

La sécurité des comptes

Objectifs de cette section

Comprendre le fonctionnement de PAM

Les différents types de comptes sur un système Linux

Les règles de bonnes pratiques concernant la sécurité des mots de passe

Qu'est-ce que sont les « shadow passwords » et pourquoi sont-ils importants.

Objectifs de cette section

Configurer des comptes et des mots de passe de manière à ce qu'ils expirent au bout d'un certain temps.

Bloquer ou débloquer des comptes.

Apprendre à utiliser les fichiers de logs concernant les connexion aux comptes.

Utiliser des méthodes d'authentification par facteurs multiples.

PAM

PLUGGABLE AUTHENTICATION MODULES

Sécurité des comptes sur Linux

Il est plus simple d'attaquer un système depuis l'intérieur que depuis l'extérieur.

Une des attaques les plus utilisées consiste à effectuer une élévation de privilèges. C'est-à-dire accéder à des zones du système auxquelles l'utilisateur ne devrait pas avoir accès.

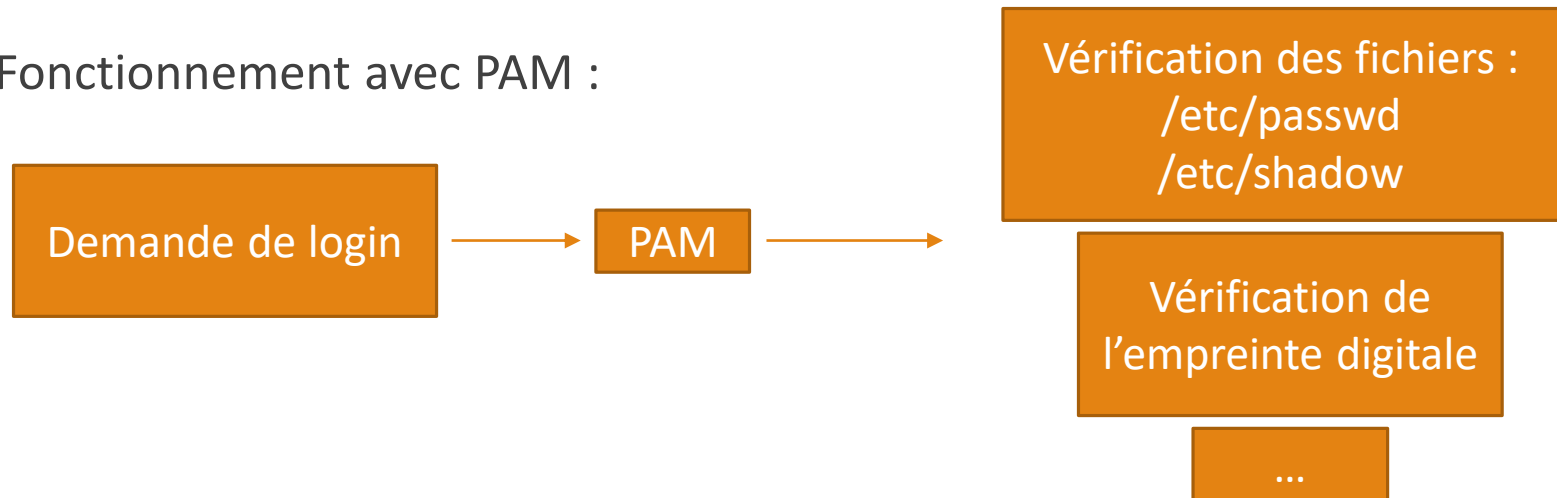
Il faut donc sécuriser les comptes sur une machine Linux de manière à ce qu'ils n'aient accès qu'aux éléments dont ils ont besoin pour travailler.

Fonctionnement de PAM

Fonctionnement standard :



Fonctionnement avec PAM :



Fichiers de configuration de PAM

Les fichiers de configuration de PAM sont localisés dans les éléments suivants :

- /etc/pam.d
- /etc/pam.d/login
- /etc/pam.d/sshd

Format des éléments de configuration de PAM

- `module_interface control_flag module_name module_arguments`

Les modules d'interface de PAM

Module Auth :

- Authentifie les utilisateurs
- Vérifie par exemple la validité du mot de passe utilisé pour l'authentification

Module Account :

- Vérifie si l'accès au système est autorisé
- Vérifie par exemple si le mot de passe n'est pas expiré, ou si le compte n'est pas bloqué
- Peut n'autoriser l'accès qu'à certaines heures de la journée

Module password :

- Change le mot de passe utilisateur

Module session :

- Manage les sessions utilisateurs
- Peut spécifier le répertoire home des utilisateurs par exemple

Les PAM Control Flags

Tous les modules PAM peuvent générer un statut réussi ou échec lorsqu'ils sont utilisés.

Les Control Flags indiquent au système ce qu'il doit faire en fonction du résultat.

Un ordre peut être défini entre les différents modules (impliquant par exemple que les modules précédents doivent renvoyer un statut réussi pour continuer).

Les PAM Control Flags (2)

Required : Le module doit renvoyer un résultat réussi pour continuer. Cependant le système continue quand même à tester les autres modules.

Requisite : Le module doit renvoyer un résultat réussi pour continuer mais cette fois, les autres modules ne sont pas testés si un échec est détecté.

Sufficient : Authentifie l'utilisateur si aucun des modules « required » n'ont échoué.

Optional : Utilisé lorsque le résultat du module doit être ignoré.

Include : Inclue les lignes de configuration provenant d'un autre fichier.

Comprendre PAM

CAS CONCRET

Utilisation de PAM sur le SSH

Nous allons utiliser le module **pam-dbus** sur notre démon SSHd.

Pour cela, il nous faut une machine Debian9 avec interface graphique sur laquelle nous allons effectuer les différentes configurations.



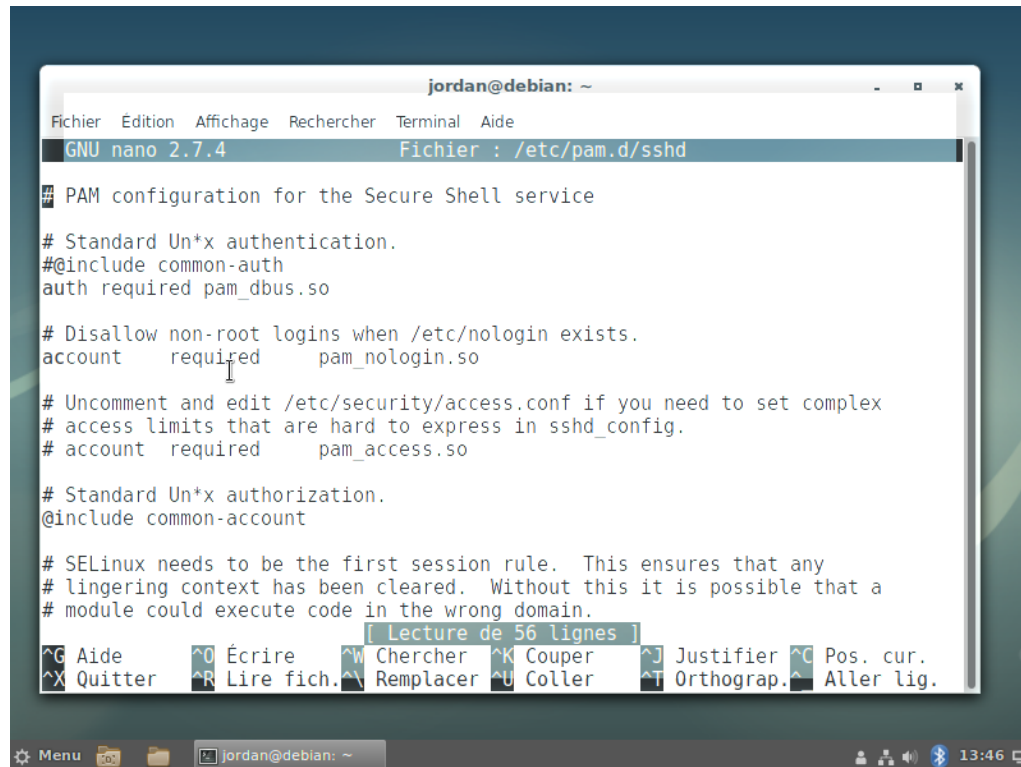
Paquets à installer

Nous avons besoin d'installer le paquet **libpam-dbus** sur notre machine debian9 grâce à la commande :

- **apt-get install libpam-dbus**

Modification de la configuration

Nous allons éditer le fichier `/etc/pam.d/sshd` pour remplacer la ligne :
« **@include common-auth** » par « **auth required pam_dbus.so** »



```
jordan@debian: ~
Fichier  Edition  Affichage  Rechercher  Terminal  Aide
GNU nano 2.7.4      Fichier : /etc/pam.d/sshd

# PAM configuration for the Secure Shell service

# Standard Un*x authentication.
#@include common-auth
auth required pam_dbus.so

# Disallow non-root logins when /etc/nologin exists.
account    required    pam_nologin.so

# Uncomment and edit /etc/security/access.conf if you need to set complex
# access limits that are hard to express in sshd_config.
# account  required    pam_access.so

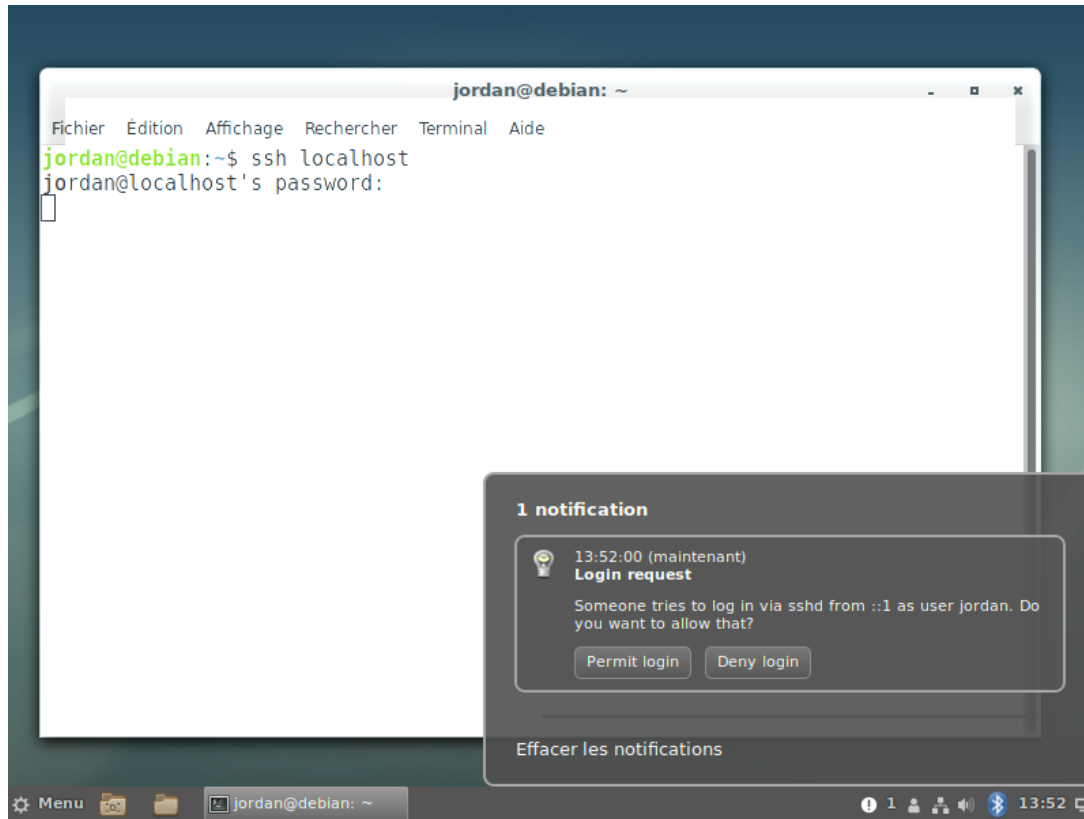
# Standard Un*x authorization.
@include common-account

# SELinux needs to be the first session rule. This ensures that any
# lingering context has been cleared. Without this it is possible that a
# module could execute code in the wrong domain.
[ Lecture de 56 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^J Justifier ^C Pos. cur.
^X Quitter   ^R Lire fich.^M Remplacer  ^U Coller    ^T Orthograp.^_ Aller lig.
```

Qu'est-ce que ça veut dire ?

L'option **auth required pam_dbus.so** signifie :

- Auth : utilisation d'un module de type « authentication »
- Required : Le module doit renvoyer un résultat réussi pour continuer.
Cependant le système continue quand même à tester les autres modules.
- Pam_dbus : Le nom du module qu'on utilise.



Conséquences

Lors d'une tentative de connexion par SSH, tous les utilisateurs locaux déjà authentifiés reçoivent un message leur demandant si le système doit accepter la demande de connexion.

Les types de comptes sur Linux

Compte root : superuser

Il s'agit du compte le plus fort sur un système Linux.

L'utilisateur root peut faire tout ce qu'il veut !

- Voir tous les fichiers du système
- Installer ou désinstaller des logiciels
- Changer la configuration réseau
- Créer, modifier et supprimer des utilisateurs

Root possède un **User ID (UID) de 0** :

- Les noms de comptes comme « root » ou « jordan » ne sont utiles que pour les humains, car le système Linux n'utilise que les UID !
- Si on met un UID de 0 à un autre compte que root, alors il possèdera également tous les privilèges possibles.

Les comptes systèmes

Il s'agit des comptes dont l'UID est différent de 0 et inférieur à 1000.

- Plutôt utilisés à des fins administratifs ou organisationnels.
- Utilisés par des services ou des applications

UID configurable dans le fichier **/etc/login.defs**

Il est possible d'indiquer qu'il s'agit d'un compte système lors de sa création en utilisant l'option « -r » ou « --system » du `useradd`.

- `useradd -r account_name`
- `useradd --system account_name`

Les comptes utilisateurs

Il s'agit de comptes dont l'UID est supérieur ou égal à 1000.

Utilisés par des humains et pour des humains.

La sécurité des mots de passe

Renforcer les politiques de sécurité

Obliger les utilisateurs à utiliser des mots de passe renforcés.

- Sinon vous risquez d'avoir des comptes avec des mots de passe du type : « azerty » ou « 12345 »

Laisser les mots de passe faibles autorisés augmente les chances qu'un attaquant puisse pénétrer votre système avec des attaques Brute Force.

Grâce au module **pam_pwquality**, lui-même basé sur le pam_cracklib (module ancien, utilisé sur les vieux systèmes).

- Fichier de configuration : `/etc/security/pwquality.conf`
- Utilisation dans PAM : `password requisite pam_pwquality.so`
- Les options possibles : `man pam_pwquality`

Utiliser le Shadow

Historiquement, les mots de passe étaient stockés sous la forme d'un « hash » ou d'un format chiffré dans le fichier : **/etc/passwd**

- Exemple : `root:$6$L3SSm1M1H5:0:0:root:/root:/bin/bash`
- Fichier visible par tous les utilisateurs par défaut
- Tous les mots de passe chiffrés étaient stockés dans ce fichier (y compris celui de root)

Lorsque les **shadow passwords** sont utilisés, les hash sont remplacés par un x dans le fichier `/etc/passwd`

- Exemple `/etc/passwd` : `root:x:0:0:root:/root:/bin/bash`
- Exemple `/etc/shadow` : `root:$6$L3ZSm1M1H5::0:99999:7:::`
- Le fichier **/etc/shadow** ne peut être lu que par l'utilisateur root.

Convertir les mots de passe

Si vous rencontrez un système qui n'utilise pas les mots de passe shadow, vous pouvez convertir les `/etc/passwd` en utilisant la commande :

- `pwconv`

Si vous voulez arrêter l'utilisation des mots de passe shadow (chose à déconseiller), vous pouvez utiliser la commande :

- `pwunconv`

Les champs dans le fichier /etc/shadow

Chaque ligne est composée d'un ensemble de champs séparés par « : » dans l'ordre suivant :

- Nom d'utilisateur
- Hash du mot de passe
- Nombre de jours depuis le dernier changement du mot de passe
- Nombre de jours restants pour un changement de mot de passe autorisé
- Nombre de jours restants avant un changement obligatoire du mot de passe
- Nombre de jours restants avant expiration du mot de passe
- Nombre de jours restants avant la désactivation du compte
- Nombre de jours depuis l'expiration du compte
- Réservé

Afficher les données d'expiration d'un compte

On peut utiliser la commande :

- `chage -l account`

```
jordan@debian:~$ chage -l jordan
Last password change                : May 30, 2018
Password expires                    : never
Password inactive                   : never
Account expires                     : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999
Number of days of warning before password expires : 7
jordan@debian:~$
```

Changer les options sur le mot de passe avec le /etc/login.defs

On peut par exemple y ajouter des options s'appliquant sur tous les mots de passe :

- `PASS_MAX_DAYS` 99999
- `PASS_MIN_DAYS` 0
- `PASS_MIN_LEN` 5
- `PASS_WARN_AGE` 7

Empêcher les utilisateurs d'utiliser les mêmes mots de passe

On peut empêcher les utilisateurs d'alterner entre les mêmes mots de passe lorsqu'ils arrivent à expiration grâce au module **pam_pwhistory** :

- `password required pam_pwhistory.so`
- Retient les 10 derniers mots de passe par défaut

Contrôler l'accès aux comptes

Verrouiller et déverrouiller un compte

Si vous avez besoin de verrouiller un compte vous pouvez utiliser la commande :

- `passwd -l nom_du_compte`

Si vous avez besoin de déverrouiller un compte vous pouvez utiliser la commande :

- `passwd -u nom_du_compte`

Verrouiller un compte avec le « nologin »

Il est possible de verrouiller un compte en modifier le shell à utiliser dans le fichier `/etc/passwd` :

- `www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin`
- `sshd:x:117:65534:./run/sshd:/usr/sbin/nologin`

Plutôt que de modifier manuellement le fichier `/etc/passwd`, vous pouvez utiliser la commande `chsh` :

- `chsh -s shell_a_utiliser nom_du_compte`
- `chsh -s /usr/sbin/nologin jordan`

Désactiver la possibilité de se loguer sur votre système

Imaginons que vous souhaitez empêcher les utilisateurs d'accéder à votre système linux pour une période de temps donnée.

Vous pouvez utiliser la module pam : **pam_nologin**

- Permet d'empêcher les utilisateurs de se loguer sur le système si le fichier `/etc/nologin` existe.
- Le contenu de ce fichier est alors affiché aux utilisateurs qui tentent de se loguer, et un refus leur est signifié.

Les logs des authentifications

Vous pouvez accéder aux logs des dernières authentifications en utilisant la commande **last** (ou fichier **/var/log/auth.log** sur debian) :

```
root@debian:/home/jordan# last
```

```
jordan pts/1          192.168.1.93      Mon Jun 18 13:58   still logged in
jordan tty7            :0                Mon Jun 18 12:58   still logged in
jordan tty7            :0                Mon Jun 18 12:49 - 12:58   (00:08)
jordan tty7            :0                Mon Jun 18 12:43 - 12:49   (00:05)
reboot system boot    4.9.0-6-amd64     Mon Jun 18 12:40   still running
jordan tty7            :0                Wed May 30 16:00 - 16:01   (00:01)
reboot system boot    4.9.0-6-amd64     Wed May 30 15:59 - 16:01   (00:01)
```

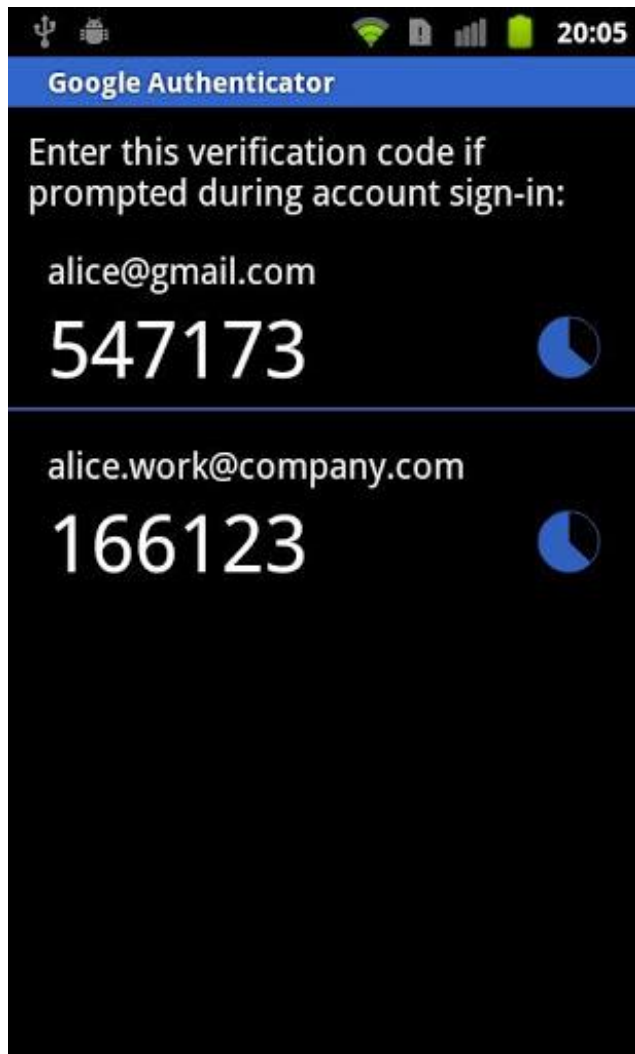
```
wtmp begins Wed May 30 15:59:55 2018
```

Intrusion Preventing System (IPS) – Fail2ban

On peut utiliser la fonctionnalité fail2ban pour bloquer spécifiquement une adresse IP qui tenterait de se loguer sur votre système.

Idéal contre des attaques de type brute force.

Typiquement, fail2ban procède à un bannissement en ajoutant une règle au pare-feu **iptables** pour bannir l'adresse IP de la source.



Authentification par multifacteurs

Il est possible d'utiliser le module PAM Google Authenticator qui envoie un code sur votre smartphone lorsque vous tentez de vous connecter au système :

<https://github.com/google/google-authenticator-libpam>

Sécurité par type de compte

Sécurité des comptes - Root

Il vaut mieux utiliser un compte « normal » pour effectuer des activités classiques.

Désactiver la possibilité de s'authentifier directement en root sur le système. Sinon cela signifie qu'un attaquant a l'opportunité d'obtenir un accès direct au compte root !

Utiliser le sudo pour effectuer des actions root, plutôt que le su.

- Permet de savoir qui effectue quelle commande

Eviter d'utiliser le même password root sur tous vos systèmes.

Vérifier que seul le compte root possède un UID de 0 dans le /etc/passwd.

Désactiver l'authentification directe en root pour le SSH

Cela se fait dans le fichier de configuration `/etc/ssh/sshd_config` où la ligne suivante doit être présente :

- `PermitRootLogin no`

Il faut ensuite redémarrer le service SSH pour que la modification soit prise en compte :

- `service ssh reload`

Comptes système / application

Par défaut, il existe de nombreux **comptes systèmes** sur votre machine Linux. Ils sont appelés **comptes applicatifs** par certains et **comptes de services** par d'autres.

Ces comptes sont associés à de nombreux services, et une règle de bonne pratique consiste à n'associer un service qu'à un seul compte.

Ces comptes sont verrouillés par défaut (on ne peut pas s'authentifier avec) et ce pour une bonne raison.

Comptes utilisateurs

Il est souhaitable de ne configurer qu'un compte par personne.

Supprimer les comptes qui ne sont plus utilisés :

- Déterminer l'UID du compte : `id nom_du_compte`
- Supprimer le compte et le répertoire home : `userdel -r`
- Rechercher les fichiers qui lui appartiennent :
 - `find / -user UID`
 - `find / -nouser`
 - En effet, si un nouvel utilisateur est créé avec le même UID, il aura accès aux fichiers qui appartenaient à l'ancien utilisateur.