

M2-DNR2I TP1-1: Accès à distance et DNS

Gaétan Richard

10 octobre 2011

1 Accès à distance

1.1 ssh

ssh permet de se connecter à une autre machine à l'aide d'une connexion sécurisée. Essayez par exemple de vous connecter sur la machine de votre voisin. à l'aide de la commande **ssh nom de machine**. Une fois connecté sur la nouvelle machine, lancez la commande *who* et comparez avec le résultat obtenu sur votre machine locale.

Cette opération est également possible depuis l'extérieur en passant par la machine *mike.info.unicaen.fr*. Pour connaître la liste des machines du domaine étudiant, il est possible de lancer la commande **host -l etu.info.unicaen.fr**.

Note : Vous comprenez maintenant pourquoi il est important de **ne pas** éteindre les machines.

Question 1 *Ouvrez trois terminaux : un sur votre machine, un sur la machine *mike* et un sur la machine de votre voisin.*

Question 2 *À l'aide de la commande **who** et **w**, indiquez quels sont les utilisateurs présents chacune des machines et ce qu'ils font.*

Normalement, **ssh** demande un mot de passe pour chaque connection. Il est cependant possible de s'en passer en utilisant une clefs ssh. Pour cela, il faut lancer la commande **ssh-keygen** qui va créer un couple clef publique *.ssh/id_rsa.pub* / clef privée *.ssh/id_rsa*. Cette commande va vous demander d'entre une "passphrase" pour protéger l'accès à votre clef. Vous pouvez laisser cette phrase vide. Une fois les clefs créés, il vous suffit d'ajouter le contenu de la clef **publique** au fichier *.ssh/authorized_keys2* de la machine distante pour se connecter sans utiliser de mot de passe.

Note : Dans le cadre des machines étudiants, comme toutes les machines partagent votre "home", il suffit de faire **cp id_rsa.pub authorized_keys2**.

Question 3 *Créez un couple de clefs sur votre machine pour pouvoir faire des **ssh** sans avoir besoin du mot de passe.*

1.2 scp

La commande **scp** permet de copier des fichiers entre différentes machines. Elle fonctionne comme la commande **cp** sauf qu'il est possible de préciser une adresse sous la forme *login@machine :chemin du fichier*. De plus, si vous avez une clef ssh, aucun mot de passe ne vous est demandé.

Question 4 *Essayez de copier un de vos fichiers dans le répertoire /tmp/ d'une autre machine. Vérifiez le résultat en vous connectant en ssh sur cette machine.*

Question 5 *Depuis votre machine, copiez le fichier précédent créé de la machine distante vers votre machine.*

Pour le transfert de fichier, il existe également des serveurs *sftp*. Par exemple, vous pouvez lancer le client **gftp**. Essayez alors de vous connecter sur *mike* sur le port 22. De tels serveurs existent aussi sous Mac OS X ou windows.

Question 6 *Utilisez **gftp** pour récupérer le fichier /tmp/M1-DNR2I sur la machine sully.*

1.3 screen

Ceci est à faire si vous avez terminé de TP en avance. Vous pouvez sauter cette sous-section lors du premier passage.

Si vous souhaitez travailler à distance via *ssh*, un problème qui se pose est celui des déconnexions intempestives. Pour pallier à ce problème, il existe l'utilitaire **screen**. Cet utilitaire lance un "shell" qui n'est pas arrêté lors d'une déconnexion.

Tapez la commande **screen** pour lancer le programme, il se termine lorsque vous quittez le shell à l'aide de la commande **exit**. La plupart des commandes se font à l'aide de la combinaison de touche **ctrl + a** suivi d'une autre touche. Une aide est d'ailleurs disponible en tapant **ctrl + a** puis **?**.

Pour "détacher" le screen, il suffit de faire **ctrl + a** puis **d**. Vous êtes alors sorti du programme qui tourne toujours en arrière plan, il est possible de le reprendre en utilisant la commande **screen -r**. Si un screen est toujours attaché, il faut alors le détacher avant à l'aide de la commande **screen -d**.

screen permet également de gérer plusieurs fenêtres en même temps. Pour plus d'information sur *screen*, vous pouvez consulter par exemple les pages <http://www.billyboylindien.com/tutos/how-to-screen.html> ou <http://news.softpedia.com/news/GNU-Screen-Tutorial-44274.shtml>.

1.4 Services

Nous allons vous présenter un utilitaire permettant de rechercher les ports ouverts sur une machine. Ce logiciel s'appelle **nmap**. Il est très pratique pour la maintenance réseau mais il est intrusif et ne doit être utilisé que sur des machines sur lesquelles vous avez l'autorisation de le faire.

Question 7 *Trouvez les ports ouverts sur :*

- *votre machine ;*
- *mike ;*
- *mail.info.unicaen.fr.*

Question 8 *Comparer le premier résultat obtenu avec le résultat de la commande **netstat -taupe**.*

1.5 DNS

Pour la suite, vous utiliserez les commandes **host**.

Question 9 *Quelle est l'adresse ip de la machine **mike** ? Quelle est la particularité de cette adresse ? Que se passerait-il si on demande cette adresse de l'extérieur ?*

Question 10 *En exploitant le champ **MX**, trouver quelle machine se charge de transmettre le mail destiné à **unicaen.fr** ?*

Question 11 *Trouver les info du domaine **unicaen.fr** et **info.unicaen.fr** à l'aide de la commande **dig** et du champ **SOA**.*

Question 12 *A l'aide de l'option **-l**, listez toutes les machines du domaine **info.unicaen.fr**, combien y en a-t-il ?*

Question 13 *Trouvez l'adresse ip de la machine hébergeant votre page web à l'université. À l'aide du **dns inverse**, trouvez le nom canonique de cette machine.*

Pour finir, récupérez les informations d'un domaine (ne faites qu'un seul essai) à l'aide de la commande **whois**.