TP1 : réseau sous Linux

Le but de ce premier TP est de vous familiariser avec les commandes Unix liées au réseau, sur une machine physique. Dans les TPs suivants, ces commandes seront exécutées sur des ordinateurs virtuels.

Cours Init Réseau sur Moodle

L'université d'Artois dispose d'une plateforme d'enseignement à distance. Vous y trouverez les supports de cours du module d'initiation aux réseaux et des pointeurs vers des informations supplémentaires.

L'accès au cours nécessite une inscription préalable. La clé d'inscription est `tcpip`.

Accès au webmail de l'université

Les étudiants de l'université disposent d'une adresse email `prenom_nom@ens.univ-artois.fr`. Les emails sont consultables à l'aide d'un navigateur. L'identifiant et le mot de passe sont ceux de votre ENT.

Il est indispensable de consulter régulièrement votre boîte aux lettres électronique universitaire car il s'agit d'un moyen de communication officiel avec l'université et l'équipe pédagogique. L'idéal est d'ajouter la consultation de vos mails universitaires sur votre téléphone en utilisant le protocole exchange. Le serveur de mail est `http://wmail-etu.univ-artois.fr/`, l'identifiant est votre adresse email au format `prenom_nom@ens.univ-artois.fr` et votre mot de passe celui de votre ENT.

Découverte de votre terminal

Sous UNIX, la commande utilisée pour connaître les informations réseau de sa machine est `ifconfig`

```
$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=3<RXCSUM,TXCSUM>
    inet6 ::1 prefixlen 128
    inet 127.0.0.1 netmask 0xff000000
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=1<PERFORMNUD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=10b<RXCSUM,TXCSUM,VLAN_HWTAGGING,AV>
    ether c8:2a:14:47:ed:1d
    inet6 fe80::ca2a:14ff:fe47:ed1d%en0 prefixlen 64 scopeid 0x4
    inet 172.17.141.139 netmask 0xffffff00 broadcast 172.17.141.255
    nd6 options=1<PERFORMNUD>
    media: autoselect (1000baseT <full-duplex,flow-control>)
    status: active
en1: flags=8823<UP,BROADCAST,SMART,SIMPLEX,MULTICAST> mtu 1500
    ether e4:ce:8f:5d:7b:e7
    nd6 options=1<PERFORMNUD>
    media: autoselect (<unknown type>)
    status: inactive
en4: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=60<TSO4,TSO6>
    ether d2:00:1c:3b:c5:80
    media: autoselect <full-duplex>
    status: inactive
en3: flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=60<TSO4,TSO6>
    ether d2:00:1c:3b:c5:81
    media: autoselect <full-duplex>
    status: inactive
```

```
fw0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 4078
    lladdr c8:2a:14:ff:fe:c3:bc:58
    nd6 options=1<PERFORMNUD>
    media: autoselect <full-duplex>
    status: inactive
bridge0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=63<RXCSUM,TXCSUM,TSO4,TSO6>
    ether ca:2a:14:74:c3:00
    Configuration:
        id 0:0:0:0:0:0 priority 0 hellotime 0 fwddelay 0
        maxage 0 holdcnt 0 proto stp maxaddr 100 timeout 1200
        root id 0:0:0:0:0:0 priority 0 ifcost 0 port 0
        ipfilter disabled flags 0x2
    member: en4 flags=3<LEARNING,DISCOVER>
            ifmaxaddr 0 port 6 priority 0 path cost 0
    member: en3 flags=3<LEARNING,DISCOVER>
            ifmaxaddr 0 port 7 priority 0 path cost 0
    nd6 options=1<PERFORMNUD>
    media: <unknown type>
    status: inactive
p2p0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 2304
    ether 06:ce:8f:5d:7b:e7
    media: autoselect
    status: inactive
```

Le champ `ether` permet de connaître son adresse MAC. Sous Linux, on trouve cette information dans le champ “‘HWaddr““.

Votre adresse MAC : `..... : ..... : ..... : ..... : ..... : .....`

Le champ `inet` permet de connaître son adresse IPv4.

Votre adresse IP (v4) : `..... . ..... . ..... . .....`

Rechercher l’adresse IP d’une machine

Les humains utilisent des noms pour identifier les machines. Le protocole Internet utilise les adresses IP.

Pour retrouver l’adresse IP associée à un nom, on utilise les commandes `host`, `nslookup` ou `dig`.

```
$ host www.univ-artois.fr
www.univ-artois.fr has address 194.254.23.3
```

```
$ nslookup www.univ-artois.fr
Server:     193.49.62.9
Address:    193.49.62.9#53

Name:   www.univ-artois.fr
Address: 194.254.23.3
```

```
$ dig www.univ-artois.fr

; <<>> DiG 9.8.3-P1 <<>> www.univ-artois.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33328
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.univ-artois.fr.        IN  A

;; ANSWER SECTION:
```

```
www.univ-artois.fr. 86400   IN  A   194.254.23.3

;; AUTHORITY SECTION:
univ-artois.fr.     86400   IN  NS  ns.univ-artois.fr.
univ-artois.fr.     86400   IN  NS  mailserv.univ-artois.fr.

;; ADDITIONAL SECTION:
ns.univ-artois.fr.  86400   IN  A   193.49.62.9
mailserv.univ-artois.fr. 86400  IN  A   193.49.62.13

;; Query time: 6 msec
;; SERVER: 193.49.62.9#53(193.49.62.9)
;; WHEN: Tue Sep 16 17:09:52 2014
;; MSG SIZE  rcvd: 124
```

Recherchez les adresses IP des machines suivantes :

- moodle.univ-artois.fr
  ....
- wmail.univ-artois.fr
  ....
- www.free.fr
  ....
- www.facebook.fr
  ....
- www.twitter.fr
  ....
- stopcovid.gouv.fr
  ....
- tousanticovid.gouv.fr ....

Pour information, les serveurs de noms principaux sont disponibles ici.

Rechercher le nom correspondant à une adresse IP

Il est aussi possible de retrouver le nom associé à une adresse IP à l'aide de `host`.

```
$ host ns.univ-artois.fr
ns.univ-artois.fr has address 193.49.62.9
$ host 193.49.62.9
9.62.49.193.in-addr.arpa domain name pointer ns.univ-artois.fr.
```

Recherchez les noms correspondant aux adresses IP suivantes :

- 212.27.48.10
  .....
- 80.87.236.17
  .....
- 193.178.244.4
  .....

Se connecter à une autre machine : Telnet

La commande `telnet` est disponible sur la plupart des systèmes d'exploitation.

Cette commande permet de créer un connexion sur une machine particulière à un port particulier.

Chaque port correspond à un service particulier :

- 22 SSH
- 23 telnet
- 25 SMTP
- 80 HTTP
- 110 POP3
- 143 IMAP4

```
$ telnet smtp.monfai.fr 25
Trying 192.168.50.4...
Connected to smtp.monfai.fr.
Escape character is '^]'.
220 smtp.monfai.fr ESMTP Postfix
HELO client
250 smtp.monfai.fr
MAIL FROM: pere@noel.org
250 2.1.0 Ok
RCPT TO: daniel.leberre@univ-artois.fr
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: liste des cadeaux

J'ai bien recu votre commande d'iphone X et d'airpods pour Noel.

.
250 2.0.0 Ok: queued as 356F74B02A2
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

La séquence d'instructions suivante correspond à l'envoi d'un email sur un serveur de mail installé spécialement pour les TPs à l'adresse IP `172.31.130.100`:

```
$telnet 172.31.130.100 25
HELO client
MAIL FROM: harry@potter.com
RCPT TO: guest1
DATA
Subject: Message à la main

Du texte,
sur autant de lignes nécessaires,
on terminera par un caractere . seul sur une ligne
.
QUIT
```

Utilisez ces commandes pour envoyer un email à guest[1-8].

Vous pouvez aussi lire vos mails à l'aide de telnet en utilisant le Post Office Protocol (POP3).

```
$ telnet monpop 110
Trying 172.x.x.x...
Connected to monpop.
Escape character is '^]'.
+OK POP3 monpop v2001.78rh server ready
USER leberre
+OK User name accepted, password please
PASS xxxx
+OK Mailbox open, 3 messages
```

```
LIST
+OK Mailbox scan listing follows
1 11418
2 14440
3 8592
.
RETR 1
+OK 11418 octets
...
QUIT
```

Les commandes importantes sont les suivantes :

- `USER` indique le nom de l'utilisateur
- `PASS` (password) donne le mot de passe pour accéder à la boîte aux lettres
- `LIST` liste les messages, en affichant leurs identifiants et leurs tailles
- `RETR <id>` (retrieve) affiche le message identifié par `id`.
- `DELE <id>` (delete) efface le message identifié par `id`
- `QUIT` termine la communication avec le serveur de mails.

Nous avons mis en place un serveur POP3 pour ce TP à l'adresse `172.31.130.100`.

Les utilisateurs suivants sont disponibles :

- guest1/guest1
- guest2/guest2
- guest3/guest3
- …
- guest8/guest8

Vous pouvez ouvrir une session ssh sur le serveur 172.31.130.100 en tant que guest1, guest2,……guest8 et utiliser la commande `mail` pour envoyer des messages.

Pour se connecter sur la machine `172.31.130.100` comme utilisateur `guest1` :

```
$ssh guest1@172.31.130.100
passwd guest1
```

Pour envoyer un mail de `guest1` à `guest2` :

```
$mail guest2
Subject: mon sujet
Mon message
Cc:
CTRL+D pour terminer
```

Utilisez la commande `telnet 172.31.130.100 110` pour vous connecter au serveur de mails avec l'un des comptes ci-dessus et effectuez les opérations suivantes :

- afficher la liste des messages
- afficher le contenu du 2ème message
- afficher le contenu du 1er message
- afficher les 3 premières lignes du 4ème message (lire la documentation)

Se déplacer de manière sécurisée sur le réseau sous Unix : ssh

Généralement, les serveurs et les machines sous Unix sont administrées à distance. Un moyen sécurisé de se connecter à distance sous Unix est d'utiliser la commande `ssh` (secure shell).

Vous avez le droit à la faculté des sciences de vous connecter à toutes les machines des salles TPs avec le même identifiant et mot de passe.

Le nom des machines est `infoXY.univ-artois.fr` où X est une lettre entre `a` et `l` et Y un entier entre `01` et `25` selon les salles.

```
$ ssh infoe11
The authenticity of host 'infoe11 (172.31.128.181)' can't be established.
ECDSA key fingerprint is be:73:c5:18:d4:fc:ef:ff:14:91:f1:dd:49:d4:30:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'infoe11,172.31.128.181' (ECDSA) to the list of known hosts.
leberre@infoe11's password:
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic i686)

 * Documentation:  https://help.ubuntu.com/
```

Pour chacune des machines suivantes, déplacez-vous sur cette machine pour déterminer son adresse IP et son adresse MAC :

- infok20
  ```
  ..... . ..... . ..... . .....
  ..... : ..... : ..... : ..... : ..... : .....
  ```
- infok13
  ```
  ..... . ..... . ..... . .....
  ..... : ..... : ..... : ..... : ..... : .....
  ```
- infok07
  ```
  ..... . ..... . ..... . .....
  ..... : ..... : ..... : ..... : ..... : .....
  ```

Échanger des données de manière sécurisée

Le but de ces manipulations est de vous montrer comment les échanges sur Internet sont sécurisés. On utilisera l'outil GNU Privacy Guard qui est disponible sur tous les systèmes d'exploitation.

Créer une paire clé publique/clé privée

Pour pouvoir signer ou chiffrer des documents, il est nécessaire de disposer d'un jeu de deux clés : une clé privée qui ne doit jamais être divulguée et une clé publique qui doit être accessible via une source sûre.

La génération de ces clés se fait à l'aide de la commande :

```
gpg --gen-key
```

Il faut répondre à diverses questions : voici un example de génération de clé sur mac OS.

```
$ gpg --gen-key
gpg (GnuPG/MacGPG2) 2.0.26; Copyright (C) 2013 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Sélectionnez le type de clef désiré :
(1) RSA et RSA (par défaut)
(2) DSA et Elgamal
(3) DSA (signature seule)
(4) RSA (signature seule)
Quel est votre choix ? 2
les clefs DSA peuvent faire entre 1024 et 3072 bits de longueur.
Quelle taille de clef désirez-vous ? (2048)
La taille demandée est 2048 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
0 = la clef n'expire pas
<n>  = la clef expire dans n jours
<n>w = la clef expire dans n semaines
<n>m = la clef expire dans n mois
<n>y = la clef expire dans n ans
Pendant combien de temps la clef est-elle valable ? (0) 3m
```

```
La clef expire le Mar 13 déc 15:36:08 2016 CET
Est-ce correct ? (o/N) O

GnuPG doit construire une identité pour identifier la clef.

Nom réel : Daniel Le Berre
Adresse électronique : daniel.leberre@demo.univ-artois.fr
Commentaire : Clé de démo
Vous utilisez le jeu de caractères « utf-8 ».
Vous avez sélectionné cette identité :
« Daniel Le Berre (Clé de démo) <daniel.leberre@demo.univ-artois.fr> »

Faut-il modifier le (N)om, le (C)ommentaire, l'(A)dresse électronique
ou (O)ui/(Q)uitter ? O
Une phrase de passe est nécessaire pour protéger votre clef secrète.

De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: Attention : certains programmes OpenPGP ne peuvent pas gérer
de clef DSA avec cette taille de hachage
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: clef A3A1AF99 marquée de confiance ultime.
les clefs publique et secrète ont été créées et signées.

gpg: vérification de la base de confiance
gpg: 3 marginale(s) nécessaire(s), 1 complète(s) nécessaire(s),
modèle de confiance PGP
gpg: profondeur : 0  valables :   3  signées :   0
confiance : 0 i., 0 n.d., 0 j., 0 m., 0 t., 3 u.
gpg: la prochaine vérification de la base de confiance aura lieu le 2016-12-13
pub   2048D/A3A1AF99 2016-09-14 [expire : 2016-12-13]
Empreinte de la clef = F09E 1E50 CEF4 2CF3 CC22  3D7B BA30 2176 A3A1 AF99
uid      [  ultime ] Daniel Le Berre (Clé de démo) <daniel.leberre@demo.univ-artois.fr>
sub   2048g/E9802827 2016-09-14 [expire : 2016-12-13]$
```

Exporter la clé publique

La clé privée doit rester sur votre machine, dans un endroit sûr. Par contre, la clé publique a vocation à être diffusée pour que l'on puisse vérifier la provenance des données.

La commande pour exporter une clé publique sous la forme d'un fichier pour être échangé est la suivante :

```
gpg --export --armor <email>
```

Pour exporter la clé générée précédemment, on utilisera la commande suivante pour stocker la clé publique dans un fichier `clepublique.txt`.

```
gpg --export --armor daniel.leberre@demo.univ-artois.fr >clepublique.txt
```

Importer une clé publique

Pour pouvoir chiffrer et signer des documents, il faut disposer d'au moins une clé publique autre que celle qui a été générée. Envoyez votre clé publique sur le répertoire temporaire de la machine de votre voisin à l'aide de la commande :

```
$ scp clepublique.txt infoXY:/tmp/clepublique.txt
```

où XY correspond au numéro de la machine de votre voisin.

Vous pouvez alors importer cette clé dans votre trousseau à l'aide de la commande

```
gpg --import  /tmp/clepublique.txt
```

Chiffrer un document

Les documents sont chiffrés avec la clé publique du destinataire. Vous devez donc l'importer avant de pouvoir chiffrer.

Chiffrer un message pour un destinataire particulier garantit que seul le destinataire pourra lire ce message (plus précisément, seule la personne possédant la clé privée associée à la clé publique utilisée).

Utilisez un éditeur de texte pour écrire un message secret à votre voisin dans un fichier `secret.txt`.

Vous pouvez chiffrer son contenu pour que seul votre voisin puisse le lire à l'aide de la commande suivante :

```
gpg -r destinaire@example.com --armor --encrypt secret.txt
```

Cette commande doit créer un fichier `secret.txt.asc` qui peut être envoyé par mail à son destinataire (destinaire@example.com) ou peut être copié sur sa machine à l'aide de la commande `scp` précédente.

Déchiffer un document

Le déchiffrement du document par le destinataire du document est très simple, car il utilise la clé privée que vous venez de générer.

```
gpg --decrypt secret.txt.asc
```

Signer un document

Chiffrer un document permet de s'assurer que seul le destinataire va pouvoir lire le message. Cependant, cela ne donne aucune garantie sur la provenance du message.

Pour vérifier l'origine d'un message, il faut vérifier sa signature.

gpg permet de signer un document avec votre clé privée à l'aide de la commande suivante :

```
gpg --clearsign secret.txt.asc
```

Un fichier `secret.txt.asc.sig` est généré.

Ce fichier doit être envoyé en même temps que le fichier `secret.txt.asc`.

Vérifier la signature d'un document

La vérification de la signature nécessite d'avoir importé auparavant la clé publique du signataire.

La vérification nécessite d'avoir dans le même répertoire les fichiers `secret.txt.asc` et `secret.txt.asc.sig`.

On peut alors vérifier la signature :

```
gpg --verify secret.txt.asc.sig
```

Ce qui donne par exemple :

```
$ gpg --verify secret.txt.asc.sig
gpg: Signature faite le Mer 14 sep 17:23:27 2016 CEST avec la clef DSA d'identifiant A3A1AF99
gpg: Bonne signature de « Daniel Le Berre (Clé de démo) <daniel.leberre@demo.univ-artois.fr> » [ultime]
```

Si le fichier `secret.txt.asc` est modifié, on obtiendra une erreur :

```
$ gpg --verify secret.txt.asc.sig
gpg: Signature faite le Mer 14 sep 17:23:27 2016 CEST avec la clef DSA d'identifiant A3A1AF99
gpg: MAUVAISE signature de « Daniel Le Berre (Clé de démo) <daniel.leberre@demo.univ-artois.fr> » [ultime]
```