

1)Quelques commandes :

- Que réalise les commandes : "echo \$PATH", "echo \$HOME", "echo \$USER", "echo \$HOSTNAME", "clear", "date", "startx" (si un XWINDOWS n'est pas actif), "uname -a", "uname -snrvmpro", "ps aux", "alias"
- Analysez la commande "find" : en utilisant le joker "*", en évitant les partitions montées manuellement, en ne considérant pas la casse, en masquant d'éventuelles erreurs sur la console.

2)Gestion des droits standards

- Que réalise les commandes "tty", "whoami", "who am i", "who", "w", "su", "sudo", "visudo"
- Pour la suite, on utilisera la commande "sudo" pour des tâches administratives.
- Décrivez les structures des fichiers "/etc/passwd", "/etc/group" et "/etc/shadow", "/etc/sudoers"
- Que réalise les commandes "id", "adduser", "groupadd", "passwd", "groups", "userdel", "groupdel", "usermod", "groupmod", "chfn", "finger"
- Observer les résultats des commandes : "file /usr/sbin/adduser", "file /usr/sbin/useradd", "file /usr/sbin/deluser", "file /usr/sbin/userdel"

3)Gestions par groupes :

- Créer un utilisateur "stud1" : adduser
- Créer un groupe de travail "lprofi" et un second "lproapp" : addgroup
- Comment rendre accessible en lecture seule une ressource au nouveau groupe, en lecture et exécution puis en lecture/écriture : chmod
- Comment ajouter des utilisateurs existant à ce nouveau groupe : usermod -a -G lprofi stud1
- Comment créer un nouvel utilisateur en lui précisant son groupe principal "lprofi"
- Créer un répertoire "/home/lp" d'échange appartenant à "root". Le rendre accessible aux membres du groupe "lprofi" en lecture/écriture et interdit de tout accès pour les autres.
- Peut-on donner l'accès en lecture/écriture à tous les membres des 2 groupes "lprofi" et "lproapp" et toujours interdit de tout accès pour les autres.
- Peut-on proposer des accès différents sur une ressource particulière à des utilisateurs différents.
- Analysez (critiquez) l'administration des droits sous Unix
- Changer les informations personnelles (téléphone, email, ...) de "stud1" : chfn
- Supprimer le compte stud1 ainsi que son home directory.
- *Remarque : le prompt # distingue le compte root d'un quelconque utilisateur, de prompt \$*

4)Visualisation des journaux

- Visualiser le contenu du fichier /var/log/auth.log
- Après une connexion réussie sur un écran virtuel (Ctrl+Alt+F1), observer le résultat dans le fichier des journaux:quand, qui, où ?
- Idem après un échec de connexion.
- Après une connexion réussie via ssh, observer le résultat dans le fichier des journaux.
- Idem après un échec de connexion.
- Idem après une déconnexion
- Que réalise la commande 'cat /var/log/auth.log.1' ?
- Que réalise la commande 'zcat /var/log/auth.log.2.gz' ?
- Que réalise les commandes 'file /var/log/wtmp', 'last', 'last -f /var/log/wtmp', 'last -F'?
- Que réalise les commandes 'lastb', 'lastb -f /var/log/btmp.1', 'lastb -f /var/log/wtmp -F' ?
- Que réalise la commande 'lastlog' ?

5)Configuration d'un serveur HTTP : Apache2

- Grâce à la commande 'ps', vérifiez que la station est serveur HTTP
- Recherchez les fichiers de configuration de ce serveur
- Recherchez le répertoire où placer les pages html que le serveur HTTP considère comme racine : rechercher la chaîne de caractères "DocumentRoot" dans le fichier "/etc/apache2/sites-enabled/000-default", "commonhttpd.conf", "httpd.conf" ou "httpd2.conf" en fonction de la distribution.
- Modifier le fichier 'index.html'
- Ajouter à la racine du serveur, un fichier 'html' proposant un formulaire. Il contiendra une zone de saisie et un bouton de soumission.
- Rechercher où placer des "cgi-bin" pour que le serveur HTTP puisse les exécuter : rechercher la chaîne de caractères "cgi-bin" dans le fichier de configuration indiquant le répertoire des cgi binaires (/usr/lib/cgi-bin)
- Réaliser la commande : sudo a2enMod cgi
- Définir si vous avez les droits d'accès à ce répertoire. Prévoir leurs modifications.
- Vérifier à l'aide d'un navigateur sur cette même station le bon fonctionnement de ces pages HTML en accédant au serveur HTTP localhost.
- Vérifier à l'aide d'un navigateur sur une autre station la possibilité d'accéder au serveur HTTP. Éventuellement modifier :
 - ◆ Dans la section : <Directory /var/www/>
 - ◆ Proposer : allow from all

5)Compilation : Exemple d'un cgi binaire.

- Créer le source C suivant "cgi.c" en ne gardant dans un premier temps que les lignes en caractère gras. Le compiler et placer l'exécutable dans le répertoire 'cgi-bin' du serveur.

```
//Pour ubuntu : gcc cgi.c -o /usr/lib/cgi-bin/cgi.cgi
//Pour ubuntu : sudo a2enmod cgi
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main( int argc, char *argv[])
{
    char *c, name[30], value[30];
    int i, j, nb;
    printf("Content-type: text/html\n\n");
    printf("<HTML><HEAD><TITLE>%s</TITLE></HEAD><BODY>\n", argv[0]);
    printf("<H1>Résultat traitement formulaire</H1>\n");
    printf("Méthode %s\n <BR>", getenv("REQUEST_METHOD"));
    printf("<H2>Données reçues par le CGI</H2>\n");
    if(strcmp(getenv("REQUEST_METHOD"), "POST")==0)
    {
        nb = atoi(getenv("CONTENT_LENGTH"));
        c = (char *) malloc(sizeof(char) * (nb +1));
        for (i=0;i<nb;i++) c[i]=getchar();
        c[i]=0;
    }
    if(strcmp(getenv("REQUEST_METHOD"), "GET")==0)
    {
        nb=strlen(getenv("QUERY_STRING"));
        c = (char *) malloc(sizeof(char) * (nb +1));
        strcpy(c, getenv("QUERY_STRING"));
    }
    printf("Chaîne CGI : <B>%s</B>\n", c);
    //Traitions les différents champs
    i=0;
    printf("<TABLE BORDER><TR><TH>NAME</TH><TH>VALUE</TH></TR>");
    do
    {
        j=0;
        while((c[i]!='\0') && (c[i]!='\n'))
        {
            if(c[i]=='+') name[j++]=' ';
            else name[j++]=c[i];
            i++;
        }
        name[j]=0;
        if(c[i]!='\0') i++;
        j=0;
        while((c[i]!='&') && (c[i]!='\0'))
        {
            if(c[i]=='+') value[j++]=' ';
            else value[j++]=c[i];
            i++;
        }
        value[j]=0;//fin de chaîne
        if(c[i]!='&') i++;
        printf("<TR><TD>%s</TD><TD>%s</TD>", name, value);
    } while(i < nb);
    printf("</TABLE>");
    printf("</BODY></HTML>\n");
    return 0;
}
```

- Vérifiez les droits d'exécution sur 2 niveaux : droits unix et droits apache dans le fichier de configuration (Allow from all)
- S'assurer que le paramètre ACTION du formulaire précédant soit positionné sur ce nouvel exécutable :


```
<form action="/cgi-bin/cgi">
<input type="submit">
</form>
```
- Vérifier en soumettant le formulaire.