Minis-projets en sécurité

Encadrement	Professeur My. Ahmed EL KIRAM
Réalisation	Yassine CHERRADI

I. Environnement

Environnement de virtualisation	Environnement de déploiment	Addresse du réseau	Type d'adapteur réseau de la VM	Addresse IP du Client	Addresse IP de l'AC Root, l'AC Intermediaire et les Serveurs Web et DNS
VirtualBox 6.1.10 Ubuntu	Ubuntu 16.04.7 LTS	192.168.56.0/24	Host-only Adapter, 'vboxnet' - Paravirtualized Network (virtio-net)	192.168.56.1	192.168.56.3

II. Passwordless SSH

1. Configuration :

1.1 Configuration du Client :

```
ssh-keygen
ssh-copy-id -i /home/user/.ssh/id_rsa.pub user@192.168.56.3
```

1.2. Configuration du Serveur :

```
vim /etc/ssh/sshd_config
    PasswordAuthentication no
    ChallengeResponseAuthentication no
    UsePAM no

systemctl restart sshd
```

III. Autorité de certification OpenSSL

1. Introduction :

OpenSSL est une bibliothèque cryptographique libre et gratuite qui fournit plusieurs outils en ligne de commande pour la gestion des certificats numériques. Certains de ces outils peuvent être utilisés pour agir en tant qu'autorité de certification.

Une autorité de certification (AC) est une entité qui signe des certificats numériques. De nombreux sites web doivent faire savoir à leurs clients que la connexion est sécurisée. Ils paient donc une AC de confiance internationale (par exemple, VeriSign, DigiCert) pour signer un certificat pour leur domaine.

Dans certains cas, il peut être plus judicieux d'agir en tant que propre AC, plutôt que de payer un AC comme DigiCert. Les cas les plus courants sont la sécurisation d'un site web intranet ou la délivrance de certificats aux clients pour leur permettre de s'authentifier auprès d'un serveur (par exemple, Apache, OpenVPN).

2. Créer une paire pour la racine

Agir en tant qu'autorité de certification (AC) signifie traiter des paires cryptographiques de clés privées et de certificats publics. La toute première paire cryptographique que nous allons créer est la paire de la racine. Celle-ci se compose de la clé de la racine (ca.key.pem) et du certificat de la racine (ca.cert.pem). Cette paire forme l'identité de votre AC.

Généralement, l'AC racine ne signe pas directement les certificats des serveurs ou des clients. Elle n'est utilisée que pour créer une ou plusieurs AC intermédiaires, auxquelles l'AC racine fait confiance pour signer des certificats en son nom. Il s'agit là d'une bonne pratique. Elle permet de garder la clé de la racine hors ligne et inutilisée autant que possible, car toute compromission de la clé de la racine est désastreuse.

2.1. Préparer le répertoire :

1. Choisissez un répertoire (/root/ca) pour stocker toutes les clés et certificats.

```
mkdir /root/ca
```

2. Créer la structure du répertoire. Les fichiers index.txt et serial agissent comme une base de données pour garder la trace des certificats signés.

```
cd /root/ca
mkdir certs crl newcerts private
chmod 700 private
touch index.txt
echo 1000 > serial
```

2.2. Préparer le fichier de configuration :

```
[ CA_default ]
# Directory and file locations.
dir               = /root/ca
countryName_default             = MA
stateOrProvinceName_default     = Marrakech-Safi
0.organizationName_default      = CHERRADI Ltd
```

2.3. Créer une clé pour la racine :

Créez la clé de la racine (ca.key.pem) et conservez-la en toute sécurité. Toute personne en possession de la clé de la racine peut délivrer des certificats de confiance. Cryptez la clé de la racine avec un cryptage AES 256-bits et un mot de passe fort.

Note: Utilisez 4096 bits pour toutes les clés de l'AC racine et intermédiaires. Vous pourrez toujours signer des certificats de serveur et de client d'une longueur plus courte.

```
cd /root/ca
openssl genrsa -aes256 -out private/ca.key.pem 4096
    // pass phrase : prcasp

chmod 400 private/ca.key.pem
```

2.4. Créer un certificat pour la racine :

Utilisez la clé de la racine (ca.key.pem) pour créer un certificat de la racine (ca.cert.pem). Donnez au certificat de la racine une longue date d'expiration, par exemple vingt ans. Une fois le certificat de la racine expiré, tous les certificats signés par l'AC deviennent invalides.

Attention: Chaque fois que vous utilisez l'outil req, vous devez spécifier un fichier de configuration à utiliser avec l'option -config, sinon OpenSSL sera par défaut configuré '/etc/pki/tls/openssl.cnf'.

```
cd /root/ca
openssl req -config openssl.cnf \
    -key private/ca.key.pem \
    -new -x509 -days 7300 -sha256 -extensions v3_ca \
    -out certs/ca.cert.pem

    // pass phrase : rcasp

    // Country Name (2 letter code) [XX]:MA
    // State or Province Name []:Marrakech-Safi
    // Organization Name []:CHERRADI Ltd
    // Organizational Unit Name []:CHERRADI Ltd Certificate Authority
    // Common Name []:CHERRADI Ltd Root CA

chmod 444 certs/ca.cert.pem
```

2.5. Vérifier le certificat de la racine :

```
openssl x509 -noout -text -in certs/ca.cert.pem
```

3. Créer une paire pour l'intermédiaire

Une autorité de certification (AC) intermédiaire est une entité qui peut signer des certificats au nom de l'AC racine. L'AC racine signe le certificat intermédiaire, formant ainsi une chaîne de confiance.

L'utilisation d'une AC intermédiaire a pour objectif principal la sécurité. La clé de la racine peut être maintenue hors ligne et utilisée aussi rarement que possible. Si la clé de l'intermédiaire est compromise, l'AC racine peut révoquer le certificat de l'intermédiaire et créer une nouvelle paire cryptographique pour intermédiaire.

3.1. Préparer le répertoire :

1. Les fichiers de l'AC racine sont conservés dans /root/ca. Choisissez un autre répertoire (/root/ca/intermediate) pour stocker les fichiers de l'AC intermédiaires.

```
mkdir /root/ca/intermediate
```

2. Créez la même structure du répertoire que celle utilisée pour les fichiers de l'AC racine. Il est également pratique de créer un répertoire `csr` pour contenir les demandes de signature de certificats.

```
cd /root/ca/intermediate
mkdir certs crl csr newcerts private
chmod 700 private
touch index.txt
echo 1000 > serial
```

3. Ajoutez un fichier `crlnumber` à l'arborescence des répertoires de l'AC intermédiaire. `crlnumber` est utilisé pour garder une trace des listes de révocation de certificats.

```
echo 1000 > /root/ca/intermediate/crlnumber
```

4. Copiez le fichier de configuration de l'AC intermédiaire de l'annexe dans `/root/ca/intermediate/openssl.cnf`. Cinq options ont été modifiées par rapport au fichier de configuration de l'AC racine :

```
[ CA_default ]
dir             = /root/ca/intermediate
private_key     = $dir/private/intermediate.key.pem
certificate     = $dir/certs/intermediate.cert.pem
crl             = $dir/crl/intermediate.crl.pem
policy          = policy_loose
```

3.2. Créer une clé pour l'intermédiaire :

Créez la clé de l'intermédiaire (intermediate.key.pem). Cryptez la clé de l'intermédiaire avec un cryptage AES 256-bits et un mot de passe fort.

```
cd /root/ca
openssl genrsa -aes256 \
    -out intermediate/private/intermediate.key.pem 4096
    // pass phrase : imcasp

chmod 400 intermediate/private/intermediate.key.pem
```

3.3. Créer un certificat pour l'intermédiaire :

Utilisez la clé de l'intermédiaire pour créer une `demande de signature de certificat` (CSR). Les détails doivent généralement correspondre à ceux de l'AC racine. Le nom commun doit cependant être différent.

Attention: Assurez-vous de spécifier le fichier de configuration de l'AC intermédiaire (intermediate/openssl.cnf).

```
cd /root/ca
openssl req -config intermediate/openssl.cnf -new -sha256 \
    -key intermediate/private/intermediate.key.pem \
    -out intermediate/csr/intermediate.csr.pem

    // pass phrase : imcasp

    // Country Name (2 letter code) [XX]:MA
    // State or Province Name []:Marrakech-Safi
    // Organization Name []:CHERRADI Ltd
    // Organizational Unit Name []:CHERRADI Ltd Certificate Authority
    // Common Name []:CHERRADI Ltd Intermediate CA
```

Pour créer un certificat pour l'intermédiaire, utilisez l'AC racine avec l'extension v3_intermediate_ca pour signer le CSR de l'intermédiaire. Le certificat de l'intermédiaire doit être valide pour une période plus courte que le certificat de la racine. Dix ans serait raisonnable.

Attention: Cette fois, spécifiez le fichier de configuration de l'AC racine (/root/ca/openssl.cnf).

```
cd /root/ca
openssl ca -config openssl.cnf -extensions v3_intermediate_ca \
    -days 3650 -notext -md sha256 \
    -in intermediate/csr/intermediate.csr.pem \
    -out intermediate/certs/intermediate.cert.pem

    // pass phrase : rcasp

chmod 444 intermediate/certs/intermediate.cert.pem
```

Le fichier index.txt est l'endroit où l'outil OpenSSL `ca` stocke la base de données des certificats. Ne pas supprimer ou modifier ce fichier à la main. Il doit maintenant contenir une ligne qui fait référence au certificat de l'intermédiaire.

3.4. Vérifier le certificat de l'intermédiaire :

Comme nous l'avons fait pour le certificat de la racine, vérifiez que les détails du certificat de l'intermédiaire sont corrects.

```
openssl x509 -noout -text \
    -in intermediate/certs/intermediate.cert.pem
```

Vérifiez le certificat de l'intermédiaire par rapport au certificat de la racine. Un OK indique que la chaîne de confiance est intacte.

```
openssl verify -CAfile certs/ca.cert.pem \
    intermediate/certs/intermediate.cert.pem
```

3.5. Créer le fichier de chaîne de certificats :

Lorsqu'une application (par exemple, un navigateur web) tente de vérifier un certificat signé par l'AC intermédiaire, elle doit également vérifier le certificat de l'intermédiaire par rapport au certificat de la racine. Pour compléter la chaîne de confiance, créez une chaîne de certificats de l'AC à présenter à la demande.

Pour créer la chaîne de certificats de l'AC, concaténez les certificats intermédiaire et racine ensemble. Nous utiliserons ce fichier ultérieurement pour vérifier les certificats signés par l'AC intermédiaire.

```
cat intermediate/certs/intermediate.cert.pem \
    certs/ca.cert.pem > intermediate/certs/ca-chain.cert.pem

chmod 444 intermediate/certs/ca-chain.cert.pem
```

Note: Notre fichier contenant la chaîne de certificats doit inclure le certificat de la racine car aucune application client n'en a encore connaissance. Une meilleure option, en particulier si vous administrez un intranet, consiste à installer votre certificat de la racine sur chaque client qui doit se connecter. Dans ce cas, le fichier de chaîne ne doit contenir que votre certificat de l'intermédiaire.

4. Signer des certificats pour serveurs et clients

Nous allons signer des certificats en utilisant notre AC intermédiaire. Vous pouvez utiliser ces certificats signés dans diverses situations, par exemple pour sécuriser des connexions à un serveur web ou pour authentifier des clients se connectant à un service.

Note: Les étapes ci-dessous sont, de votre point de vue en tant qu'autorité de certification. Un tiers peut cependant créer sa propre clé privée et sa propre demande de signature de certificat (CSR) sans vous révéler sa clé privée. Il vous donne sa CSR, et vous lui rendez un certificat signé. Dans ce scénario, ignorez les commandes `genrsa` et `req`.

4.1. Créer une clé :

Les paires de nos racine et intermédiaire sont de 4096 bits. Les certificats des serveurs et des clients expirent normalement après un an, nous pouvons donc utiliser 2048 bits à la place.

Note: Bien que 4096 bits soit légèrement plus sûr que 2048 bits, il ralentit `les poignées de main TLS` et augmente considérablement la charge du processeur. C'est pourquoi la plupart des sites web utilisent des paires de 2048 bits.

Si vous créez une paire cryptographique à utiliser avec un serveur web (par exemple, Apache), vous devrez entrer ce mot de passe chaque fois que vous redémarrerez le serveur web. Vous pouvez omettre l'option -aes256 pour créer une clé sans mot de passe.

```
cd /root/ca
openssl genrsa -aes256 \
    -out intermediate/private/yacyncherradi.net.key.pem 2048

chmod 400 intermediate/private/yacyncherradi.net.key.pem
```

4.2. Créer un certificat :

Utilisez la clé privée pour créer une demande de signature de certificat (CSR). Les détails de la CSR ne doivent pas nécessairement correspondre à ceux de l'AC intermédiaire. Pour les certificats de serveur, le nom commun doit être un FQDN (par exemple, www.example.com), tandis que pour les certificats de client, il peut s'agir de n'importe quel identifiant unique (par exemple, une adresse électronique). Notez que le `nom commun` ne peut pas être le même que celui de votre certificat racine ou intermédiaire.

```
cd /root/ca
openssl req -config intermediate/openssl.cnf \
    -key intermediate/private/yacyncherradi.net.key.pem \
    -new -sha256 -out intermediate/csr/yacyncherradi.net.csr.pem

    // pass phrase : ycsp

    // Country Name (2 letter code) [XX]:MA
    // Organization Name []:Yacyn Ltd
    // Organizational Unit Name []:Yacyn Ltd Web Services
    // Common Name []:yacyncherradi.net
```

Pour créer un certificat, utilisez l'AC intermédiaire pour signer le CSR. Si le certificat doit être utilisé sur un serveur, utilisez l'extension `server_cert`. Si le certificat doit être utilisé pour l'authentification d'un utilisateur, utilisez l'extension `usr_cert`. Les certificats ont généralement une validité d'un an, bien qu'une autorité de certification accorde généralement quelques jours supplémentaires pour des raisons de commodité.

```
cd /root/ca
openssl ca -config intermediate/openssl.cnf \
    -extensions server_cert -days 375 -notext -md sha256 \
    -in intermediate/csr/yacyncherradi.net.csr.pem \
    -out intermediate/certs/yacyncherradi.net.cert.pem

chmod 444 intermediate/certs/yacyncherradi.net.cert.pem
```

Le fichier `intermediate/index.txt` doit contenir une ligne faisant référence à ce nouveau certificat.

4.3. Vérifier le certificat :

```
openssl x509 -noout -text \
    -in intermediate/certs/yacyncherradi.net.cert.pem
```

Utilisez le fichier de la chaîne de certificats de l'AC que nous avons précédemment créé (ca-chain.cert.pem) pour vérifier que le nouveau certificat possède une chaîne de confiance valide.

4.4. Déployer le certificat :

Vous pouvez maintenant soit déployer votre nouveau certificat sur un serveur, soit distribuer le certificat à un client. Lors du déploiement vers une application serveur (par exemple, Apache), vous devez mettre à disposition les fichiers suivants :

```
ca-chain.cert.pem
yacyncherradi.net.key.pem
yacyncherradi.net.cert.pem
```

Si vous signez un CSR d'un tiers, vous n'avez pas accès à sa clé privée. Il vous suffit donc de lui rendre le fichier de la chaîne (ca-chain.cert.pem) et le certificat (yacyncherradi.net.cert.pem).

5. Annexe

1. Root CA configuration file
2. Intermediate CA configuration file
3. Configuration du serveur Web et DNS

IV. Sources

- OpenSSL PKI Tutorial v1.1 par Stefan H. Holek
- OpenSSL Certificate Authority par Jamie Nguyen
- Mise en place d'une Autorité de certification par Thomas Boutry
- How to Create SSL Certificates for Development par Patrick Kalkman