Évolution de l'authentification

1. Les premiers systèmes Unix et le fichier passwd

a. Mots de passe dans le fichier /etc/passwd

Depuis le début de leur existence, les systèmes Unix utilisent le fichier **/etc/passwd** comme base de comptes des utilisateurs. Ce fichier est utilisé naturellement pour les ouvertures de session sur le système. Comme son nom l'indique encore, il contenait en plus des identifiants utilisateurs leurs mots de passe cryptés. Si des éléments logiciels autres que l'ouverture de session ont besoin des informations de compte (connexion ftp, ouverture de session distante, etc.), ils vont également consulter ce fichier. Dans cette situation originelle simple, on a affaire à une base de compte unique et des applications multiples qui exploitent cette base de compte. Toutes les applications doivent reconnaître le format de cette base d'information.

b. Mots de passe dans le fichier /etc/shadow

Avec l'évolution des techniques d'attaques des mots de passe, le besoin est venu de placer les mots de passe dans un fichier non accessible aux utilisateurs ordinaires. Ils sont alors stockés dans un fichier **/etc/shadow** fermé aux utilisateurs. Les paramètres d'authentification avec shadow sont gérés par un fichier **/etc/login.defs**. Les paramètres présents par défaut dans ce fichier sont en général satisfaisants.

Gestion des erreurs d'authentification dans le fichier login.defs

Parmi les nombreux paramètres du fichier login.defs, ceux concernant le login sont les plus fréquemment modifiés.

```
toto@ubuntu:~$ grep LOGIN /etc/login.defs
LOGIN_RETRIES      5
LOGIN_TIMEOUT      60
toto@ubuntu:~$
```

2. D'autres bases d'informations

Pour la consultation des éléments d'identification, la situation s'est compliquée quand il a fallu intégrer d'autres bases de comptes, différentes du fichier **passwd** et surtout plus complexes. Ces bases d'identités sont souvent centralisées, comme c'est le cas pour NIS (*Network Information Server*) ou LDAP (*Leightweight Directory Access Protocol*). La première solution envisagée fut naturellement de réécrire les programmes qui exploitaient initialement le fichier **/etc/passwd** afin qu'ils soient capables de consulter les bases centralisées sur le réseau. Cette méthode manquait cruellement de souplesse, puisqu'elle obligeait à reprendre beaucoup de programmes en profondeur à chaque fois qu'une modification était apportée au mode de stockage des bases centralisées.

3. NSS

NSS (*Name Service Switch*) est une première réponse à la multiplicité des bases d'information locales ou centralisées. NSS a pour objet de normaliser la résolution de nom au sein d'un système. NSS permet de résoudre un nom en une autre information associée, comme par exemple un nom d'utilisateur et son uid, un nom de groupe et son gid, ou encore un nom d'hôte et son adresse IP.

Dans un fonctionnement NSS, un fichier **/etc/nsswitch.conf** détermine pour différents types de résolutions la source d'information à privilégier, et les applications ayant besoin de ces informations vont consulter les sources dans l'ordre imposé par le fichier **nsswitch.conf**. La résolution s'appuie alors sur des bibliothèques NSS (**libnss_*X*.so** où *X* représente le service de résolution employé), et les applications n'ont pas besoin de connaître directement la méthode de résolution employée.

Format du fichier nsswitch.conf

```
résolution: source_1 source_n
```

nsswitch.conf : format du fichier

résolution	Le type de résolution à effectuer.
source_1	Obligatoire. La première source de résolution à employer.
source_n	Facultatif. La ou les autres sources de résolution possibles à utiliser après la première.

Exemple de fichier nsswitch.conf

On voit dans cet exemple que les résolutions de type passwd, group et shadow feront leur résolution grâce à la bibliothèque libnss_compat.so, alors que la résolution de noms d'hôtes se fera par les bibliothèques libnss_files.so et libnss_dns.so. Ce qui veut dire que les éléments d'identification des utilisateurs seront trouvés dans les fichiers locaux de /etc, alors que la résolution de noms d'hôtes s'appuiera d'abord sur le fichier local (/etc/hosts) avant de se reporter sur un service dns.

```
passwd:         compat
group:          compat
shadow:         compat

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Sur un système Linux moderne, NSS n'est plus utilisé que pour des opérations d'identification, c'est-à-dire trouver des informations sur une identité. Tout ce qui relève de l'authentification est dévolu à un mécanisme plus élaboré : PAM.

4. Modules d'authentification

Si NSS représente déjà un progrès par rapport aux fichiers statiques utilisés dans les premiers temps, la révolution viendra avec PAM (*Pluggable Authentication Module*). PAM est un mécanisme complémentaire de NSS qui assure une authentification sur mesure par l'exécution de modules au choix de l'administrateur.

Lors d'une ouverture de session Linux, l'utilisateur va présenter un identifiant et un mot de passe. Grâce à la résolution NSS, on en déduira les identifiants uid/gid, ainsi que les autres paramètres nécessaires (date d'expiration, etc.). PAM de son côté va en fonction de sa configuration exécuter des modules pour assurer l'authentification mais aussi éventuellement pour effectuer certaines tâches liées à l'ouverture de session, comme la définition de variables par exemple.