Sécurité

Le système d'exploitation doit assurer les points suivants :

- Confidentialité des données : un utilisateur non autorisé ne doit pas pouvoir accéder à certaines données
- Intégrité des données : un utilisateur ne doit pas pouvoir supprimer des fichiers s'il n'est pas autorisé
- Disponibilité du système : un utilisateur ne doit pas bloqué tout le système et le rendre inutilisable

I – le chiffrement

Chiffrement assyrien 600 av JC : bande de papyrus enroulé sur scytale.

1) Définition

On appelle texte brut le message ou fichier non crypté et texte chiffré celui qui a été converti de sorte à être incompréhensible. Les algorithmes de chiffrement, et leurs paramètres, appelés clés, peuvent être publics ou secrets (autorisation spéciale pour classer un algorithme « secret défense » par exemple).

Il existe plusieurs types de chiffrement :

On appelle chiffrement à clef symétrique, ou chiffrement à clef secrète, ou encore chiffrement à clé privé, un chiffrement où la clé de déchiffrement se déduit de la clé de chiffrement en appliquant le processus inverse.

Un exemple simple de chiffrement à clé symétrique est la substitution mono-alphabétique, la clé de chiffrement est la suite de lettres correspondant à l'alphabet, la clé de déchiffrement la correspondance inverse.

Exemple (réduit) :

Lettre de texte de départ : A B C D E
Nouvelles lettres : B C E D A
La clé de chiffrement : B C E D A
La clé de déchiffrement : E A B D C

Il existe des algorithmes de chiffrement à clé symétrique suffisamment sûrs (clé de 1024 bits). Un des inconvénients du chiffrement à clé secrète est que l'expéditeur et le destinataire doivent se rencontrer pour échanger la clé.

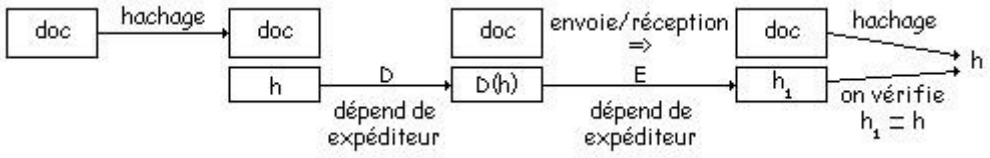
Une alternative est le chiffrement à clé asymétrique, appelé aussi chiffrement à clé public, dans ce cas l'algorithme de chiffrement est connu, mais l'algorithme de déchiffrement est difficile à trouver ou long à mettre en place (on calcule par exemple facilement le carré d'un nombre, mais plus difficilement la raciné carré du résultat).

Les fonctions f telles que le calcul de f(x) soit aisé, mais que le calcul de x connaissant f(x) soit pratiquement impossible sont appelées des fonctions à sens unique.

2) Signature numérique

Pour signaler numériquement un document, on procède souvent comme suit : on applique un algorithme de hachage au document, on obtient un résultat h de taille fixe (par exemple 16 octets pour MD5, Message Digest, 20 octets pour SHA, Secure Hash Algorithm), appelé la valeur de hachage ou empreinte, on envoie ensuite le document avec en plus D(h) (où D est une fonction avec une clé privée dépendant de l'utilisateur). Celui qui reçoit le document le fait repasser par l'algorithme de hachage et connaît une fonction E telle que E(D(h)) = h, il vérifie alors que les deux résultat coïncident.

Illustration :



3) Sténographie

Une image couleur sur 3x7 bits (RVB) est indifférenciable à l'œil d'une image 3x8 bits. On peut utiliser un des bits de l'octet codant chaque couleur pour cacher un message.

II – Quelques aspects système d'exploitation

1) Authentification

Il existe plusieurs façon d'authentifier un utilisateur pour ouvrir une session. Si on prend l'exemple du mot de passe, le système d'exploitation facilite plus ou moins le travail d'un intrus qui cherche à se connecter selon :

- Qu'il accepte un mot de passe simple ou non
- Qu'il affiche le nombre de caractère du mot de passe
- Qu'il réagit à un mauvais nom d'utilisateur avant d'avoir saisit le mot de passe
- Qu'il ne permette qu'un nombre fini d'essai
- La façon brut ou chiffré dont il stocke la table des paires utilisateur/mot de passe
- Qu'il accepte un mot de passe à durer limité ou non

L'authentification peut se faire également par défi-réponse, par l'usage de carte (on parle alors d'authentification physique), par biométrie.

2) Domaines de protection

Un domaine de protection est un ensemble de couples (objets, droits), les objets pouvant être d'ordre matériel : segments mémoire, lecteur de disque, imprimante, …, ou d'ordre logiciel : processus, fichiers, …, et les droits des opérations possibles sur ces objets. Un domaine peut concerner un utilisateur précis ou un groupe d'utilisateur. A chaque instant, un processus s'exécute dans le cadre d'un domaine bien déterminé et peut changer de domaine (lors d'un appel système par exemple) au cours du temps.
Il existe deux approches pour définir les domaines :

- La liste de contrôle d'accès
- La liste des capacités

On prend comme objet l'exemple des fichiers. Dans la liste des contrôles d'accès, ou ACL (Access Control List), on associe à chaque fichier une ACL contenant les utilisateurs autorisés avec les actions permises, par exemple :

Fichier F1 => UiD1 : RW, UiD2 : R, GiD1 : R
Fichier F2 => UiD1 : RWX, UiD2 ; RX (R : Read (lire), W : Write (écrire), X : eXecute (exécuter))

Dans la liste des capacités, ou C-list (Capacity list), à chaque utilisateur, on associe la liste des fichiers avec les droits autorisés, on reprend l'exemple :

UiD1 : F1 : RW, F2 : RWX
UiD2 : F1 : R, F2 : RX
GiD1 : F1 : R

La C-list permet une vérification rapide des droits lors de l'exécution d'un processus, et la suppression d'un utilisateur est facile à gérer, par contre la suppression d'un fichier est plus facile avec l'ACL. On appelle matrice de protection l'ensemble des droits indépendamment de la présentation, ici par exemple on obtient :

	UiD1	UiD2	GiD1
F1	RW	R	R
F2	RWX	RX	

Certains systèmes peuvent avoir une liste d'états autorisés, et vérifier régulièrement que la matrice de protection correspond à un état autorisé.

3) Sécurité multiniveau

Lorsque l'accès est contrôlé de façon individuelle, on parle de contrôle d'accès discrétionnaire. De plus, dans des environnements hautement sécurisés (armée, laboratoires, les individus appartiennent à des classes hiérarchisées qui n'ont pas toutes les mêmes droits, on parle de sécurité multiniveau.

Dans le modèle Bell - La Padula, conçu pour l'armée (secret des données), un processus lancé par un utilisateur prend le niveau de sécurité de l'utilisateur. La circulation des informations suit les deux principes ci-dessous :

- Propriété de sécurité simple : un processus ne peut lire que des objets de son niveau ou de niveau inférieur
- Propriété * : un processus ne peut écrire que dans des objets de son niveau ou de niveau supérieur

Dans le modèle BIBA, conçu pour des entreprises (intégrité des données), on a les propriétés inverses :

- Propriété d'intégrité simple : un processus ne peut écrire que dans des objets de son niveau ou de niveau inférieur
- Propriété d'intégrité * : un processus ne peut lire que des objets de son niveau ou de niveau supérieur

Il existe des normes standards de sécurité qui classe les systèmes d'exploitation suivant leur caractéristiques en matière de sécurité (livre orange).

III – Les attaques

1) Attaques internes

Le cheval de Troie consiste à mettre dans un programme en apparence inoffensif du code qui modifiera, supprimera ou copiera des données ou du code.

Un classique de l'usurpation d'identité est le remplacement de la fenêtre de connexion par une fenêtre d'aspect identique, mais qui stockera le nom d'utilisateur et le mot de passe dans un endroit accessible à l'usurpateur, ensuite on génère la commande de saisie non valide qui renverra un message d'erreur et redemandera la saisie pour la connexion.

Il existe d'autres attaques qui peuvent être introduites par le personnel de l'entreprise. les bombes logiques sont des programmes visant par exemple à supprimer des données si le nom de l'employé ne figure plus dans la liste des utilisateurs. Les trappes (ou entrées de service) permettent de contourner le système de sécurité de connexion en rajoutant par exemple une ligne de code qui n'entraîne aucune vérification si on saisit une chaîne particulière comme nom d'utilisateur.

2) Virus

Il existe plusieurs types de virus : les virus compagnons (lignes de code rajoutées à un exécutable), les virus d'écrasement (remplace le code d'un exécutable), les virus résidents en mémoire (se place en mémoire au démarrage et s'effectue par exemple à chaque appel système), les virus du secteur d'amorçage (écrase le MBR ou le 1er secteur de la partition active), virus de pilote de périphérique, virus macros, les virus polymorphes (le code se modifie pour échapper aux anti-virus)…

3) Ver

Un ver, contrairement à un virus classique, est autonome : il n'a pas besoin de parasiter un programme ou des données, et ne nécessite aucune action particulière de l'utilisateur pour pouvoir agir. Il est autorépliquant. Il contient en général deux parties : l'amorce (qui sera envoyée à d'autres destinataires sans aucune action spécifique de l'utilisateur (via les tables de routage par exemple), et la partie virus proprement dite.