impressions

- CUPS gère les impressions sur le parc du département informatique
- les outils usuels de Linux (lpr & Co) sont compatibles avec CUPS
- Exemples:
  - a2ps -2 -Pc121 monFichierTexteaMoi.txt : imprime le fichier sur l'imprimante c121 en 2 pages par pages
  - lpstat -a : donne la liste des imprimantes
  - lpq -Pc121: liste des travaux en attente d'envoi à l'imprimante
  - lprm -Pc121 idTravail : annule le travail IdTravail (-: tous les travaux)

sauvegardes

- archivage : tar
- compression:
  - compress (historique),
  - gzip: le standard (géré aussi par les outils windows)
  - bzip2: plus récent, compresse en général mieux que gzip
- combiner les deux :
  - tar cvf - fichiersAArchiver | gzip -9 > archive.tar.gz
  - tar czvf archive.tar.gz fichiersAArchiver (compression avec gzip)
  - tar cjvf archive.tar.gz fichiersAArchiver (compression avec bzip2)

Sauvegarde (suite)

- zip/unzip: outil en ligne de commande pour générer des fichiers zip
- ark: outil graphique pour générer des archives compressés. supporte de nombreux formats
- explorateur de fichier de KDE :
  - support natif de nombreux formats
  - pour créer des archives, s'y déplacer, les désarchiver
- le support (décompression) du format rar suppose l'installation de l'outil unrar (il l'est)

réseau

- présentation faite directement au tableau
- On a rappelé
  - la notion de couches
  - les couches 2, 3 et 4
  - les notions d'adresses ip et de ports
  - le routage IP

X window

- X window ou X11: gestion du clavier/souris/écran (désigné par la variable DISPLAY)
- peut en gérer plusieurs sur un même poste
- ne gère pas la décoration => nécessité d'un gestionnaire de fenêtre
- capable d'afficher des applications graphiques distantes
- gestion minimale de la sécurité
- client/serveur: le serveur (de ressources graphiques) est la machine où s'affiche l'application

X window & ssh -X

- ssh -X fait transiter X 11 via le tunnel sécurisé de ssh
- méthode sure et simple (ssh fait tout). ssh positionne :
  - le MIT-magic-cookie
  - la variable DISPLAY
- l'option -X doit être autorisée :
  - sur le serveur distant (configuration du serveur sshd éalisée par l'administrateur du poste: activée au dept informatique)
  - sur le client

ssh

- secure shell
  - un protocole : le protocole ssh (version courante: version 2.0)
  - des produits ou commandes implantant ce protocole
- pour répondre aux problème de sécurité des outils traditionnels (telnet, rsh, rcp, ...)
  - clients et serveur s'authentifient (pas de mim ou d'usurpation)
  - login, mot de passe et données passent dans un tunnel chiffré (imperméable à l'utilisation d'analyseur de trames, de sniffers)
  - ssh permet de relier deux machines sûres à travers un réseau non sûr.

ssh (suite)

- une suite d'outils s'appuyant sur un procole securisé: ssh, scp, sftp (v2)
- des outils de gestion de clefs: ssh-keygen, ssh-add, ssh-agent
- des fonctionnalités
  - redirection de ports
    - rediriger un port de la machine locale vers un port d'une machine distante : permet l'accès à des applications/ressources de machines distantes via le tunnel ssh
    - idem avec un port de la machine à laquelle on est connecté vers un port local: permet l'accès à des ressources locales depuis des machines distantes via le tunnel ssh
  - possibilité de compression des données transmises

clef privées/clefs publiques

- rappel sur les notions
  - chiffrement symétrique
  - de clefs privée/publiques
  - de l'impact de tout ça
- fait en live au tableau

cryptographie et ssh

- clefs d'hôtes: couple clefs privée/publique, sert à garantir que l'on dialogue avec la bonne machine
- clef de session: clef calculée par ssh pour chiffrer la session. algorithme symétrique (une seule clef)
- clef d'authentification utilisateur: méthode alternative d'authentification. Remplace la fourniture du mot de passe de connexion

authentifier les hôtes

- principe général :
  - couple clefs publique/clef privée
    - la clef publique est diffusée par un canal sur aux machines B1, B2, ...
    - en mode non paranoïaque, lors de la première connexion, la machine B1 récupère la clef publique de A
  - la clef privée de A est gardée en lieu sur A
  - à l'aide de la clef publique de A, les machines B1, B2, ... peuvent authentifier la machine A
    - si A fait une connexion ssh vers B1
    - si B1 fait une connexion ssh vers A
  - lors d'une connexion ssh de A vers B :
    - A doit avoir la clef publique de B pour l'authentifier
    - B doit avoir la clef publique de A pour l'authentifier
  - l'échange de clef: première phase de la connexion ssh

Clef de session

- deuxième phase du procole
- la première phase a permis l'échange d'informations chiffrées entre les machines et la négociation des protocoles utilisés (chiffrement, compression, ...)
- Clef de session : une clef pour algorithme symétrique, chiffre l'ensemble des données de la session
- Clef de session: change à chaque session
- Les algorithmes de chiffrement symétriques sont plus rapides

Authentification de l'utilisateur

- 3 méthodes: le serveur indique au client les méthodes qu'il accepte. Le client essaie dans l'ordre fourni par le serveur celles qu'il supporte
  - par login/mot de passe (transitent dans un tunnel chiffré)
  - par clefs privée/publique
    - possibilité de protéger la clef privée par une passphrase
    - possibilité de limiter l'accès avec une clef depuis certains hôtes (cf « man sshd »)
  - par reconnaissance d'hôtes
    - on définit (fichier .shosts) des machines dont les utilisateurs ayant un compte local pourront ouvrir une session localement sans mot de passe
    - ATTENTION: **très** déconseillé pour la sécurité

Authentification des utilisateurs par clefs privée/publique

- permet des connexions sans mot de passe
  - utilise un couple clefs privée/publique
  - clef privée sur le poste client
  - clef publique de l'utilisateur sur le serveur dans ~/.ssh/authorized_keys
  - la clef privée permet à l'utilisateur de s'authentifier
  - possibilité de restreindre les postes depuis lesquels la clef est utilisable (cf man sshd)
  - la clef privée est protégée par une « passphrase »
  - se génère avec « ssh-keygen -t dsa » (par exemple)
  - il existe des outils de gestion de clefs (pour éviter de fournir trop souvent la « passphrase »)

Redirection de ports

- dans une version ultérieure de ce document

Outils windows pour faire du ssh:

- putty/pscp: équivalent graphique de ssh
- filezila: pour réaliser des transferts par sftp (déconseillé en cas d'utilisation de clefs privée/publiques)
- winscp (conseillé): outil graphique permettant de faire du transfert de fichier via sftp/ssh
- doc sur l'accès fichier au département: http://dept.lami.univ-evry.fr/

TP

- connexion ssh sur une machine distante de la salle et lancement d'une application graphique (xeyes)
- idem mais en enchaînant deux connexion ssh de suite A->B->C
- on souhaite pouvoir se connecter sans mot de passe d'une machine du dept à une autre
  - citer les étapes pour y parvenir en indiquant sur quelle machine est à réaliser quelle action
  - mettre en application

rsync

- outil de transfert de fichier optimisé:
  - rsync ne transfère que les fichiers modifiés ou nouveaux
  - il ne transfère que la partie modifiée des fichiers modifiés
- s'appuie sur ssh
- compatible avec l'accès extérieur à vos fichiers du département informatique
- plus d'info:
  - http://samba.anu.edu.au/rsync/tech_report/ : rapport technique sur l'algorithme utilisé par rsync
  - http://samba.anu.edu.au/rsync/documentation.html : site de rsync

Rsync: options utiles

- -r : copie récursive
- -a: archive mode, préserve autant que possible les propriétés des fichiers copiés (sauf les liens physiques, option -H). équivaut à -rlptgoD (cf man)
- -v: affichage verbeux (notamment la liste des fichiers transférés)
- -vv, -vvv: encore plus verbeux
- -z: compresse les fichiers lors du transfert
- --progress: affiche une barre de progression lors du transfert de chaque fichier

Rsync: exemples

- rsync -av --progress petit@asr270-23 :/home/petit/Test Old/ : transfère le dossier /home/petit/Test situé sur la machine asr270-23 dans le dossier local Old. La connexion ssh se fera en tant qu'utilisateur petit. Le dossier Old contiendra un sous-dossier Test
- rsync -av --progress petit@asr270-23 :/home/petit/Test/ Old/ : idem mais le contenu de Test est copié directement dans Old. Aucun sous-dossier Old/Test n'est créé

Rsync: TP

- on utilisera les options --stats, --progress et -v pour avoir des informations sur le comportement de rsync
- recopier l'arborescence créée lors du TP shell dans /tmp
- modifier un fichier de la source et relancer la copie
- créer un fichier texte un peu gros sur la source avec la commande « dd bs=1024k count=10 if=/dev/zero of=bigFile »
- relancer la copie de l'arborescence
- modifier le début de bigFile avec un éditeur de texte et relancer la copie