Généralités sur la sécurité UNIX

- La sécurité informatique est un problème extrêmement complexe, et de surcroît mal compris par une grande partie des intervenants. Des problèmes très différents sont recouverts par ce terme :
  - ✓ Protection contre les dommages et les sinistres informatiques (qu'ils soient accidentels ou malintentionnés), non-divulgation de données confidentielles, authentification d'un correspondant, isolation des différents utilisateurs d'un même système informatique.

- Le système UNIX est devenu de plus en plus sécuritaire.

- Plus de 90% des problèmes de la sécurité d'UNIX sont dus à des erreurs de la part des utilisateurs et des administrateurs de système.

☞ Pourquoi se soucier de la sécurité?

- L'information peut être deviné (*mauvais mot de passe*).

- Les ressources ne sont pas inépuisables (*espaces sur disque, temps de calcul...*).

- Votre système n'est *jamais* seul:
  - ✓ Système d'information global, constitué de nombreux systèmes interconnectés.
  - ✓ Partage de l'information et des ressources.
  - ✓ Préjudice :
    - Préjudice pour vos données personnelles,
    - Préjudice pour votre entreprise

☞ Sécurité physique et logique :

- La sécurité physique est la sécurité des locaux dans lesquels est installé le système informatique.

- Elle concerne les protections contre les intrusions, le vol, les accidents, les risques naturels (inondations, incendie, fumée...).

- Elle peut conduire à l'utilisation de clés, badges, cartes d'accès (électrique, magnétique), gardes, détecteurs...

- La sécurité logique est la sécurité fournie par le système d'exploitation de la machine et les logiciels de base.

- Elle est sous la responsabilité de l'administrateur de système.

- Cependant, en tant qu'utilisateur, vous avez quand même un rôle à jouer en rapportant à votre administrateur de système vos soupçons sur d'éventuelles violations de sécurité.

- La protection physique est aussi importante que la protection logique:
  - ➔ *L'une n'a pas de sens sans l'autre.*

- Sans perdre de vue la sécurité physique, c'est sur la sécurité logique que nous allons travailler.

☞ En quoi consiste la sécurité logique?

- La sécurité logique concerne la protection de l'information dans l'ordinateur et le contrôle d'accès à ses ressources:
  - ✓ La protection contre la divulgation non autorisée de l'information (la *confidentialité*). Menaces potentielles: intrusions, chevaux de Troie, accidents, espions.
  - ✓ La protection contre la destruction ou la modification non autorisée de l'information (l'*intégrité*). Menaces potentielles: intrusions, bombes logiques, chevaux de Troie, bogues, accidents, sabotages.
  - ✓ Protection contre le refus ou la dégradation de service (la *disponibilité*). Menaces potentielles: intrusions, bombes logiques, chevaux de Troie, bogues, accidents, sabotages.
  - ✓ Protection contre les incohérences du système d'information (l'*homogénéité*). Menaces potentielles: intrusions, bombes logiques, bogues.
  - ✓ Protection contre l'accès interdit, contre l'intrusion (le *confinement*). Menaces potentielles: intrusions, vers.
  - ✓ Capacité à tracer les modifications et les problèmes en revenant à la source (la *traçabilité*).

☞ Les mots de passe "Les clés de votre système"

- Avant de pouvoir utiliser un système UNIX, il faut se connecter à l'aide d'un identificateur (*username*) et d'un mot de passe (*password*). Ces deux éléments combinés avec le répertoire d'accueil représente le point d'entrée ou compte (*account*) d'une personne sur le système.
- Les fichiers */etc/passwd* et /etc/shadow sont utilisés par le système pour définir ses utilisateurs. Chaque ligne du fichier correspond à un identificateur et décrit un certain nombre de paramètres qui lui sont associés. La structure d'une ligne de ce fichier se présente sous la forme:

  ➔ Login:*:UID:GID:nom:répertoire_accueil:shell (passwd)
  ➔ Login:mot_de_passe_crypté:::::: (shadow)
  ➔ C'est le mot de passe qui est important, et non pas le login qui est facilement devinable.
  ➔ Il existe des méthodes pour percer les mots de passe.

☞ Comment choisir les bons mots de passe?

- Les bons mots de passe sont faciles à se rappeler, ne sont pas des noms et ne se trouvent pas dans aucun dictionnaire.
- Ce qui rend les mots de passe faciles à se rappeler est l'utilisation de termes mnémoniques qui associent un modèle de lettres et de nombres à des expressions qui vous sont familières. Par exemple, "Il était une fois trois petits cochons..." devient "ie1x3pc" qui peut être utilisé comme un bon mot de passe.

☞ Comment abandonner votre terminal?

- Ne jamais laisser votre terminal sans surveillance. Déconnectez-vous ou utilisez un programme de verrouillage approuvé pour protéger votre identificateur d'utilisateur.

- Un programme de verrouillage demande soit un mot de passe, soit il utilise le mot de passe que vous utilisiez quand vous êtes connecté.

- Le programme de verrouillage n'est pas standard sur la plupart des systèmes UNIX. Les stations de travail Sun fournisseur l'utilitaire *lockscreen* pour protéger l'affichage à l'écran. Sur d'autres systèmes UNIX, vous pouvez installer des programmes provenant du domaine public (par exemple, *screenblank*), mais seulement des programmes agréés, car un programme cheval de Troie de verrouillage peut vous voler votre mot de passe.

☞ Cheval de Troie *"en anglais trojan horse"*

- On appelle "Cheval de Troie", un programme informatique effectuant des opérations malicieuses à l'insu de l'utilisateur. Le nom "Cheval de Troie" provient d'une légende narrée dans l'*Iliade (de l'écrivain Homère)* à propos du siège de la ville de Troie par les Grecs.

  - La légende veut que les Grecs, n'arrivant pas à pénétrer dans les fortifications de la ville, ils eurent l'idée de donner en cadeau un énorme cheval de bois revêtu d'or en offrande à la ville en abandonnant le siège.

  - Un cheval de Troie (informatique) est donc un programme caché dans un autre qui exécute des commandes sournoises, et qui généralement donne un accès à la machine sur laquelle il est exécuté, par extension il est parfois nommé *troyen* par analogie avec les habitants de la ville de Troie.

- Un peu comme un virus, le cheval de Troie est un code (programme) nuisible placé dans un programme sain (imaginez une fausse commande de listage des fichiers, qui détruit les fichiers au-lieu d'en afficher la liste).

  - Un cheval de Troie peut par exemple
    - ✓ **voler des mots de passe**
    - ✓ **copier des données sensibles**
    - ✓ **exécuter tout autre action nuisible ...**

☞ Le contrôle d'accès

- La procédure de *login* fournit la première ligne de défense à votre système UNIX. Les procédures de contrôle d'accès procurent la deuxième en déterminant qui peuvent accéder à quels fichiers.

- Les sujets sont divisés en trois catégories: l'utilisateur (propriétaire de l'objet), le groupe et tous les autres

- Les types d'accès sont lire *r*, écrire *w* et exécuter *x*

- Ensemble, les trois catégories et les trois types d'accès procurent neuf droits d'accès distincts. La commande *ls -l* affiche ces droits, le propriétaire et le groupe.

- Le contrôle d'accès est *discrétionnaire* en ce sens que le propriétaire d'un objet (fichiers, répertoires) peut en faire n'importe quoi. Il peut ajuster les permissions d'accès de sorte que lui-même ne puisse y accéder. Dans System V, le propriétaire peut même donner l'objet à un autre utilisateur. Une fois l'objet cédé, le propriétaire originel n'a aucun contrôle sur les autorisations qui lui sont associés.

- Chaque utilisateur peut être un membre de groupes multiples définis dans le fichier système */etc/group*. L'appartenance à un groupe donne droit aux privilèges accordés au groupe.

- En particulier, dans les systèmes UNIX basés sur System V, la commande *newgrp* permet de changer de groupe.

☞ La sécurité du réseau

- Il est de forte chance que votre station de travail UNIX est connectée directement à un réseau local qui, à son tour, est relié à un autre réseau plus vaste, plus global comme le réseau Internet, par exemple. Dans ce contexte, votre station UNIX devient un des maillons du grand réseau. Et si jamais quelqu'un, quelque part trouve des failles de sécurité dans votre système et les exploite pour poser des actes illicites sur les systèmes des autres, vous serez responsable de ses agissements

☞ Les mécanismes du réseau

- Les informations qui circulent sur les réseaux informatiques sont transmises en morceaux appelés *paquets*.
  - ✓ Un paquet contient les données réelles à transmettre accompagnées d'un emballage. L'emballage comporte normalement les adresses d'origine et de destination du paquet, et des informations sur le type de paquet (par exemple, TCP, IP, etc.).

  - ✓ Pendant le voyage du paquet à travers des réseaux, les machines hôtes connectées aux réseaux doivent lire l'emballage de chaque paquet pour voir si le paquet leur est adressé.

  - ✓ Ceci soulève le premier problème de sécurité de réseau: toute machine hôte peut lire n'importe quel paquet qui défile sur le réseau. Une machine qui lit les données dans des paquets qui ne lui sont pas destinés est connue sous le nom d'*écouteur indiscret*.

- ✓ De plus, pour une question pratique, les données qui circulent sur le réseau ne sont pas chiffrées. C'est donc dire que, du moins en principe, l'information est potentiellement utilisable par quiconque fait de l'écoute de réseau.

- ✓ Il y a un autre problème lié cette fois-ci à l'identification de la source d'une communication.

- ✓ Le protocole TCP/IP d'UNIX utilise bien une adresse *Internet Paquet* (IP), un quadruplet d'octets dont la valeur varie entre 0 et 256, pour supposément pouvoir identifier d'une façon unique chaque machine hôte connectée au réseau.

- ✓ En réalité, toutefois, quiconque a le privilège de super utilisateur peut modifier à sa guise l'adresse IP de sa propre machine. Cette possibilité d'usurpation d'identité soulève des problèmes lorsqu'on voulait authentifier des requêtes de services provenant des autres machines.

- ✓ Certaines tentatives ont été effectuées pour améliorer l'authentification, mais elles ne sont pas encore universellement disponibles. Le système Kerberos du MIT en est un exemple. D'autres fabricants d'ordinateur essaient également de renforcer le mécanisme d'authentification de leurs systèmes en introduisant une marque de date chiffrée dans le paquet comme le *Secure RPC* de Sun Microsystems, par exemple.

☞ Communication "Les machines hôtes de confiance"

- ➢ Certaines fonctions UNIX, sont développées pour permettre aux utilisateurs distants, soit de se connecter (*rlogin*), soit d'exécuter des commandes (*rsh, rcp, rexec*) à travers le réseau. Normalement, ces utilisateurs doivent fournir un code d'accès *login* et un mot de passe valable sur la machine locale avant de pouvoir se connecter.

- ➢ Certains emploient la notion **de machines (hôtes) de confiance**, dans ce cas il faut bien faire très attention au choix des machines auxquelles vous faites confiance. Sachez aussi que la confiance est transitive en ce sens que si la machine A fait confiance à la machine B et que B fait de même pour la machine C, alors un utilisateur sur C peut effectuer un *rlogin* sans mot de passe vers B, puis un autre *rlogin*, toujours sans mot de passe, vers A et ce, même si A ne fait explicitement confiance à la machine hôte C.

- ➢ Les machines hôtes de confiance sont définies dans deux fichiers différents sur chaque machine hôte. Le fichier ***/etc/hosts.equiv*** du système fournit une définition des machines de confiance et les fichiers ***$HOME/.rhosts*** permettent aux utilisateurs de définir d'autres utilisateurs et machines de confiance. Le contrôle se fait en cascade: d'abord au niveau du fichier */etc/hosts.equiv*, ensuite par les fichiers *$HOME/.rhosts*.

- ➢ Le fichier */etc/hosts.equiv* permet aux utilisateurs qui travaillent sur une machine répertoriée d'utiliser *rlogin, rsh* ou *rcp* sans fournir de mot de passe. L'utilisateur distant doit posséder un identificateur dans le fichier */etc/passwd* de la machine hôte locale, sinon un nom de *login* et un mot de passe seront toujours exigés. Le fichier se compose d'une liste de noms, un par ligne. Chaque ligne se présente sous la forme:

[+|-] nom_machine [***nom_login***]

Si [***nom_login***] est absent, tous les utilisateurs sur **nom_machine** sont de confiance.
Autrement dit, autoriser ou interdire à des machines et à des utilisateurs l'utilisation des commandes r (telles que rlogin, rsh ou rcp) sans donner de mot de passe.

☞ En conclusion cette notion est très dangereuse, elle pose beaucoup de problèmes de sécurité, donc à éviter.

☞ La détection des intrusions

- ✎ Votre système UNIX n'est pas construit avec des alarmes qui peuvent vous signaler des intrusions.
- ✎ Votre meilleur système d'alarme est votre vigilance dans la surveillance des activités d'inhabituel qui sont en train de se produire.
- ✎ Vous devez savoir comment déterminer que quelque chose est de travers.
- ✎ Vous devez également essayer de découvrir qui est coupable et, si possible, boucher par vous-même le trou par lequel l'intrus est rentré (déconnecter la connexion, changer immédiatement votre mot de passe...).
- ✎ Rapportez vos soupçons à votre administrateur de système ou de sécurité et demandez-lui de faire le nécessaire.

☞ Outils pour surveiller votre système

- ✎ Les outils de surveillance sur les systèmes UNIX ne sont pas réservés aux initiés.
- ✎ Vous allez vite vous familier avec ces outils en vous en servant le plus souvent possible.
- ✎ Bien sûr, il y a des outils dont l'exécution demande des privilèges *root*; d'autres, par contre, peuvent être exécutés par des utilisateurs ordinaires.
- ✎ Ce sont ces derniers que nous allons décrire
  - ✓ La commande *who* est peut-être la première commande que tout nouvel utilisateur UNIX doit apprendre. Elle affiche les noms des utilisateurs, les noms des ports de connexion, et les dates de *login*. Cette information est obtenue à partir du fichier */usr/adm/utmp* (sur BSD) et elle est entretenue par les programmes *init* et *login* et la commande *date*. L'utilisation de l'option *-u* de la commande affiche plus d'informations (temps d'inactivité, PID)
  - ✓ La commande *ps* affiche l'état des processus. L'information est présentée en colonnes et comprend toujours les points suivants: le numéro d'identification du processus, le port utilisé par l'utilisateur qui a créé le processus, le temps CPU utilisé par ce processus, et le nom du fichier exécutable utilisé comme base par ce processus. Par défaut, seuls les processus appartenant à l'utilisateur qui lance la commande *ps* est affichés. Sous System V, l'option *-e* listera tous les processus. Sous BSD, les options *-ax* doivent être utilisées. L'ajout de l'option *-l* affichera plus d'informations sur chaque processus.

☞ Sécurisation des accès réseau

- Il s'agit de l'étape la plus importante de l'activité de l'expert en sécurité : on distingue quatre phases.
  1. auditer son réseau pour rechercher les problèmes de sécurité :

  a) les services privatifs qui ne doivent pas être accessibles de l'extérieur (exemple type : une imprimante en réseau). TCP/IP étant un protocole qui permet de faire abstraction de la différence entre le réseau local et le reste du monde, cette étape est importante et le vendeur du système d'exploitation ne peut pas la faire à la place de l'administrateur.

  b) Les protocoles non sûrs. Certains protocoles réseau véhiculent des mots de passe en clair (exemples typiques : telnet et ftp), d'autres utilisent l'adresse IP de l'appelant comme moyen d'authentification, ce qui est notoirement faible (rlogin, rsh, NIS et NFS). En fonction des besoins de sécurité du site, ces protocoles devront être réservés à certaines machines ou bien carrément supprimés.

  c) Les vulnérabilités des logiciels réseau, c'est-à-dire les bogues qui sont exploitables par un attaquant malintentionné. Il convient de différencier celles qui sont accessibles depuis l'extérieur du réseau de celles qui ne le sont pas. Dans de nombreux cas, en effet, le scénario de sécurité est du type «nous contre eux» et un certain niveau de confiance peut être accordé aux ordinateurs du réseau interne.

  2. Remédier aux défauts constatés en installant des correctifs pour les vulnérabilités (soit des mises à jour de logiciel, soit des modifications de configuration par défaut) et des procédures de filtrage adéquates pour les services privés.

  3. Mettre en place un système de surveillance qui allie récupération des alertes, détection précoce des intrusions et surveillance des machines et des réseaux ainsi sécurisées.

☞ Les virus

- ✒ Cela peut paraître incroyable, mais il n'y a pas de virus sous UNIX ! Cela tient bien sûr à sa philosophie de conception : la sécurité fait partie du noyau du système et aussi de la mentalité des programmeurs.

- ✒ Pas question, sous Linux par exemple, de programmer un logiciel de traitement du courrier électronique qui interpréterait des macros commandes dans les pièces jointes...

- ✒ Et pour ce qui est des virus plus classiques, qui s'attaquent au système d'exploitation et aux binaires des programmes, il n'est pas tout à fait inenvisageable d'en fabriquer ; cependant l'avis de l'auteur est que la compétence nécessaire pour réaliser cela est hors de portée d'un programme automatique (et de surcroît minuscule) comme un virus.

  La diversité dans le monde UNIX est suffisamment importante pour qu'un virus ne puisse pas trouver de niche écologique suffisamment uniforme entre différentes distributions, différentes versions d'un même programme, etc.

☞ Infections informatiques

- ➢ Une infection informatique est une modification ou un ajout non autorisé d'un objet ou d'un sujet sur un système informatique.

- ➢ Les virus, les chevaux de Troie, les vers, le passage secret, les bombes, les bactéries sont les infections informatiques.

- ➢ Un virus est une petite partie de code qui se recopie toute seule à l'intérieur d'un autre programme en le modifiant et qui ne peut s'exécuter que lorsque le programme hôte est lui-même en cours d'exécution. Son mode de reproduction lui permet d'infecter tous types de supports: disques, disquettes, mémoire, cassettes... et de se propager à travers un réseau de communication. Son objectif principal est d'altérer et de modifier le comportement des programmes qu'il touche et des systèmes dans lesquels il sévit.

- ➢ Un virus sous UNIX ne peut se propager qu'en exploitant les mauvais accès du système.. Il faudrait qu'il arrive à écrire dans un programme (permission d'écriture *w*), et ensuite que d'autres utilisateurs exécutent ce programme (permission d'exécution *x*) avec leurs propres privilèges. Pour ces raisons, il est très difficile de réaliser un virus sous UNIX.

- ➢ Un cheval de Troie, est conçu et fonctionne comme un virus, mais en se faisant passer pour un autre programme. Il porte le même nom qu'un programme utilitaire banal du système et si possible fréquemment utilisé (*su, login*, par exemple). Il donne à l'utilisateur l'impression de fonctionner normalement et réalise pendant son exécution toutes sortes d'opérations généralement dommageables pour le système.

- ➢ Un ver (*worm*) est un programme illicite autonome qui se reproduit tout seul avec une très grande rapidité. L'objectif du ver est d'immobiliser à son profit le maximum de ressource de la machine, qui demeure dès lors difficilement utilisable.

- Le passage secret est un moyen pour le concepteur ou le développeur d'un système de se réserver une voie accès pour s'introduire dans le système une fois ce dernier opérationnel. Ce n'est pas forcément néfaste, car il peut servir à dépanner, surveiller ou corriger le comportement du programme qui le contient. Les options *debug, wiz* dans les anciennes versions de *sendmail* en sont des exemples.

- Il existe d'autres variétés de nuisibles plus ou moins apparentées aux infections informatiques présentées ici. Il n'y a pas de remèdes miracles, il existe toutefois quelques précautions et règles de base à respecter pour diminuer les risques de contamination:

  - ✓ Ne pas installer de programmes à provenance douteuse;
  - ✓ N'installer que les programmes utiles (il ne s'agit pas de ramasser des collections);
  - ✓ En cas de doute sur l'intégrité du disque dur, redémarrer (*boot*) le système à partir d'un médium externe de provenance sûre et protégé en écriture;

☞ Les sauvegardes

- Il est très facile de faire des sauvegardes sous UNIX, nous ne nous étendrons pas sur ce point. Qu'il suffise de rappeler quelques utilitaires : *find* permet de rechercher sur tout le disque dur des fichiers possédant certaines caractéristiques (par exemple, par nom, ou par date de modification) ; et cpio ou *tar* permet d'en faire une archive. Il existe aussi un certain nombre d'outils pour effectuer des sauvegardes incrémentales comme backupPC (logiciel libre, UNIX, Linux, Mac OS Windows), BackInTime et TimeShift logiciels libres aussi, équivalent à Time Machine sur Mac OS.