OpenSSL - pkcs8

Outil de conversion de clé privée au format pkcs#8

Cette commande traite les clé privées au format pkcs#8. Elle peut gérer le format PrivateKeyInfo et EncryptedPrivateKeyInfo avec divers algorithmes PKCS#5 (1.5 et 2.0) et PKCS#12.

OPTIONS

-topk8 Normallement une clé privée PKCS#8 est attendue en entrée et un format de clé privée traditionnel est écris. Cette option inverse la situation

-inform DER|PEM Spécifie le format d'entrée

-outform DER|PEM Spécifie le format de sortie

-passing arg Source du mot de passe du fichier d'entrée

-out filename Fichier de sortie

-passout arg source du mot de passe du fichier de sortie

-iter count En créant un nouveau conteneur PKCS#8, utilise le nombre d'itérations donné sur le mot de passe pour dériver la clé de chiffrement.

-nocrypt Les clé PKCS#8 générées sont normallement des structures EncryptedPrivateKeyInfo. Cette option créé des structure PrivateKeyInfo.

-2 alg Active l'utilisation des algorithmes PKCS#5 v2.0. Normalement les clés privées PKCS#8 sont chiffrées avec un mot de passe basé sur l'algorithme de chiffrement pbeWithMD5AndDES-CBC qui utilise DES 56-bits. PKCS#5 v2.0 permet d'utiliser des algorithmes tel que 3DES 168bits, ou RC2 128bits.

-v2prf alg Définis l'algorithme PRF à utiliser avec PKCS#5 v2.0. Une valeur typique est hmacWithSHA256

-v1 alg Définis l'algorithme PKCS#5 v1.5 ou PKCS#12 à utiliser.

-nooct Génère des clé privées RSA dans un format cassé que certains logiciels utilisent

-embed Génère des clé DSA dans un format cassé.

-nsdb Génère des clé DSA dans un format cassé compatible avec les base de clé privée Netscape.

-engine id pkcs8 va tenter d'obtenir une référence fonctionnelle du moteur spécifié.

-scrypt Utilise l'algorithme scrypt pour le chiffrement de la clé privée en utilisant les paramètres par défaut. Actuellement N=16384, r=8 et p=1 et AES en mode CBC avec une clé 256 bits. Ces paramètres peuvent être modifiés avec -scrypt_N, -scrypt_r, -scrypt_p et -v2

Notes

Le format chiffré des fichiers PKCS#8 encodés PEM utilisent les en-tête et fin suivant :

```
---BEGIN ENCRYPTED PRIVATE KEY---
---END ENCRYPTED PRIVATE KEY---
```

La version non-chiffrée :

```
---BEGIN PRIVATE KEY---
---END PRIVATE KEY---
```

Les clé privées chiffrées en utilisant PKCS#5 v2.0 et un compteur d'itération élevé sont plus sécurisés que ceux chiffrés en utilisant les formats compatible SSLeay traditionnels. Le chiffrement par défaut est seulement de 56bits parce que c'est le chiffrement le plus courant dans les implémentations supportant PKCS#8.

Certains logiciels peuvent utiliser des algorithmes de chiffrement basé sur mot de passe PKCS#12 avec des clés privées au format PKCS#8 : ils sont gérés automatiquement mais il n'y a pas d'option pour les produire.

Il est possible d'écrire des clé privée encodées DER au format PKCS#8 parce que les détails de chiffrement sont inclus au niveau ASN1 alors que le format traditionnel les inclus au niveau PEM.

Algorithmes PKCS#5 v1.5 et PKCS#12

Divers algorithmes peuvent être utilisés avec l'option -v1, incluant PKCS#5 v1.5 et PKCS#12 :

PBE-MD2-DES PBE-MD5-DES Ces algorithmes sont inclus dans la spécification PKCS#5 v1.5. Ils offrent une protection 56-bits vu qu'ils utilisent DES

PBE-SHA1-RC2-64 PBE-MD2-RC2-64 PBE-MD5-RC2-64 PBE-SHA1-DES Ces algorithmes ne sont pas mentionnés dans la spécification PKCS#5 v1.5 mais utilisent le même algorithme de dérivation de clé et sont supportés par certains logiciels. Ils sont mentionnés dans PKCS#5 v2.0. Ils utilisent soit rs2 64bits, ou DES 56bits

PBE-SHA1-RC4-128 PBE-SHA1-RC4-40 PBE-SHA1-3DES PBE-SHA1-2DES PBE-SHA1-RC2-128 PBE-SHA1-RC2-40 Ces algorithems utilisent l'algorithme de chiffrement de mot de passe PKCS#12 et utilisent 3DES ou rc2 128bits.

Exemples

Convertir une forme privée traditionnelle en PKCS#5 v2.0 avec 3DES
openssl pkcs8 -in key.pem -topk8 -v2 des3 -out enckey.pem
Convertir un forme privée traditionnelle en PKCS#5 v2.0 avec AES 256 en mode CBC et hmacWithSHA256 PRF
openssl pkcs8 -in key.pem -topk8 -v2 aes-256-cbc -v2prf hmacWithSHA256 -out enckey.pem
Convertir une clé privée en PKCS#8 utilisant PKCS#5 1.5 (DES)
openssl pkcs8 -in key.pem -topk8 -out enckey.pem
Convertir un clé privée en PKCS#8 en utilisant un algorithme compatible PKCS#5 1.5 (DES)
openssl pkcs8 -in key.pem -topk8 -out enckey.pem -v1 PBE-SHA1-3DES
Lire une clé privée PKCS#8 DER non-chiffré
openssl pkcs8 -inform DER -nocrypt -in key.der -out key.pem
Convertir une clé privée depuis un format PKCS#8 dans un format traditionnel
openssl pkcs8 -in pk8.pem -out key.pem
Convertir une clé privée au format PKCS#8, la chiffrer avec AES-256 et un million d'itération
openssl pkcs8 -in raw.pem -topk8 -v2 aes-256-cbc -iter 1000000 -out pk8.pem

Standards

Le format des clé privée DSA PKCS#8 (et d'autres) ne sont pas bien documentés : c'est caché dans PKCS#11 v2.01, section 11.9. Openssl se conforme à ce standard.