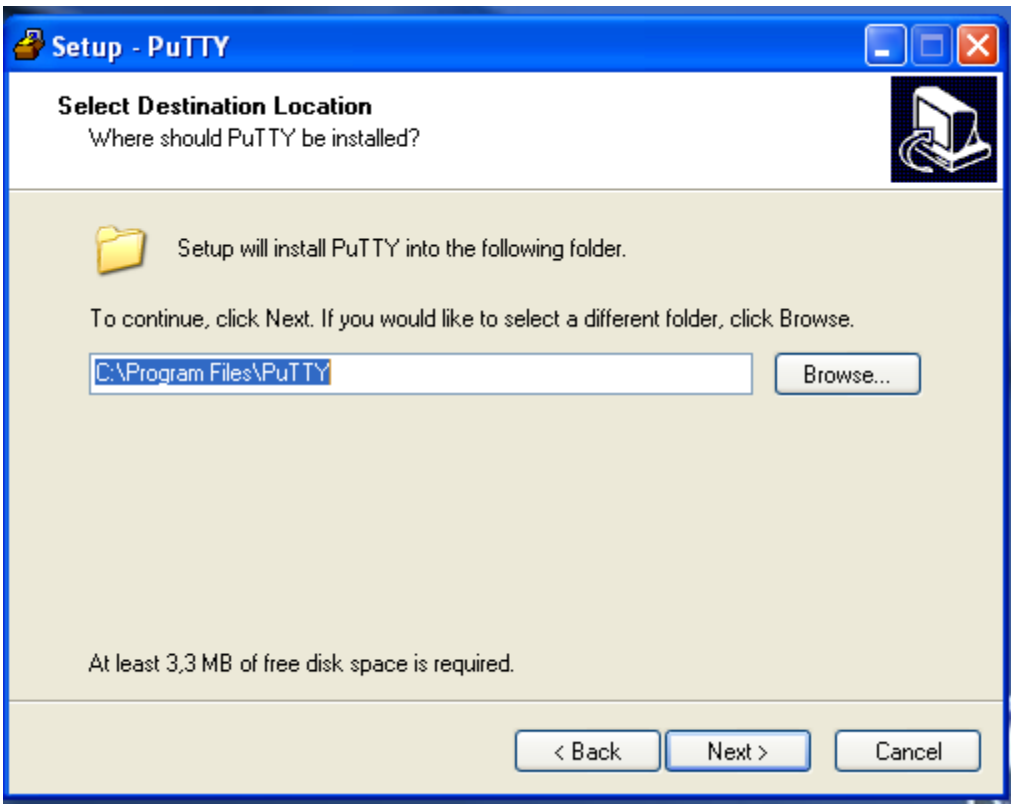
II-1-b-Installation de putty :

Démarrer l'installation de PuTTY en double-cliquant sur l'icone, et suivre les étapes suivantes :

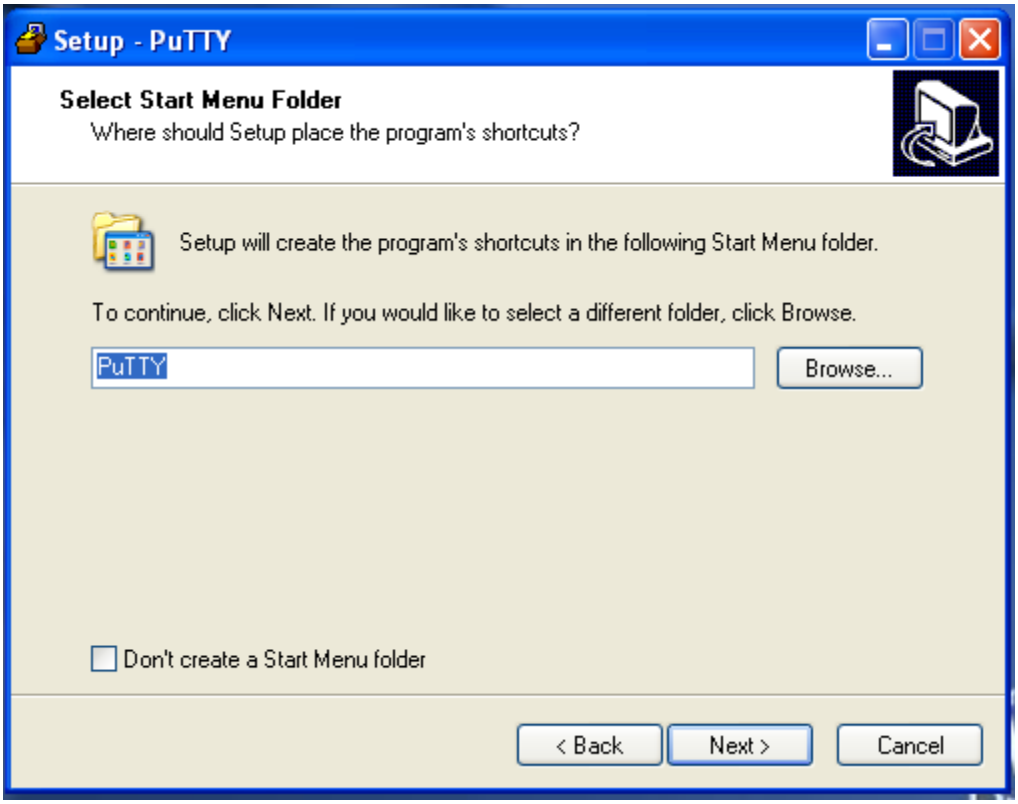
a) Fenêtre de démarrage de l'installation qui donne la version de putty à installer.



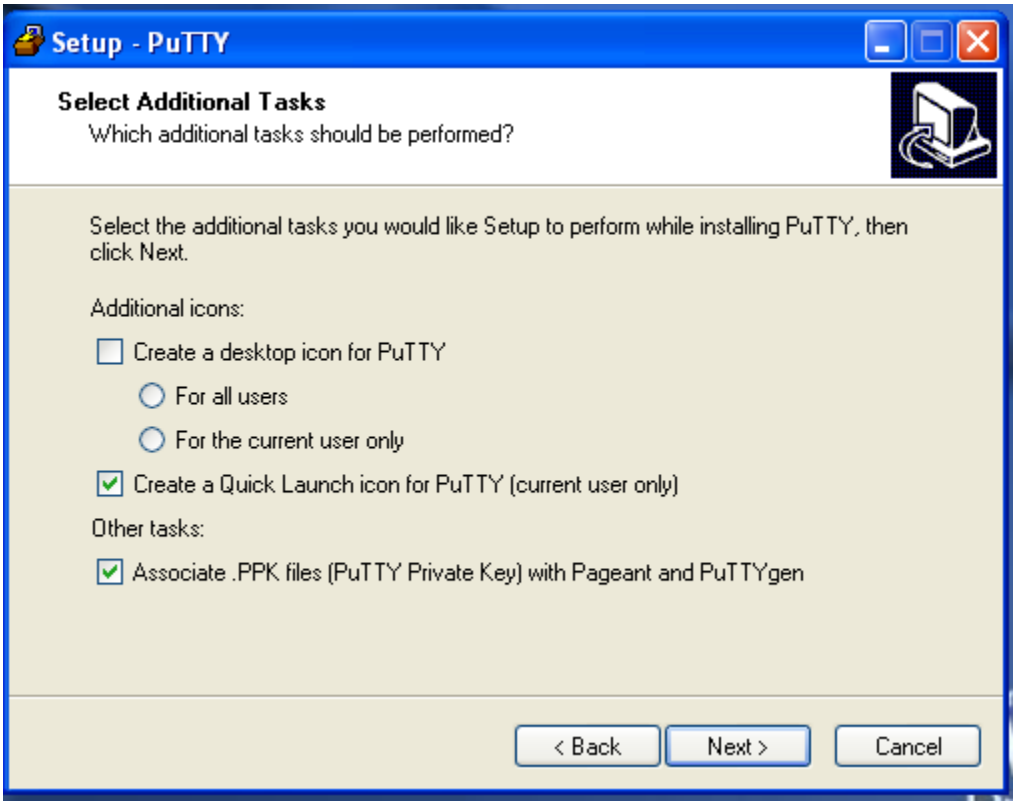
b) Fenêtre « select destination location » : choisissez le chemin où vous voulez installer putty.



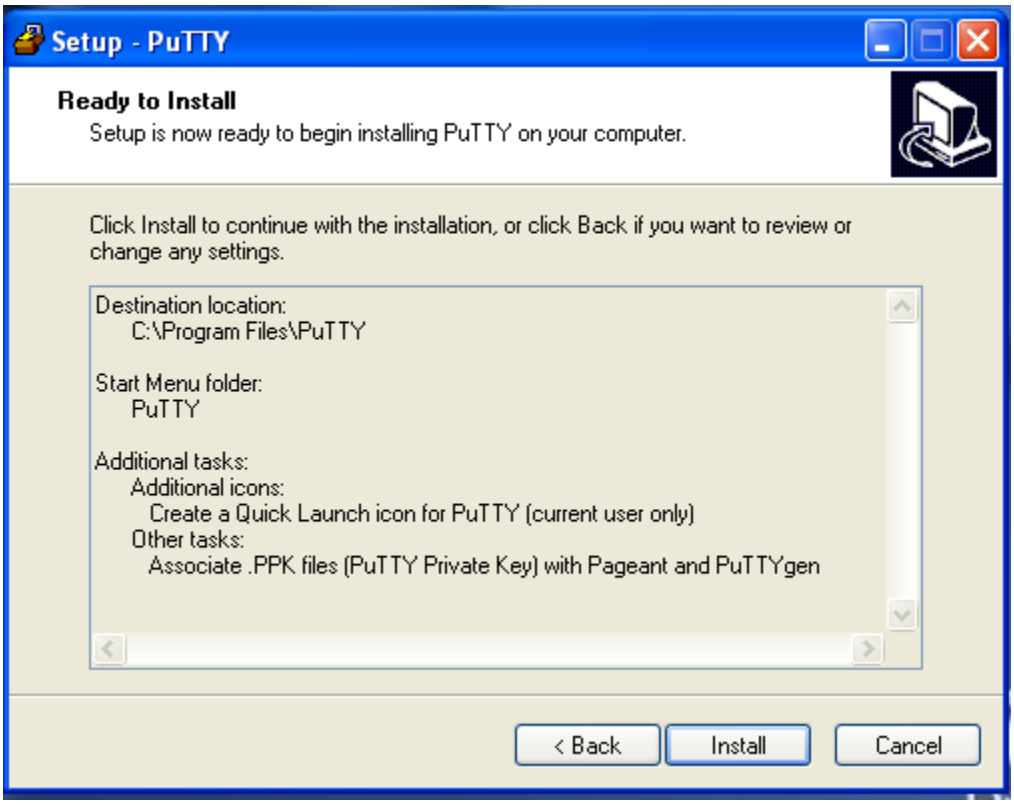
c) Fenêtre « select start menu folder » : choisissez où vous placez le programme putty.



d) Fenetre « select additional tasks » : pour créer un raccourci sur le bureau, et assossiez **.ppk files** avec pageant et puttygen.

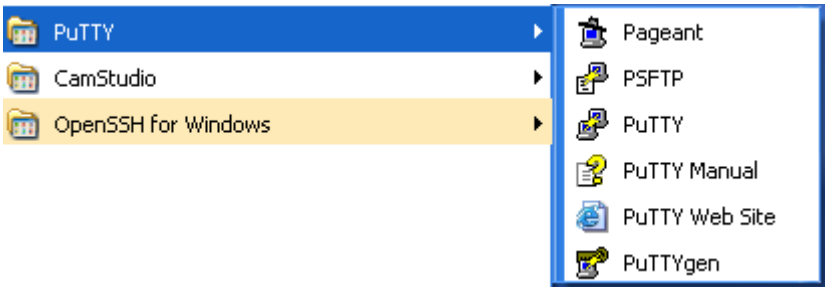


e) Fenêtre « ready to install » : pour lancer l'installation, cliquez sur install.



PuTTY est maintenant installé.

Vous trouvez dans le menu « démarrer-tous les programmes » putty et d'autre logiciel qu'il l' accompagne :



putty : client SSH évolué, permet de créer des sessions et de les configurer de façon graphique.

puttygen : utilitaire permettant de créer, modifier, générer des clés.

pageant : utilitaire permettant de communiquer les clés quand c'est nécessaire.

psftp : client sftp en ligne de commandes ; transfert de fichier sécurisé.

II-2-Installer et configurer un serveur SSH sous Windows XP :

II-2-a-Définition :

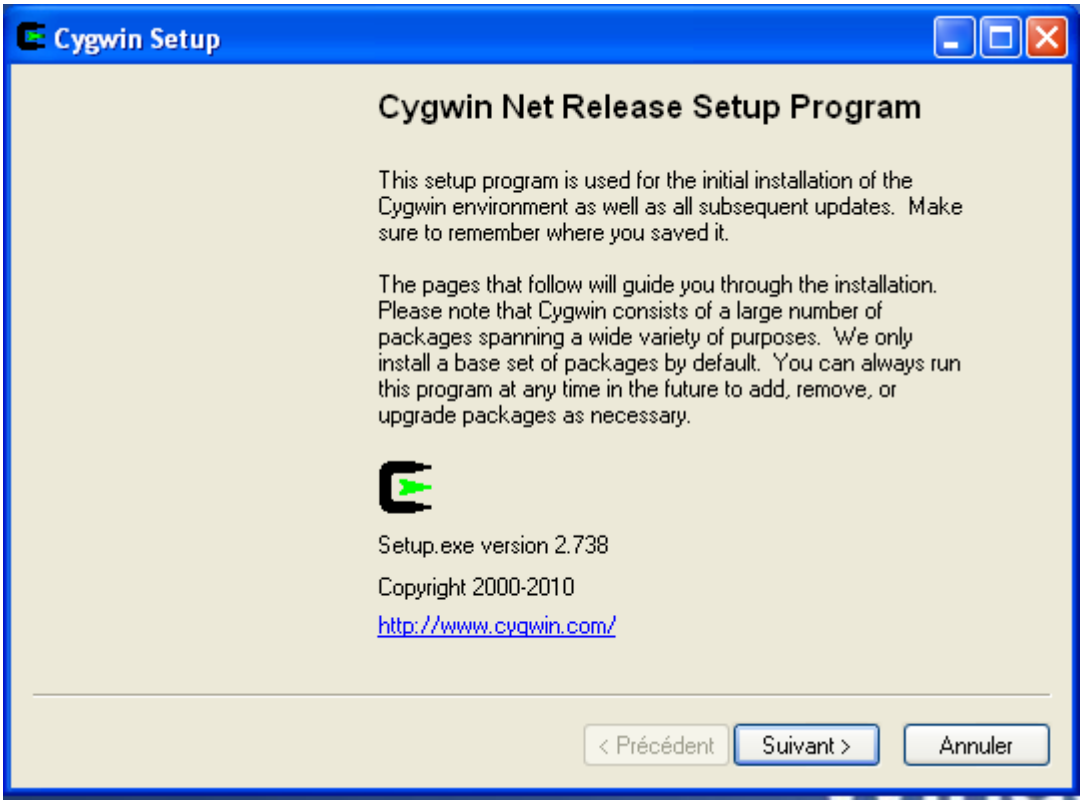
Cygwin est un environnement Unix complet pour Windows. Il dispose d'un serveur graphique et de la plupart des outils Unix (tels que sftp et SSH) en ligne de commande.

Cygwin veille à maintenir ses packages à jour, ce qui est important du point de vue sécurité. De plus l'installeur Cygwin permet de mettre à jour OpenSSH, openssl et toutes les librairies sans avoir à tout réinstaller.

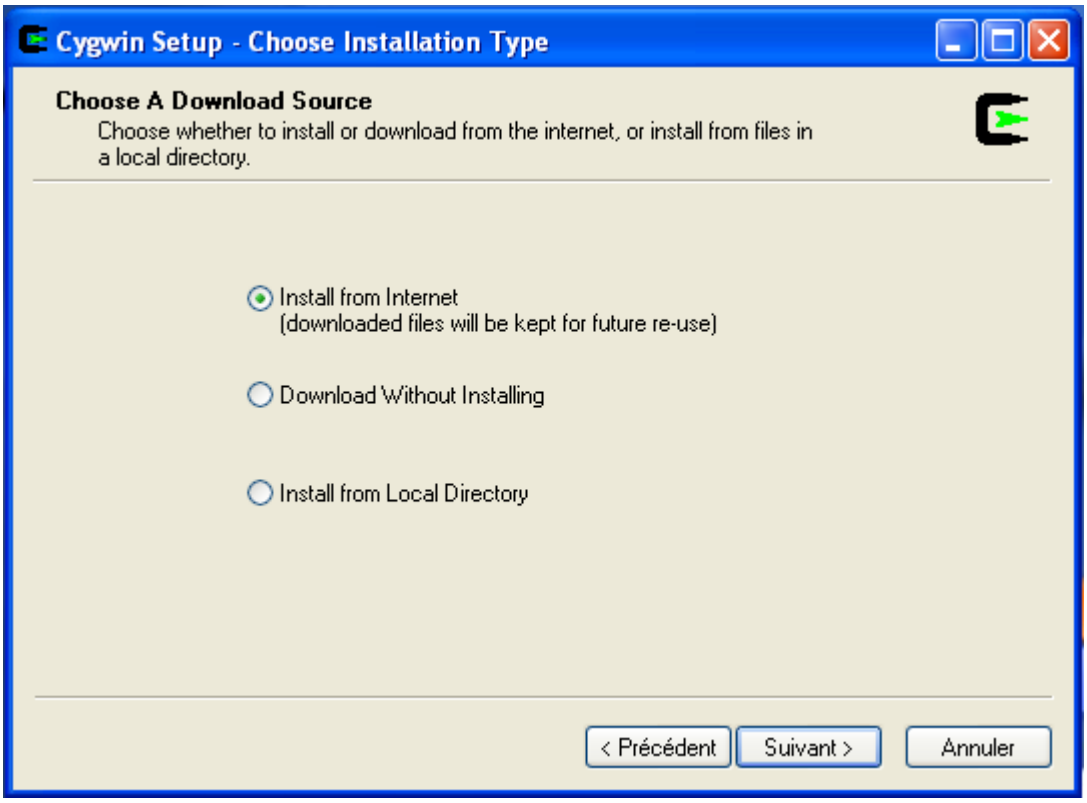
II-2-b-Installation de cygwin :

Téléchargez le fichier 'setup.exe' et exécutez-le. Le fichier contient le programme d'installation, La procédure d'installation démarre.

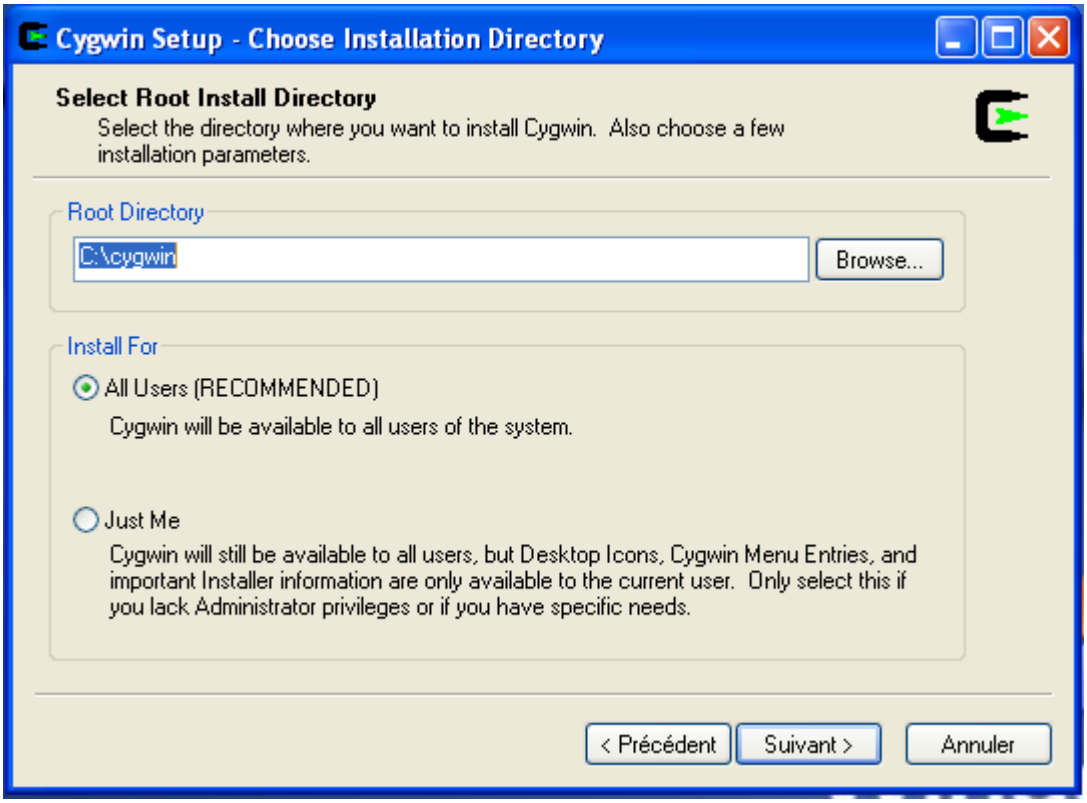
a) Fenêtre de démarrage de l'installation qui donne la version de cygwin à installer.



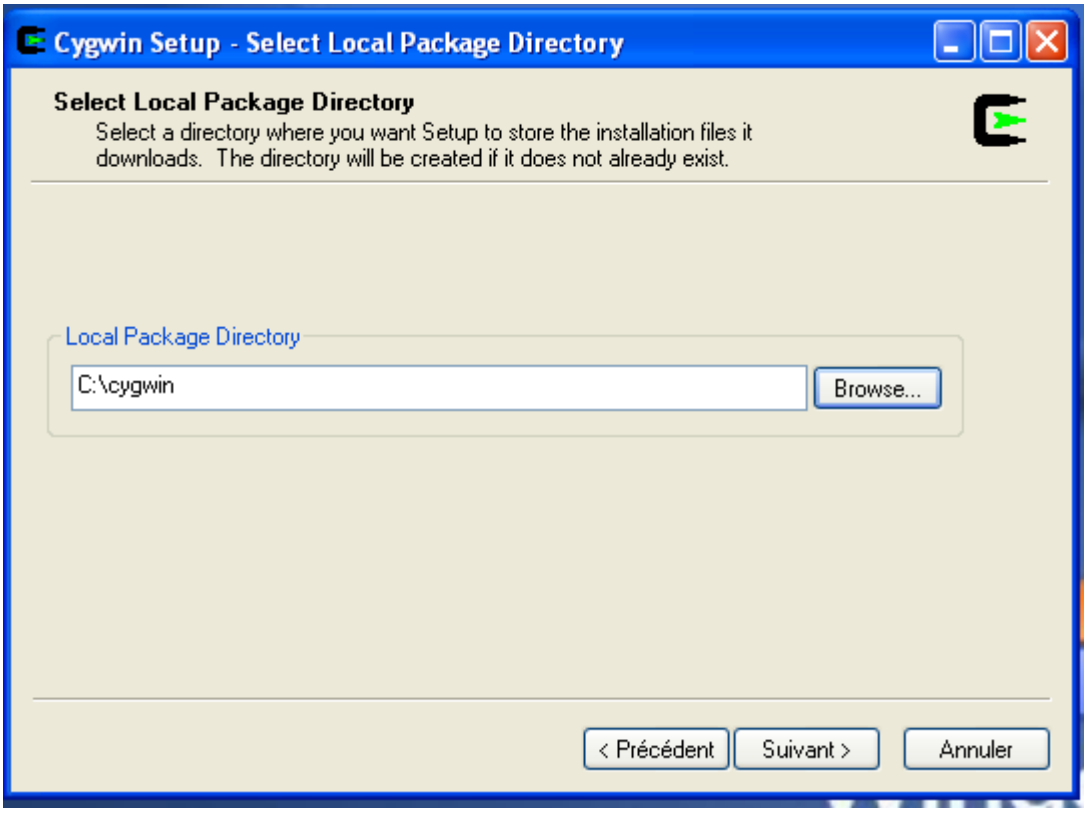
b) Fenêtre "Choose installation type": Choisissez "Install from internet"



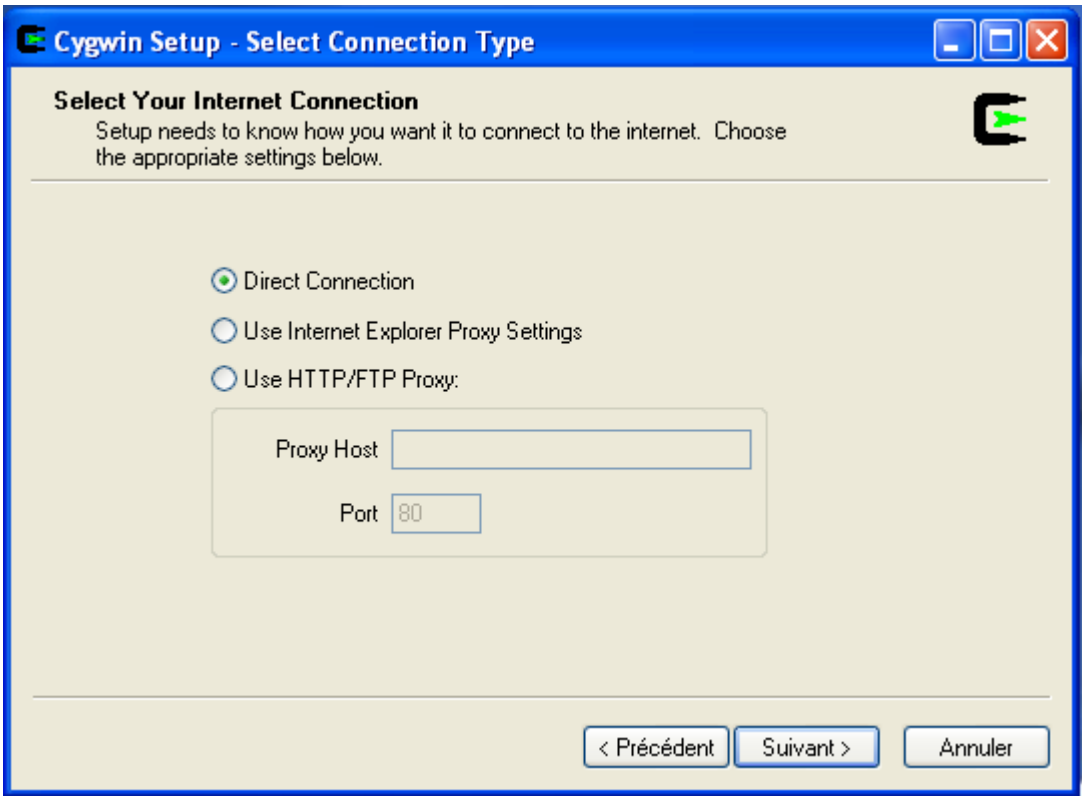
c) Fenêtre "Choose Installation Directory", choisissez le chemin où vous installez cygwin.



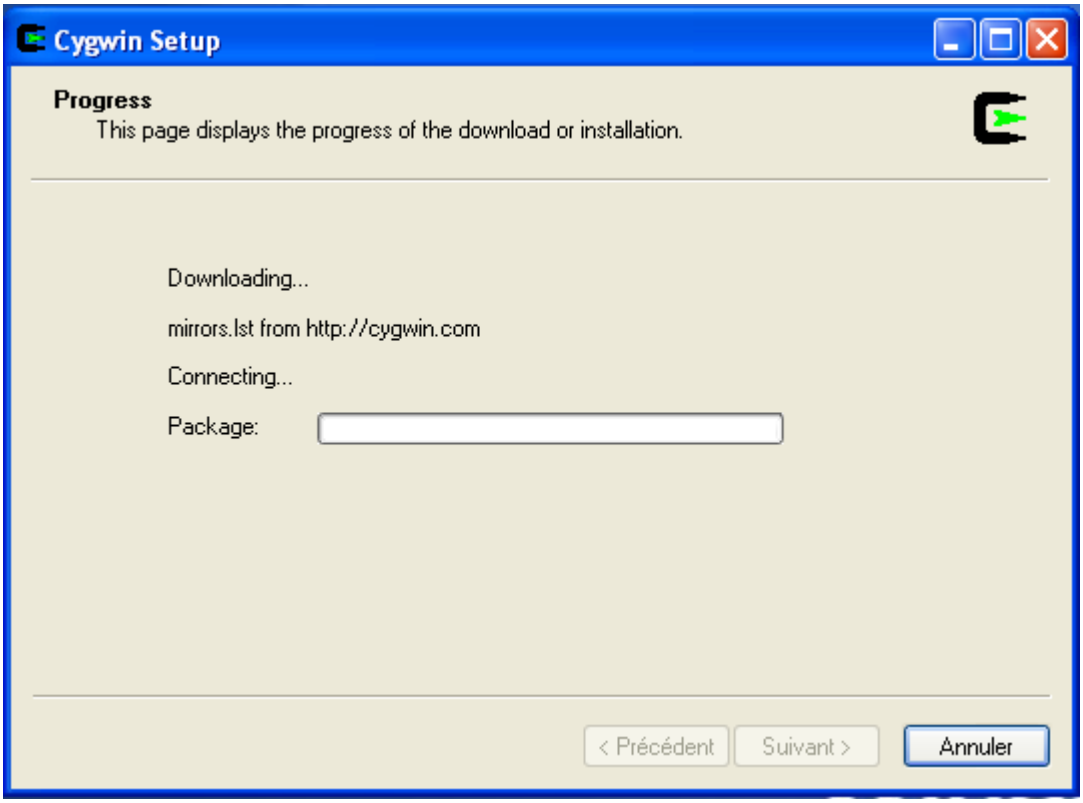
d) Fenêtre "Select Local Package directory", choisissez où vous voulez installer les packages de cygwin.



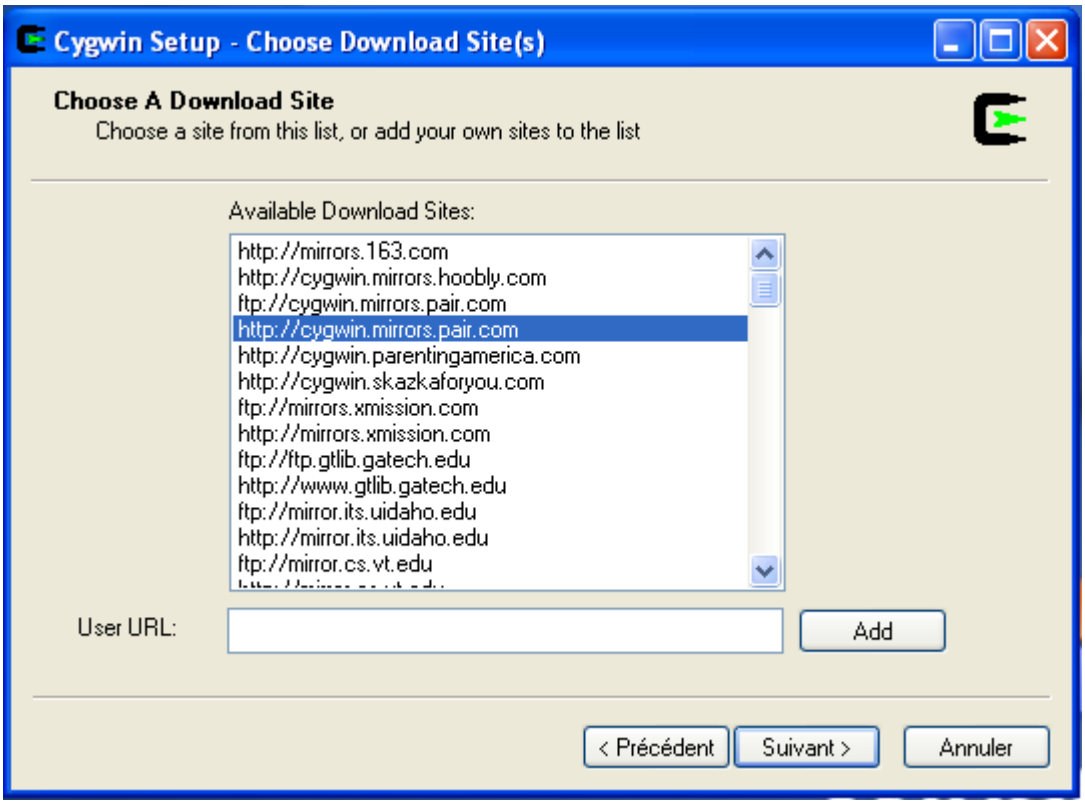
e)Fenêtre "Selection connection type": Entrez d'éventuels paramètres de proxy si vous en utilisez un.



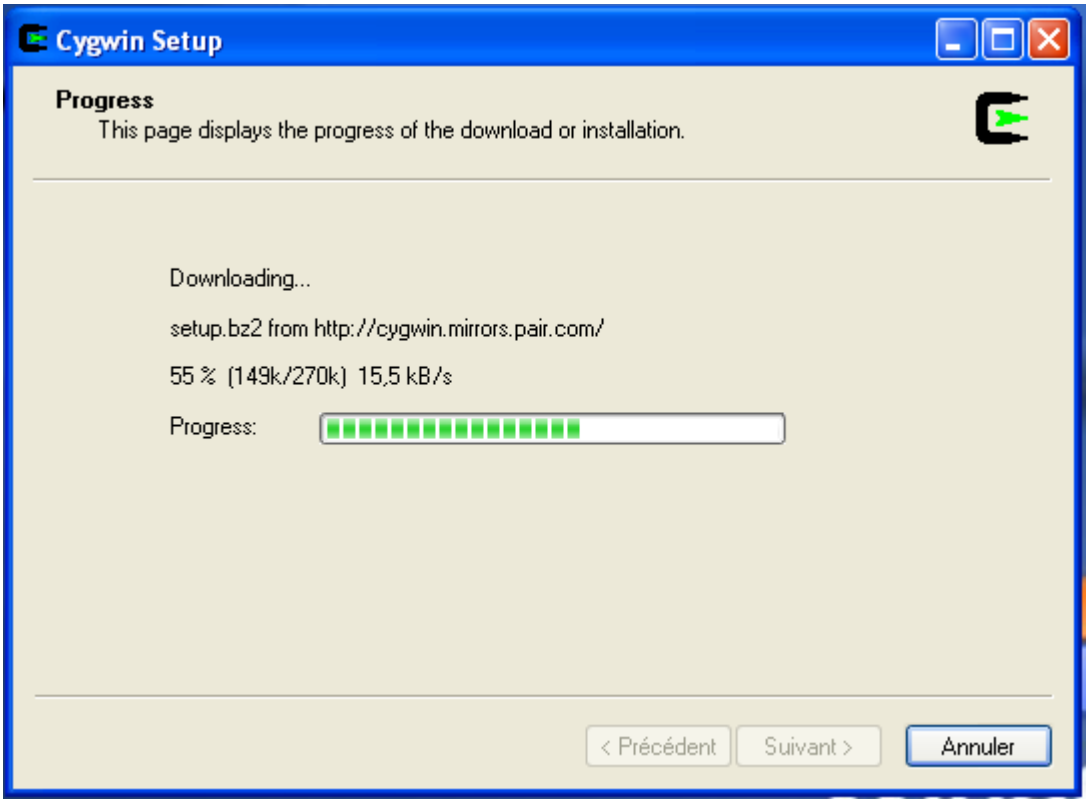
f) Il va se connecter directement et charge les packages nécessaire.



g) Fenêtre "Choose Download Site(s)": Choisissez un site de téléchargement proche de chez vous.



h) Après il va se connecter au serveur désirer :



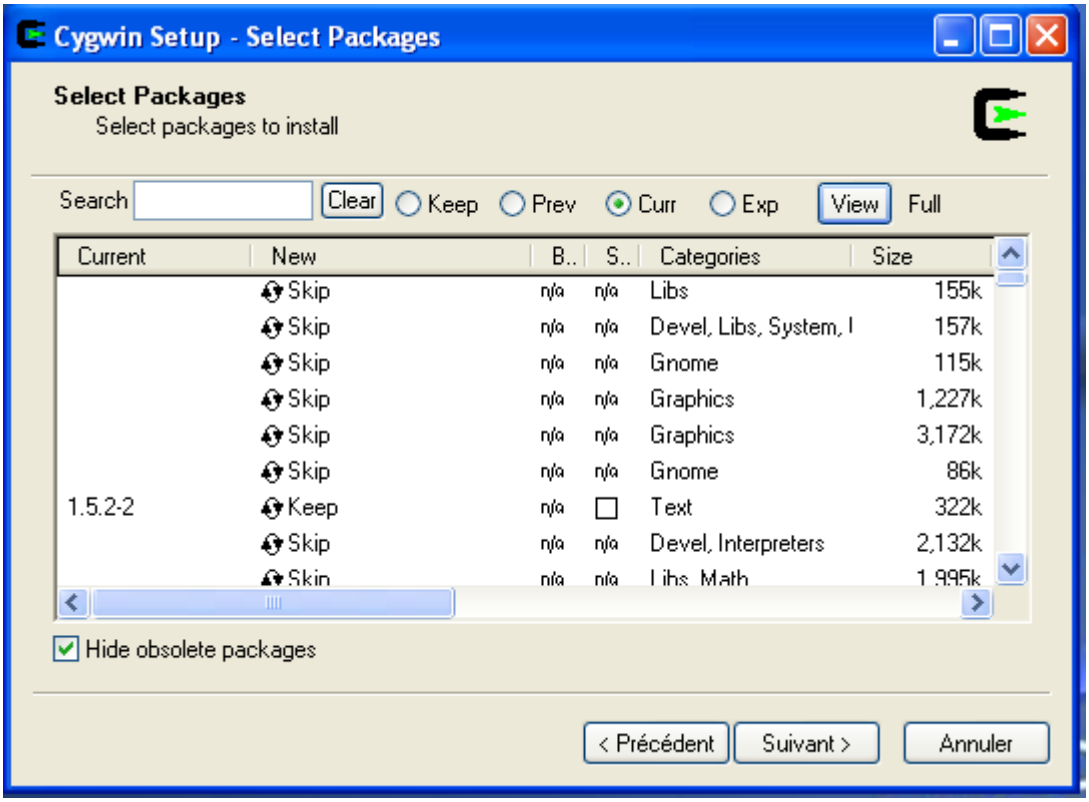
i) Fenêtre "Select packages", cliquez sur le bouton "View" pour voir la liste des paquets. Descendez dans la liste pour trouver "openssh: The OpenSSH server '' et cliquez sur le mot "Skip" pour sélectionner ce package. Le mot "Skip" doit être remplacé par la version d'openssh.

D'autres packages doivent êtres sélectionnés :

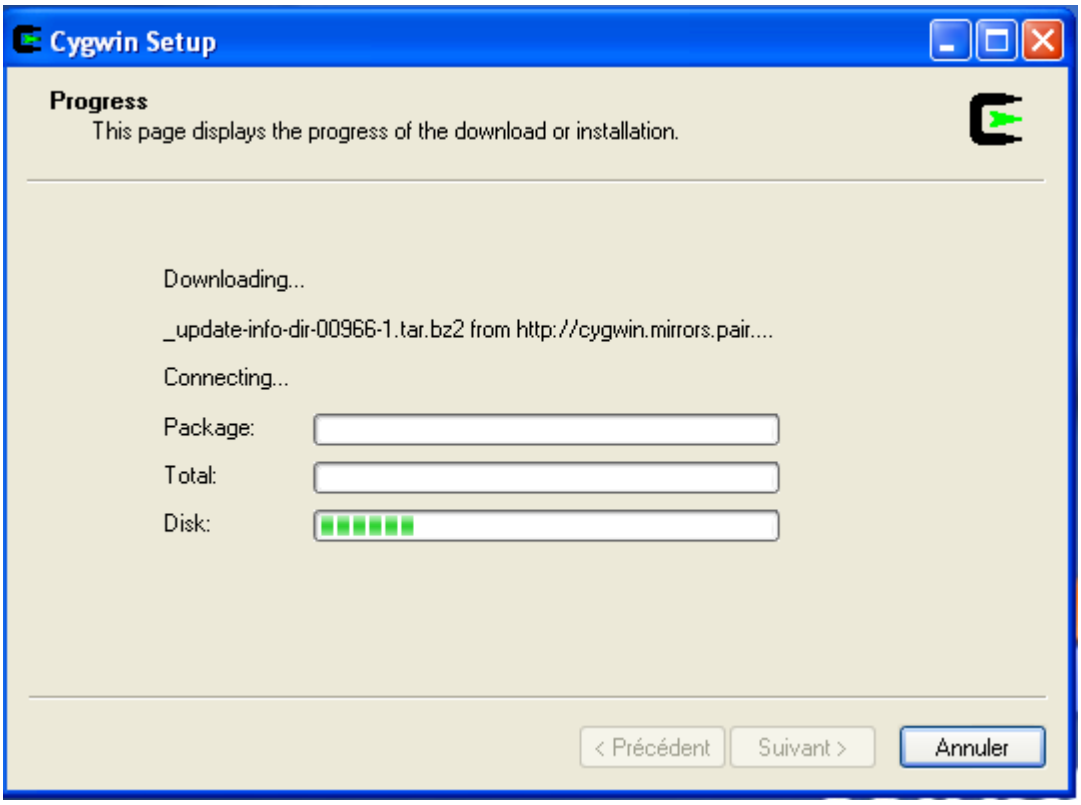
SSH, unison (sauvegarde de fichier) et wget (téléchargement de fichiers, il suffit d'indiquer une adresse HTTP ou FTP d'un fichier à télécharger), qui sont les trois lignes de commandes qui manque le plus sous windows. Ainsi que :

Perl lors de l'installation, pour des applications tiers (ssh copy-id…).

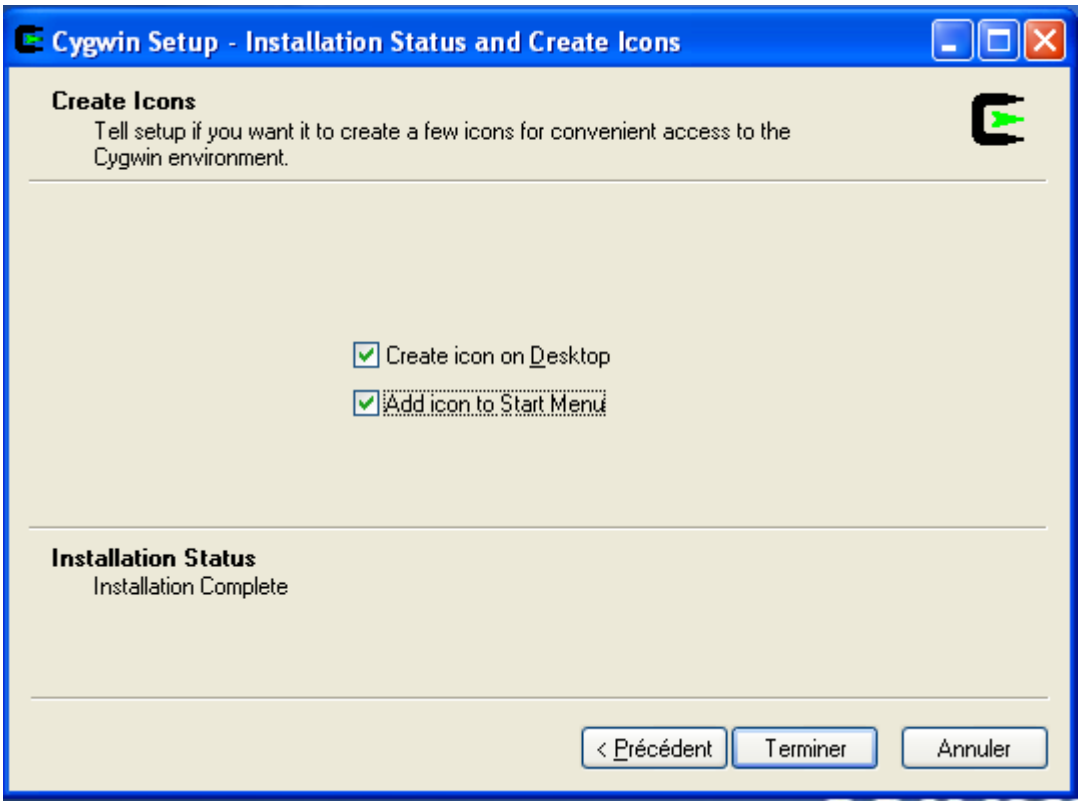
Zlib pour la compression des flux de données.



j) Cliquez sur "suivant": Le téléchargement commence, Il va télécharger environ 17 Mo de fichiers.



k) Fenêtre "Create icons", qui permet de créer les raccourcis et de terminer l'installation de cygwin.



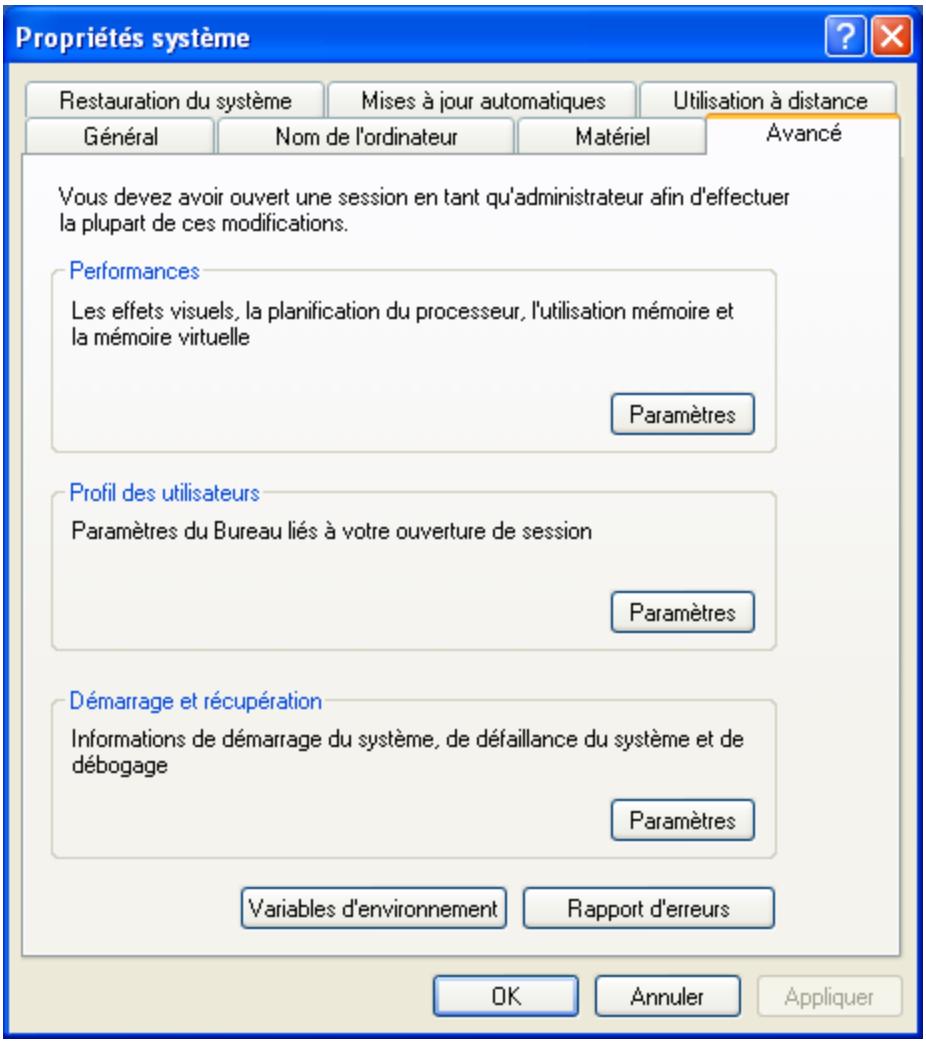
l) Cliquez sur "terminer". L'installation de Cygwin est terminée.

II-3-Configuration du serveur SSH :

II-3-a-Modification de l'environnement :

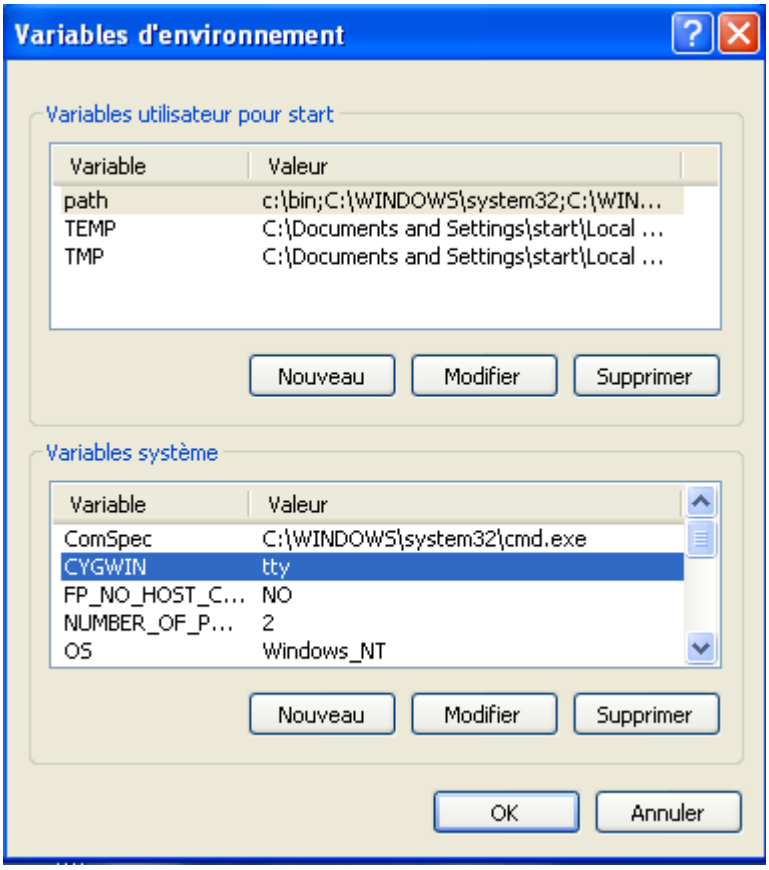
Windows XP dispose des variables d'environnement, notamment le PATH. Il est nécessaire d'éditer le PATH et de rajouter le répertoire des commandes pour Cygwin.

Clique-droit sur le Poste de travail --> Propriétés --> Avancé --> Variables d'environnement

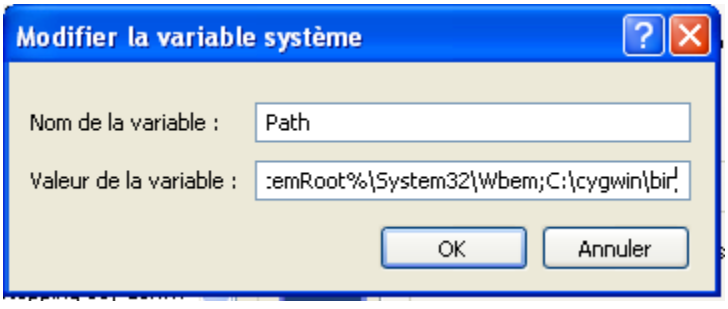


Fenêtre des propriétés système de Windows XP, dans la partie "Variables système", cliquez sur "Nouveau" :

- Le nom de la variable à ajouter est : CYGWIN
- La valeur de cette variable est : ntsec tty



Dans la fenêtre Variable système, sélectionnez PATH dans la liste, cliquez sur "modifier" et ajoutez à la fin du PATH : C:\Cygwin\bin.

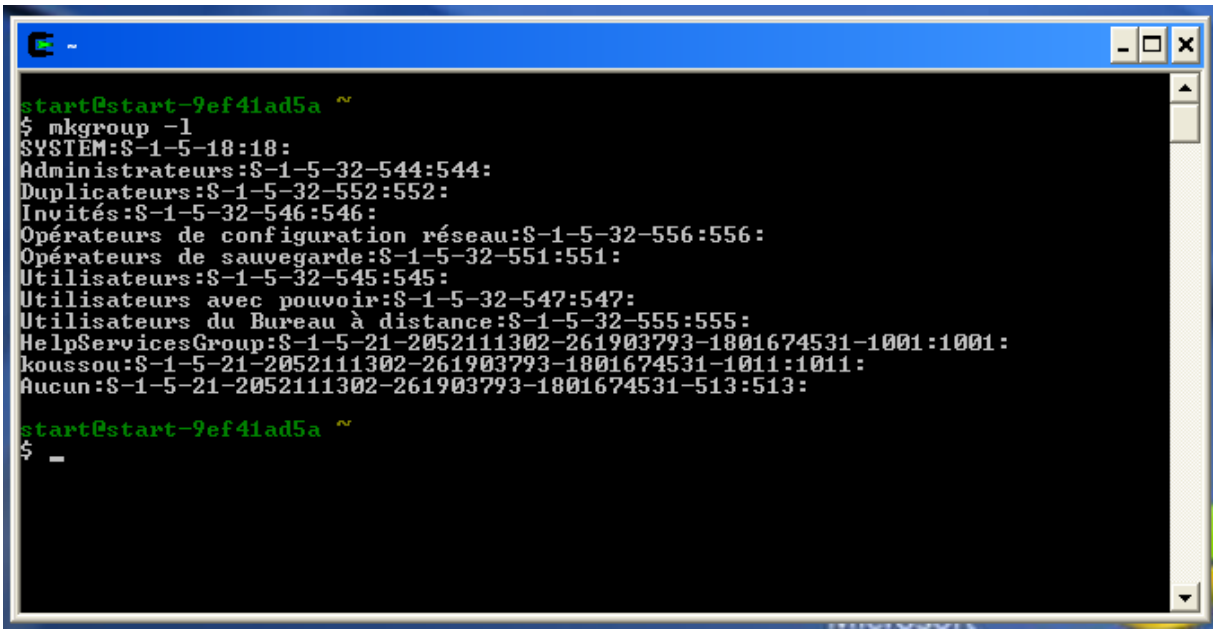


II-3-b- Création des groupes et utilisateurs :

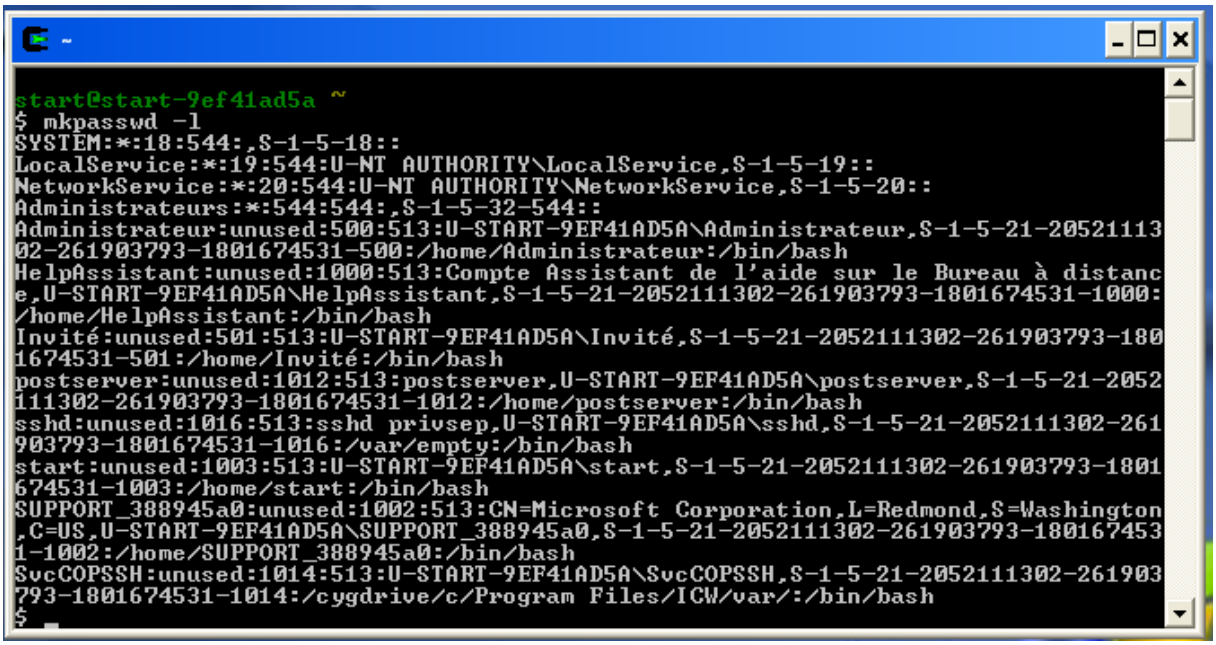
Ouvrez la fenêtre Cygwin.

Créez les groupes et les utilisateurs:

- Groupes: mkgroup -l



- Utilisateurs: mkpasswd -l



Cela va prendre les users et les groupes de Windows et les créer dans les fichiers correspondants à Cygwin.

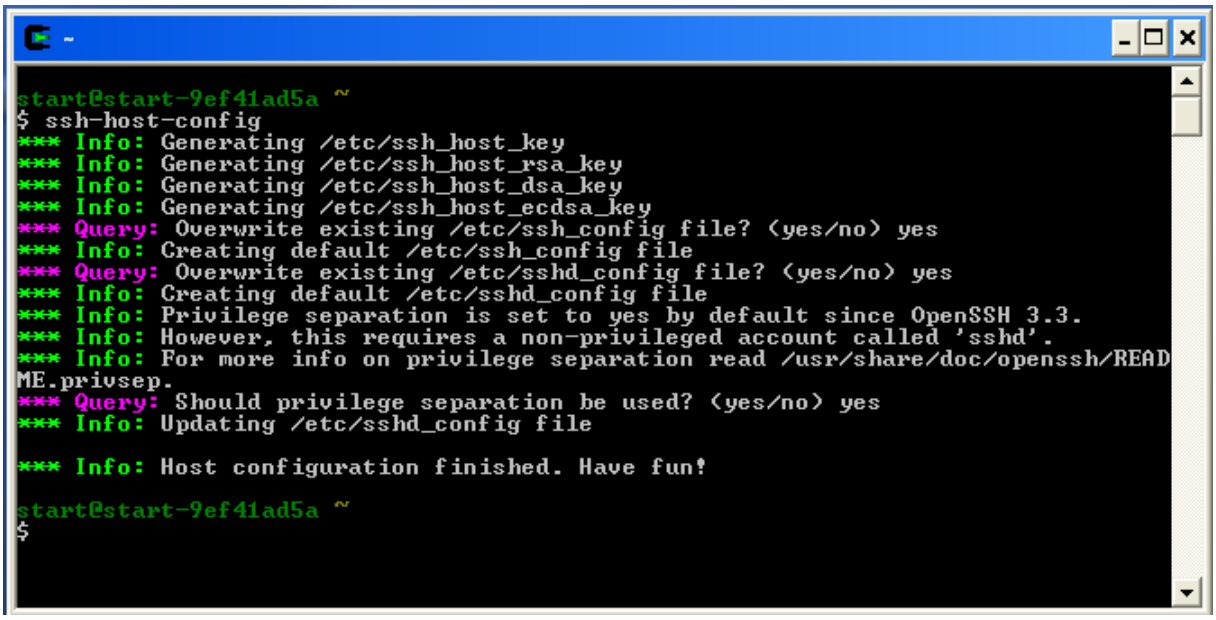
Il faut contrôler le contenu des fichiers passwd et group. Si ces fichiers sont vides, le serveur SSH ne fonctionnera pas.

Si un utilisateur et son groupe ne sont pas déclarés dans ces deux fichiers, il ne pourra pas se connecter.

Si vous avez un message d'erreur sur mkpasswd ou mkgroup, inutile de poursuivre l'installation: vous devez d'abord résoudre ce problème avant de continuer.

II-3-c-Lancement du service sshd:

Pour lancer le service sshd, vous tapez la commande suivante : **ssh-host-config**



Les fichiers de clés du serveur sshd :

ssh_host_key : ssh v1, RSA1

ssh_host_ rsa _key : v2, RSA

ssh_host_ dsa _key_: v2, DSA

ssh_host_ ecdsa _key_: v2, ECDSA

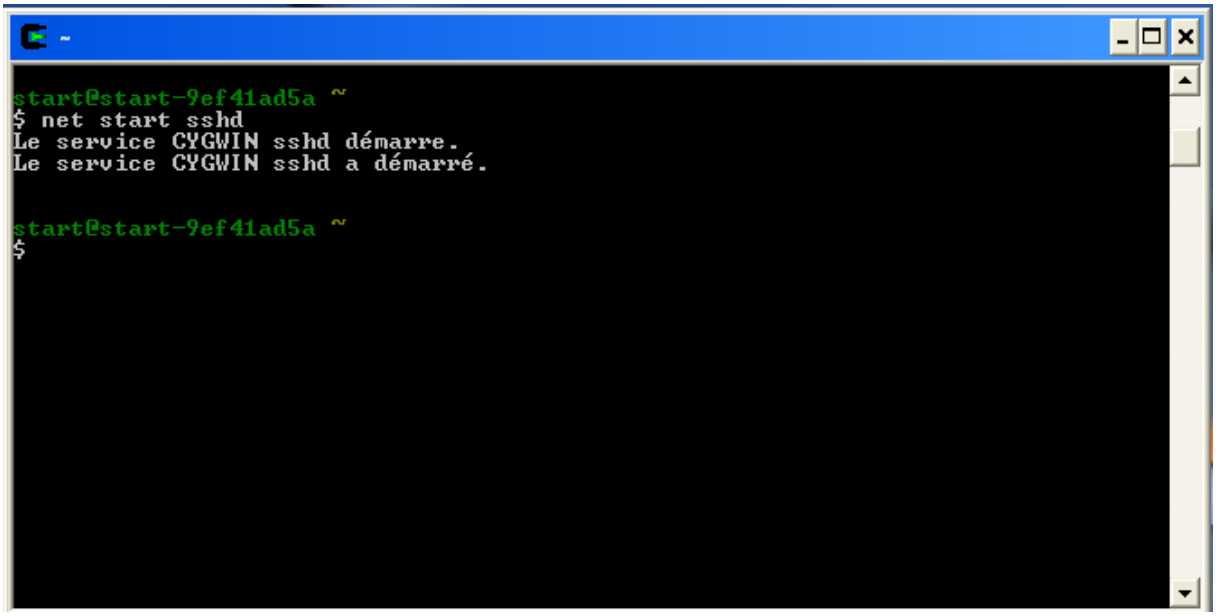
ssh_config: vérifie que la clé publique de l'hôte distant existe sur l'hôte qui cherche à se connecter et ensuite autorise la demande de connexion. Une non connaissance de la clé ou un changement de clé du serveur implique alors un échec de la connexion SSH.

sshd_config: Filtrage d'utilisateurs ou des groupes. Permettre l'accès à certains utilisateurs et pas à d'autres pour un accès SSH.

Le service sshd est en principe configuré pour démarrer automatiquement, mais il n'est pas encore démarré.

Pour le démarrer tapez la commande: **net start sshd**

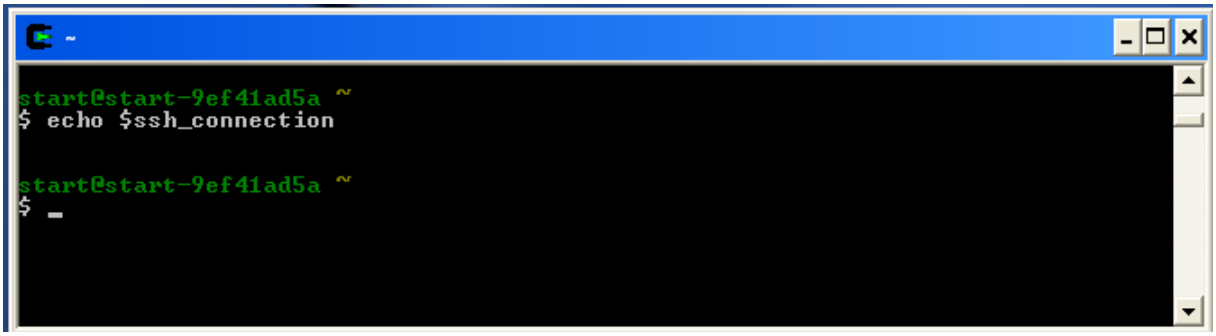
(le service démarrera automatiquement au prochain redémarrage de Windows ; Vous n'aurez plus à taper cette commande).



Vous pouvez arrêter le service sshd avec la commande : **net stop sshd**

II-3-d- Testez le service sshd:

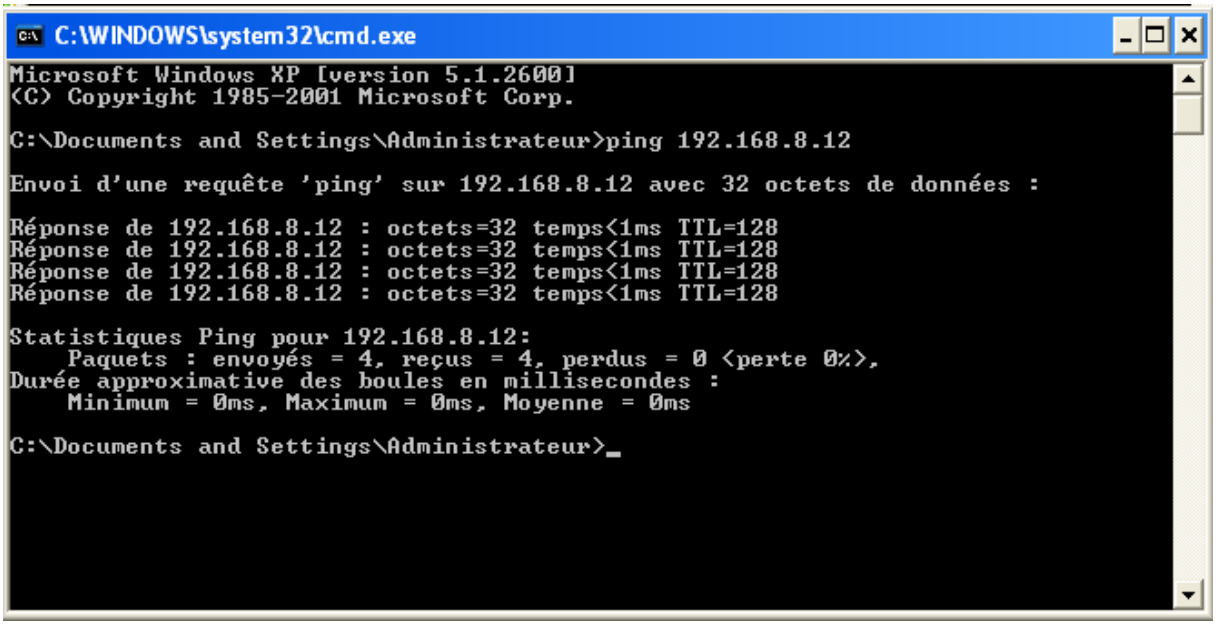
Vous pouvez voir la connexion en tapant: echo $SSH_CONNECTION



S'il n'ya pas un message d'erreur, le serveur SSH est connecté au client.

Vous disposez maintenant d'un serveur SSH basique. Passons maintenant au test de fonctionnement.

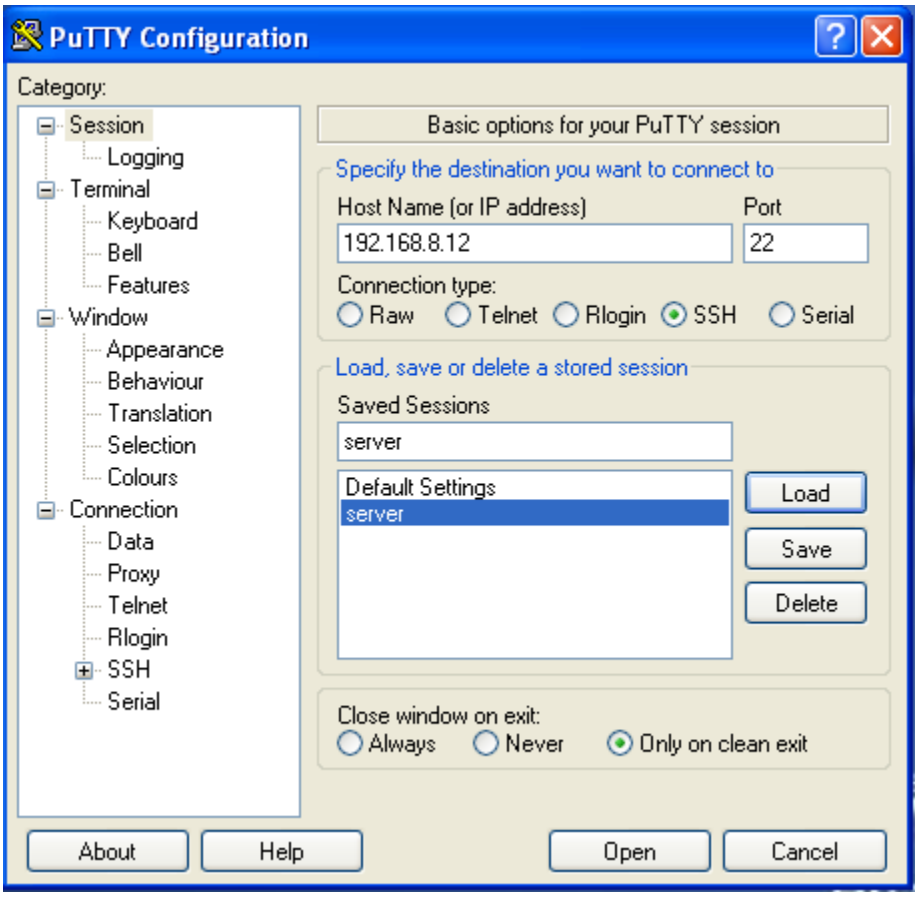
Avant de se connecter avec le client, commencez à vérifier que le serveur répond au ping :



II-3-e- La méthode d'authentification par mot de passe :

Vous pouvez connecter au serveur en utilisant un identifiant et un mot de passe valide. Pour se faire, suivez la procédure suivante :

1. Lancez PuTTY.
2. Dans le champ **hostname (or IP address)** entrez l'IP du serveur SSH.
3. Cliquez sur le bouton **SSH** s'il ne l'est pas par défaut.
4. Dans le champ **saved sessions** entrez un nom pour cette connexion, puis cliquez sur **save** pour l'enregistrer.
5. Cliquez sur **Open.**

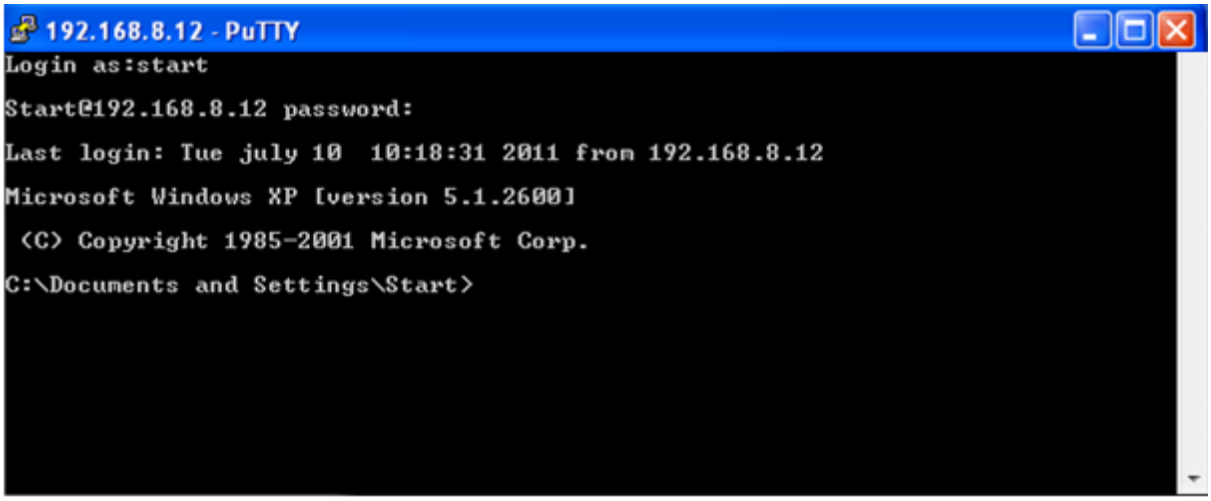


6. Si un avertissement apparaît, cliquez sur **OUI.**

Putty donne l'empreinte (fingerprint) du serveur. Pour confirmer que c'est le serveur désirer, cliquez sur **Oui**.

On ne vous reposera plus la question. Par contre, si le fingerprint change, un gros message d'avertissement s'affichera. Cela signifiera soit que le serveur a été réinstallé, soit que quelqu'un est en train de se faire passer par le serveur (c'est ce qu'on appelle une attaque man-in-the-middle).

7. A l'invite, entrez un identifiant du serveur.
8. Entrez ensuite le mot de passe du serveur.
9. Si tout se passe bien, vous devez obtenir l'accès à l'invite de commande du serveur.



Quand vous tapez le password, il n'ya ni point ni caractère qui s'affiche.

II-3-f- La méthode d'authentification par clés :

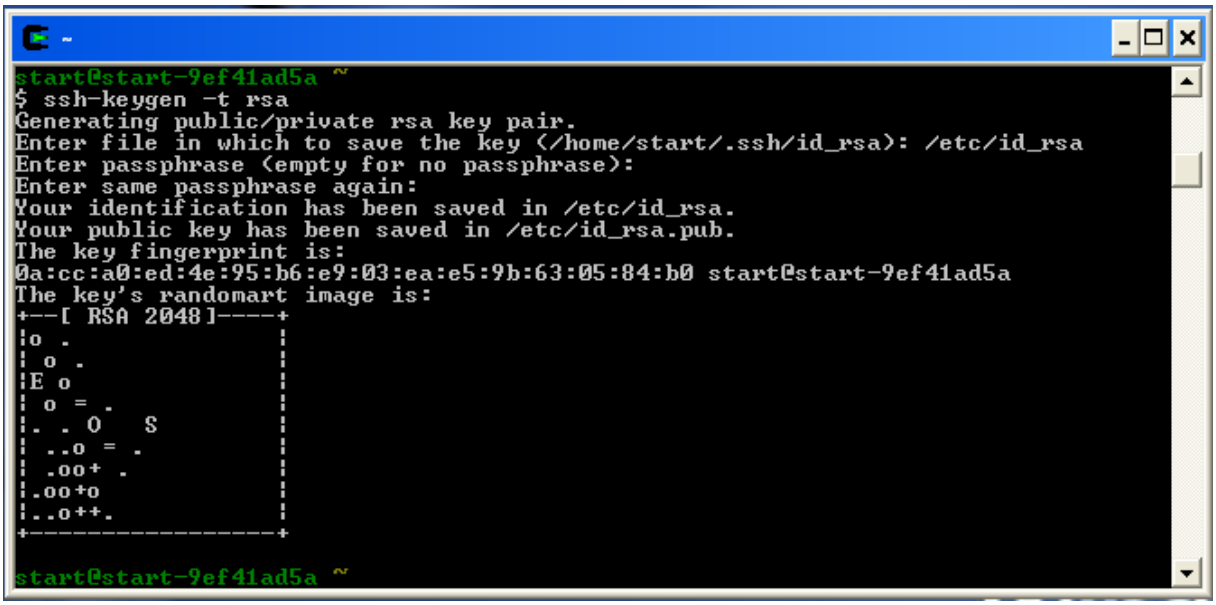
II-3-f-1-Génération de la paire de clés avec cygwin :

Ouvrez la fenêtre de cygwin, tapez la commande : **ssh-keygen -t rsa** afin de générer une clé de type SSH2 RSA.

ssh-keygen : Création des paires de clés.

- t (algorithmes) : choix de l'algorithme (rsa1, rsa et dsa)

ssh-keygen -t rsa : Définir la phrase d'identification pour protéger la clé privée.



Lorsque le programme vous demande où stocker la clé, entrez **/etc/id_rsa**

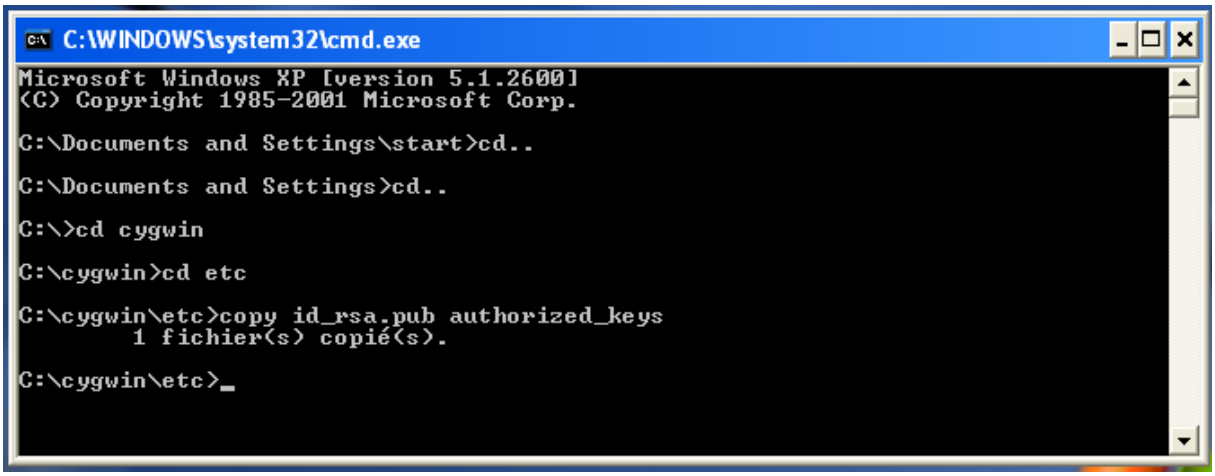
Openssh vous demande d'entrer une passphrase, pour protéger la clé privée.

Deux fichiers sont alors créés :

- Id_rsa : la clef privée que les clients utilisent.
- Id_rsa.pub : la clef publique qui restera sur votre serveur.

Chaque paire de clés a une empreinte unique.

Dans l'invite de commande « démarrer-exécuter-cmd-ok », tapez **cd cygwin** pour entrer dans le répertoire cygwin, puis **cd etc** pour entrer dans le répertoire **etc,** après taper la commande **copy id_rsa.pub authorized_keys** pour copier la clé publique dans le fichier **authorized_keys.**



Authorized_keys : Liste les clefs publiques (RSA ou DSA) utilisables pour se connecter avec le compte de l'utilisateur.
Copiez le fichier id_rsa sur le client (par une disquette, CD, clé USB).

II-3-f-1-a-Configuration de PuTTY afin d'utiliser la clé OpenSSH :

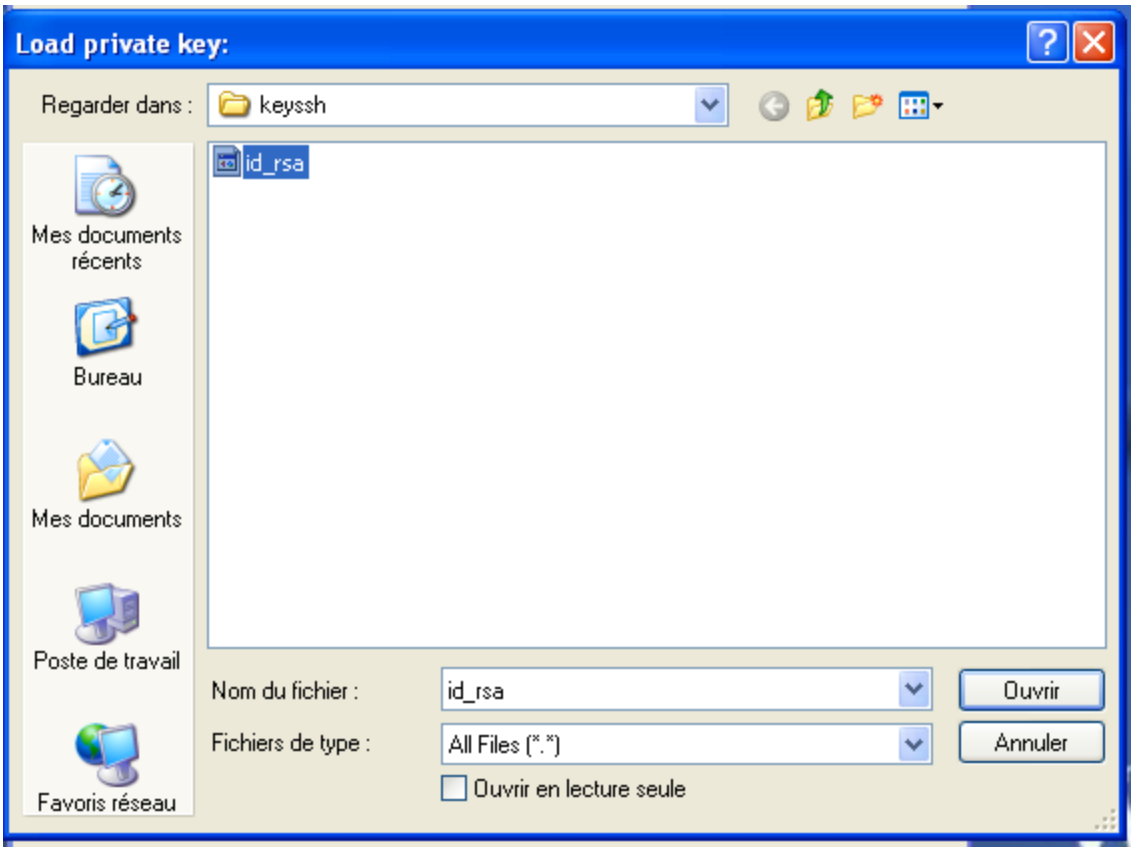
Sur le poste client, lancez le programme **puttygen.**

Choisissez le type de clé et l'algorithme de cryptage, ainsi que la longueur de la clé (il faut choisir les mêmes caractéristiques que la clé créer par le serveur).

Par exemple, une clé de type rsa SSH-2 de longueur 2048 bits.

Cliquez sur le bouton **Load** puis changez le type de fichiers à rechercher par fichiers de type : **All Files (*.*)**

Entrez le chemin où vous avez placé le fichier id_rsa, puis sélectionnez le. Cliquez ensuite sur **Ouvrir**.

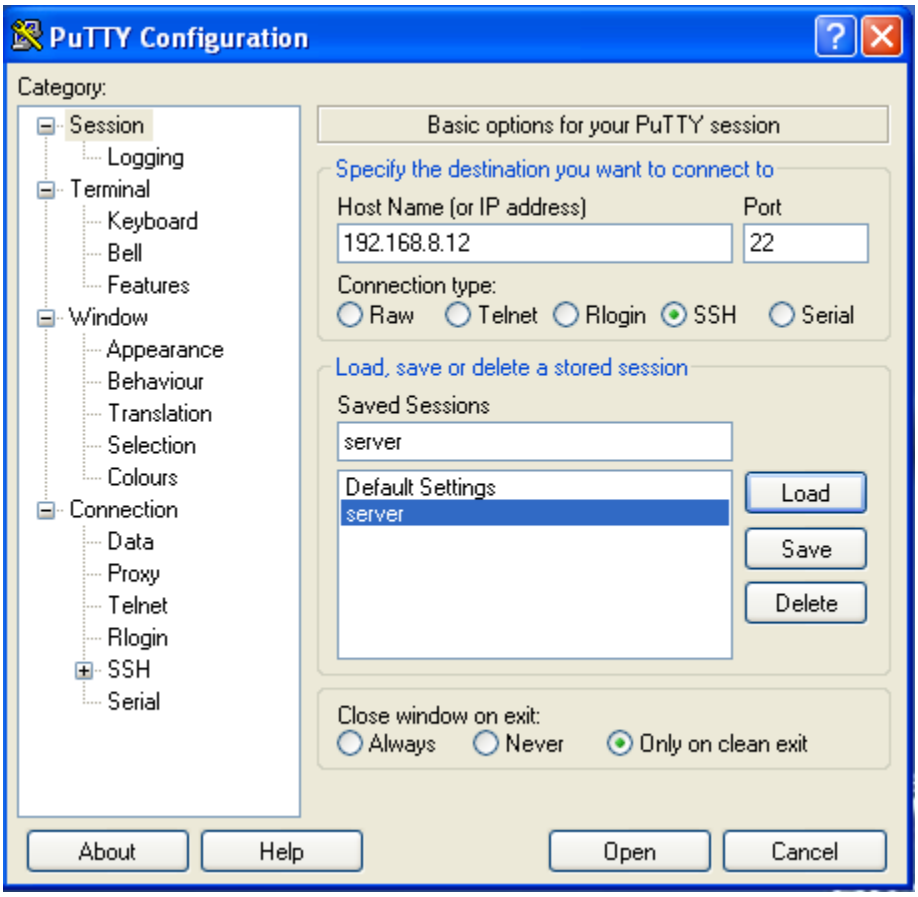


Vous obtenez un message de réussite d'importation, cliquez sur **Ok**, puis sur **Save private key**. Spécifiez le chemin du dossier où stocker la clé. Par défaut, elle se nommera id_rsa.ppk. Fermez ensuite PuTTYGen.

II-3-f-1-b-Utilisation de la clé avec PuTTY :

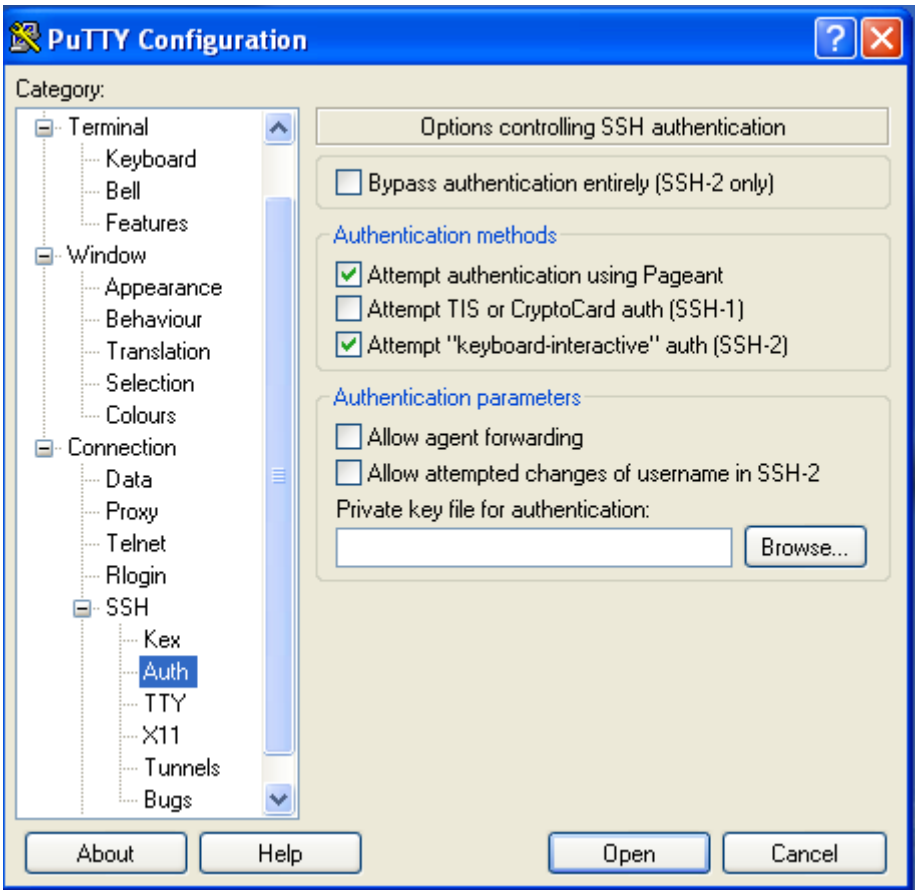
Il faut maintenant créer une connexion PuTTY avec utilisation de la clé nouvellement importée, pour se faire, suivez la procédure suivante :

• Lancez PuTTY.
• Dans le champ **Host Name or IP Address** , entrez l'IP du serveur.
• Assurez vous que le bouton SSH est sélectionné.
• Dans le champ **''Saved Sessions''**, entrez un nom pour cette connexion.

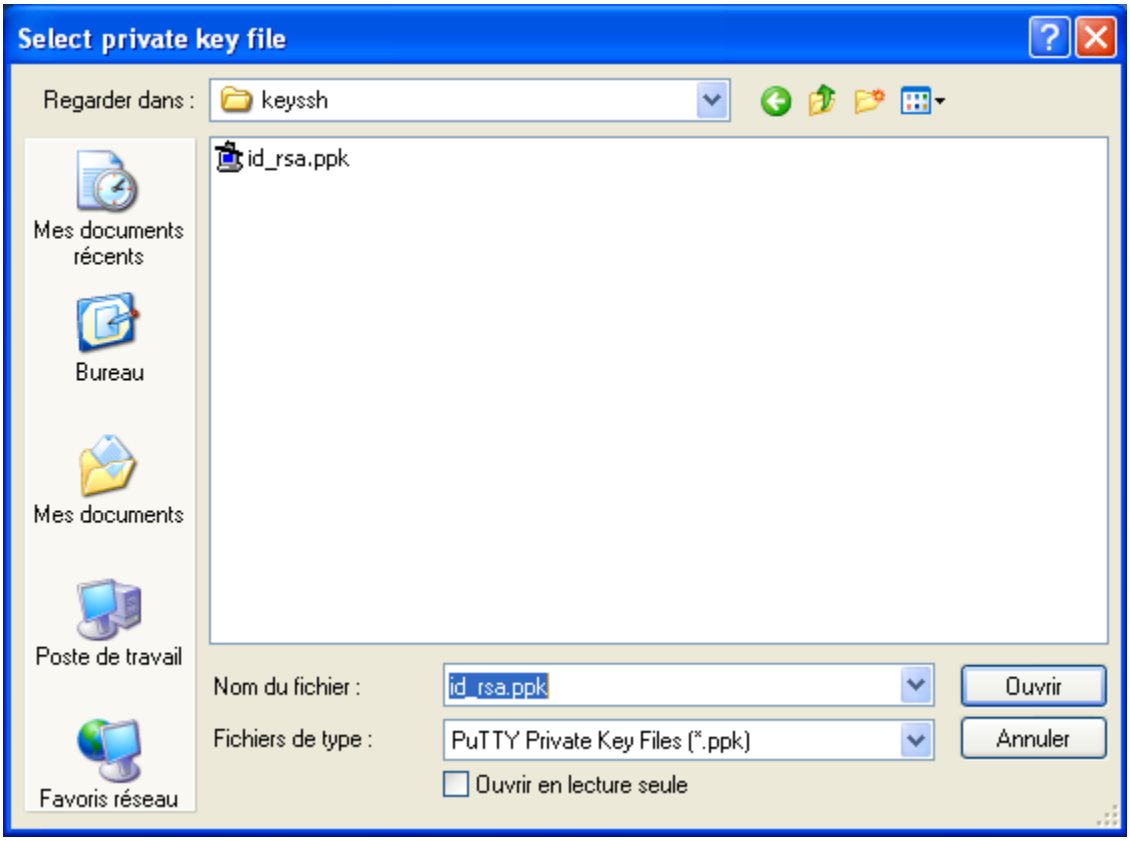


• Dans la fenêtre de gauche, dépliez l'arborescence **''Connection''** puis **''SSH''**, cliquez sur **Auth**.

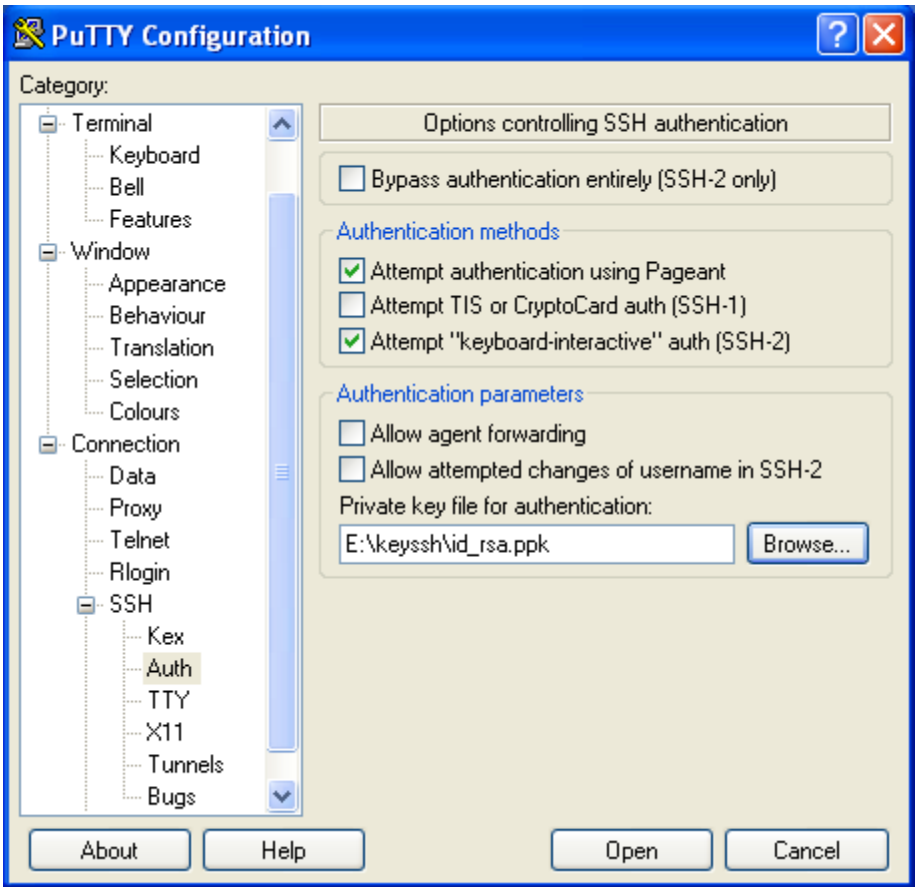
• Dans le champ **Private key file for authentication**, cliquez sur **Browse**.



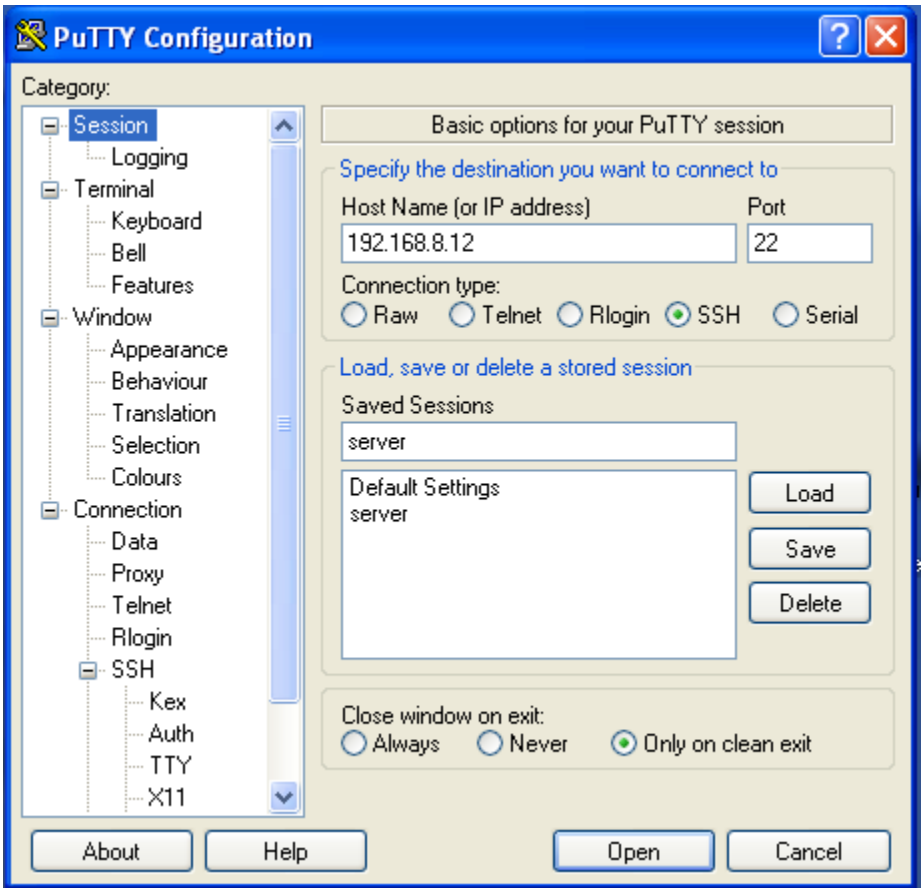
Spécifier le chemin où vous avez enregistré **id_rsa.ppk.**



Cliquez sur **Ouvrir**.

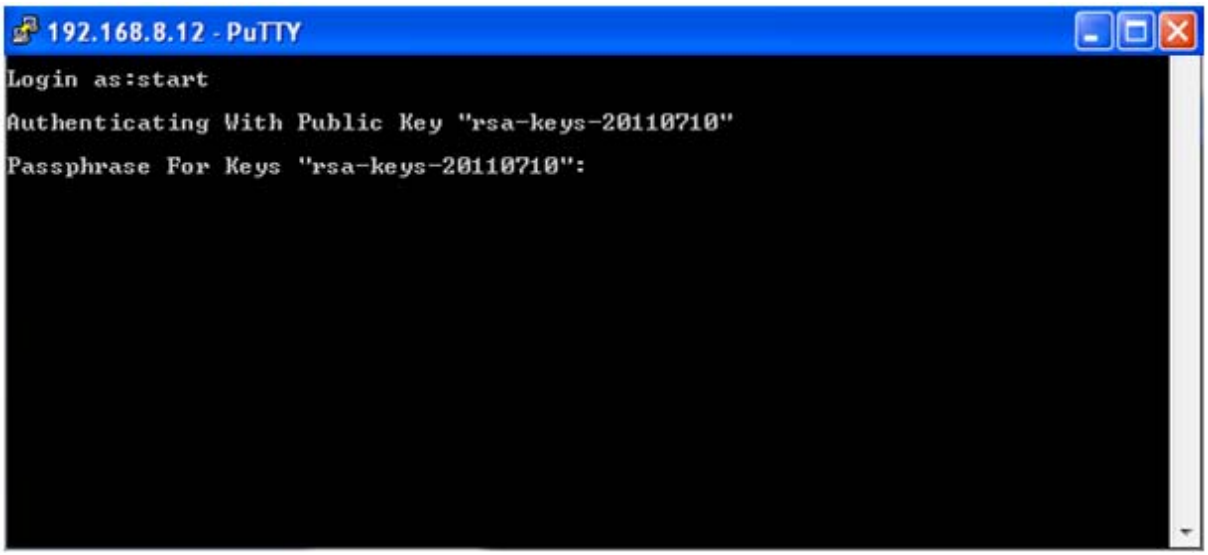


• Retourner au menu **''Session''**, cliquez sur **Save** pour enregistrer votre session et ses paramètres.



• Cliquez sur **Open**, une fenêtre s'ouvre et vous demandez l'identifiant (nom du serveur) et la passphrase.

• Vous êtes maintenant connecté au serveur.

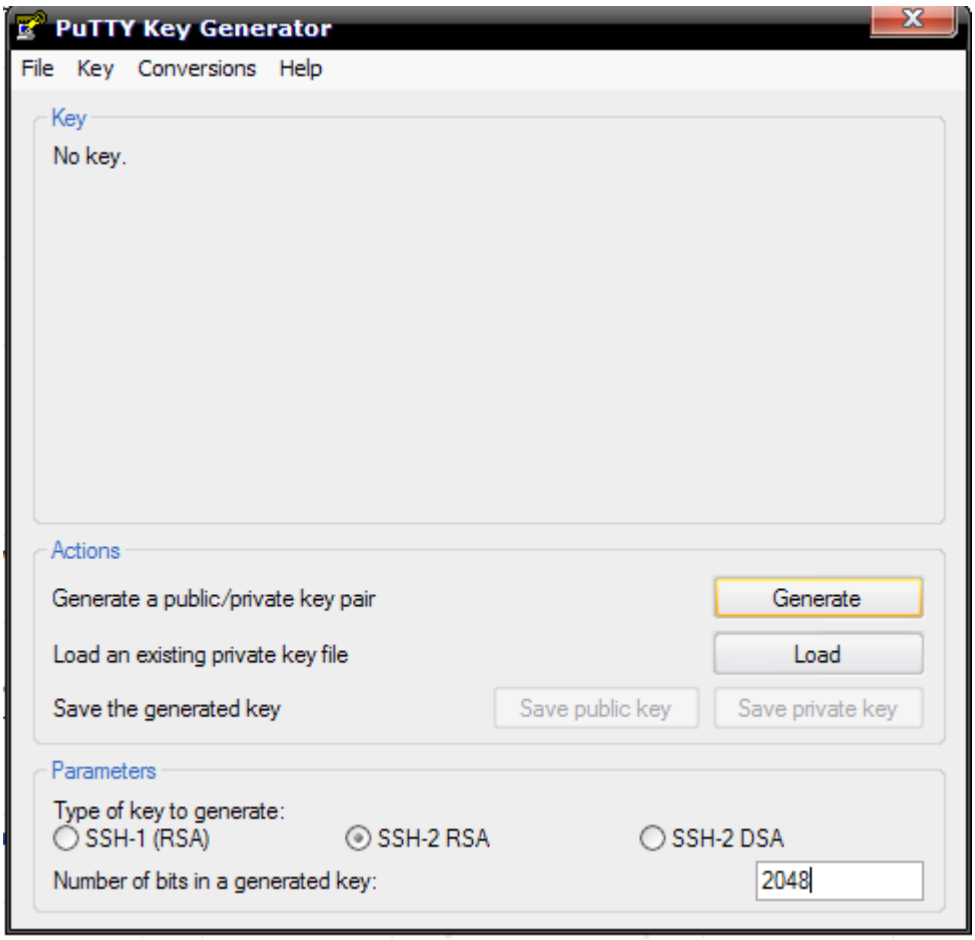


II-3-f-2-Génération de la paire de clés avec putty :

Générer une paire de clés sur le client, puis les envoyés au serveur. Nous retrouverons aussi un équivalent de l'agent SSH pour éviter d'avoir à rentrer la passphrase à chaque fois.

Commençez par la génération des clés (paire de clés publique et privée) avec Puttygen.

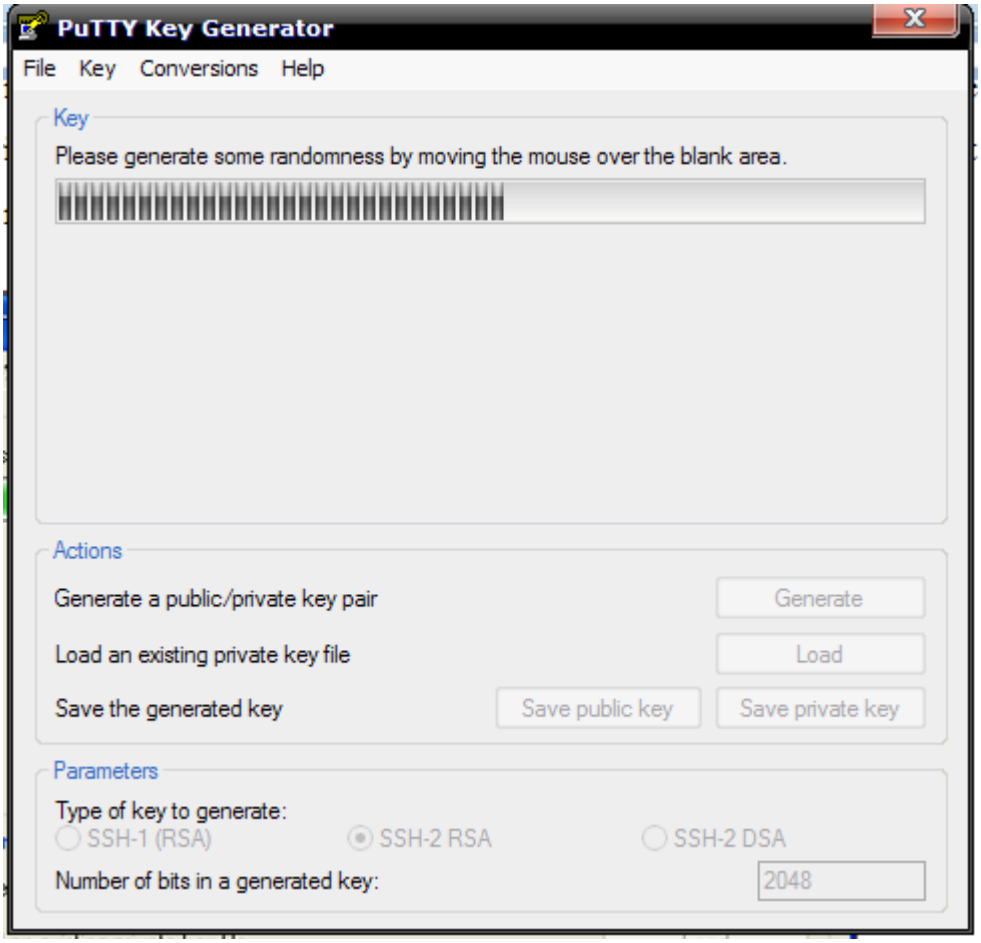
- Lancer puttygen :



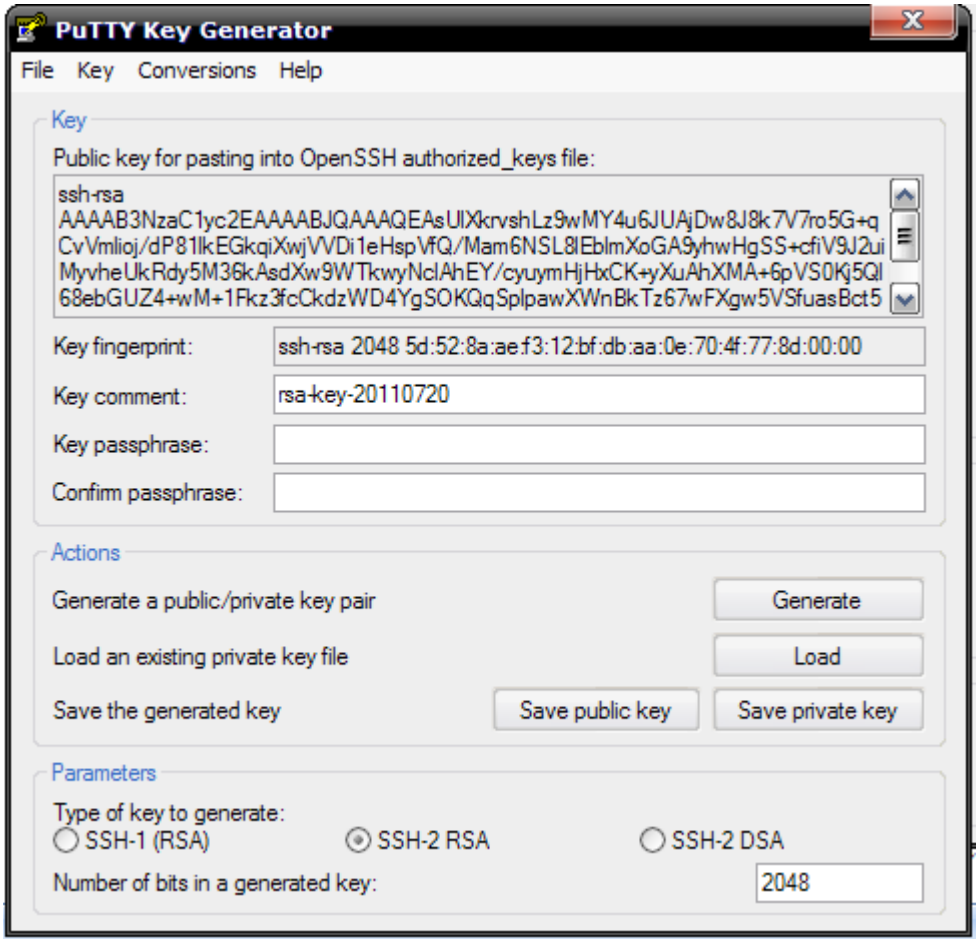
Choisissez l'algorithme (ssh-2 rsa) et la puissance du cryptage (2048bits).

Cliquez sur le bouton "Generate". Le programme va générer une paire de clés (publique et privée).

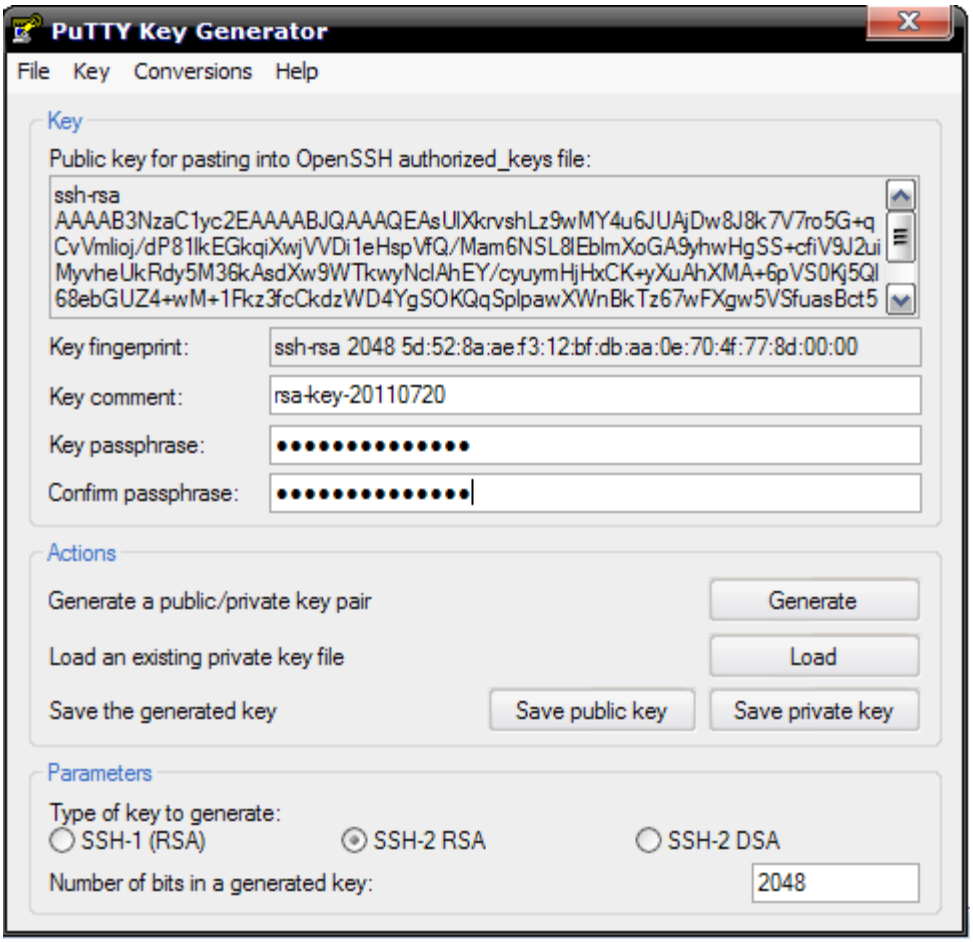
Pour aider le programme à générer cette paire de clé, il vous propose de bouger la souris dans la fenêtre. Vous allez le faire aléatoirement, cela aidera Puttygen à générer les clés.



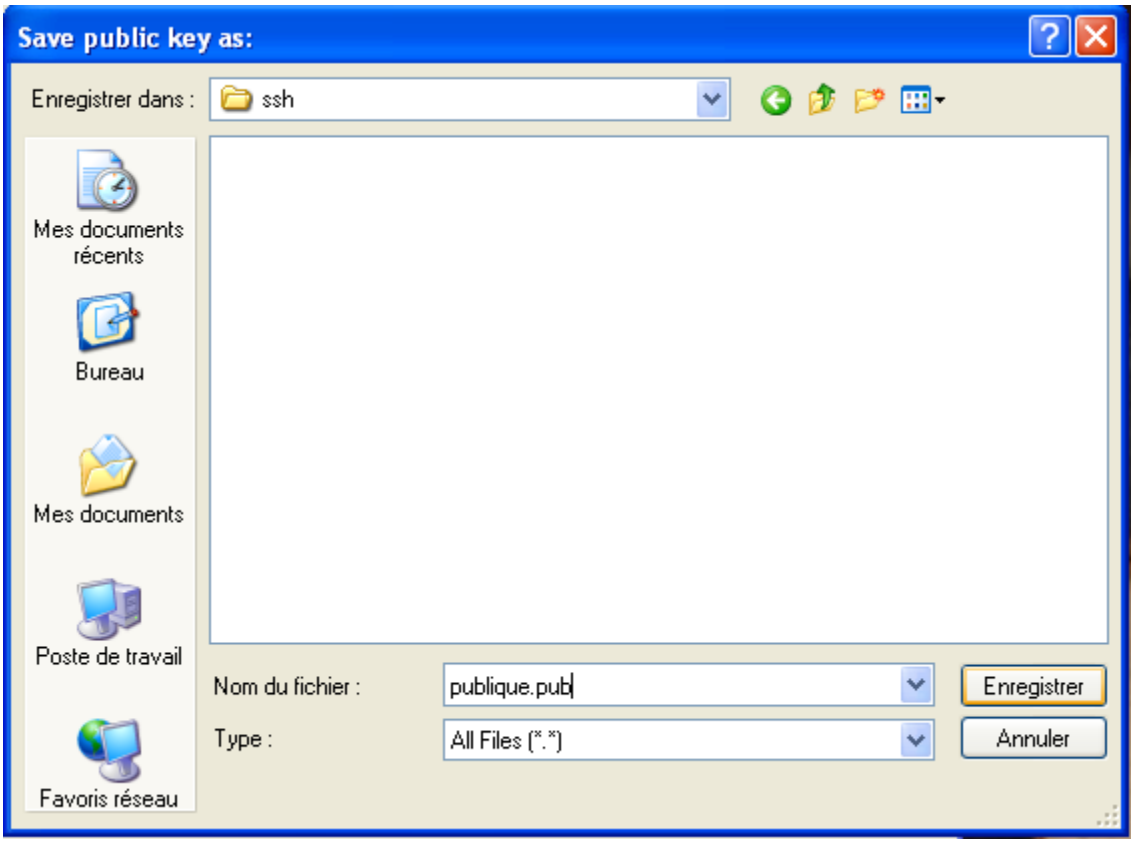
Après la génération des clés, puttygen affiche la clé publique. Par contre la clé privée doit rester secrète.



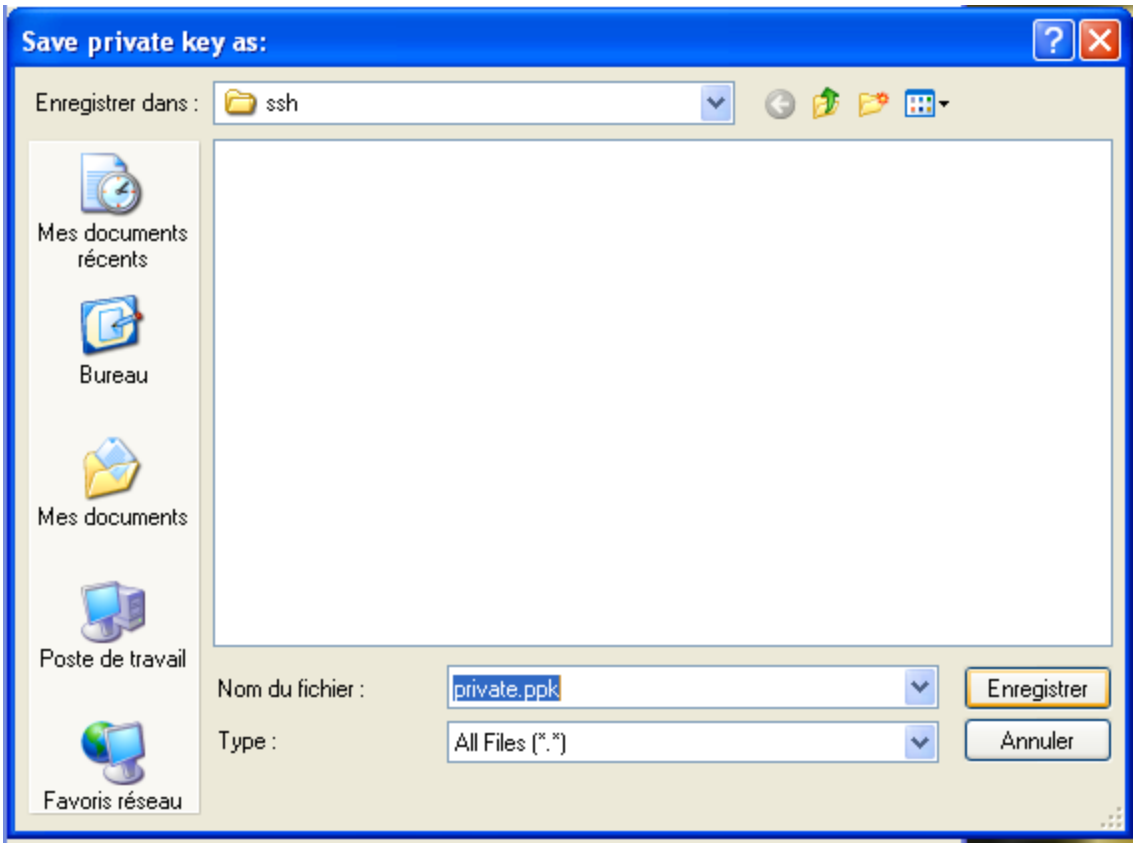
Saisissez la passphrase dans les champs "Key passphrase" et "Confirm passphrase", pour crypter la clé privée.



Enregistrez la clé publique dans un fichier en cliquant sur "Save public key". Vous pouvez nommer ce fichier comme vous voulez, par exemple ssh Enregistrez-le où vous voulez.



Puis, enregistrez la clé privée en cliquant sur "Save private key". Donnez-lui l'extension **.ppk** (private.ppk, par exemple).

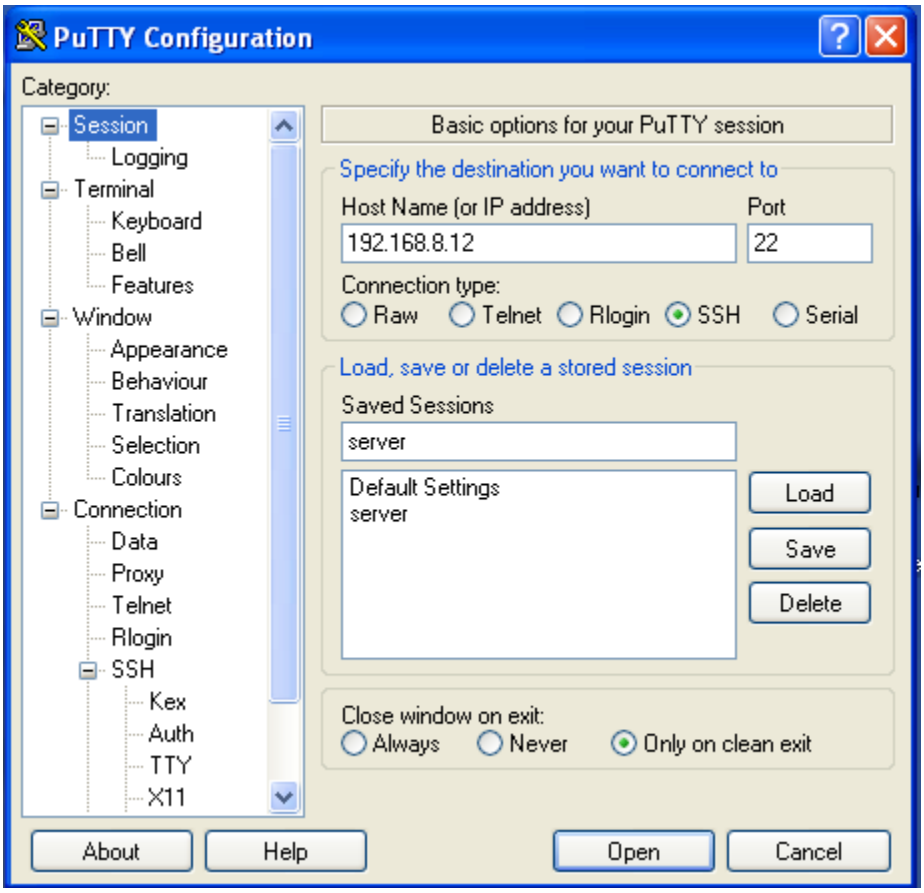


Ne fermez pas Puttygen.

II-f-2-a-L'envoie de la clé publique au serveur :

Il faut envoyer la clé publique au serveur pour qu'il vous autorise de connecter par clé. Le problème, c'est qu'il n'y a pas de commande pour le faire automatiquement depuis Windows. Il va falloir ajouter la clé à la main dans le fichier authorized_keys.

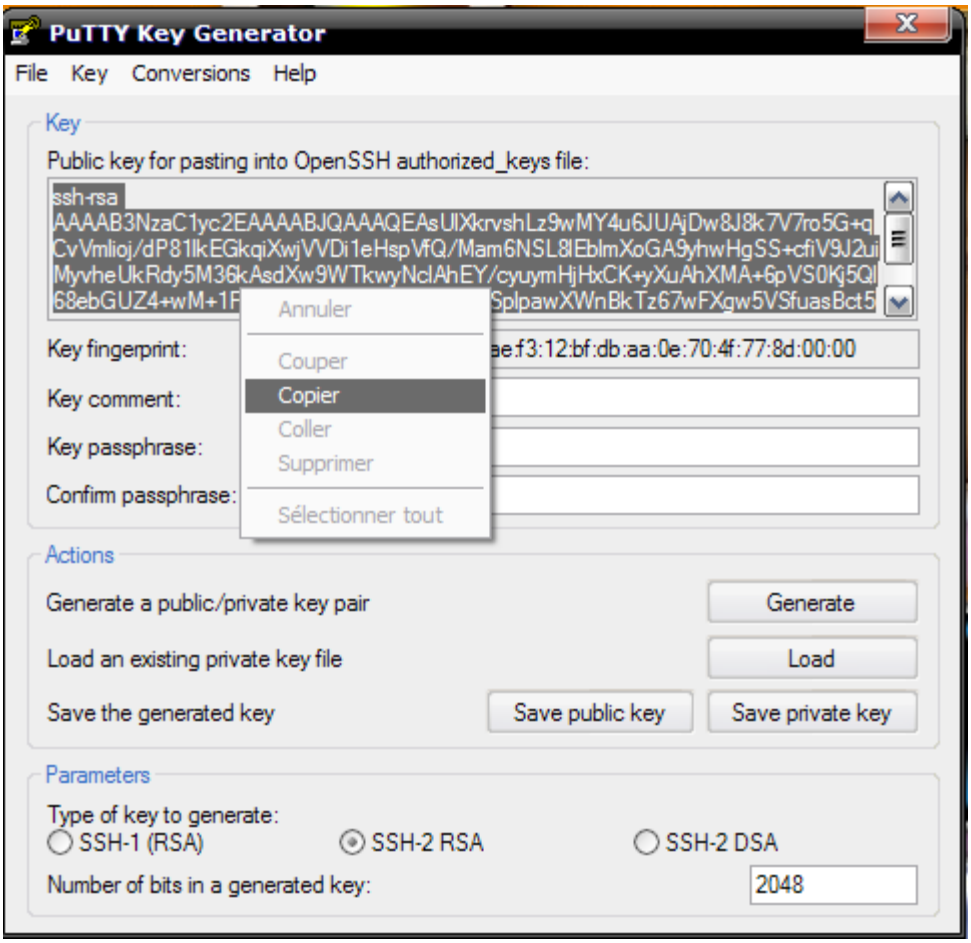
Ouvrez Putty et connectez-vous au serveur comme auparavant (en rentrant le mot de passe).



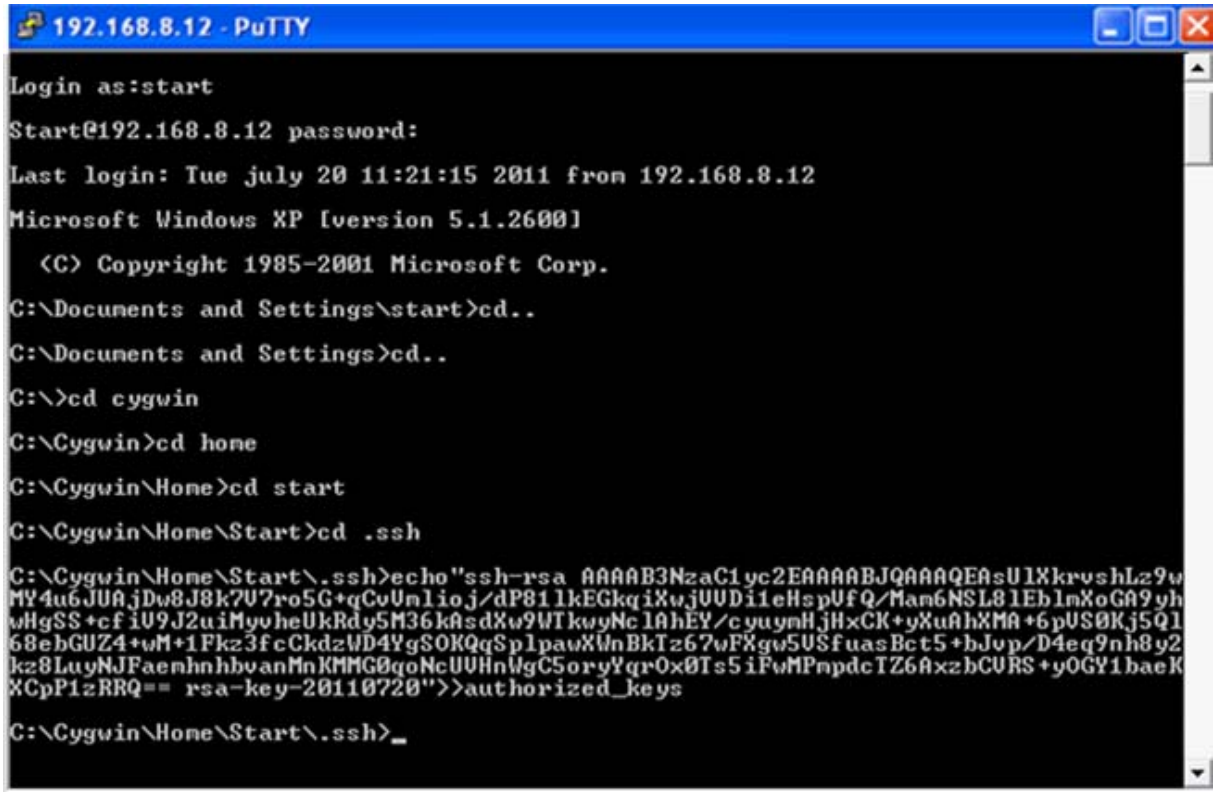
Pour rentrer dans le fichier « .ssh », tapez la commande : **cd .ssh**

Si le dossier .ssh n'existe pas, créez-le avec la commande : **mkdir .ssh**

Faire un copier\coller de la clé publique :



Rajoutez votre clé publique à la fin du fichier authorized_keys (s'il n'existe pas il sera créé). Vous utilisez la commande suivante : **echo "votre clé">>authorized_keys**

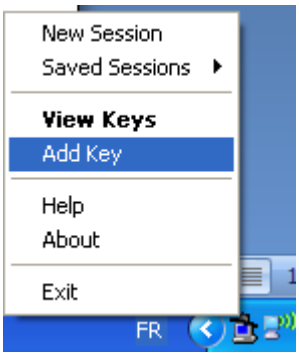


Maintenant le client peut se connecter avec le serveur par clé.

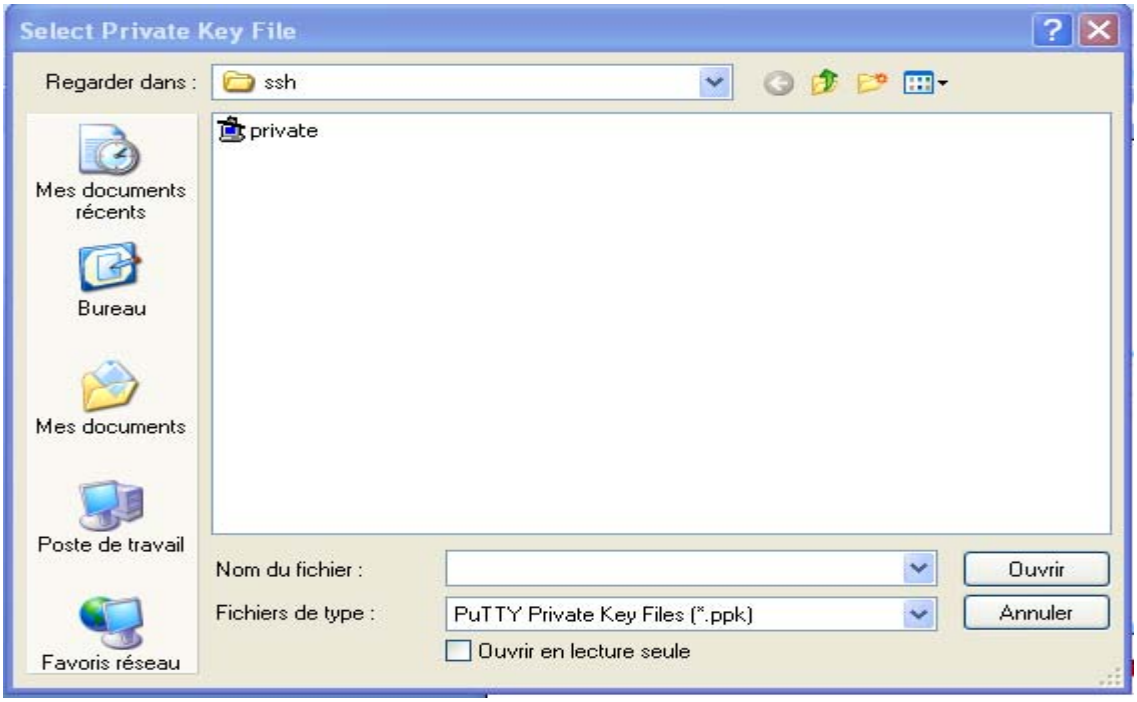
II-3-f-2-b-L'agent SSH « Pageant»:

L'agent SSH installé avec Putty s'appelle "Pageant". Il faut le lancer au démarrage de l'ordinateur automatiquement (il ne prend que 4 Mo en mémoire), Lorsque vous lancez Pageant, une petite icône d'un ordinateur avec un chapeau s'ajoute dans la barre des tâches :

Faites un clic droit dessus, puis cliquez sur "Add key" pour ajouter les clés privés.

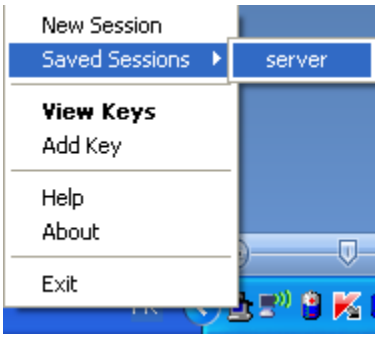


Il vous demande où se trouve la clé privée (clé.ppk).



Rentrez ensuite la passphrase.

Vous avez juste besoin de le faire une fois. Vous pouvez connecter au serveur en cliquant droit sur l'icône puis en sélectionnant "Saved Sessions".

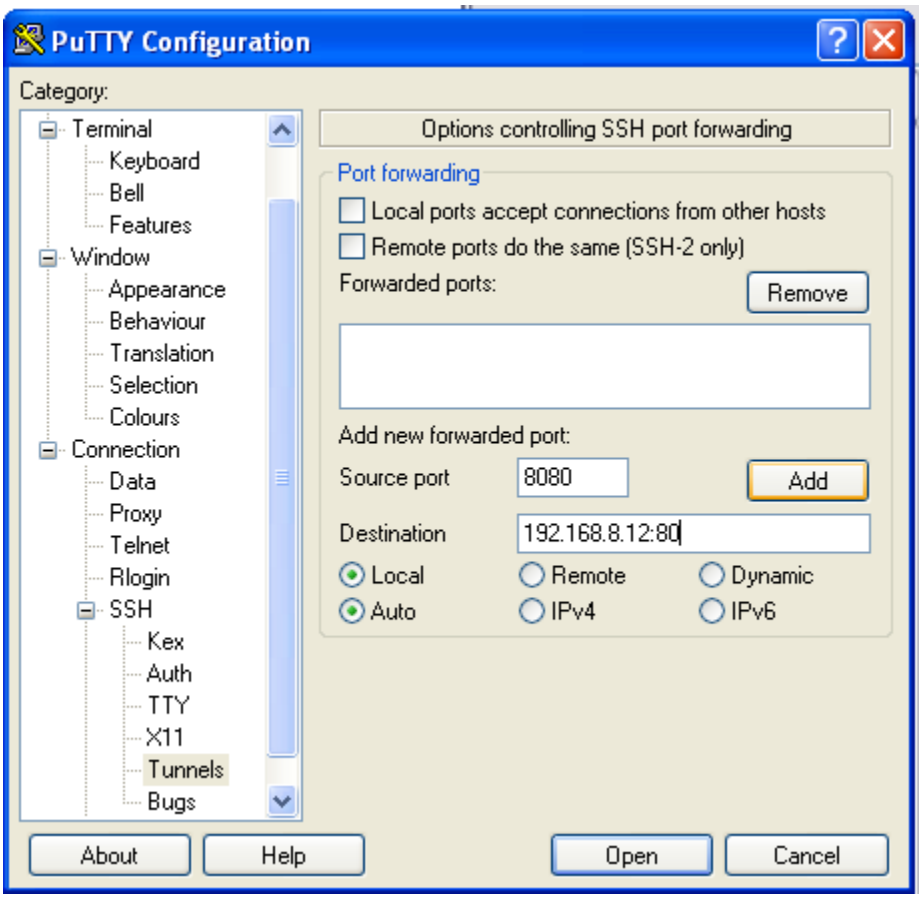


III-Création d'un tunnel SSH :

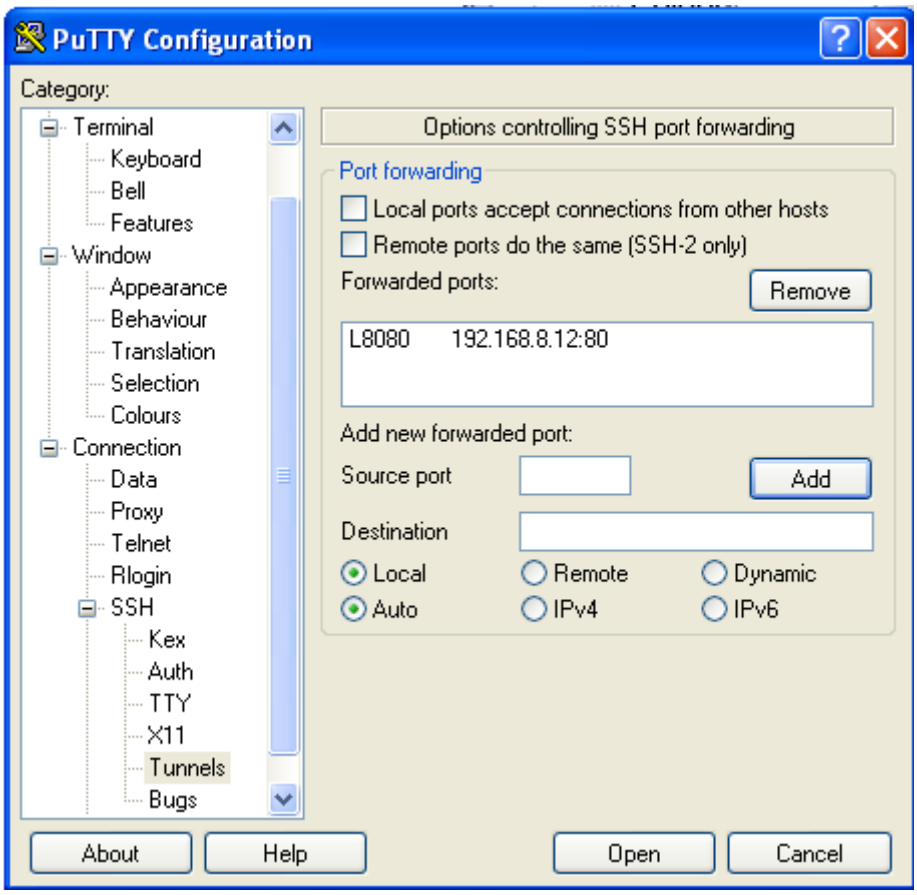
Pour créer un tunnel SSH, utilisez putty.

Suivez la procédure suivante :

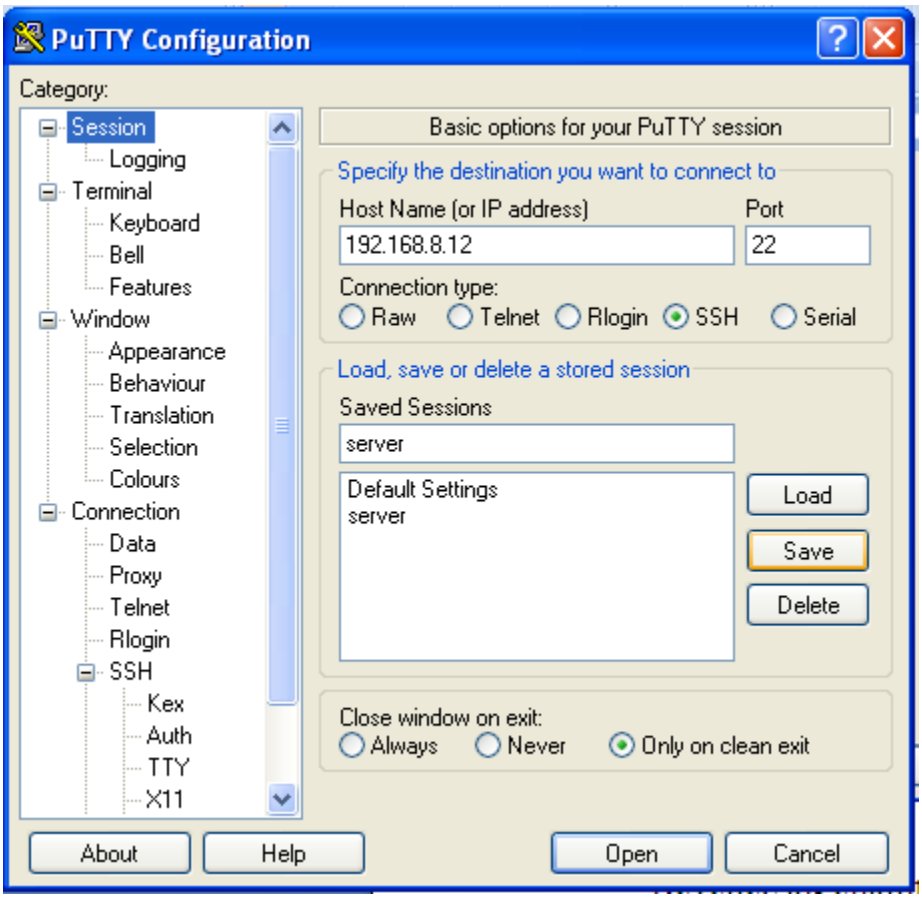
- Lancez putty.
- Allez dans le champ **connection/ssh/tunnel.**
- Remplissez le champ « **source port** », c'est le port local qui sera redirigé vers la machine distante.
- Dans le champ destination tapez l'adresse IP du serveur suivi de « **:** » et le numéro du port du serveur web.



- Cliquez sur **Add.**



- Revenez au champ **session** et choisissez un nom pour **saved sessions.**
- Cliquez sur **save** pour mémoriser cette session.



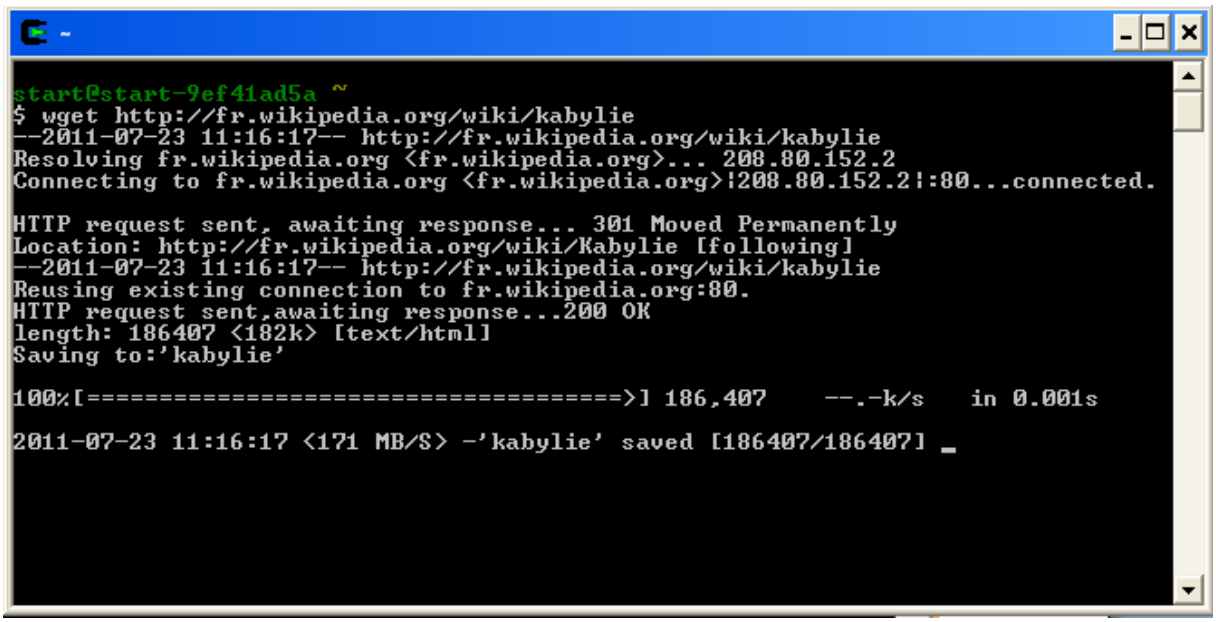
- Cliquez sur **open.**

Par la suite c'est les mêmes étapes qu'on a décrit déjà dans l'authentification par mot de passe, ou par clés si vous chargez la clé lors de la création du tunnel.

IV-Téléchargements de fichiers avec cygwin:

Téléchargez des fichiers à partir d'une adresse http avec la commande « wget», par exemple :

Téléchargez le fichier « wikipedia-kabylie » à partir d'une adresse http://fr.wikipedia.org/wiki/kabylie.

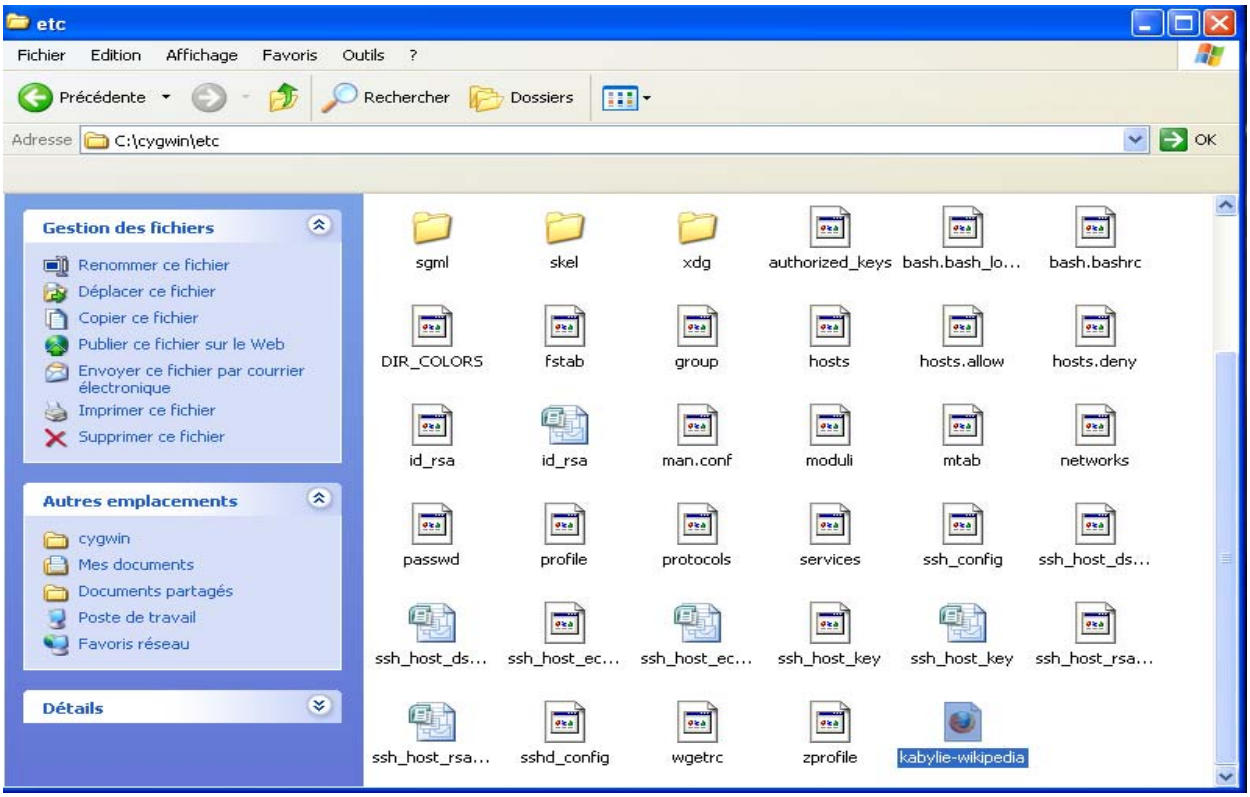


Les premières lignes représentent des informations sur la façon de communication du programme avec le serveur qui possède les fichiers.

Les deux dernières lignes représentent :

- ✓ Une barre de progression qui se met à jour (Vous pouvez arrêter le téléchargement en utilisant la combinaison Ctrl + C et le reprendre avec l'option -c).
- ✓ Le nombre d'octets téléchargés (186407 octets).
- ✓ La vitesse de téléchargement (171 MB/s).
- ✓ Le temps restant estimé (0.001s).

Vous trouvez le fichier téléchargé dans le répertoire c:\cygwin\etc :



V-Conclusion :

Le protocole SSH permet à un administrateur de pouvoir répondre à n'importe quel problème sur internet. De plus cette communication est cryptée, les mots de passe et autres applications ne peuvent être vu en clairs sur les réseaux.

Il a été expliqué, dans ce projet, le fonctionnement du protocole SSH et les différents algorithmes qu'il emploie dans le but d'assurer l'authentification, l'intégrité et la confidentialité des données. Ce protocole sécurisé est une réelle évolution quand on le compare au protocole Telnet ou rlogin. En plus d'offrir un accès au Shell de manière sécurisée, on peut profiter des avantages de SSH pour faire du multi-tunneling applicatif et ainsi permettre de transmettre des données très confidentielles sur le réseau. D'autre part, le tunneling permet également de limiter le nombre de ports à ouvrir sur le garde-barrière et donc masquer les services qui tournent sur un serveur en interne ce qui est un point vraiment important dans l'établissement d'une architecture sécurisée.

Nous avons également abordé la mise en œuvre d'un serveur SSH sous Windows même si cela pose certains problèmes ; SSH est capable de contourner de nombreuses menaces de sécurité liées au réseau.

Cependant, il est vulnérable aux attaques par déni de service, héritant ainsi les faiblesses de TCP/IP sur lequel il repose. En outre, suivant l'environnement SSH est sensible à certaines méthodes d'attaques, comme l'analyse et le détournement de trafic.

Comme perspectives, l'installation d'un serveur SSH sous Unix, Linux ou BSD est plus fiable et sans aucune comparaison (pas de DLL d'émulation d'une partie d'un système d'exploitation (cygwin.dll)).

Ce projet nous a permis d'étudier le protocole SSH du point de vue logiciel et avoir une idée sur les différentes méthodes d'authentification entre le client et le serveur sous Windows XP.

AES : Advanced Encryption Standard.

Algorithme : Un jeu de règles ou de procédures bien défini qu'il faut suivre pour obtenir la solution d'un problème dans un nombre fini d'étapes. Un algorithme peut comprendre des procédures et instructions algébriques, arithmétiques, et logiques, et autres. Un algorithme peut être simple ou compliqué. Cependant un algorithme doit obtenir une solution en un nombre fini d'étapes.

Anneau à jeton : Token Ring, mis en place par IBM, a commencé à se développer au début des années 70. C'est aujourd'hui le deuxième type de réseau derrière Ethernet.

ARP: Address Resolution Protocol.

ASCII : American Standard Code for Information Interchange ; C'est une norme de codage de 128 caractères alphanumériques sur 7 bits. Les versions étendues sur 8 bits, 256 caractères sont adaptées suivant les pays.

Attaque : n'importe quelle action qui compromet la sécurité des informations.

Authentification : l'identité des acteurs de la communication est vérifiée.

Bit : C'est l'unité binaire de quantité d'information qui peut représenter deux valeurs distinctes : 0 ou 1.

Blowfish : algorithme cryptographique conçu par Bruce Schneier en 1993, chiffrant les données par blocs de 64 bits, avec une clé allant de 32 à 448 bits. Il est rapide et simple, et surtout, il est dans le domaine public.

BSD : Berkeley Software Distribution, il désigne en particulier une famille de versions d'Unix issue de l'université de Berkeley en Californie.

CA : autorité de certification.

Chiffrement : Procédé grâce auquel on peut rendre la compréhension d'un document impossible à toute personne qui n'a pas la clef d'encodage.

Clé : Code constitué de symboles permettant les opérations de chiffrement et décryptage.

Clé d'hôte (host key): biclé asymetrique, crée par l'administrateur du serveur lors de l'installation et de la configuration, elle permet l'authentification du système lors de l'installation.

Clé publique : Clef permettant de coder un message c'est à dire de le crypter.

Clé secrète : Clé permettant de décoder un message c'est à dire de le déchiffrer.

Clé de session : session key, clé secrète destinée à être utilisée par l'algorithme de chiffrement symetrique chiffrant le canal de communication.

Client : Une application, installée sur votre ordinateur, qui récupére des informations sur un serveur distant

Codage : Ensemble de règles définissant une correspondance biunivoque entre des informations et leur représentation par des caractères, des symboles ou des éléments de signal.

Compression : C'est un processus qui consiste à réduire le volume de données

confidentialité : les données (et l'objet et les acteurs) de la communication ne peuvent pas être connues d'un tiers non-autorisé.

Connexion : Une communication logique définie par une paire de sockets.

Cryptographie : La cryptographie permet d'assurer les fonctions principales de la sécurité des systèmes d'informations .

Cygwin : ensemble des logiciels libres permettant de compiler des applications en provenance du monde unix sous windows.

Décodage : Action visant à découvrir le sens d'un texte chiffré dont on ignore la clé.

DES : Data Encryption Standard

DH : Diffie-Hellman, permet d'échanger une clé symétrique, mais ne permet pas de chiffrer ; permet à deux entitées de communiquer sans se rencontrer.

Disponibilité : les acteurs de la communication accèdent aux données dans de bonnes conditions.

DMZ : Demilitarized zone

DSA : Distributed Network Architecture

Ethernet : Ethernet est une technologie universelle qui dominait déjà les réseaux locaux bien avant le développement de l'Internet (Ethernet a été standardisé sous le nom IEEE 802.3).

FDDI : Fiber Distributed Data Interface

Flux : Définition du mot FLUX, Vidéo diffusée en streaming sur un réseau.

FTP : File Transfer Protocol

Homme du lieu : le contrôle d'un equipement du reseau se place au milieu d'une communication.

HOST : Ordinateur distant qui reçoit les appels d'autres machines (connexions sur un site Web, par exemple).

HTTP : HyperText Transfer Protocol

IBM: International Business Machines.

ICMP: Internet Control Message Protocol.

IDEA : International Data Encryption Algorythm

Identification : Opération par laquelle on retrouve l'identificateur qui est lié à un élément de dessin ou d'image désigné par un opérateur.

IETF : Internet Engineering Task Force

intégrité : les données de la communication n'ont pas été altérées.

IP : Internet Protocol

ISO : International Standart Organisation

LAN : Local Area Network

Login :Nom d'utilisateur permettant d'identifier un utilisateur qui se connecte

sur ordinateur ou un site internet**.**

MAN: Metropolitan Area Network.

Mot de passe : Suite de caractères entrée par un utilisateur pour pouvoir accéder (login) à son environnement informatique, une application ou des données dont l'usage est soumis à des autorisations ou des contraintes de confidentialité.

Multiplexage : Le multiplexage consiste à faire transiter sur une seule et même ligne de liaison, dite voie haute vitesse, des communications appartenant à plusieurs paires d'équipements émetteurs et récepteurs

Netscape : L'éditeur du navigateur le plus populaire et le plus répandu sur le Web avant l'apparition de la version 4 d'Internet Explorer de Microsoft.

Openssh : est une implémentation libre du protocole SSH, permettant une connexion interactive comme pour Telnet mais de manière sécurisée.

Openssl : est une boite à outils de chiffrement comportant deux bibliothèques (une de cryptographie générale et une implémentant le protocole SSL) ainsi qu'une commande en ligne.

OSI : Open System Interconnection.

pageant : utilitaire permettant de communiquer les clés quand c'est nécessaire.

Paquet : Unité d'information utilisée pour communiquer sur le réseau.

Pare-feu : Le firewall est l’outil de sécurité le plus performant à l’heure actuelle. Il permet de se protéger du réseau global en isolant le réseau local. Il n’exclut toutefois pas les risques de piratage interne, mais solutionne le problème d’intrusion depuis l’extérieur.

Passphrase : c’est un mot de passe sous forme de phrase et qui sert à protéger la

clé privée asymétrique de l’utilisateur.

PATH : chemin d'accès à un fichier, c'est-à-dire la liste des répertoires qu'il faut traverser pour l'atteindre.

PGP: Pretty Good Privacy.

PKI : Public Key Infrastructure.

Point d’accès : Installation qui permet à un utilisateur de se connecter par une liaison radio (RLAN) en 2,4 GHz ou en 5 GHz à un réseau haut débit

POP : Port Office Protocol

Port : La portion d'un socket qui définit l'entrée ou la sortie "logique" qu'utilise un processus pour véhiculer les données.

PPP : Point-to-Point Protocol.

Protocole : un protocole est une description formelle de règles et de conventions à suivre dans un échange d'informations, que ce soit pour acheminer les données jusqu'au destinataire ou pour que le destinataire comprenne comment il doit utiliser les données qu'il a reçues.

Proxy: Les proxy permettent de rompre avec le modèle classique client-serveur d’une communication en interdisant une connexion directe du client au serveur.

Psftp : client sftp en ligne de commandes.

PuTTY : client SSH évolué,Permet de créer des sessions et de les configurer de façon graphique.

Puttygen : utilitaire permettant de créer/modifier des clés.

rcp : Remote Copy Protocol.

RC: Rivest Cipher.

Requête : est une demande de traitement. Le terme est notamment employé dans le contexte.

RIP : Routing Information Protocol

rlogin: Désigne un Remote Login. On peut par cette méthode se connecter à un ordinateur distant.

Routeur: une machine connectée sur deux réseaux locaux différents qui se charge de faire passer les données de l'un à l'autre.

RSA : Rivest, Shaman, Adleman.

rsh: Remote Shell, Programme permettant de faire tourner un shell à distance, via TCP/IP.

SAN: System Area Network.

scp : Secure Copy Protocol.

SEPP: Secure Electronic Payement Protocol.

Serveur : Ordinateur dédié à l'administration d'un réseau informatique, il gère l'accès aux ressources et aux périphériques et les connexions des différents utilisateurs. Il est équipé d'un logiciel de gestion de réseau.

Session : c'est l'exécution d'un programme pour un utilisateur donné. L'exécution du programme est alors paramétrée par les informations du profil de l'utilisateur (ses caractéristiques, ses préférences, l'historique de ses interactions avec le programme).

SET: Secure Electronic Transaction.

SFTP: Secure File Transfer Protocol.

S-http: Secure http.

SMTP : Simple Mail Transfert Protocol.

Sniffer : Sniffeurs de trames; Les sniffeurs de trames permettent depuis

n’importe quelle machine reliée au LAN de voir ce qui transite dans les paquets du réseau.

SSH: Secure Shell.

ssh-agent qui assure la fonction d'agent forwarding.

ssh-add qui permet d'ajouter des clés à l'agent d'authentification précédent.

SSHd : le serveur permettant à l'utilisateur de se logger.

ssh-keygen : qui permet de générer les clés.

ssh-keyscan : qui permet de récupérer des clés publiques d'hôtes à partir d'un ensemble d'hôtes.

ssh-keysign : assure une aide à l'authentification basée sur l'hôte.

SSL: Secure Socket Layer.

STT: Secure Transaction Technology.

TCP: Transmission Control Protocol.

TCP/IP : Transmission Control Protocol/ Internet Protocol.

Telnet :(Network Terminal Protocol) Le protocole Telnet est un protocole standard d'Internet permettant l'interfaçage de terminaux et d'applications à travers Internet. Ce protocole fournit les règles de base pour permettre de relier un client (système composé d'un affichage et d'un clavier) à un interpréteur de commande (côté serveur).

Trame: Définition du mot TRAME, Unité d'information transportée sur un réseau, constituée d'une série de bits de taille variable.

UDP: User Datagram Protocol.

VNC:Virtual Network Computing

VPN: virtual Private network.

WAN: Wide Area Network.

X11: X-window system Version 11, la première et seule Version de X à avoir quitté les laboratoires pour être massivement utilisée.

X-windows: Interface graphique associée à Unix et Linux.

X.509: est une norme de cryptographie de l'Union internationale des télécommunications pour les infrastructures à clés publiques (PKI).