Système d’exploitation

**31/03/2014**

bastien.guerry@ens.fr

## Introduction

Internet : toutes les données accessibles via des ordinateurs publics – structure des ordinateurs connectés entre eux. C’est l’ensemble des connexions des réseaux.

Intranet : données privées

Internet est plus grand que le web

Web : un des services accessible sur internet

Services : (fondamental pour connaitre les ordinateurs entre eux)

* Web (port 80)
* telnet (programme pour accéder à n’importe quel service sur un ordinateur) – telnet n\_d\_m 21
* ftp (port 21)
* smtp : sortant
* pop
* secure web port 443

Chaque service est associé à un protocole.

http://

ftp://

smtp:// Chaque ressource doit être unique

https://

Le numéro de port intervient dans où dans url ?

File : protocole

:// => Chemin d’accès

/home/guerry : chemin d’accès dans le programme dans la machine

Ex : [www.lemonde.fr/index](http://www.lemonde.fr/index).html = n’est pas l’url

L’url est : <http://www.lemonde.fr/index.html>

Web : partie populaire de l’internet

Structure client – serveur (2ème notion importante)

* Localhost : n’accepte que les requêtes qui viennent de l’ordinateur même.

La structure client serveur est la même quand on est sur une même machine.

* Daemon : programme qui tourne en tâche de font. Le serveur de mail va envoyer la requête au daemon -> fonctionne comme un serveur : traite les données. Il est interrogé par un autre logiciel : il a un filtre spam par exemple, il va rajouter une en tête spam par exemple

🡪 spam c : spam détecté côté client

🡪 spam d : spam détecté côté daemon

Quand on veut surveiller des tonnes de mail : on utilise pas l’interface client (thunderbird), on utilise le terminal et les lignes de commandes.

Sur linux spam = X – SPAM. 0,7 (probablement du spam)

Un fichier source : sera toujours du texte brut. EN tant qu’administrateur réseau, on cherche des données précises rapidement, et c’est les lignes de commandes qui nous le permettent.

Ex : installation d’un logiciel debian, besoin de passer par des lignes de commandes ? non car il existe des interfaces graphiques pour ça.

Dès que l’on utilise grep et find il faut utiliser les lignes de commandes : obligatoire.

🡪 faire des recherches là dessus

Cloud : service sur Internet qui existe depuis 5 ou 6 ans.

Internet existe depuis les années 70 : premier service d’internet : mail.

HTML origine du WWW

Cloud : moins de données sur les machines, plus de données sur les serveurs.

SaaS : software as a service : on peut faire tourner excel sur notre ordi mais aussi sur google drive ! C’est ça SaaS.

L’utilisateur passe sur le web pour utiliser des services identiques.

Hébergement : louer un espace sur un serveur

* Mutualisé : un serveur pour plusieurs sites internet, on ne peut pas changer la configuration
* Dédié : root sur notre machine, administrateur de notre machine
* Virtualisé : machine virtuelle par client, on peut même avoir une centaine de machines virtuelles sur une même machine : ressemble à un mutualisé car on peut avoir plusieurs clients sur la même machine, autre avantage, on est en mode dédié : car on est en root sur notre machines, administrateur de notre machine, mais partagé avec plusieurs utilisateurs

PaaS : plateform as a service

IaaS : infrastructure as a service

OpenShift : faire des recherches

Ex : amazon : stockage de données, propose des machines virtuelles

Infrastructure : quand on a un très gros site web, on doit avoir plein d’instances de serveur entre lesquelles on échange entre plein de serveur web.

Logiciel libre :

* Gratuit (NON)
* Modifiable
* Opensource (à code source ouvert)
* Libre de droit (NON)
* Copyright
* Droit d’auteur

Exemples : Linux, Unix, openoffice, firefox, thunderbird, notepad ++, vlc

Différence code source compilé et interprété.

Dans la propriété intellectuelle on a :

* DM : droit des marques (firefox par exemple)
* Brevets : on peut breveté un intervention technique sur les forces de la nature, or maintenant on peut breveté qqch qui vient de la nature (gènes, adn, que l’on a trouvé)
* DA : droit d’auteurs (notion d’œuvre, pas intuitif qu’un logiciel soit une œuvre)

Bletchley Park (Alan Turing), Enigma 🡪 faire des recherches

50-70 : IBM : vendre ordi aux universités et aux grandes entreprises.

1969 : état américain intervient car IBM vend ordi, logiciel et maintenance (service + logiciel) 🡺 pratiques qui entrainent le monopole, du coup ils ont été obligés de vendre le service à part (logiciel) de l’ordinateur.

IBM protège ses logiciels avec le droit d’auteur.

Window 95 : explorer servait à explorer tout.

Copyleft : all right reversed : tous droits reversés à l’utilisateur (l’utilisateur a les plein pouvoirs)

Free software fondation : association

En 84 : mouvement du logiciel libre : Free Software Foundation

* GNU : Gnu is Not Unix (système d’exploitation que n’importe qui peut modifier
* Licence qui définit l’usage copyleft de l’utilisateur
* GPL : Gnu private licence
* Linux : noyau qui fait intervenir le système d’exploitation avec la machine

Ce mouvement est pour contrer le fait qu’on le dépose beaucoup de droit d’auteurs et de licences.

On peut installer des logiciels client dans une distribution

**Debian est une distribution :**

Thunderbird

X window

Distribution : écosystème sur lequel peuvent être installés des logiciels et il y a un noyau à l’intérieur

Dans debian on a thunderbird

Window manager peut faire partie du système d’exploitation

Iceweasel : navigateur sur debian, c’est firefox avec un autre nom et design différent

Virtualbox, télécharger un fichier iso et installer linux.

* Ubuntu
* Fedora
* LinuxMint
* ArchLinux
* Debian

Quand on télécharge une file iso, on aura une version plus récente de Firefox.

Lecteur :

C:/

D:/

Disque dur physique : on peut avoir lecteur c et lecteur d sur le même disque dur physique.

Sur unix (base de mac et de linux, alors que GNU a prit unix comme base) tout est dans la même racine

Max OS X existe depuis les années 95

Se base sur free BSD, même chose que Gnu linux qui s’inspire de UNIX, c’est de l’open source.

Il va créer à partir de la Mac OS X.

MAC OS X n’est pas un logiciel libre car les logiciels dérivés ne sont pas libres.

La licence BSD est la plus proche du domaine public.

Unix

Free BSD GNU/linux : GNU GPL

(ouest) MIT Licence

Mac OS X BSD Domain Public

Berkeley software dev

Quand on veut installer un serveur on installe un BSD

1991 : Linus petit mec qui veut utiliser un système unix sur sa petite machine. Il lui manque juste un noyau : Minix, Hurd, noyaux qui ne marche pas

Il crée un truc qu’il met sous licence gpl.

Ssh / crypto

Git

Clé publique (boite dans laquelle les gens peuvent mettre des messages), clé privée (clé qui ouvre ou ferme la boite 🡪 celle que l’on garde toujours sur son ordinateur)

Shell : ensemble de commandes disponibles, par exemple echo est une commande qui fait partie de l’interpréteur shell.

Terminal : environnement de ligne de commande

Bash : born again shell

Telnet x 21

Ssh  user @x : connecte toi à un terminal sur la machine distante, on ouvre un terminal sur la machine distante. Si le serveur est dans la même pièce on a pas besoin de ssh

Si on donne sa clé publique au serveur distant, ça veut dire qu’à tout moment le serveur distant nous connaît et va pouvoir déchiffrer la transaction

A FAIRE :

* GNU/linux (virtualbox)
* Créer une clé publique
* Ssh-keygen
* Créer un compte openshift.com
* Uploader votre clé publique sur OS (openshift)
* Créer un compte github.com
* Uploader clé publique sur Github

Gestionnaire de versions :

* Git
* Subversion : le plus utilisé
* Mercurial

L’ancêtre de tous ces systèmes de version est CVS.

Forge : service web permettant de créer des sauvegardes de systèmes de versions que l’on a sur sa machine SUR LE WEB 🡪 sauvegarde versionning sur le web et permet de partager avec d’autre personnes

* Google code
* Github
* Bitbucket
* Gitorious
* TortoiseSVN : dérivé de SVN

Dans le temps on envoyait un fichier RCS, pour partager les infos.

Le web permet maintenant de faire ça de manière plus simplifiée.

Comment fonctionne le versionning :

On a un dépôt central où les modifications sont enregistrées de telle sorte que la ligne au dessus et en dessous de la modification sont enregistrées.

**28/04/2014**

Pourquoi utiliser les lignes de commandes ?

* Parce que l’on peut récupérer des « mouvements » que l’on a déjà fait en ligne de commandes grâce à la flèche.

**Langage compilé**: en java on écrit un code, on met ça dans des fichiers d’extension et on compile, ce n’est qu’à partir de là que l’on peut lancer le programme.

**Langage interprété**: BASH on a un programme qui s’appelle un interpréteur 🡪 REPL : read eval print loop 🡺 Lecture de ce qui est saisi au clavier, l’interpréteur évalue et print affiche le résultat et loop continue d’exécuter le code.

Cette deuxième famille et notion de REPL : pourquoi ?

BASH est un interpréteur. Quand on est sur une console avec un terminal, on est sous un interpréteur qui est BASH. Ce qu’on exécute ce sont des lignes de code dans un langage de programmation (ls par exemple ou cd)

BASH interpréteur de SHELL

Quand on est sous bash on peut interpréter n’importe quel programme.

Nom relatif :

Nom absolu : chemin pour ouvrir un fichier à partir de la racine

Quand on écrit un programme sous shell on écrit uniquement le nom du fichier mais bash doit savoir où se trouve se programme

On peut changer la liste de répertoire : variable : PATH

ATTENTION : minuscules et majuscules sont importantes

Convention de base : $PATH

**Variable d’environnement**: variable qui permet de donner à bash les éléments nécessaires pour interpréter ce que l’on veut faire

Variable : nom auquel on associe une variable

Comment acceder à une variable : $nomvariable

BASH s’attend à avoir une commande avant le reste de ce que l’on demande, il faut toujours mettre une commande sinon bash ne peut pas interpréter ni deviner ce que l’on veut faire.

En faisant $PATH on lui a demander d’exécuter la variable PATH, mais pas de l’afficher. Si on veut afficher la valeur on doit écrire la commande suivante : echo

2 façons d’interpréter un langage de programmation :

BASH : domain specific language : bash n’interprète que les chaines de caractère : ex une variable contient une chaine de caractère.

EX : PATH dans un autre langage que BASH, on aurait pu avoir un tableau : PATH contient un ensemble de chaine de caractère

Editeur de texte : permet d’editer, gérer, créer modifier du texte brut

* JEdit
* Imax

Traitement de texte : pour faire du code

Complétion automatique

Etc password correspond à des personnes qui veulent se connecter

Commandes de base :

* Editor : sélectionner un editeur de texte par défaut
* Man ‘nom de la commande’ : manuel de dpc : permet d’avoir la doc d’une commande
* Ls : lister un contenu
* Cd accéder à un dossier, se déplacer
* Nano : ouvrir un éditeur de texte
* Cut : **cut -d : -f 1,7 /etc/passwd** :lister le dernier élément pour chaque fichier du document etc passwd dans le nano (editeur de texte)
* cd ~ : revenir au départ
* Sort : permet de classer le contenu d’un fichier
* Uniq permet de ressortir les éléments uniques dans le fichier
* Uniq –c permet d’afficher le nombre de fois que le nom de fichier dans notre fichier apparaît
* Sort –n permet de trier les fichiers par rapport à leur ordre d’apparition par ordre croissant
* Curl nom du lien : télécharger un site ou un fichier ou un logiciel sur mac
* Wget : télécharger sur windows
* Ctrl D : fin de fichier quand on a exécuté une commande qui ne répond pas pendant longtemps
* Nom du fichier.txt > nom du fichier.txt : permet d’enregistrer le fichier ailleurs dans les documents dans un fichier séparé et renommé
* Pwt : nous ramène à tout moment dans le chemin
* Git add \* ou . ou –A
* Git commit –m ‘message’
* Git push (mettre en ligne)
* Git pull (récupérer ce qu’il y a en ligne)

Différence type windows et type linux :

ATTENTION QUESTION EXAM A TRAVAILLER :

Prendre un fichier texte, sortir les mots qui sortent le plus souvent dans l’ordre d’apparition avec le nombre de fois que le mot apparaît dans le texte

**Exercice : trouver la liste des shells utilisés sur la machine que l’on utilise**

cut -d : -f 1,7 /etc/passwd

* Lister le premier et le dernier élément (-f 1,7) pour chaque fichier du document etc/passwd dans le nano (editeur de texte)
* Il faut d’abord définir le délimiteur qui est ici les deux points ( : ), on aurait pu mettre le délimiteur « y » ou autre mais dans le cas particulier de notre fichier on a les deux points qui nous arrangent.

(pour afficher le contenu du fichier on tape : nano /etc/passwd

cut -f 7 -d : /etc/passwd | sort | uniq –c | sort –n

Cette ligne de commande permet d’afficher tous les shells utilisés dans etc/passwd dans le nano (éditeur de texte) par ordre d’utilisation croissant sans doublons (c’est pour cela qu’on utilise sort et uniq) 🡪 voir plus haut pour l’utilisation de la commande

Mode append : on utilise ce symbole >>

Mode overwrite : <

>

curl www.gutenberg.org/cache/epub/45495/pg45495.txt | tr -s -c [:alpha:] '\n'

Cette commande permet de télécharger un fichier et de morceler ce fichier par mot et de mettre à la ligne via ‘ \ n ‘ chaque mot. La ligne de commande va regarder quand il n’y a pas de lettre (les espaces) et passe donc à la ligne on fait ça grace à la commande tr.

curl www.gutenberg.org/cache/epub/45495/pg45495.txt | tr -s -c [:alpha:] '\n' | tr "A-Z" "a-z" | sort | uniq –c | sort –n

Permet de classer l’ordre d’apparition des mots par ordre de fréquence croissant

BASH est un langage interprété avec des variables avec des opérations d’entrée et de sortie

**05/05/2014**

Notation du cours :

* Commit effectués sur le dépôt (commit intéressants)
* Demander à kiagoisepsi2014 et pusher ses notes faites pendant les cours

Ce qu’il faut savoir faire et ce à quoi il faut savoir répondre :

* Comment utiliser path des logs
* Comment utiliser Grep des logs
* Plateforme as a service (openshift gg doc)
* Software as a service (gmail)
* Qu’est-ce qu’un cloud ?

2ème partie de l’exam :

* Questions à propos de l’exam
* Savoir faire des scripts
* Define something about incoming data and do someting about output
* Ex : parsing data trying to receve data
* Faire une liste de website sur lesquels on va et trouver les données qui sont cachées derrière et les intéractions qui sont effectuées

Commande : Export : permet d’exporter un

**12/05/2014**

Questions précises sur Bash :

* Comment créer une variable ? 🡪 Comment définir une variable ?
* Comment parser des données ?
* // Comment créer un dossier ?

mkdir

* Comment faire des comparateurs en BASH ?

#!/bin/bash points="point" **if [ "$points" = 'point' ];** then echo "ok" else echo "pas ok"

* Comment faire pour envoyer des mails en BASH ?
* mail -s "sujet" monadresse@yahoo.fr < fichier.txt
* Comment faire pour annuler une sauvegarde ?
* Comment gère-t’ont les boucles if ou for en BASH ?

Ctrl A : retour à la ligne

Ctrl i : aller à la fin

Ctrl l : nettoyer le terminal

Ctrl c

Exit ou Ctrl d

Ctrl r : rechercher une commande écrite dans le terminal pendant une session

Sudo : super user do

Su pseudo : on switche d’un user à un autre

Su ; cd

Visudo

**Installer des éléments sur un serveur :**

Créer un fichier texte (ssh.txt) sur git avec nom, login que l’on veut utilisee sur le serveur et notre clé publique

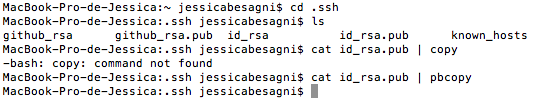
On crée ensuite des users (l’administrateur du serveur va le faire)

On récupère des repository : faire un script

Trouver tous les fichiers qui sont exécutables : dash f find

Avec ces fichiers exécutables, on cherche ensuite à les lancer : exec

Afficher sa clé publique github :



ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDLf5fJPspYUThLoB0aig5EDTCRdxPNwpcEmpiCaVBE/nrxV18WkZ2eRHkE61kJLt8dKMV4OTAubL1V4OwciR7SV2ffuwtlbgi57/2D47D8U7vleafOSo9kA276cR9go0H7nrUzjyerlbayqzl8rJPbGmGj477diiPNQCmh+CMpEwUOxlt04oJ91ed5muCZpgwTjW5dC0K+uKXyhRKp3rkB9pLqKZCHPIDdoHS6M5PrK5rJbeoOL6IJQTlh1xrESlMw637VpliRmE0AE8f6aieGB/E+b5U8gEC4pfQlnThor4McZ5OXjjj+fHWmJGERMRkyuZPX4VJFw+HnkgXDGp2d jbesagni@juniorisep.com