Sécurité passwd - libpam-pwquality

- Objet du tuto : Utiliser et configurer les outils libpam-pwquality pour améliorer la qualité de ses mots de passe pour les sessions root et utilisateurs
- Niveau requis :
  avisé
- Commentaires : À partir de buster Debian 10
- Débutant, à savoir : Utiliser GNU/Linux en ligne de commande, tout commence là !.
- * Suivi :
  à-tester
  - Création par gilles 04/01/2021
  - Testé par <...> le <...>
  - Commentaires sur le forum : C'est ici[1)]

Remerciement à l'équipe Debian-France

Remerciement à Frédéric Lenquette de Debian-France https://france.debian.net/ et à l'équipe qui a organisé la Mini-DebConf Marseille 2019 sans lequel ce tuto n'aurait pas été envisagé.

Introduction

Les outils libpam-pwquality permettent de paramétrer le refus des mots de passe trop faibles pour les sessions root et utilisateurs. Ils permettent aussi de les évaluer en fonction de leur caractère aléatoire apparent et de les comparer à un dictionnaire des mots de passe trop courants. Ils reprennent ainsi les fonctions du paquet pam_cracklib en restant compatibles tout en ajoutant des nouvelles fonctions.

Sources :

- Mini-DebConf Marseille 2019 Hardening Debian 10 (Buster) par Frédéric Lenquette de Debian-France : https://youtu.be/N-pvLMHtRSA?t=705
- http://deer-run.com/~hal/linux_passwords_pam.html
- man pwquality.conf (Version debian 10 du 2017-05-26 Red Hat, Inc.)
- man pam_pwquality (Version debian 10 du 2017-05-26 Red Hat, Inc.)

Installation

```
apt-get update && apt-get install libpam-pwquality libpwquality-tools
```

Un message par une fenêtre ncurses indiquera peut-être que pam_cracklib était déjà installé et qu'il sera désactivé. En effet, un seul gestionnaire de mot de passe peut être activé.

Si vous passez d'un gestionnaire de mot de passe à un autre par une installation ou une désinstallation ou si vous voulez savoir lequel est en vigueur, utilisez la

commande :

```
pam-auth-update
```

Visualisation des options par défaut et fichier à modifier

Les options par défaut de libpam-pwquality peuvent être visualisées par :

```
cd /etc/pam.d/
```

suivi de :

```
grep 'pwquality' *
```

Dans Debian 10 buster et Debian 11 bulleyes, vous aurez cette réponse :

```
common-password:password    requisite            pam_pwquality.so retry=3
```

Le fichier à modifier est où se trouve mentionné la bibliothèque partagée *pam_pwquality.so*, c'est à dire :
`etc/pam.d/common-password`
et à la ligne où est présent le terme `pam_pwquality.so`

L'option par défaut présente : *retry=3*, c'est à dire que qu'au maximum 3 tentatives ratées de changement de mot de passe seront autorisées avant que la commande `password` ne s'arrête.

Configuration

Comme indiqué on modifie le fichier `/etc/pam.d/common-password` et à la ligne où est présent le terme `pam_pwquality.so` en indiquant les options que vous choisissez dans le tableau des options suivant :

```
nano /etc/pam.d/common-password
```

Tableau des options

La description du comportement des options dcredit, ucredit, lcredit, ocredit est sujette à caution, les descriptions de ces options rédigées dans man pwquality.conf sont différentes de celles rédigées dans man pam_pwquality. Ne sachant pas leur comportement réel, nous ne les utiliserons pas en ne les mentionnant pas dans le fichier de configuration. À la place, nous fixerons un nombre minimum de types de caractères différents présent dans le mot de passe avec l'option minclass.

D'autre part, un mot de passe long avec peu de types de caractères différents est plus facile à mémoriser et donc à ne pas noter à côté du clavier qu'un mot de passe court mais hyper-complexe et compliqué.

Par exemple un mot de passe du type *mac-enroe-maradona-2-champions* est préférable à ceux du type *_[r[3!_G{j0,K*

mots-clés	action/description
difok=N	Nombre de caractères du nouveau mot de passe qui ne sont pas présents dans l'ancien, par défaut difok=1
minlen=N	Taille minimum du nouveau mot de passe. Cependant un bonus d'un caractère en plus est rajouté si un type de caractères différent de plus est présent dans le mot de passe.
dcredit=N	Si dcredit < 0, dcredit est l'opposé du nombre minimum de chiffres dans le nouveau mot de passe, exemple si dcredit = -5, il faut au moins 5 chiffres dans le mot de passe.
ucredit=N	Si ucredit < 0, ucredit est l'opposé du nombre minimum de lettres majuscules dans le nouveau mot de passe, exemple si ucredit = -4, il faut au moins 4 lettres majuscules dans le mot de passe
lcredit=N	Si lcredit < 0, lcredit est l'opposé du nombre minimum de lettres minuscules dans le nouveau mot de passe, exemple si lcredit = -10, il faut au moins 10 lettres minuscules dans le mot de passe. Si lcredit > 0, alors le nombre de caractères minimum utilisé par le mot de passe diminue de minlen à (minlen - lcredit). (par défaut 0)
ocredit=N	Si ocredit < 0, ocredit est l'opposé du nombre minimum de caractères spéciaux dans le nouveau mot de passe, exemple si ocredit = -4, il faut au moins 4 caractères spéciaux dans le mot de passe. Si ocredit > 0, alors le nombre de caractères minimum utilisé par le mot de passe diminue de minlen à (minlen - ocredit). (par défaut 0)
minclass=N	Le nombre minimum de types de caractères requis pour le nouveau mot de passe (chiffres, majuscules, minuscules, autres). (par défaut 0)
maxrepeat=N	Si maxrepeat=N, alors un caractère ne pourra pas être présent plus de N fois
maxclassrepeat=N	Rejete les mots de passe contenant plus de N caractères consécutifs du même type. La valeur par défaut est 0, ce qui signifie que la vérification est désactivée
gecoscheck=N	Si différent de zéro, vérifie si les mots de plus de 3 caractères des champs GECOS du mot de passe de l'utilisateur sont contenues dans le nouveau mot de passe. Cette vérification n'est pas effectuée si gecoscheck=0, qui est aussi la valeur par défaut. On peut avoir une idée rapide de ce que sont les champs GECOS en tapant : `chfn --help`
dictcheck=N	Si dictchech est différent de 0, vérifie si le mot de passe est présent dans le dictionnaire cracklib des mots de passe trop courants. Si l'on veut créer son propre dictionnaire des mots à bannir comme mot de passe, regarder et interpréter les résultats de la commande : `find / -iname '*cracklib*' -print | grep dict`
usercheck=N	Si usercheck est différent de 0, vérifie si le mot de passe contient le nom de l'utilisateur $USER. Cette vérification n'est pas effectuée pour les noms d'utilisateur de moins de 3 caractères.
usersubstr	à faire

enforcing=N	Si N=0, il ne sera pas tenu compte du reste des vérifications effectuées par les autres options, le mot de passe sera accepté quelque soit sa qualité, seulement un message d'avertissement sera émis. Si N différent de 0, un mot de passe qui échoue à remplir les conditions posées par les autres options est rejeté. C'est le comportement par défaut.
dictpath	Définit le choix du chemin dans l'arborescence (*PATH*) du dictionnaire des mots de passe bannis, par défaut, il s'agit du dictionnaire des mots de passe bannis fourni par l'outil gestionnaire de mot de passe cracklib. Pour savoir où est ce dictionnaire, interprétez les résultats de : `find / -iname '*cracklib*' -print | grep dict`
retry=N	Nombre maximum de tentatives ratées de connexion au compte
enforce_for_root	Cette option renverra une erreur en cas d'échec de la vérification même si l'utilisateur qui modifie le mot de passe est root. Cette option est désactivée par défaut, ce qui signifie que dans ce cas seul le message concernant l'échec de la vérification est affiché mais que root peut quand même changer le mot de passe. Pour la sécurité, il vaut mieux donc activer cette option. Notez qu'à root on ne demande pas un ancien mot de passe, donc les vérifications qui comparent l'ancien et le nouveau mot de passe ne seront pas effectuées.
local_users_only	à faire

Utilisation et exemples expliqués

Les explications sont aussi sous forme de commentaires dans le code, ce qui permettra ultérieurement de les comprendre si vous le relisez plus tard.

Configuration trop succincte, les règles ne s'appliquent pas à root

/etc/pam.d/common-password

```
# here are the per-package modules (the "Primary" block)
# Trois tentatives, 12 caractères minimum
password        requisite                       pam_pwquality.so
retry=3 minlen=12
```

Si vous ne respectez pas la consigne :

```
passwd utilisateur0
Nouveau mot de passe :
MOT DE PASSE INCORRECT : Le mot de passe comporte moins de 12 caractères
```

Configuration basique recommandée, les règles s'appliquent aussi à root

/etc/pam.d/common-password

```
# here are the per-package modules (the "Primary" block)
# Trois tentatives, 12 caractères minimum, 2 types de caractères différents, les règles s'appliquent aussi à root
password        requisite                       pam_pwquality.so retry=3 minlen=12 minclass=2 enforce_for_root
```

Configuration renforcée avec vérification avec un dictionnaire des mots de passe courants, les règles s'appliquent aussi à root

/etc/pam.d/common-password

```
# here are the per-package modules (the "Primary" block)
# Trois caractères différents entre l'ancien et le nouveau mot de passe, trois tentatives, 12 caractères minimum, 2 types de caractères différents, vérification avec un dictionnaire de mots de passe courants,les règles s'appliquent aussi à root
password        requisite                       pam_pwquality.so difok=3 retry=3 minlen=12 minclass=2 dictcheck=1 enforce_for_root
```

Tester et générer ses mots de passe

Tester

- Vous pouvez déjà évaluer la complexité des mots de passe que vous projetez d'utiliser avec ce site :

http://inforisque.fr/fiches-pratiques/tester-mot-de-passe.php

Générer

- Si vous séchez complètement pour créer un mot de passe la commande `pwmake` vous permettra d'en créer avec une cible d'entropie déterminée.

Exemple : Vous voulez un mot de passe conforme aux règles que vous avez définies dans `/etc/pam.d/common-password` et d'une entropie de 96 bits :

```
pwmake 96
```

Pour en savoir plus

Le site du projet : https://github.com/libpwquality/libpwquality

Plus d'informations sur les bibliothèques libpam ici :

- https://debian-facile.org/doc:systeme:pam?s[]=pam
- https://www.formatux.fr/formatux-securite/module-020-pam/index.html

Comparaisons entre pam pwquality et cracklib :

- https://searx.gnous.eu/search?q=pam%20pwquality%20versus%20cracklib&categories=general&language=en
- https://duckduckgo.com/?t=ffab&q=pam+pwquality+versus+cracklib&ia=web

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !