

Préparation TP Système et réseau : mise en oeuvre et exploitation

Procédure à réaliser avant chaque TP :

Sous Windows :

Démarrer/Panneau de configuration/Connexion Réseau

- Propriétés/Onglet général
- Propriétés du protocole TCP/IP, vider les champs
- Décocher Protocole internet TCP/IP
- Revenir sur l'icone dans Démarrer/Panneau de configuration/Connexion Réseau et désactiver la carte.

Vider les adresses en dessous de 127.0.0.1 localhost dans le fichier
c:\windows\system32\drivers\etc\hosts

Arrêter le service telnet (panneau de configuration/outils d'administration/services)

Sous Linux :

Système/Administration/Reseau/

- Propriétés
- Vider les champs IP et Masque, désactiver la carte
- Désactiver la connexion

Dans un terminal : sudo nano /etc/hosts

- Supprimer les lignes dessous 127.0.0.1 localhost ...
- Pour supprimer ligne par ligne ctrl + k
- Pour enregistrer ctrl + o puis valider
- Pour sortir ctrl + x

Autre :

Dé-cliquer les boutons uplink sur les hubs.

Machine Linux : dormeur / Machine Windows : grincheux

2.1 Configuration physique du réseau (couche physique-liaison)

(Q1) Schéma du réseau (fig1), les noms des machines peuvent être distribués de toute autre façon

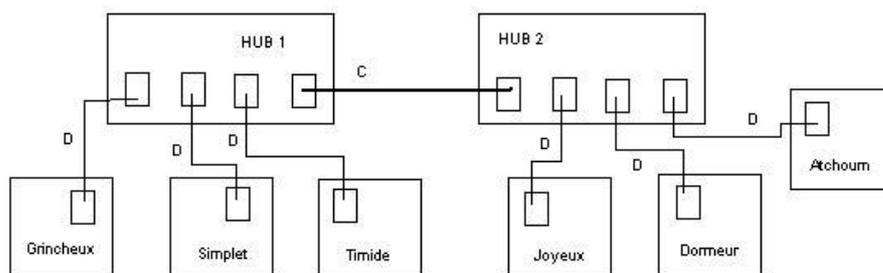


Fig 1 : plan du réseau

(Q2) Le câble croisé (C) devrait servir à brancher les deux hubs entre eux.

Il n'y a pas de câble croisé à disposition mais que des câbles droits (D), mais le hub est dit cascadable c'est à dire que l'on peut inverser l'une des sorties et du coup croiser l'entrée et la sortie d'un câble droit. Concrètement l'un des hubs à son bouton uplink enfoncé et pas l'autre.

Lorsque l'on a tout branché, si les lumières clignotent c'est qu'il y a une détection automatique du réseau par les machines.

2.2 Configuration des machines (couche transport-réseau)

Sous windows :

(Q3) Masque 255.255.0.0

Ici le masque a peu d'importance puisqu'il n'y a d'envoi de paquet que sur un seul réseau. Normalement le masque sert à identifier les différents sous réseaux suivant leurs adresses IP.

Configuration effective : (fig 2)

- Démarrer/Panneau de configuration/Connexion Réseau
- Propriétés/Onglet général
- Propriétés du protocole TCP/IP, remplir les champs
- Pas de passerelle puisque pas d'autre sous réseau à atteindre, on n'utilise pas de serveur de nom non plus.
- Cocher Protocole internet TCP/IP

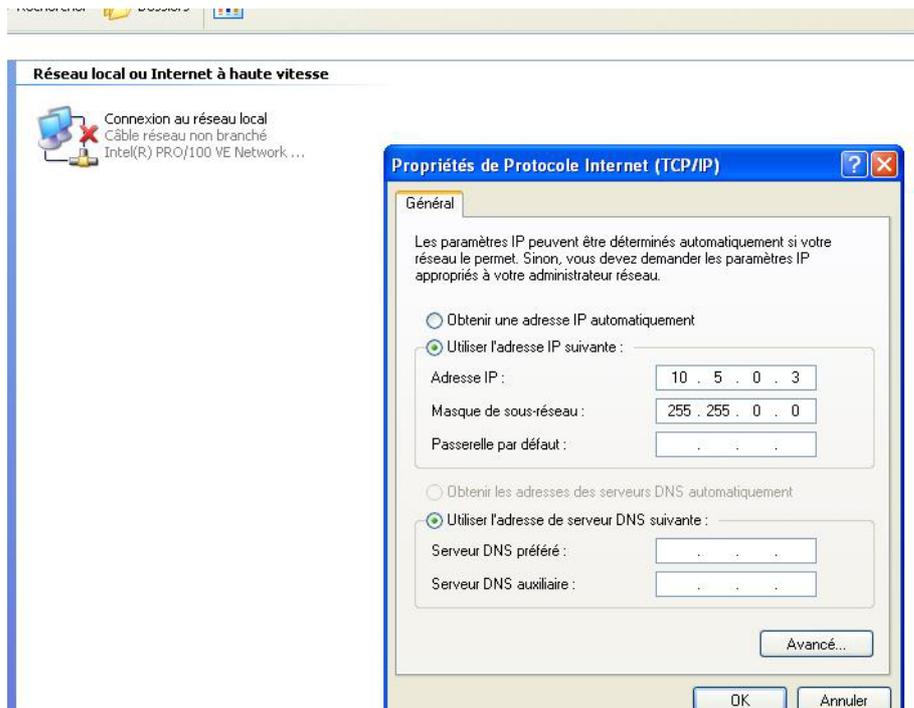


Fig. 2 : configuration réseau sous windows

Sous Linux :

(fig. 3)

- Système/Administration/Reseau/
- Propriétés
- Activer la carte
- Remplir les champs IP et Masque (pour le masque commencer à taper avant le 0 puis supr le 0)
- Activer la connexion

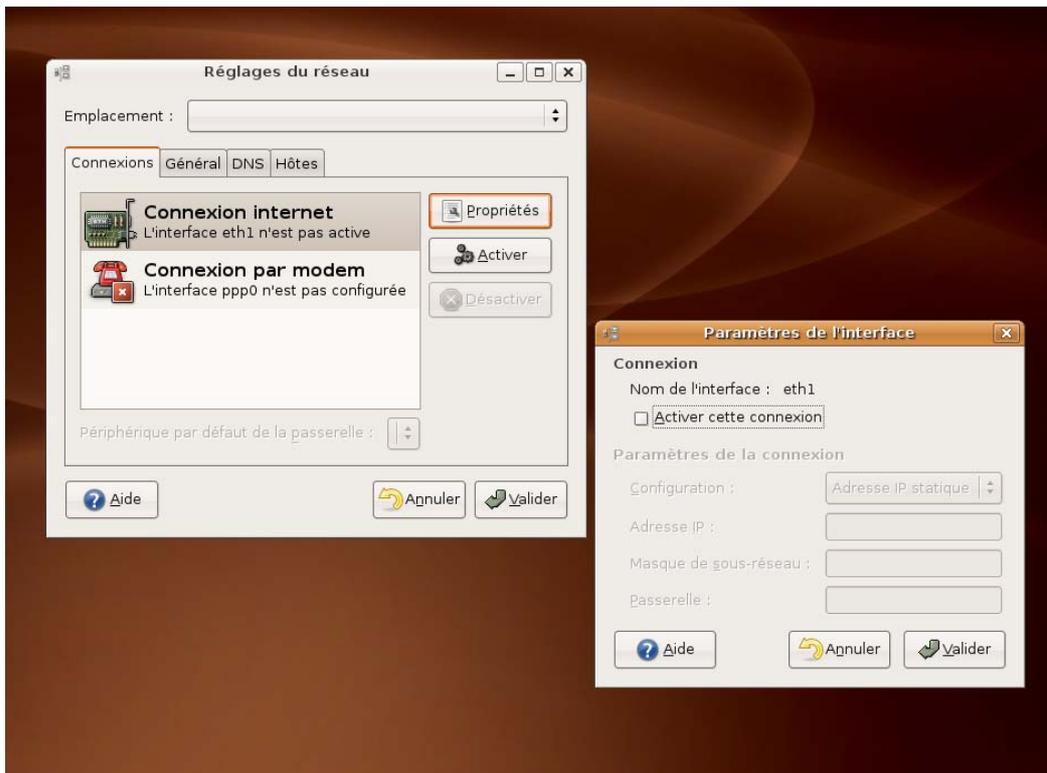


Fig. 3 : configuration réseau sous linux

Test de l'activation de la connexion Windows et Linux

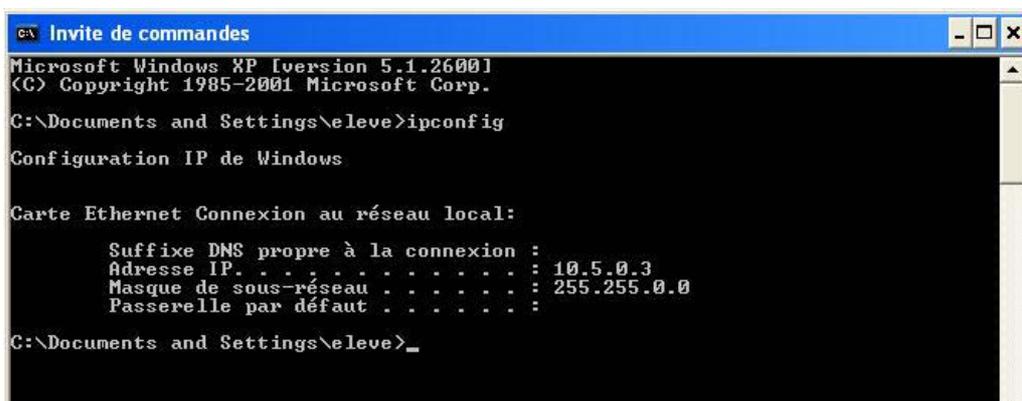


Fig. 4 : Test de configuration réseau sous windows

Windows

Suffixe DNS propre à la connexion : c'est l'adresse du serveur distant éventuel que nous n'avons pas ici. (fig.4)

Si on veut plus d'information ipconfig /all, il se peut qu'on se retrouve avec plein de cartes réseau, ce sont des cartes virtuelles, c'est un protocole (IPV6 à priori) qui les cré.

```

eleve@dormeur: ~
┌───┴───┐
│ Fichier  Édition  Affichage  Terminal  Onglets  Aide │
└───┬───┘
eleve@dormeur:~$ sudo nano /etc/hosts
eleve@dormeur:~$ ifconfig
eth1      Lien encap:Ethernet  HWaddr 00:09:6B:F4:7D:54
          inet  adr:10.5.0.4  Bcast:10.5.255.255  Masque:255.255.0.0
          adr inet6: fe80::209:6bff:fef4:7d54/64  Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reçus:0 erreurs:0 :0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reçus:0 (0.0 b) Octets transmis:936 (936.0 b)

lo        Lien encap:Boucle locale
          inet  adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128  Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          Packets reçus:27625 erreurs:0 :0 overruns:0 frame:0
          TX packets:27625 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          Octets reçus:8329280 (7.9 MiB) Octets transmis:8329280 (7.9 MiB)

eleve@dormeur:~$ █

```

Fig. 5 : Test de configuration réseau sous linux

Linux

lo : look back, correspond au besoin par Linux d'une interface réseau -en particulier pour ce qui est graphique- donc une carte réseau (virtuelle ?) en plus. C'est une nécessité sous linux. (fig. 5)

2.3 Déclaration des machines (couche application)

(Q5) Le ping dormeur depuis grincheux ne passe pas (Attention ne pas utiliser l'IP)

Édition des fichiers hosts :

Sous windows.

Ajouter les adresses des autres machines avec le nom qui leur est associé en dessous de 127.0.0.1 localhost dans le fichier c:\windows\system32\drivers\etc\hosts (fig. 6)

```

hosts - Bloc-notes
┌───┴───┐
│ Fichier  Edition  Format  Affichage  ? │
└───┬───┘
# Copyright (c) 1993-1999 Microsoft Corp.
#
# Ceci est un exemple de fichier HOSTS utilisé par Microsoft TCP/IP
# pour windows.
#
# Ce fichier contient les correspondances des adresses IP aux noms d'hôtes.
# Chaque entrée doit être sur une ligne propre. L'adresse IP doit être placée
# dans la première colonne, suivie par le nom d'hôte correspondant. L'adresse
# IP et le nom d'hôte doivent être séparés par au moins un espace.
#
# De plus, des commentaires (tels que celui-ci) peuvent être insérés sur des
# lignes propres ou après le nom d'ordinateur. Ils sont indiqués par le
# symbole '#'.
#
# Par exemple :
#
#      102.54.94.97      rhino.acme.com          # serveur source
#      38.25.63.10      x.acme.com              # hôte client x
#
127.0.0.1 localhost

```

Fig. 6 : Édition du fichier hosts sous windows

Sous Linux.

Dans un terminal : sudo nano /etc/hosts (fig. 7)

Ajouter les adresses des autres machines avec le nom qui leur est associé en dessous 127.0.0.1 localhost ...

Pour enregistrer ctrl + o puis valider

Pour sortir ctrl + x

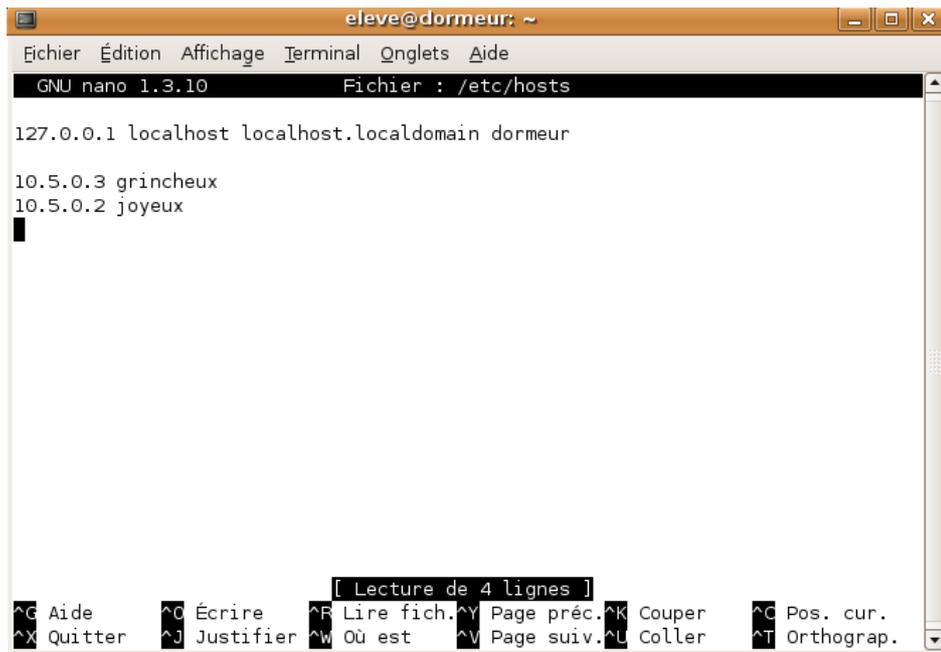


Fig. 7 : Edition du fichier hosts sous linux

(Q6) Maintenant la commande ping fonctionne (fig. 8), on reçoit bien une réponse de grincheux. "icmp_seq=1" numéro de la trame en cours d'envoi.

"TTL" nombre de noeud que la trame a traversé, ici au départ 64 (ça peut être configuré à 32 ou 128 aussi) puis le nombre est diminué de un à chaque élément traversé (ordinateur, hub...) Cette mesure permet de savoir le nombre maximal d'éléments que l'on peut avoir avoir la source et la destination du transfert.

```

eleve@dormeur:~$ ping grincheux
PING grincheux (10.5.0.3) 56(84) bytes of data:
64 bytes from grincheux (10.5.0.3): icmp_seq=1 ttl=128 time=2.39 ms
64 bytes from grincheux (10.5.0.3): icmp_seq=2 ttl=128 time=0.178 ms
64 bytes from grincheux (10.5.0.3): icmp_seq=3 ttl=128 time=0.179 ms
64 bytes from grincheux (10.5.0.3): icmp_seq=4 ttl=128 time=0.178 ms
64 bytes from grincheux (10.5.0.3): icmp_seq=5 ttl=128 time=0.171 ms

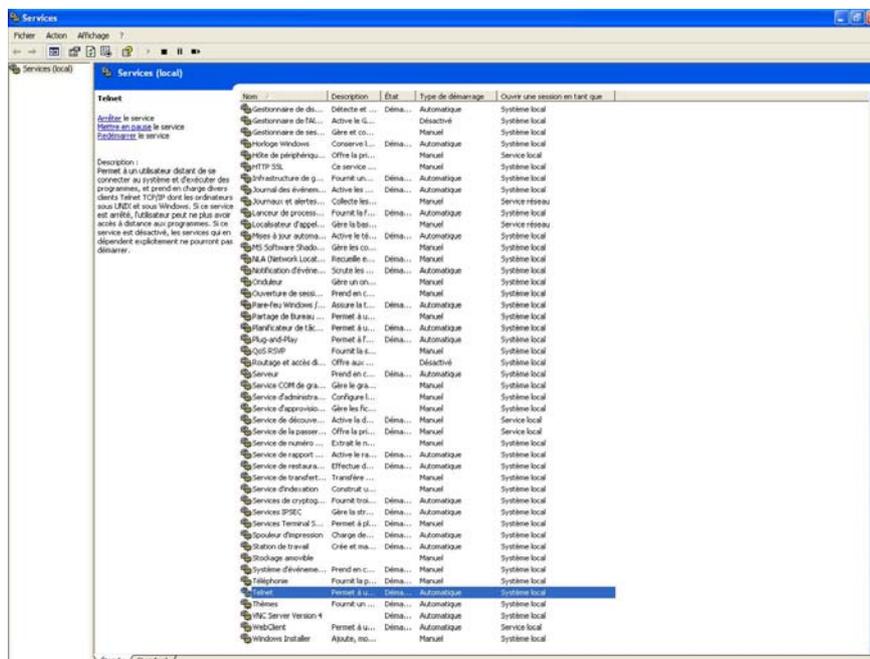
--- grincheux ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 0.171/0.619/2.392/0.886 ms
eleve@dormeur:~$

```

Fig. 8 : Test ping sous linux

2.4 La connexion telnet

Sous windows, démarrer le service telnet (panneau de configuration/outils d'administration/services)



Sous Linux, établir une connexion via un client telnet, attention comme on se connecte depuis dormeur sur grincheux, les login et mot de passe sont ceux de grincheux ! (fig. 9)

```

eleve@dormeur: ~
-----
Fichier  Édition  Affichage  Terminal  Onglets  Aide
-----
eleve@dormeur:~$ telnet grincheux
Trying 10.5.0.3...
Connected to grincheux.
Escape character is '^]'.
Welcome to Microsoft Telnet Service

login: eleve
password:

*****
Bienvenue Microsoft Telnet Server.
*****
C:\Documents and Settings\eleve>mkdir LeNouveauDossier
C:\Documents and Settings\eleve>

```

Fig. 9 : Connexion telnet au serveur telnet sous windows depuis linux

(Q7) La commande telnet sous grincheux a établi que la machine serait serveur, donc dormeur qui se connecte sur ce serveur est son client. Sur Dormeur c'est une console distante qui s'affiche. (fig. 10)

```

C:\ Invite de commandes
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\eleve>dir
Le volume dans le lecteur C n'a pas de nom.
Le numéro de série du volume est 001D-E0DF

Répertoire de C:\Documents and Settings\eleve

03/07/2009  14:16    <REP>          .
03/07/2009  14:16    <REP>          ..
14/11/2008  15:39    <REP>          Bureau
09/01/2009  15:26    <REP>          c
21/11/2008  15:38    <REP>          fabrice
12/09/2008  10:04    <REP>          Favoris
12/09/2008  15:40    <REP>          Lambda
28/06/2007  18:05    <REP>          Menu Démarrer
12/09/2008  10:04    <REP>          Mes documents
19/11/2008  15:55    <REP>          test
02/07/2009  14:46                91 test.txt
             1 fichier(s)                91 octets
             10 Rép(s)  37 143 285 760 octets libres

C:\Documents and Settings\eleve>dir
Le volume dans le lecteur C n'a pas de nom.
Le numéro de série du volume est 001D-E0DF

Répertoire de C:\Documents and Settings\eleve

03/07/2009  14:17    <REP>          .
03/07/2009  14:17    <REP>          ..
14/11/2008  15:39    <REP>          Bureau
09/01/2009  15:26    <REP>          c
21/11/2008  15:38    <REP>          fabrice
12/09/2008  10:04    <REP>          Favoris
12/09/2008  15:40    <REP>          Lambda
03/07/2009  14:17    <REP>          LeNouveauDossier
28/06/2007  18:05    <REP>          Menu Démarrer
12/09/2008  10:04    <REP>          Mes documents
19/11/2008  15:55    <REP>          test
02/07/2009  14:46                91 test.txt
             1 fichier(s)                91 octets
             11 Rép(s)  37 143 285 760 octets libres

C:\Documents and Settings\eleve>_

```

Fig.10 : : Résultat d'une création de dossier sous windows depuis linux via une connexion telnet.

(Q8) Un serveur TelNet établit facilement des connexions réseaux avec des ordinateurs distants.

2.5 La connexion ssh (couche session-application)

Vérification du serveur ssh [sous Linux](#) (fig. 11) : `ps -aux | grep sshd`, la première commande liste les processus en cours la seconde en extrait le démon ssh (sous linux un processus est appelé démon) Normalement il s'affiche deux processus, celui du `grep sshd` que l'on est en train d'effectuer et le processus `sshd` que l'on cherche.

Si ce n'est pas le cas : `sudo /etc/init.d/ssh start` démarrage du processus ssh (`sudo /etc/init.d/ssh stop` arrêt)



```
eleve@dormeur: ~  
Fichier Édition Affichage Terminal Onglets Aide  
eleve@dormeur:~$ ps -aux|grep sshd  
Warning: bad ps syntax, perhaps a bogus '- '? See http://procps.sf.net/faq.html  
root      4575  0.0  0.1  4768 1024 ?        Ss   19:55   0:00 /usr/sbin/sshd  
eleve    6032  0.0  0.1  2932  824 pts/0  S+   20:29   0:00 grep sshd  
eleve@dormeur:~$
```

Fig. 11 : Vérification du serveur ssh sous Linux

[Sous windows](#), on se connecte via Putty (fig. 12&13) (répertoire sur le bureau) à la station dormeur. Putty est un logiciel qui permet de se connecter à distance à des serveurs en utilisant le protocole ssh, Telnet ou rlogin (TCP).

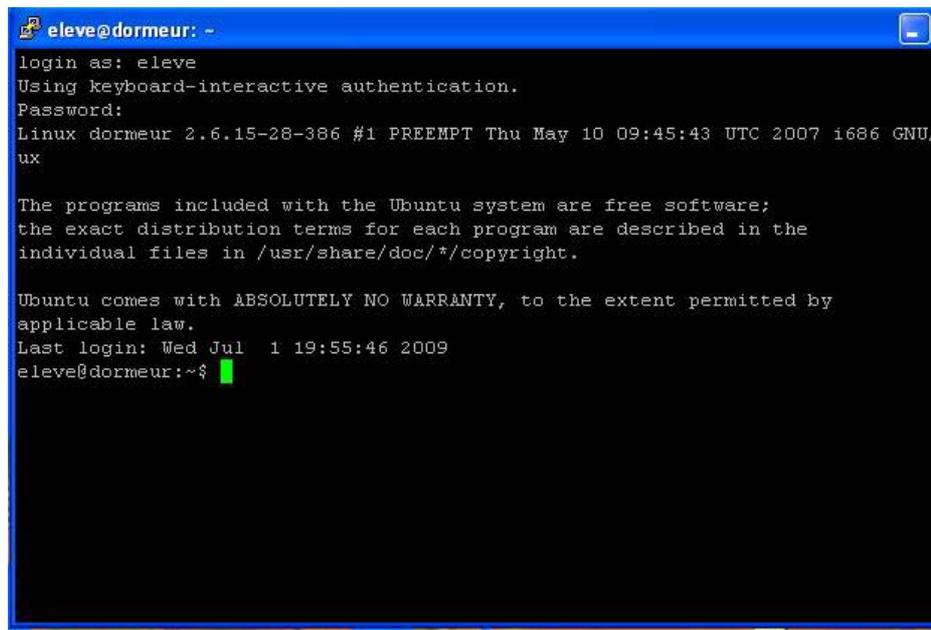
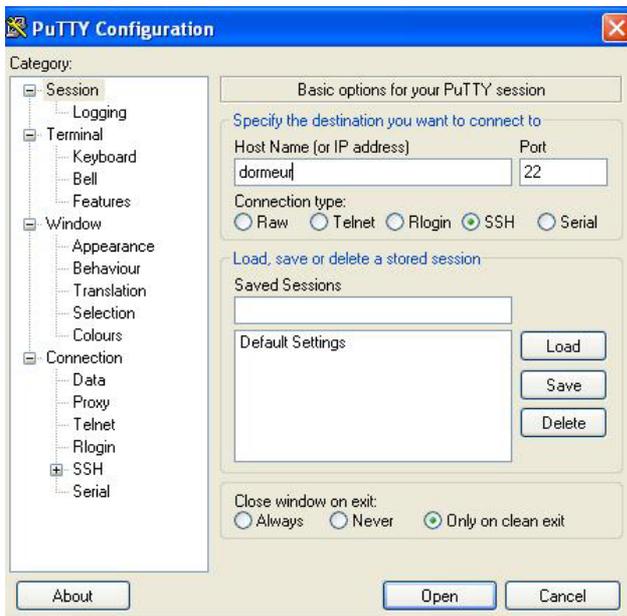


Fig. 12 et 13 : Connexion en ssh au serveur sous linux depuis windows via putty.

(Q9) Sous Linux, la commande `who` (fig. 14) affiche une liste des utilisateurs connectés au serveur ssh.

On dénote :

- le terminal graphique physique de dormeur :0
- un pseudo terminal qui correspond à la console graphique lancée en cours pts/0
- un pseudo-terminal qui correspond à la console lancée sur la machine distante grincheux pts/1

```
eleve@dormeur:~$ who
eleve  :0          2009-07-01 19:55
eleve  pts/0      2009-07-01 20:23 (:0.0)
eleve  pts/1      2009-07-01 20:35 (grincheux)
eleve@dormeur:~$ █
```

Fig. 14 : Résultat de la commande `who` sous linux.

Pour s'en convaincre (fig. 15) on peut lancer un autre terminal et refaire un `who`, un nouveau pts est apparu sur dormeur (:0.0)

Par ailleurs la console de la machine peut être vu avec `ctrl + alt + F1` (`ctrl + alt + f7` pour sortir) SAUF SUR LA MACHINE DORMEUR QUI A UN BUG.

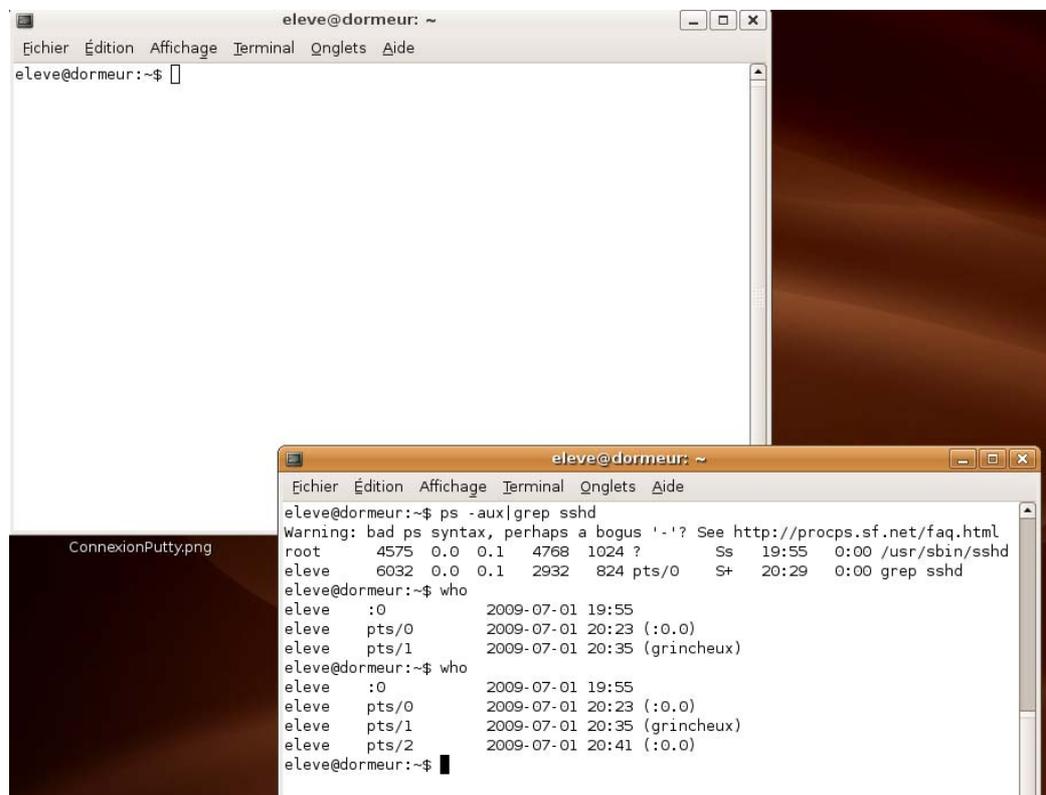


Fig. 15 : Nouveau test de la commande `who` avec 2 terminaux en cours d'exécution

La trame TCP/IP

3.1 Scan de port avec Nmap

(Q10) Ports ouverts sur grincheux : sous linux nmap grincheux (fig. 16)

service -> définition

telnet -> voir paragraphe relatif au telnet

msrpc -> associe le nom des machines aux adresses du réseau

netbios.ssn -> le système de résolution et de transfert windows dans un réseau

microsoft.ds -> partage de fichiers windows entre autre

vnc-http -> voir suite du TP

vnc -> voir suite du TP

```

eleve@dormeur: ~
Fichier Édition Affichage Terminal Onglets Aide
eleve@dormeur:~$ nmap grincheux

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2009-07-01 20:47 CEST
Interesting ports on grincheux (10.5.0.3):
(The 1657 ports scanned but not shown below are in state: closed)
PORT      STATE SERVICE
23/tcp    open  telnet
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5800/tcp  open  vnc-http
5900/tcp  open  vnc

Nmap finished: 1 IP address (1 host up) scanned in 1.502 seconds
eleve@dormeur:~$ █

```

Fig. 16 : Détermination des ports ouverts sous windows depuis linux.

3.2. Analyse de trames TCP/IP

Telnet

Sous Linux :

sudo ethereal (Attention ne pas mettre le & comme sur l'énoncé sinon on ne peut pas entrer le password du super utilisateur)

Lancer un enregistrement des trames dans ethereal (1er bouton en partant de la gauche du logiciel, sélectionner la capture de eth1) et une session telnet depuis Linux (Dans un terminal : telnet grincheux) simultanément. (fig. 17&18)

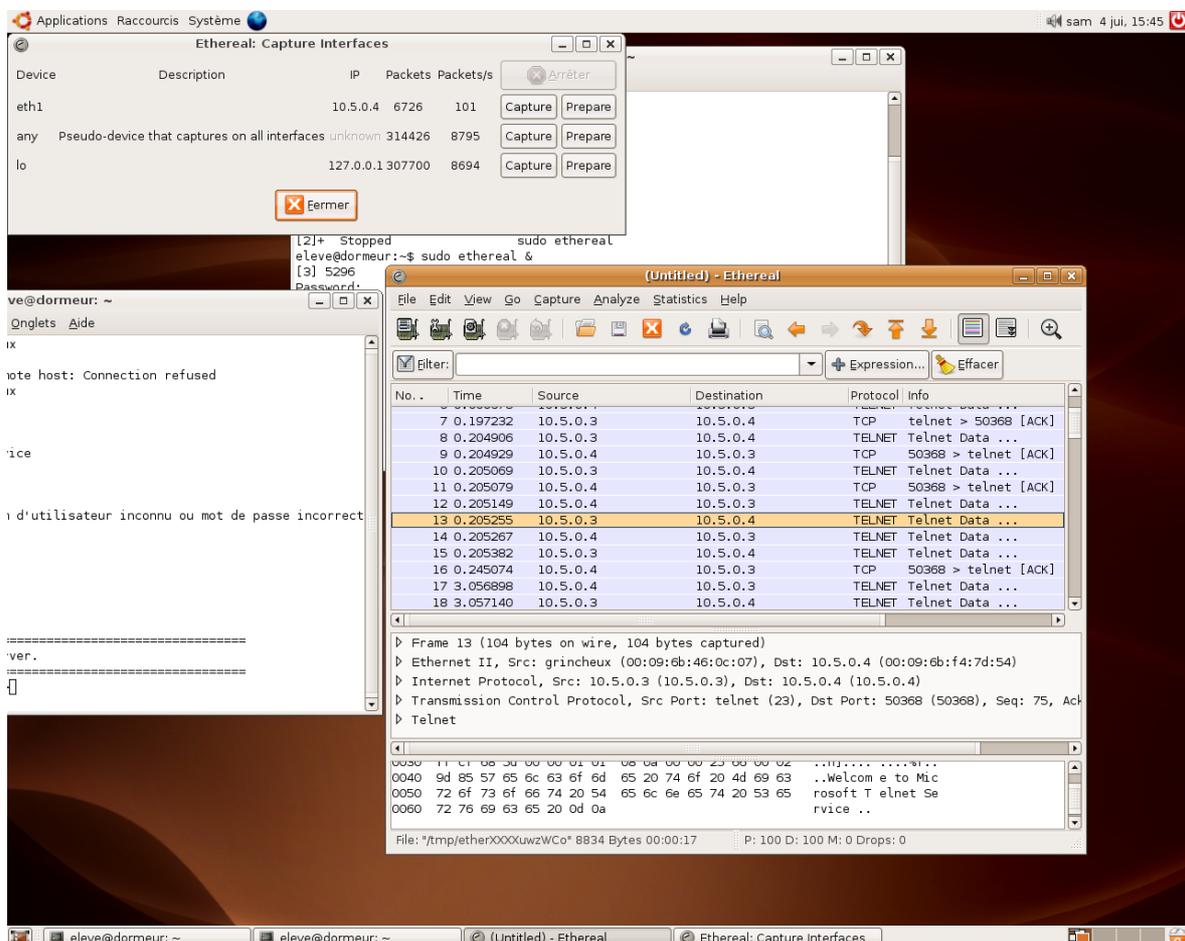


Fig. 17 : Sous l'enregistreur de trames ethereal, on remarque le message en clair dans les données sous une connexion telnet.

(Q11) Les paquets "sniffés" proviennent de la transmission entre grincheux et dormeur via la connexion telnet.

(Q12) On utilise des hubs, éléments passifs du réseau qui retransmettent à tout le réseau et c'est le destinataire qui seul est censé lire les données. Le switch ne transmet qu'au destinataire.

(Q13) Analyser les trames enregistrées, il s'agit de trouver dans les données des trames qui transitent des informations concernant la connexion telnet entre grincheux et dormeur. Par exemple, l'invite de connexion telnet : "Welcome to Microsoft Telnet Service..." (fig. 17) ou la demande de login (fig. 18)... le mot de passe ... qui peut se lire lettre par lettre (du au protocole telnet, comme il ne connaît pas la taille de la donnée à transmettre il la lit caractère par caractère pour ne pas en perdre) **(Q14)**

Problème : Tout ceci est en clair ! Problème de sécurité.

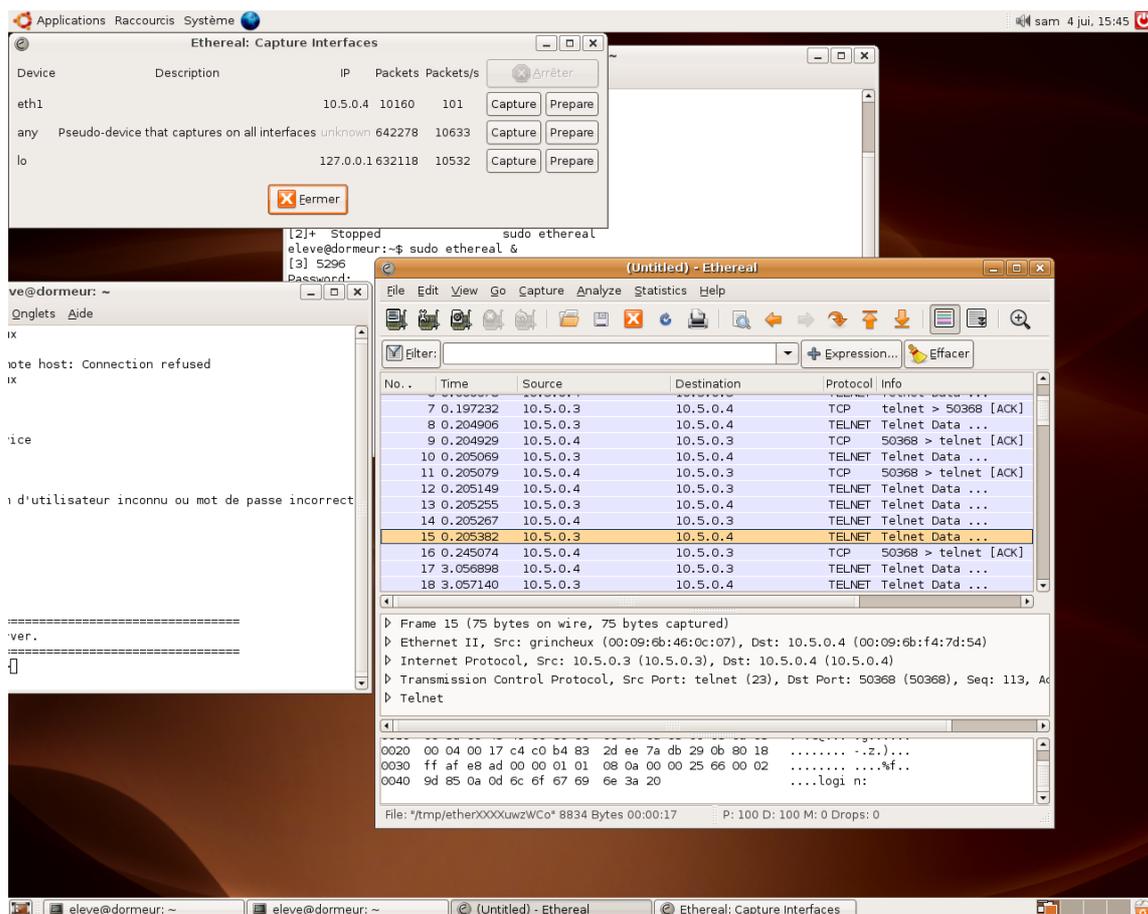


Fig. 18 : Sous l'enregistreur de trames ethereal, on remarque message en clair « login » dans les données sous une connexion telnet.

Ssh

Connexion sous [windows](#) à dormeur avec putty en ssh.
En simultanée, lancer un enregistrement de trames [sous linux](#).

(Q15) Analyse des trames : la longueur des paquets augmente car il y a un accroissement de la sécurité, chaque bit encodé est multiplié par la taille de la clé. Tout y est crypté sauf les clés publiques (Normal on en a besoin pour crypter les messages et renvoyer ensuite, c'est la clé privée qui décode et est détenu par l'envoyeur seulement). Le protocole ssh est plus sécurisé que le protocole telnet

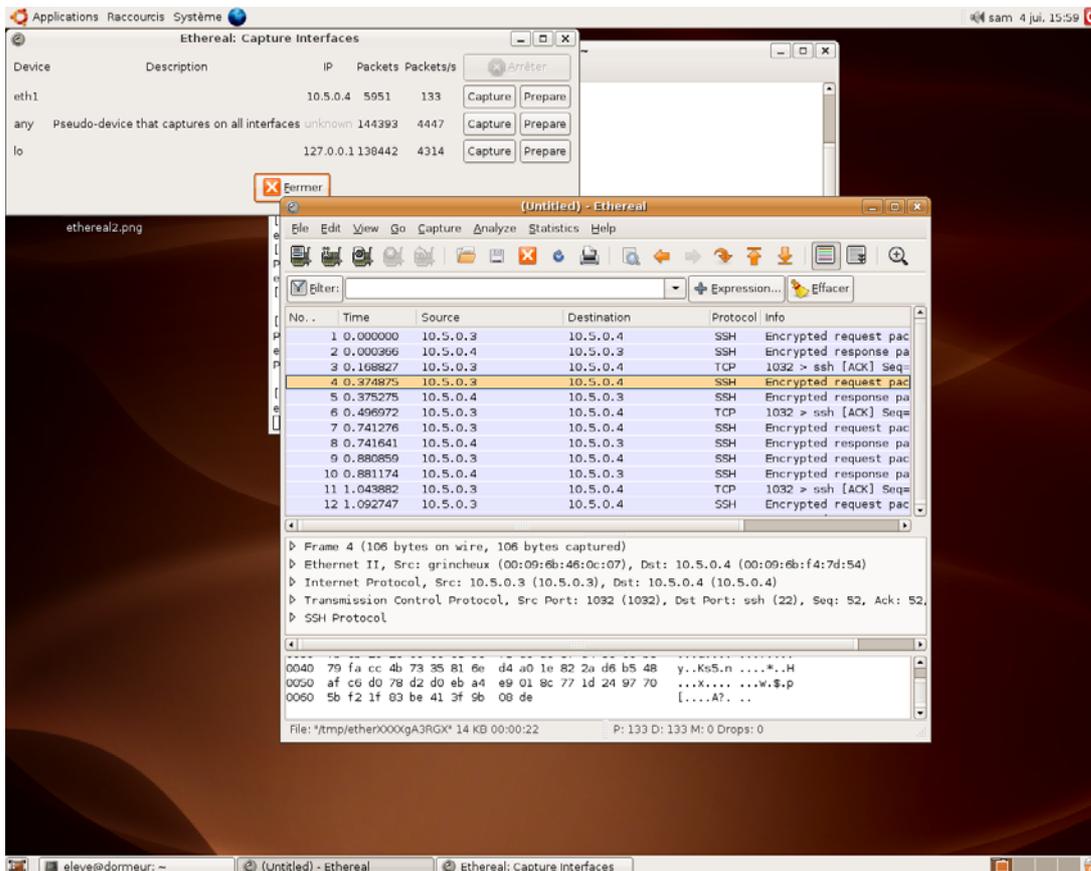


Fig. 19 : Sous l'enregistreur de trames ethereal, on remarque le message crypté dans les données sous une connexion ssh.

Plus loin avec Linux

4.1. "Follow the white rabbit".

sous linux : `sudo cat /dev/psaux` bouger la souris, la sortie de la souris s'affiche à l'écran.

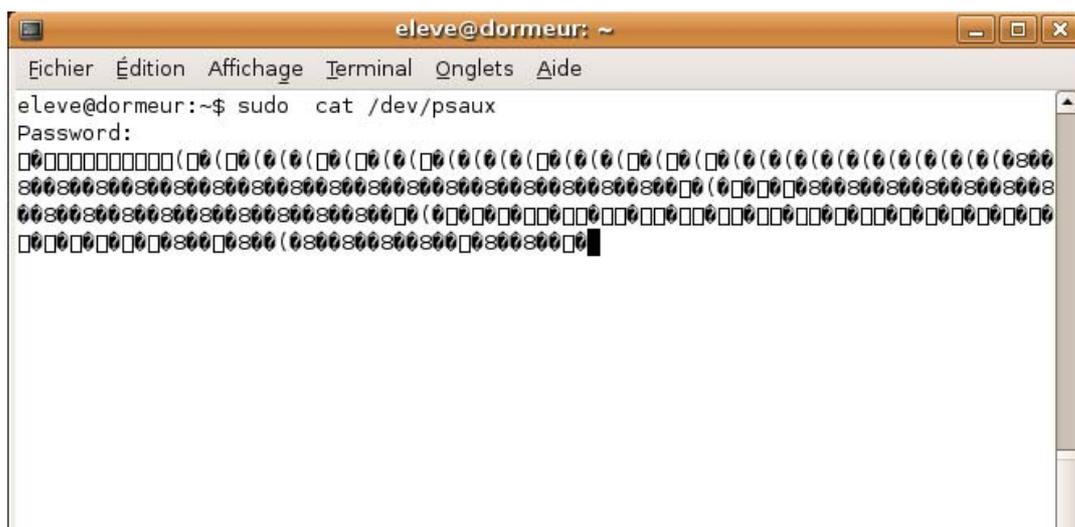


Fig. 20 : Accès au périphérique de la souris sous linux

(Q16) Commande `sudo cat reseau.pdf > /dev/lp0` signifie que l'on souhaite imprimer le fichier `reseau.pdf` sur l'imprimante `lp0`.

(Q17) commande who, les consoles sont désignées sous /dev par pts/0 et pts/2 .

(Q18) Voir résultats [partie 2.5](#)

pts : pseudo terminal interface graphique uniquement, connexion uniquement sur machine distante.

echo "mon message" > /dev/pts/1

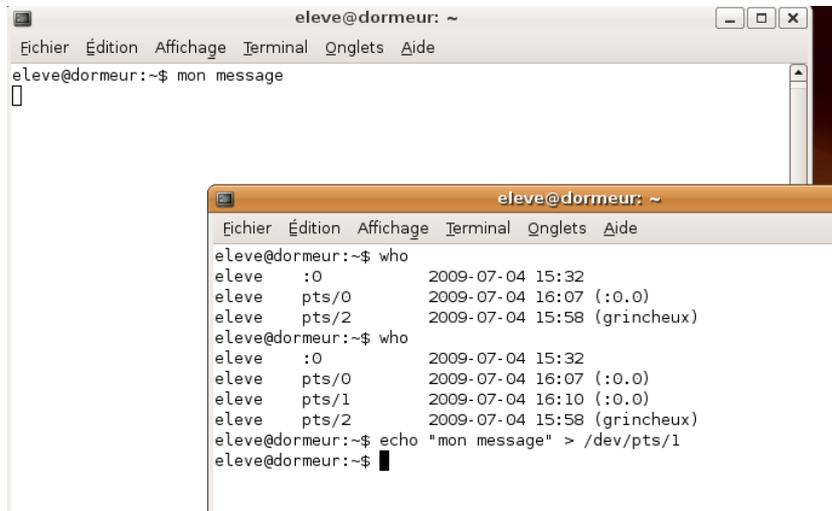


Fig. 21 : Ecrire un message sur un autre terminal que celui sur lequel on se trouve.

Connexion ssh atchoum (une autre machine Linux)

...

Test avec des commande tels que eject, reboot sur les machines distantes...

4.2 Affichage distant

Sous Linux

echo \$DISPLAY renvoie dormeur:0.0

(fig. 22)

dormeur : adresse machine

0 : numéro du serveur graphique

0 : numéro de l'écran

(Q19) Les applications s'afficheront sur le serveur local à la machine sur laquelle on se trouve.

export DISPLAY="joyeux:0.0" sans espace entre joyeux et :

(Q20) Cela s'affiche sur notre écran (Done + nom application éventuellement faire entrée ou lancer autre chose pour que ça s'affiche)



Fig. 21 : Affichage et modification de la variable DISPLAY qui définit les propriétés d'affichage d'une application graphique.

4.3 Prise en main à distance

Normalement il n'y a pas besoin de configurer le serveur sous windows (fig. 22). Le mot de passe n'est pas indispensable. Au cas où clic droit sur l'icone en bas à droite/option

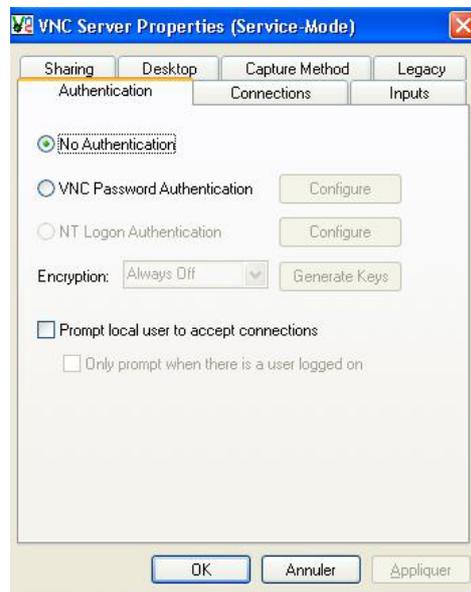


Fig. 22 : Configuration de vnc sous windows.

Sous linux on tape : vncviewer grincheux

(Q21) On peut prendre conscience de la prise en main de l'ordinateur avec le passage de l'icone vnc du blanc au noir. Il existe des logiciels plus discrets... (fig. 23)



Fig. 23 : détection d'une connexion à distance sur la machine vue depuis la machine linux.

(Q22) Les données des trames collectée avec ethereal ne sont pas exploitables, il n'y a que des données brutes mais pas cryptées.

(Q23) Ceci est dû au fait qu'il ne transite que des données liées au graphisme (afficher tel

(Q24) La prise de contrôle par vnc n'est pas sécurisée, et elle surcharge le réseau.