Filière: DUT-Informatique / S3

Série 1

Administration des Systèmes d'exploitation

I. Pour assurer la sécurité des mots de passé, Unix a changé la structure de stockage des informations propres à chaque utilisateur. Ici, nous citons un exemple d'une ligne de stockage dans le fichier **/etc/passwd**:

bob:x:1000:1000:BOB,,,:/home/bob:/bin/bash

I.1. Donner la signification de chaque champ de cette ligne en précisant son impact sur la sécurité des mots de passe sous Unix.

I.2. Chercher les informations d'authentification propres à votre système d'exploitation ;

II. Le résultat de cryptage d'un mot de passe dans le fichier **/etc/shadow** ressemble à:

1etNnh7FA$OlM7eljE/B7F1J4XYNnk81

II.1. Déterminer les deux parties composant ce mot de passe crypté ;

II.2. Citer les deux fonctions utilisées pour régénérer ce mot de passe en précisant leurs impacts sur la sécurité de ce mot de passe régénéré ;

II.3. Trouver le mot de passe original saisi par l'utilisateur ;

III. Gestion des groupes et des utilisateurs

Pour protéger son environnement de travail, le système d'exploitation Linux scinde les utilisateurs en trois classes, également, ses droits d'accès:

III.1. Donner ces trois classes des utilisateurs ;

III.2. Créer un nouveau compte utilisateur portant votre nom ;

III.3. Verrouiller le compte utilisateur créé ;

III.4. Changer le nom et l'emplacement de ce compte utilisateur créé.

III.5. Créer un nouveau groupe supplémentaire portant votre prénom (GID : 2021);

III.6. Ajouter ce compte utilisateur à ce groupe ;

III.7. Modifier le nom de ce groupe des utilisateurs ;

III.8. Lister les groupes d'utilisateur ;

III.9. Supprimer l'utilisateur et le groupe créés ;

III.10. **EXERCICE :**

1) Créer deux groupes groupG1 (GID : 2020) et groupG2 (GID : 2021);
2) Créer quatre utilisateurs user1, user2, user3 et user4 ;
3) Rendre les utilisateurs dans les groupes créés :
   a. Les premier et deuxième utilisateurs sont membres du premier groupe ;
   b. Les troisième et quatrième utilisateurs sont membres du second groupe ;
   c. Le deuxième utilisateur est aussi membre du second groupe ;
   d. Le quatrième utilisateur est aussi membre du premier groupe ;
4) Vérifier les membres du groupe groupG2 ;
5) Créer deux répertoires rep1, rep2 et rep3 en seul ligne ;
6) Créer dans rep1 un fichier nommé fich11 et dans rep2 un répertoire nommé rep21 ;
7) Déplacez-vous au répertoire rep21 ;
8) Copier le rep1 et son contenu dans le répertoire courant ;
9) Copier l'arbre rep2 dans le répertoire rp3 ;
10) Visualiser le contenu de rep3 de façon détaillée ;
11) Supprimer l'arbre rep3 ;

IV.Etude de RSA

Le RSA a été inventé par Rivest, Shamir et Adleman en 1978. C'est l'exemple le plus courant de cryptographie asymétrique, toujours considéré comme sûr, avec la technologie actuelle, pour des clefs suffisamment grosses (1024, 2048 voire 4096 bits). D'ailleurs le RSA128 (algorithme avec des clefs de 128 bits), proposé en 1978 par Rivest, Shamir et Adleman, n'a été "cassé" qu'en 1996, en faisant travailler en parallèle de nombreux ordinateurs sur internet.

IV.1 Décrire la description du RSA.

IV.2 Donner toutes les fonctions composantes du RSA.

IV.3 Donner toutes les opérations utilisées dans le RSA.

Etant donné deux nombres premiers p et q .

IV.4 Supposons $p = 5$ et $q = 7$.

IV.4.1 Déterminer l'ensemble des nombres inférieurs à 24 et premiers avec 24.

IV.4.2 Montrer que pour tout nombre premier avec 24 et inférieur à 24 admet son propre inversible modulo 24.

IV.4.3 Montrer, dans ce cas, l'algorithme RSA est symétrique.

IV.5 Supposons $p = 5$ et $q = 11$.

IV.5.1 Déterminer toutes les clefs publiques possibles ;

IV.5.2 Déterminer toutes les clefs publiques pour que RSA soit symétrique.

IV.5.3 On en déduire les clefs publiques pour que RSA soit asymétrique.

IV.5.4 Supposons que $e = 7$.

a. RSA est il symétrique ou asymétrique ? Justifier votre réponse.

b. Calculer la clef privée.

c. Chiffrer votre nom en précisant toutes les étapes.

d. Déchiffrer le message obtenu en précisant les étapes.

IV.5.5 Refaire la question (IV.5.3) en posant $e = 9$.

V. RSA ET OPENSSL

OpenSSL est un outil cryptographique open source particulièrement pratique. Dans la partie consacrée aux références, vous trouverez toutes les informations nécessaires pour le télécharger, mais sachez que la plupart des distributions GNU/ Linux le proposent par défaut. Nous allons donc l'utiliser ici pour configurer un environnement de test dans lequel l'attaque à l'algorithme RSA sera lancée. Pour cela on procède comme suit :

V.1. Créer votre message à crypter avec RSA (nommé plain.tex).

V.2. Exécuter les commandes suivantes puis les interpréter :

V.2.1 openssl genrsa -out rsa_privkey.pem m (m un nombre à préciser).

V.2.2 cat rsa_privkey.pem.

V.2.3 openssl rsa -in rsa_privkey.pem -pubout -out rsa_pubkey.pem.

V.2.4 cat rsa_pubkey.pem.

V.2.5 openssl rsautl -encrypt -pubin -inkey rsa_pubkey.pem -in plain.txt –out cipher.txt.

V.2.6 openssl rsautl -decrypt -inkeyrsa_privkey.pem -in cipher.txt.

V.2.7 openssl rsa -in rsa_pubkey.pem -pubin -text –modulus.

VI. Certificats d'authentification pour les sites web sécurisés : **Firefox**

Un certificat d'authentification numérique aide Firefox à déterminer si le site, que vous consultez, est bien celui qu'il prétend être. Ainsi, si l'adresse web demandée par un utilisateur commence par **https et affiche une clé de couleur vert,** alors, le serveur web demandé assure la sécurité des données échangées. Ici, la confidentialité de la communication avec ce serveur est assurée par un certificat d'authentification numérique. Pour chiffrer la communication, le serveur présentera à **Firefox** un certificat numérique pour s'identifier.

Nous vous demandons de suivre les étapes suivantes afin d'extraire les paramètres de certification propres au site web **www.google.com:**

1. Ouvrir votre navigateur Firefox ;
2. Consulter le site web sécurisé : **www.google.com**;
3. Cliquer sur la clé verte ;

4. Afficher les détails de la connexion ;
5. Lire attentivement les informations affichées sur la page demandée ;
6. Afficher le certificat numérique ;
7. Donner le nom du certificat et l'hiérarchie des certificats;
8. Chercher les informations suivantes :
    a. Entité émettrice du certificat numérique ;
    b. Organisme qui utilise ce certificat numérique ;
    c. Période de validation du certificat numérique ;
    d. Empreintes numériques et ainsi les fonctions utilisées pour calculer ces empreintes ;
9. Citer quelques permissions proposées par votre navigateur au site web https://www.google.com;
10. Donner le protocole utilisé par votre navigateur pour sécuriser la communication ;
11. De point de vu sécurité, citer les fonctionnalités assurées par ce protocole ;
12. Refaire la question VI pour le site web **www.facebook.com.**

VII. Donner le rôle et un exemple d'utilisation de chacune des commandes suivantes:

- **find, sleep, ps, declare, type, userdel;**

VIII. Expliquer la différence entre :

- La suppression et le verrouillage des comptes utilisateurs sous Linux.
- La commande chmod et umask.

IX. Supposant que je suis connecté en tant que administrateur, par défaut, la création d'un répertoire donne les permissions suivantes : **r-- rw- -w-.**

IX.1 Trouver le masque actuel défini pour tous vos fichiers et vos répertoires.
IX.2 Définir un nouveau masque de protection pour tous vos fichiers et vos répertoires (en mode logique et décimale) : 153.
IX.3 Créer un nouveau fichier et un nouveau répertoire.
IX.4 Donner les permissions pour un fichier créé selon le nouveau masque de protection ;

IX.5 Interpréter leurs nouvelles permissions (Fichier et Répertoire).

IX.6 Au niveau sécuritaire :

- Quelles sont vos recommandations sur ce masque ?
- Proposer un masque permettant d'éviter les attaques de suppression ou de modification des fichiers exercées par les membres du groupe ou bien les autres ;
- Quels sont les avantages de votre masque ?

X. PROCESSUS

On considère le tableau des processus suivants :

Proc	Date	Durée
P1	**0**	**6**
P2	**1**	**5**
P3	**3**	**2**
P4	**4**	**3**
P5	**4**	**3**
P6	**5**	**5**

Indiquez dans un diagramme de Gantt le résultat d'un ordonnancement de type :

- FIFO;
- Plus court d'abord ;
- Round Robin, Q=2;

XI. Ecrire un script Shell:

XI.1 Qui affiche tous les fichiers existent dans le répertoire courant.

XI.2 Qui affiche les jours de la semaine (en identifiant les jours de 1 jusqu'à 7) :

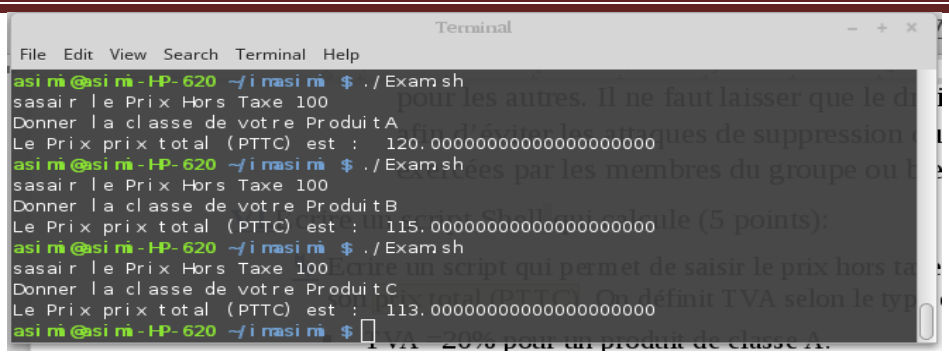
Exemple : $JourSemaine 1 ⟶ le résultat est : Lundi

XI.3 qui détermine si un nombre entier n saisi au clavier est pair ou impair.

XII. Ecrire un script qui permet de saisir le prix hors taxes (PHT) d'un article et de calculer son prix total (PTTC). On définit TVA selon le type de produit:

- TVA =20% pour un produit de la classe A ;
- TVA =15% pour un produit de la classe B ;
- TVA =13% pour un produit de la classe C ;
- TVA =12% pour un produit de la classe D ;
- TVA =10% pour un produit de la classe E.

Exemple d'exécution de ce script :



XIII. Créer un script qui demande à l'utilisateur de saisir une note et qui affiche un message en fonction de cette note :

- "Très bien" si la note est entre 16 et 20 ;
- "Bien" lorsqu'elle est entre 14 et 16 ;
- "Assez bien" si la note est entre 12 et 14 ;
- "Moyen" si la note est entre 10 et 12 ;
- "Insuffisant" si la note est inférieur à 10.

XIV. Ecrire un script Shell qui calcule :

- La somme des carrés de n premiers paramètres non nuls passés en argument à un script :
  - Exemple : $**somCarre** 1 3 6 0 5 9 ⟶ le résultat est : 46.
- Le quotient et le reste de la division euclidienne d'un entier donné x par y (avec x et y sont deux entiers saisis par l'utilisateur).