TP2
HTTPS : certificats X509

Objectifs

- utiliser SSL pour créer et signer des certificats cryptographiques
- déployer une application sécurisée

Matériel

- 1 machine virtuelle Linux (VirtualBox) et 1 carte réseau.
- 1 client Windows

Déroulement

Un compte-rendu de TP est à rendre pour chaque TP réalisé (1 compte-rendu par binôme).

A envoyer par mail au format PDF à l'enseignant au plus tard 7 jours après le TP.

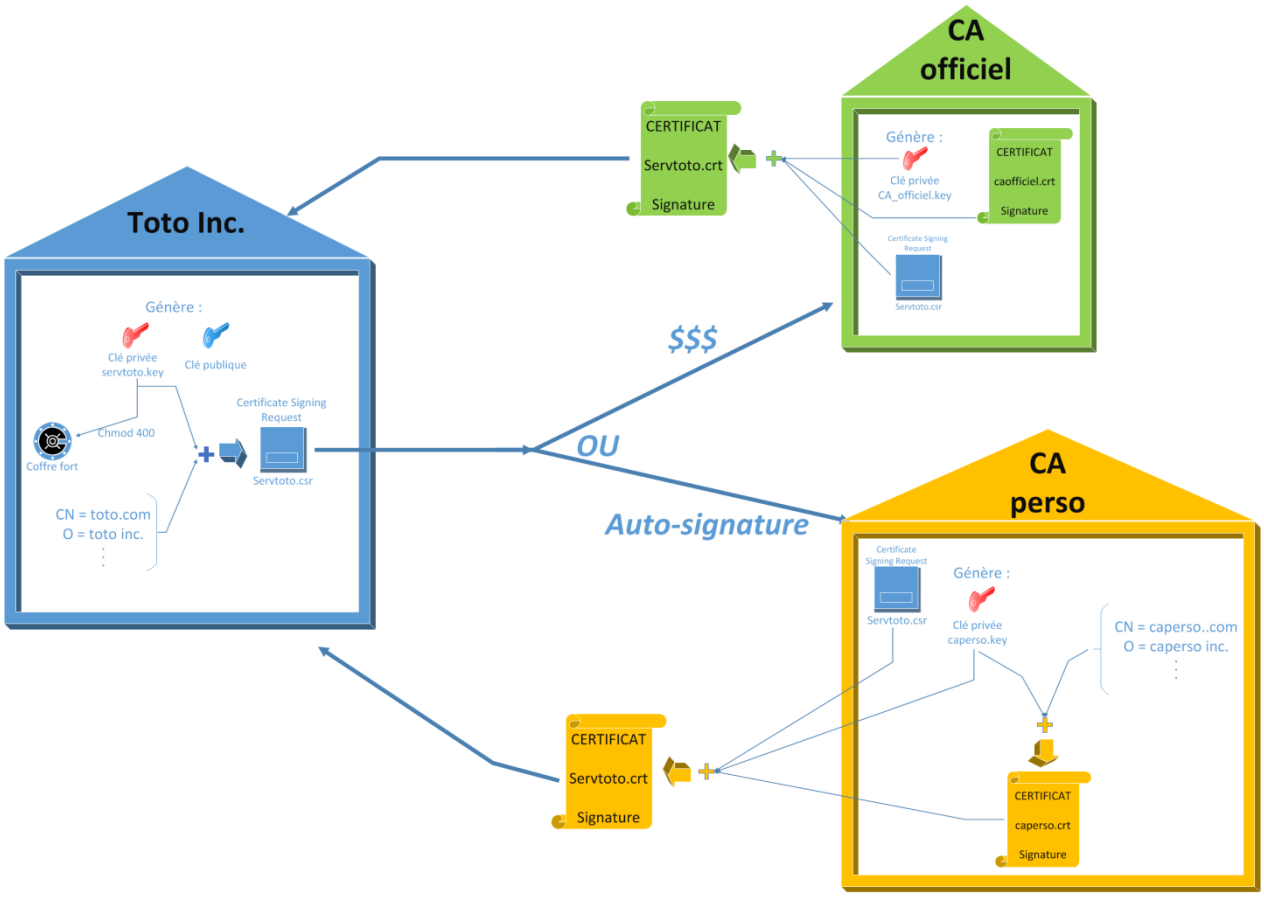
Le sujet du mail doit contenir un copier-coller du nom du fichier mis en pièce jointe. Le nom du fichier doit respecter le format qui vous est imposé par la license professionnelle ASUR. Tout mail qui ne respecte pas ce format sera automatiquement redirigé vers le dossier « Corbeille » !

Ce qui est attendu dans vos comptes rendus:

- Des explications claires et concises sur vos manipulations. Des schémas peuvent aider, ainsi que des captures d'écrans commentées.
- Les réponses aux questions posées (bien sûr).
- Vous pouvez ajouter à l'annexe si nécessaire des listings / scripts réalisés.

1 Principe

La création et l'installation d'un certificat obéit au processus suivant (schéma de Frédéric Michel):



Un « jeu » est constitué :

- d'un certificat faisant autorité (CA = Certificate Authority) ou certificat racine (Root Certificate) : *.ca
- d'un certificat client (certificat) : *.crt ou *.cer
- d'une clé privée (PK = Private Key) : *.key

Au passage, un document appelé (CSR = Certificate Signing Request) est produit.

Un certificat contient un certain nombre d'éléments clés appelés champs:

- organisation : O = organization
- unité d'organisation : OU = Organization Unit
- pays : C = country
- état / département : S = State
- ville : L = Locality
- nom commun : CN = Common Name (CN).

La notion de « Common Name » est souvent mal comprise. C'est la concaténation du nom d'hôte et du nom de domaine. Il DOIT correspondre au nom du site sur lequel est effectuée la requête.
En général, un certificat généré pour « domain.com » NE peut PAS être utilisé pour « www.domain.com »

2 DNS et domaine

Pour la suite, votre machine doit faire partie d'un domaine et avoir un nom d'hôte. L'enseignant vous affectera un numéro *X* qui va déterminer votre nom d'hôte « www*X* » et votre adresse IP 172.24.100.100+*X* sur le domaine « iut-valence.net ».
Ce domaine est servi par un DNS à l'adresse 172.24.100.100.

Configurez votre machine pour qu'elle utilise ce DNS et l'adresse IP qui vous a été donnée. Pensez aussi à :
- changer le nom de la machine (/etc/*hostname*),
- mettre à jour la liste des hôtes (/etc/*hosts*),
- et vérifiez que tout est en ordre (*nslookup*).

QUESTION 1 : donnez dans votre compte-rendu le contenu des 3 fichiers modifiés.

3 Préparation

Installer SSL, si pas déjà fait :

```
apt-get install openssl
```

SSL met à disposition :
- le fichier de configuration « /etc/ssl/openssl.cnf »
- trois répertoires constituants les dépôts de certificats :
  - /etc/ssl/certs
  - /etc/ssl/private
  - /etc/ssl/ca (**à créer**).

4 Création d'un certificat auto-signé

Le fichier « /etc/ssl/openssl.cnf » doit être modifié au préalable selon l'annexe 1. Le common name sera votre FQDN.

QUESTION 2 : qu'est-ce que le format X.509 ?

Dans le répertoire de votre choix (home, root, …), créer un répertoire « crypto » qui contiendra votre travail. Dans ce répertoire, créer le répertoire « demoCA » puis un répertoire « demoCA/newcerts ».

```
crypto
      +-- create.sh
      +-- purge.sh
      +-- clean.sh
      +-- demoCA
            +-- newcerts
                  +-- *
            +-- *
```

Recopier le script shell ci-dessous (purge.sh) et le lancer :

```
        cd demoCA; rm *; cd newcerts; rm *; cd ..; touch index.txt; echo "01" > serial; cd ..;
```

A l'issue de cette première manipulation, vous devriez avoir une arborescence propre.

QUESTION 3 : que contient l'arborescence ?

Commencez par créer le certificat racine (à faire une fois pour toute). Répondre « ENTREE » à toutes les questions vous serons posées :

```
#openssl req -nodes -new -x509 -keyout ca.key -out ca.crt -days 3650
```

Le certificat obtenu doit être rangé en lieu sûr avec des permissions restrictives (r--------).

QUESTION 4 : détaillez les options passées en paramètre. Jetez un coup d'œil au contenu des différents fichiers générés que vous placerez dans votre compte-rendu.

Créez une demande certificat client (CSR). Répondre « ENTREE » à toutes les questions vous serons posées. Il est possible de saisir un mot de passe pour déverrouiller le certificat ultérieurement ce que l'on ne fera pas ici :

```
openssl req -nodes -new -keyout iut-valence.key -out iut-valence.csr -days 3650
```

QUESTION 5 : détaillez les options passées en paramètre. Jetez un coup d'œil au contenu des différents fichiers générés que vous placerez dans votre compte-rendu.

Signez le certificat client avec le certificat racine:

```
openssl ca -cert ca.crt -keyfile ca.key -out iut-valence.crt -in iut-valence.csr
```

QUESTION 6 : détaillez les options passées en paramètre. Jetez un coup d'œil au contenu des différents fichiers générés que vous placerez dans votre compte-rendu.

QUESTION 7 : détailler la structure du certificat client (iut-valence.crt) et expliquer les différents champs.

Les fichiers obtenus sont "jolis" mais inutilisables tels quels avec Windows ou avec un navigateur qui ne comprend que le format PKCS12 :

```
   openssl pkcs12 -export -in iut-valence.crt -inkey iut-valence.key -out iut-valence.p12 -name "Iut-valence's certificate" -certfile ca.crt
```

QUESTION 8 : qu'est-ce que le format PKCS#12 ?

Enfin, on les installe pour pouvoir les partager avec d'autres applications (Apache, DoveCot, Postfix, …) :

```
cp iut-valence.key     /etc/ssl/private/iut-valence.key
cp iut-valence.crt     /etc/ssl/certs/iut-valence.crt
cp ca.crt              /etc/ssl/ca/iut-valence.ca      (répertoire à créer)
```

QUESTION 9 : à l'aide d'un schéma, décrivez et expliquer le processus de création d'un certificat client.

5 Configuration du serveur web Apache

Vérifier que le serveur Apache fonctionne...

Par défaut, le support SSL est désactivé dans Apache. Pour l'activer :

```
a2enmod ssl
```

Par défaut, Apache écoute sur le port 80 (http) mais pas sur le port 443 (https). Ce comportement est paramétré dans le dossier « /etc/apache2/sites-available ».

Pour activer Apache sur le port 443 :

```
a2ensite default-ssl.conf
service apache2 reload
```

Vérifiez que la connexion vers https://wwwN.iut-valence.net fonctionne…

QUESTION 10 : que se passe-t-il ?

Nous allons maintenant activer les certificats dans Apache. Pour cela, éditer le fichier « /etc/apache2/sites-available/default-ssl » :

```
SSLEngine       on
SSLCertificateFile    /etc/ssl/certs/iut-valence.crt
SSLCertificateKeyFile /etc/ssl/private/iut-valence.key
SSLCACertificatePath  /etc/ssl/ca/
SSLCACertificateFile  /etc/ssl/ca/iut-valence.ca
#SSLVerifyClient require
#SSLVerifyDepth  1
```

Retentez une connexion :

- « Je comprends les risques »
- « Ajouter une exception »
- « Obtenir le certificat »
- « Voir »
- et quitter SANS confirmer l'exception de sécurité.

QUESTION 11 : que se passe-t-il maintenant ?

Récupérer les fichiers « ca.crt » et « iut-valence.p12 » depuis la machine virtuelle et le mettre sur le bureau du client Windows. Double-cliquer dessus et suivre les instructions…

Ensuite, ouvrir Firefox et faire « Outils > Options > Avancé > Certificats > Afficher les certificats ».
Dans l'onglet « Autorités », importer le certificat « ca.crt » et cocher les 3 cases. Vous venez d'installer une nouvelle autorité de certicats. Quitter le navigateur et recharger la page.

QUESTION 12 : pourquoi, malgré cette opération, cette autorité n'est pas valide ?

QUESTION 13 : à l'aide d'un schéma, décrivez et expliquer le processus d'authentification et d'échange de certificats entre le serveur Apache et le navigateur du client.

6 Sécurisation du serveur web Apache

La configuration d'Apache s'effectue au travers du fichier **httpd.conf** (sa localisation dépend de la distribution).

On peut restreindre l'accès au site d'une par les directives **deny** et **allow** du fichier **httpd.conf**, mais aussi par la directive **AccessFileName** permettant de définir le nom du fichier qui indique qu'un répertoire est protégé. Par défaut il s'agit de .**htaccess**.

Voici le rôle des directives de contrôle d'accès de **.htaccess** :
AuthName : nom de la fenêtre d'authentification.
AuthType : type d'authentification.
AuthUserFile : fichier contenant les utilisateurs et mots de passe.
AuthGroupFile : fichier contenant la liste des groupes et des utilisateurs.
Require défini la liste des personnes autorisées.

La commande « ***htpasswd –cb nonfic*** *nom_user répertoire* » permet de générer un fichier avec mot de passe nommé par ***AuhtUserFile***.

Utilisez ce type d'authentification pour un utilisateur et répertoire donné. Lancez Wireshark pour effectuer une capture entre le client et le serveur. Effectuez un accès au répertoire protégé à partir d'un client externe.

QUESTION 14 : que remarquez-vous sur les traces ? Mettez en place une solution SSL pour remédier au problème.

Pour cette dernière partie, vous écrirez un compte rendu détaillé de toutes les opérations que vous avez effectuées, en particuliers, vous commenterez les différentes options des fichiers de configurations utilisés, comment vous avez effectué la création des clés et des certificats, ou sont-ils stockés, et démontrerez par des traces bien choisies et commentées le bon fonctionnement de votre serveur une fois sécurisé.

Webographie :

[1] http://www.linux-france.org/prj/edu/archinet/systeme/ch24s03.html
[2] http://info.ssl.com/article.aspx?id=10048

Annexe 1 : /etc/ssl/openssl.cnf

```
[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = FR
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Drome

localityName                    = City
localityName_default            = Valence

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = "IUT de Valence"

# we can do this but it is not needed normally :-)
1.organizationName              = Second Organization Name (eg, company)
1.organizationName_default      = " Universite Pierre Mendes France (UPMF)"

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default      ="Universite de Grenoble”

commonName                      = Common Name
commonName_default              = "wwwN.iut-valence.net"
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_default            = admin@iut-valence.net
emailAddress_max                = 64

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 0
challengePassword_max           = 16

unstructuredName                = An optional company name
```

Annexe 2 : create.sh

```
#!/bin/sh

# create cert for CA (once)
#openssl req -nodes -new -x509 -keyout ca.key -out ca.crt -days 3650

# create server cert
openssl req -nodes -new -keyout iut-valence.key -out iut-valence.csr -days 3650

# sign cert request with CA
openssl ca -cert ca.crt -keyfile ca.key -out iut-valence.pem -in iut-valence.csr
mv iut-valence.pem iut-valence.crt

#exporting
openssl pkcs12 -export -in iut-valence.crt -inkey iut-valence.key -out iut-
valence.p12 -name "Iut-valence's certificate" -certfile ca.crt
```

Annexe 3 : purge.sh

```
#!/bin/sh
cd demoCA; rm *; cd newcerts; rm *; cd ..; touch index.txt; echo "01" > serial;
cd ..;
```

Annexe 4: clean.sh

```
#!/bin/sh
rm -f iut-valence.*
```

Annexe 5: structure du DNS

named.conf.local :

```
zone "iut-valence.net" IN {
     type master;
     file "/etc/bind/iut-valence.net.zone";
     allow-query { any; };
     allow-transfer { any; };
};

zone "0.100.24.172.in-addr.arpa." IN {
     type master;
     file "/etc/bind/iut-valence.rev.zone";
     allow-query { any; };
     allow-transfer { any; };
};
```

iut-valence.net.zone :

```
$TTL 86400
@      IN    SOA   ns.iut-valence.net.     root.iut-valence.net. (
       2016020805
       21600
       3600
       604800
       86400
       );

       NS    ns.iut-valence.net.

ns     A     172.24.100.100

www1   A     172.24.100.101
www2   A     172.24.100.102
www3   A     172.24.100.103
www4   A     172.24.100.104
www5   A     172.24.100.105
www6   A     172.24.100.106
www7   A     172.24.100.107
www8   A     172.24.100.108
www9   A     172.24.100.109
www10  A     172.24.100.110
www11  A     172.24.100.111
www12  A     172.24.100.112
www13  A     172.24.100.113
www14  A     172.24.100.114
www15  A     172.24.100.115
www16  A     172.24.100.116
```

iut-valence.rev.zone :

```
$TTL 86400
@     IN    SOA   ns.iut-valence.net.     root.iut-valence.net. (
      2016020805
      21600
      3600
      604800
      86400
      );

@     NS    ns.iut-valence.net.

100   PTR   ns.iut-valence.net.
101   PTR   www1.iut-valence.net.
102   PTR   www2.iut-valence.net.
103   PTR   www3.iut-valence.net.
104   PTR   www4.iut-valence.net.
105   PTR   www5.iut-valence.net.
106   PTR   www6.iut-valence.net.
107   PTR   www7.iut-valence.net.
108   PTR   www8.iut-valence.net.
109   PTR   www9.iut-valence.net.
110   PTR   www10.iut-valence.net.
111   PTR   www11.iut-valence.net.
112   PTR   www12.iut-valence.net.
113   PTR   www13.iut-valence.net.
114   PTR   www14.iut-valence.net.
115   PTR   www15.iut-valence.net.
116   PTR   www16.iut-valence.net.
```

Annexe 6: fichier /etc/resolv.conf

Dans l'ordre, mettre :

```
domain iut-valence.net
search iut-valence.net
nameserver 172.24.100.100

domain iut-valence.fr
search iut-valence.fr
nameserver 172.30.1.1
```

Annexe 7: fichier /etc/hosts

Pour éviter un warning chaque fois que vous relancez le serveur Apache :

```
172.24.100.10N    wwwN.iut-valence.net    wwwN
```

Annexe 8: RX_RECORD_TOO_LONG error !

Ce message apparaît lorsque la connexion HTTPS n'est pas activée et que la transaction se fait en HTTP.

Pour vérifier :

telnet localhost 443

- GET /

Si la réponse est en clair, c'est que le VirtualHost sur le port 443 n'est pas activé.