Système et réseau : mise en oeuvre et exploitation

Géraldine Del Mondo - Mathieu Petit

Version. grincheux et dormeur

1 Introduction

Objectif

La finalité de ce TP est double :

- Vous familiariser avec l'environnement Unix.
- Vous faire analyser une communication entre ordinateurs.

Il s'agira d'effectuer la mise en réseau des 6 machines, de vérifier les protocoles de communication *telnet* et *Ssh* ainsi que d'explorer des mécanismes d'accès et d'exploitation de machine à distance.

A la fin du TP, vous me rendrez le questionnaire fourni en début de TP contenant vos éléments de réflexion. **Ceux-ci seront notés**... J'attends donc de votre part **un document rédigé** et répondant aux questions. **Justifiez vos réponses**.

Remarque

Prenez soin de répondre au questionnaire au fur et à mesure que vous rencontrez les questions correspondantes dans l'énnoncé.

Matériel

La démarche que nous vous proposons consiste à analyser les communications entre ordinateurs. Certains ont comme système d'exploitation Linux (joyeux, dormeur, atchoum"), les autres utilisent Windows ("grincheux", "simplet", "timide"). Les machines seront montées en réseau Ethernet. Dans un premier temps, nous allons provoquer des communications entre les machines. Puis nous analyserons les communications entre celles-ci. Enfin nous exploiterons quelques possibilités plus subtiles d'Unix et Windows.

En utilisant les logiciels de la distribution Ubuntu et des logiciels Windows mis à votre disposition, vous allez réaliser la démarche présentée dans les sections qui suivent.

Remarque

Lors de l'écriture des commandes, sous Linux en particulier, faîtes bien attention aux espaces et à la casse des caractères.

2 Quelques vérifications

Configuration physique du réseau (couche physique-liaison)

Cette opération est a effectuer en concertation avec les autres binomes présents. Les machines sont livrées en état "sortie d'usine" et aucune configuration du réseau n'est effectuée. En premier lieu, et **avant d'allumer les ordinateurs** :

1. Câblez l'ensemble du réseau. Utilisez les câbles fournis ainsi que les 2 hubs.

Question 1 *Présentez un schéma du câblage effectué.*

Question 2 *(1) Théoriquement quel type de cable doit-on mettre entre deux hubs ? (Figure 1[a]) (2) Ici, quel type de cablage avez vous effectué entre les deux hubs (Soyez précis) ?*

[a] Image issue de `http://www.memoclic.com`.

2. **Faites valider votre câblage par l'encadrant !**

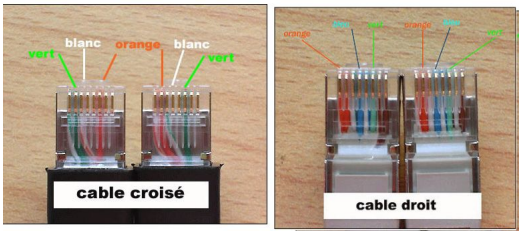

Figure 1: Différences physiques entre cable croisé et cable droit.

Configuration des machines (couche transport-reseau)

Les comptes ouverts sur chaque machine sont du type "eleve" + "nom_de_machine" comme mot de passe ; vous êtes administrateur sur chaque machine. Nous allons configurer les cartes réseau pour pouvoir échanger des trames.

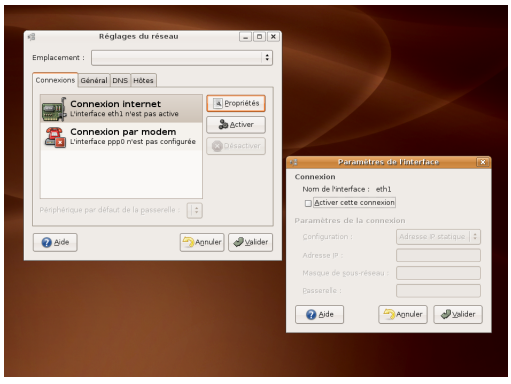

Figure 2: Configuration réseau sous Linux

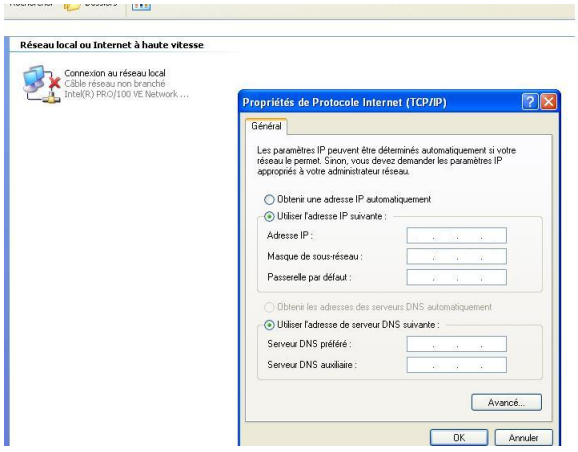

Figure 3: Configuration réseau sous Windows

1. Démarrez les machines et logez vous.

2. Sous Windows, Figure 3, configurez le réseau à partir du panneau de configuration. L'IP est 10.5.0.3, le réseau 10.5.0.0/16.

Question 3 *Précisez le masque de sous réseau*

3. Faites de même sous Linux, Figure 2, à partir du menu *Système→administration→ réseau*. L'adresse IP est 10.5.0.4, le masque est identique à celui de la machine Windows.

4. Testez l'activation de la connexion depuis la console avec les commandes *ipconfig* et *ifconfig*, respectivement sous windows et Linux, respectivement.[1]

Question 4 *Détaillez les informations retournées par ces commandes, comparez ce qui est renvoyé entre Linux et Windows.*

La déclaration des machines (couche application)

Il faut s'assurer du bon dialogue entre les machines. Les machines sur lesquelles vous travaillez s'appellent respectivement "grincheux" (Windows) et "dormeur" (Linux)

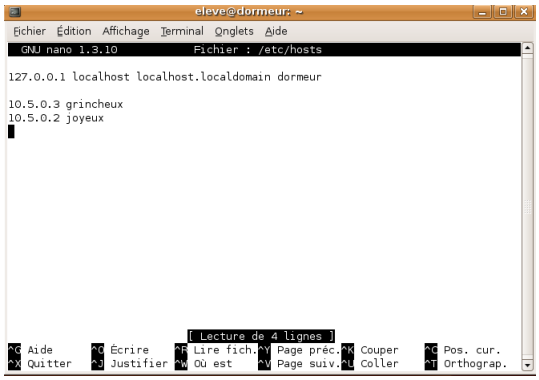

Figure 4: Déclaration des machines sous Linux

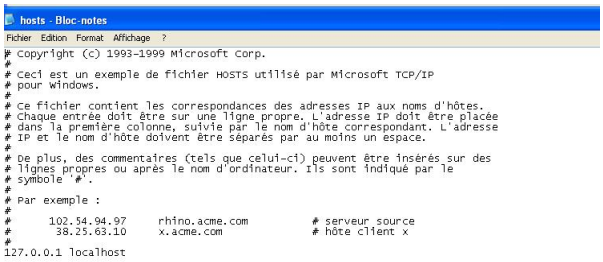

Figure 5: Déclaration des machines sous Windows

1. De **grincheux**, après avoir ouvert une boite de commande MS-DOS, essayez de "pinger" **dormeur**. avec la commande *ping dormeur*

Question 5 *Que se passe-t-il ?*

En effet, pour une bonne communication entre les machines, il faut les déclarer.

2. Sous (Windows), Figure 5, éditez le fichier *c:\windows\system32\drivers\etc\hosts*. **Sans rien modifier d'autre**, ajoutez à la fin du fichier les lignes suivantes (un simple espace suffit entre l'adresse IP et le nom):

[1] Rappel : sous Linux, vous trouvez la console dans *applications→accessoires*, sous Windows, c'est menu *démarrer→tout les programmes→accessoires*.

10.5.0.1	simplet
10.5.0.2	joyeux
...	
10.5.0.6	atchoum

➥ Ne déclarez pas votre propre machine dans la liste !

3. Sous Linux, Figure 4, ouvrez une console et ouvrez le fichier */etc/hosts* en mode super-utilisateur (*sudo*) à l'aide de l'éditeur *nano*, soit la commande : *sudo nano /etc/hosts*. Comme vous exécutez une commande en mode super-utilisateur vous allez devoir entrer votre mot de passe.

   Faites les mêmes modifications que précédemment sous Windows.

> **Remarque**
> Attention, sous l'éditeur *nano*, **tout se fait à partir du clavier**, pour enregistrer vous devez utiliser la combinaison de touche ctrl + o puis entrée, cette commande et d'autres sont détaillées en bas de la fenêtre de l'éditeur.

4. Essayez à nouveau de "pinger" **dormeur**.

> **Question 6** *(a) Que se passe-t-il ? (b) Détaillez les champs des trames reçues sous Windows. (c) Donnez la différence avec les trames reçues dans le cadre d'un ping sous Linux.*

5. Vérifiez que vous arrivez à joindre les stations des autres binômes.

La connexion telnet

On considère maintenant qu'il n'y a plus de problèmes sur le réseau. Nous allons essayer de réaliser une connexion de type **telnet** entre **grincheux** et **dormeur**. Pour cela, nous allons ouvrir un serveur telnet **sur la machine Windows** :

1. Démarrez le service **telnet** sous Windows (voir dans le *panneau de configuration*→ *Outils d'administration* →*services*).
2. Depuis la machine Linux, connectez vous via un client telnet (commande *telnet nom_du_serveur*).

> **Question 7** *Que se passe-t-il ?*

3. A partir de Linux, créez un nouveau répertoire Lambda sur le disque c:\ (commandes MsDOS *cd* et *mkdir nom_du_répertoire*).
4. Vérifiez sur **grincheux** la présence du répertoire nouvellement créé.

> **Question 8** *En quelques mots, à quoi sert un serveur Telnet ?*

La connexion Ssh (couche session-application)

Pour réaliser une connexion Ssh, il est nécessaire qu'un serveur sécurisé soit installé. Le protocole Ssh permet une connexion sécurisée entre deux machines. À l'inverse du point précédent, nous installerons le serveur sur la machine Linux.

1. Vérifiez que le serveur est bien lancé (commande *ps -aux|grep sshd*). **Si ce n'est pas** le cas, la commande pour le lancer est : *sudo /etc/init.d/ssh start*
2. Depuis la machine Windows, connectez vous à la station **dormeur** avec le logiciel **Putty** (qui se trouve sur le bureau dans le dossier TP).
3. Une fois connecté, tapez sous Linux la commande *who*.

> ? **Question 9** *(a) Que constatez-vous ? (b) À quoi sert cette commande ?*

3 La trame TCP/IP

On sait maintenant se connecter sur une machine distante. On va, maintenant, passer au reniflage de notre réseau.

Scan de port avec Nmap

Le commande que nous allons utiliser est **nmap**. Elle s'utilise en ligne de commande, *man nmap* vous donne sa syntaxe.

> ? **Question 10** *(a) Quels sont les ports ouverts sur* ***grincheux*** *? (b) Qu'est-ce que cela signifie pour ces ports particuliers ?*

Analyse de trames TCP/IP

Pour visualiser les trames qui transitent sur le réseau, je vous propose d'utiliser sous Linux : Ether-Real. C'est un outil graphique qui permet de visualiser les trames TCP.[2]

3.1 Protocole telnet

1. Démarrez le logiciel depuis la console **en mode super-utilisateur** (*sudo ethereal*)
2. Lancez un enregistrement de trames depuis Ethereal.
3. **Puis**, ouvrez une session **telnet** sur **grincheux** à partir de **dormeur**.
4. Arrêtez l'enregistrement, fermez la connexion telnet. Les trames reçues s'affichent dans la fenêtre principale de Ethereal (Figure 6).

> ? **Question 11** *D'où proviennent les paquets "sniffés" ?*

[2] Pour information, le même produit existe pour Windows.

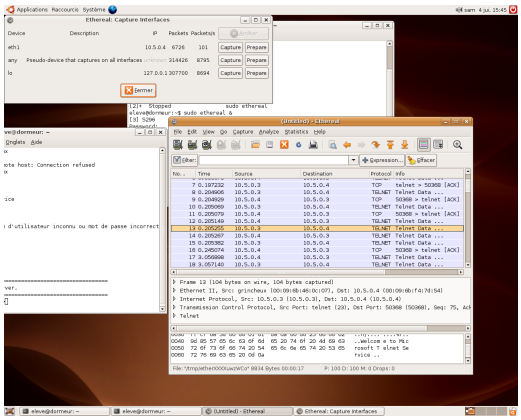

Figure 6: Utilisation du logiciel de collecte de trames Ethereal sous Linux.

5. Triez les trames par "destination" et analysez les paquets envoyés à **grincheux** (on connaît son adresse IP)

> ?
>
> **Question 12** *(a) Retrouvez vous le login et le mot de passe en clair ? (b) Si oui, comment avez vous procédé ?*
>
> **Question 13** *Pourquoi le mot de passe est-il découpé lettre par lettre ? (non, ce n'est pas une question de sécurité...)*
>
> **Question 14** *Pourquoi l'utilisation de switchs à la place de hubs permettrait un niveau de sécurité plus élévé ?*

3.2 Protocole ssh

Nous allons reprendre la même manipulation, avec Ssh et dans le sens Windows→Linux.

1. Lancez une nouvelle collecte de trames avec Ethereal sous Linux.
2. Avec putty, connectez vous à **dormeur**.
3. Arrêtez l'enregistrement, fermez la connexion ssh et passez à l'analyse des paquets reçus.

> ?
>
> **Question 15** *(a) Que remarquez vous ? (b) Concluez sur la sécurité de SSH par rapport à Telnet.*

4 Plus loin avec Linux

À priori, des serveurs SSH doivent tourner sur les 3 machines Linux. Vous connaissez les noms et les mots de passe de ces stations, tout les ingrédients sont là pour permettre une connexion.

"Follow the white rabbit"

Vous devez avoir sur **dormeur** une console en avant plan. À ce point nous pouvons assumer qu'il en est de même sur les stations Linux des autres binomes.

Le but dans cette section va être d'afficher depuis **dormeur** des messages sur la console ouverte sous **atchoum**. Pour cela, prenons un moment pour introduire la gestion des périphériques sous Linux :

Vous utilisez en ce moment même des périphériques d'entrée-sortie... Clavier, souris, écran viennent à l'esprit, mais il faut aussi considérer les disques, cdroms, cartes son, imprimantes, etc. comme des périphériques. Sous Linux, un périphérique est toujours représenté par un fichier rangé dans le répertoire */dev/*.

1. Ainsi, pour accéder au données de la souris, il vous suffit de lire le fichier correspondant dans */dev/*.
2. Dans une nouvelle console, lancez la commande sudo *cat /dev/psaux* et bougez la souris.
3. Voila, nous accédons au périphérique. De la même manière, pour les périphériques en sortie, il suffit d'écrire dans le fichier correspondant. Par exemple, pour imprimer un fichier, on peut utiliser *sudo cat nom_du_fichier > /dev/lp0* ou */dev/lp0* désigne l'imprimante Linux n° 0 (Linux printer 0).

Question 16 ***Décomposez****, c'est à dire détaillez tous les champs de la commande : sudo cat reseau.pdf > /dev/lp0*

Pour des raisons historique, les consoles sont des logiciels considérés comme des périphériques d'affichage physique. On retrouve donc des fichiers d'accès aux consoles dans l'arborescence */dev/*. Les consoles sont de 2 types : "les vrais" terminaux, représentés par les fichiers */dev/tty[1-63]* et les "pseudo" terminaux, représentés par */dev/pts/[0-N]*.

1. Fermez les consoles ouvertes, sauf la console de traffic reseau que vous pouvez nettoyer (commande *clear*).
2. Lancez la commande *who*.

Question 17 *(a) Quels sont les consoles ouvertes ? (b) Par quels fichiers sont elles représentés dans /dev/ ?*

3. Ouvrez une console supplémentaire et dans la console fixe, relancez *who*.

Question 18 *(a) Des changements ? (b) Qu'est-ce qu'un "pseudo" terminal désigne ?*[a]

[a] Vous devez constater la présence d'un vrais terminal dans la liste des connectés, il s'agit de la console qui a servi a lancer l'interface graphique de Linux et qui apparaît brièvement lors de la séquence de démarrage. On peut y accéder avec la combinaison de touches *[ctrl]+[alt]+[F1]* (*[ctrl]+[alt]+[F7]* pour revenir).

4. Écrivez un message depuis la console fixe vers la console nouvellement ouverte : *echo "message transmis">/dev/pts/2*. Celui-ci s'affiche dans l'autre console.

Avec ces bases techniques, vous pouvez maintenant effectuer **la même manipulation sur un ordinateur distant**, en l'occurrence **atchoum**.

1. Connectez vous à l'ordinateur distant par SSH depuis une nouvelle console Linux (commande *ssh nom_ordi*)

2. Lancez la commande *who* et repérez le fichier lié à la console fixe affichée sur **atchoum**. Au passage, vous remarquez que votre propre connexion SSH est identifiée...
3. Effacez le contenu de la console (*commande clear > /dev/pts/[n° _de_ console]*)
4. Envoyez un message ("Wake up Neo" ou ce que vous voulez) et guettez la réaction ou allez constater de visu que le texte s'est bien affiché sur l'ordinateur distant.

Affichage distant

Le serveur graphique de Linux, héritage de XWindow, permet une utilisation en réseau. Ces possibilités ne sont plus exploitées de nos jours mais étaient essentielles dans les années 80-90 où des ordinateurs centraux distribuaient leurs affichages graphique sur une multitude de terminaux. La couche réseau est toujours disponible, et c'est une source potentielle de nuisances... Nous allons effectuer un export d'affichage depuis **dormeur** vers **joyeux** pour faire "surgir" des fenêtres d'application sur l'écran de vos camarades.

La variable d'environnement *DISPLAY* définit les propriétés de l'affichage d'une application graphique. C'est une chaîne de caractères qui s'écrit de la façon suivante :
<adresse_machine>:<n° serveur_graphique>.<n° écran> .

1. **En local**, affichez le contenu de *DISPLAY* (commande *echo $DISPLAY*)

> ? **Question 19** *Sur quel serveur et quel écran s'afficheront les application ?*

2. Vérifiez cela en lançant une application graphique **depuis la console** (commande *xeyes* & par exemple)

Nous allons maintenant déporter l'affichage vers **joyeux** en modifiant le contenu de DISPLAY.

1. Lancez *export DISPLAY="joyeux:0.0"* et vérifiez que la variable à bien été modifiée
2. Lancez une application graphique depuis votre console[3] et guettez les réactions ...

> ? **Question 20** *Comment savez-vous que vos camarades ont fermé les applications que vous avez pu lancer depuis* ***dormeur*** *? (il sera peut-être nécessaire que vous tapiez la touche Entrée)*

Prise de main à distance

Dans cette dernière partie du TP, nous allons utiliser VNC pour prendre complètement la main sur la session Windows **depuis** Linux.

1. Il n'y a **rien à configurer** sous **grincheux**.
2. Lancez ensuite le client **vncviewer** sur **dormeur**. Entrez le nom du serveur.

> ? **Question 21** ***Sur grincheux,*** *comment prendre conscience de la prise en main à distance ? (non ce n'est pas parce que la souris clignote)*

[3] Vous pouvez d'ailleurs en ouvrir autant que vous voulez (dans la mesure du raisonnable) : *xeyes* & *firefox* & *xcalc* & etc.

3. Une fois la session VNC en cours. Lancez une session Ethereal depuis **dormeur** et collectez quelques trames.

Question 22 *(a) Les trames sont elles exploitables ? (b) Pourquoi ? (réfléchissez au type du contenu transféré)*

Question 23 *À la vue des trames collectées, quels sont les inconvénients de la prise de contrôle par VNC ?*

5 Conclusion

Vous avez parcouru en trois heures quelques possibilités de l'utilisation d'ordinateurs en réseau. L'analyse des trames doit vous faire prendre conscience des dangers du manque de sécurité des transmissions qui est malheureusement généralisé aujourd'hui. En effet, toutes les transmissions Web reposent sur le protocole HTTP, qui tout comme telnet transmet ses informations en clair ! Tout les formulaires d'abonnements, les pages de forums, vos éventuels mots de passe et logins transitent à la vue de qui veut bien prendre la peine d'y regarder (nous avons vu que c'est techniquement à la portée de tous). Prendre conscience des faiblesses des réseaux, c'est déjà faire la moitié du chemin vers la sûreté informatique, l'autre moitié vous sera dispensée en cours de voie d'approfondissement.