Documentation du Script de Création de l'Autorité de Certification Racine

LARINOUNA Mohamed, BLOT Thomas, DREZEN Corentin, FAVRE Leo, GUEYE Mouhamadou

April 2024

Table des matières

1 Introduction

Ce script Python utilise la bibliothèque PyOpenSSL pour créer une Autorité de Certification (CA) racine. Cela inclut la génération d'une clé privée pour la CA, la création d'un certificat auto-signé pour la CA, et l'exportation de la clé privée et du certificat. Voici une explication détaillée de chaque étape du script.

2 Prérequis

Python 3.x

Bibliothèque PyOpenSSL

3 Étapes du Script

3.1 1. Génération de la Clé Privée de la CA Racine

```
ca_key = crypto.PKey()
ca_key.generate_key(crypto.TYPE_RSA, 2048)
```

crypto.PKey() : Crée un objet de clé privée. 1

generate_key() : Génère une clé RSA. Le premier argument spécifie le type de la clé (RSA), et le deuxième argument spécifie la taille de la clé en bits (2048 bits), offrant un bon équilibre entre sécurité et performance.

3.2 2. Création du Certificat de la CA Racine

```
ca_cert = crypto.X509()
ca_cert.set_serial_number(1)
ca_cert.gmtime_adj_notBefore(0)
ca_cert.gmtime_adj_notAfter(10*365*24*60*60)
```

3.4 4. Ajout des Extensions de Certificat

```
ca_cert.add_extensions([
    crypto.X509Extension(b"basicConstraints", True, b"CA:TRUE, pathlen:0"),
    ...
])
```

Les extensions fournissent des informations supplémentaires sur l'utilisation du certificat. Ici, basicConstraints indique que le certificat appartient à une CA, keyUsage spécifie les usages autorisés de la clé, et subjectKeyIdentifier fournit un identifiant unique pour la clé du sujet.

3.5 5. Signature du Certificat

```
ca_cert.sign(ca_key, 'sha256')
```

Le certificat est signé avec la clé privée de la CA en utilisant l'algorithme de hachage SHA-256, ce qui complète le processus de création du certificat.

3.6 6. Exportation de la Clé Privée et du Certificat

```
with open("ca_private_key.pem", "wb") as f:
    f.write(crypto.dump_privatekey(crypto.FILETYPE_PEM, ca_key))

with open("ca_cert.pem", "wb") as f:
    f.write(crypto.dump_certificate(crypto.FILETYPE_PEM, ca_cert))
```

Les fonctions dump_privatekey() et dump_certificate() convertissent respectivement la clé privée et le certificat en format PEM, qui est ensuite écrit dans des fichiers pour une utilisation ultérieure.