AUTORITÉ DE CERTIFICATION OpenSSL

Contributeurs

(SISR2-2016) Benjamin Landais

** Testé et validé sur DEBIAN 8 **

**Problème entre les services sur
DEBIAN 7**

Pourquoi installer une autorité de certification ?

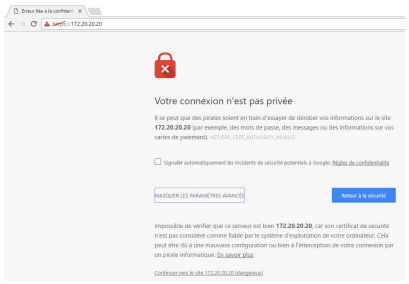
L'autorité de certification va permettre de signer des demandes de certificats, mais aussi d'en révoquer.

Son rôle est aussi de sécuriser plusieurs serveurs / services en proposant une connexion chiffrée.

Il permet de délivrer des certificats et ainsi d'authentifier les correspondants.

Dans la section SIO nous avons plusieurs serveurs qui demandent une vérification lors de l'accès via le navigateur. Le VPN, le Portefeuille de connaissance, et les serveurs Proxmox par exemple. Dans l'avenir toutes les applications de la section devraient passer en mode sécurisées.

L'installation de l'autorité de certification permettra donc d'éviter ce genre de message.



Prérequis

Besoins pour installer & tester le serveur d'autorité de certification :

L'autorité de certification sera installée sur un serveur Linux, sous Debian 8 (ou plus récent).

- J'utiliserai les scripts du paquet d'OpenSSL, qui est une « boîte à outils » de chiffrement.
- Vous allez avoir besoin de serveurs sur lesquels installer vos certificats.
- Vous aurez besoin d'un client avec un navigateur pour tester vos certificats.
- IL FAUDRA PENSER A ADAPTER LES VALEURS.

1- Installation d'openssl :

```
apt-get update
apt-get install openssl
```

2- Création de l’autorité de certification via openssl :

Modification du fichier « openssl.cnf » qui contient des informations sur notre CA : Chemin :

```
nano /etc/ssl/openssl.cnf
```

On peut changer la valeur du default_days = 3652 ici notre durée correspond à 10ans

Ensuite les différentes informations concernant l'autorité de certification :

- Country Name : FR
- StateOrProvinceName : Normandie
- LocalityName : Caen
- OrganizationName : BTS SIO

Exécution du scipt CA.sh dans

```
cd /usr/lib/ssl/misc/
sudo ./CA.sh -newca
```

Ou modifier le fichier CA.pl si CA.sh n'existe pas :

```
sudo ./CA.pl -newca
```

C’est ce fichier qui va générer notre PKI (Private Key Infrastructure). Une infrastructure à clés publiques fournit des garanties permettant de faire a priori confiance à un certificat signé par une autorité de certification grâce à un ensemble de services.

Les services de la PKI sont les suivants :

- Génération de certificats (nouveaux certificats)
- Renouvellement de certificats (quand ils arrivent à terme)
- Révocation de certificats (suppression)
- Publication de certificats

Il faut faire entrer pour créer le fichier.

Il va commencer par nous demander d’entrer une PEM pass phrase qui est une phrase de sécurité. Ici la pass phrase suivante est :
PEM pass phrase : EXEMPLE_EXEMPLE

Le fichier va ensuite demander des informations concernant la localité (pays, région…)

```
- Country Name : FR
```

```
- StateOrProvinceName : Normandie
- LocalityName :  Caen
- OrganizationName : BTS SIO
- CommonName : inforostand14   (NE PAS METTRE LE .net sous peine d'erreurs par la suite)
```

Une fois tous les renseignements complétés il va falloir indiquer le challenge password. \\ **Challenge password : EXEMPLE_EXEMPLE**

Le certificat de notre autorité est maintenant créé. Nous pouvons retrouve le fichier dans :

```
ls -l /usr/lib/ssl/misc/demoCA/
```

Il s'agira du fichier cacert.pem

3- Création d'une demande de signature :

```
openssl req -new -nodes -keyout /etc/ssl/private/www-key.pem -out /tmp/www-req.pem -days 3650
```

- **REQ** permet de créer et traiter les demandes de certificats (format PKCS#10) - **NEW** permet de générer la demande de certificat - **NODES** désactive le chiffrement de la clé privé - **KEYOUT** donne le nom ou le fichier ou la clé privée sera crée - **OUT** nom du fichier de sortie (cela correspond au certificat) - **DAYS** nombre de jour ou le certificat est valide

Le fichier est en donc en train de ce générer.

Il redemandera des informations que nous avons pu saisir avant dans le fichier CA.sh il faut réinscrire les même.

Mais à la ligne « Common Name (e.g server FQDN or YOUR NAME) [] : » penser à répondre : inforostand14.net le certificat sera valable pour le domaine.

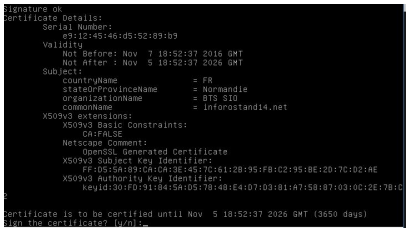
On pense aussi à ré-indiquer le Challenge password : **EXEMPLE_EXEMPLE**

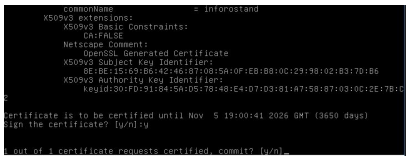
En effectuant la commande « ls » dans le dossier « temp » vous devriez normalement voir le fichier www-req.pem

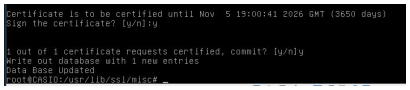
- Création du certificat depuis la demande de signature :

```
cd /usr/lib/ssl/misc/
openssl ca -out /etc/ssl/certs/www-cert.pem -infiles /tmp/www-req.pem
```

A la suite de cette commande il demandera d'entrer le PEM PassPhrase que nous avons précédemment défini par **EXEMPLE_EXEMPLE**.







Le certificat est donc crée et signé ! Il se peut que l'erreur suivante apparaisse (elle m'est apparue) :

```
Certificate is to be certified until Nov 07
17:36:37 2015 GMT (3650 days)
Sign the certificate? [y/n]:y

failed to update database
TXT_DB error number 2
```

Par défaut, la signature de plusieurs certificats créés avec les même paramètres de sujet (countryName, stateOrProvinceName, organizationName, organizationalUnitName, commonName, emailAddress) n'est pas autorisée.

Pour des certificats machine, le "commonName" désigne le nom de la machine (souvent dans son entier nom.domaine) tandis que pour les utilisateurs il désigne leurs "Prénom NOM".

Dans mon cas l'erreur s'était produite car le CommonName était le même dans le fichier CA.sh et dans le fichier de demande de signature. En l'occurrence j'avais deux fois inforostand14.net, alors qu'il le fallait seulement une seule fois lors de la demande de signature.

Si les paramètres décrits précédemment ne sont pas respectés l'erreur décrite apparaîtra.

4- Installation du certificat sur le serveur :

Serveur de test 172.20.89.101 root/....

Installation apache

```
Apt-get update
Apt-get install apache2
```

Ensuite rdv sur http://172.20.89.101 la page web devrait s'afficher.

Maintenant configuration de la partie ssl du serveur.

```
a2enmod ssl
a2ensite default-ssl
service apache2 reload
service apache2 restart
```

Maintenant notre site est accessible via l'adresse suivante https://172.20.89.101

Avant de transférer des fichiers vers le serveur cible, nous allons convertir le certificat cacert.pem en cacert.cer pour qu'il puisse être validé sur des navigateurs utilisé sous un os windows.

Commande à exécuter :

```
cd /usr/lib/ssl/misc/demoCA
openssl x509 -inform PEM -in cacert.pem -outform DER -out cacert.cer
```

Nous allons maintenant transférer sur le serveur 4 fichiers : cacert.pem, cacert.cer, www-cert.pem et www-key.pem Nous allons les transférer dans le dossier /home afin de les retrouver plus facilement.

J'utilise la commande :

```
scp –r –p /usr/lib/ssl/misc/demoCA/cacert.pem root@172.20.89.101:/tmp
```

(il faudra penser à adapter le chemin pour les autres fichiers.) Le serveur nous demande alors le mot de passe, entrez le et le transfert du fichier va s'effectuer en quelques secondes.

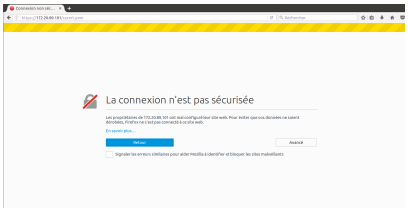
Ensuite une fois les fichiers sur le serveur ont doit transférer les fichiers dans les dossiers respectifs :

```
cp /tmp/cacert.pem /var/www/
cp /tmp/cacert.cer /var/www/
cp /tmp/www-cert.pem /etc/ssl/certs/
cp /tmp/www-key.pem /etc/ssl/private/
```

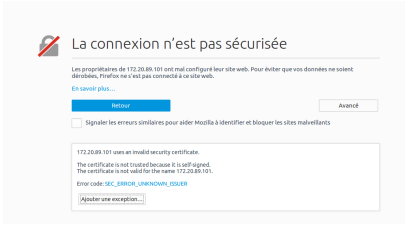
Maintenant que le certificat et la clé sont sur le serveur nous devons vérifier que les navigateurs prennent bien en compte notre certificat :

Sous LINUX :

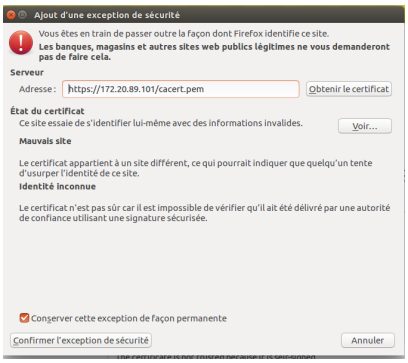
Ouvre le navigateur par exemple Mozilla Firefox : Connectez vous au serveur à l'adresse suivante : https://172.20.89.10/cacert.pem



Un message indiquant que la connexion n'est pas sécurisée apparaît, cliquez sur "avancé"



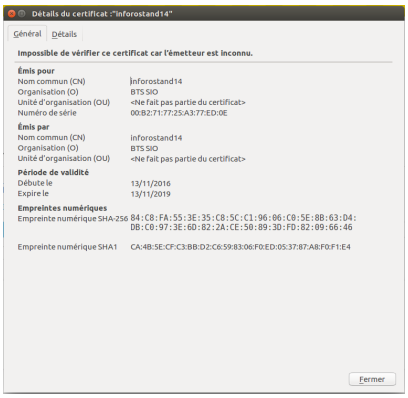
Le message indique qu'il y a un certificat mais qu'il n'est pas en règle, ce qui est un message d'erreur normal car le certificat est auto-signé actuellement par le serveur et non pas par le CA. Nous allons donc faire « Ajouter une exception »



Confirmer alors l'exception de sécurité, le message suivant apparaît :



En cliquant sur « voir » les détails du certificat s'afficheront :



La connexion est maintenant sécurisée.



Sur Windows :

Il faut se connecter à https://172.20.89.101/cacert.cer. Le message de sécurité s'affiche alors il faut cliquer sur « avancé » puis « ajouter une exception ». La demande téléchargement du certificat va alors s'afficher.

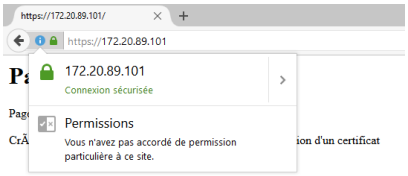
Rendez-vous maintenant dans le dossier de téléchargement, et double cliquez sur le fichier cacert.cer pour l'installer.

Cliquez sur « installer un certificat »

Sélectionner ensuite « placer tout les certificats dans le magasin suivant » et dans la fenêtre qui s'affiche choisir « autorités de certification racines de confiances »

Faire suivant, et enfin terminer, le message indiquant que l'importation est réussis devrait apparaitre.

Actualiser la page :



La connexion est donc sécurisée, l'importation du certificat s'est bien passée. Nos paramètres de configuration sont corrects.

From:
https://wiki.sio.bts/ - **WIKI SIO : DEPUIS 2017**

Permanent link:
https://wiki.sio.bts/doku.php?id=openssl&rev=1664373053

Last update: **2022/09/28 13:50**