TP 5 Usages du chiffrement : certificats SSL et HTTPS

Objectifs du TP

L'objectif de ce TP est de comprendre comment les certificats sont utilisés pour assurer des connexions sécurisées avec HTTPS. Nous verrons également comment configurer apache pour mettre en place un serveur HTTPS.

1 Certificats

1.1 Introduction

Un certificat électronique signé contient trois blocs de données :
— Une clé publique de chiffrement ou déchiffrement (selon les usages) ;
— Des informations sur l'identité du propriétaire de la clé ou du site web, une durée de validité, etc ;
— Une signature électronique.

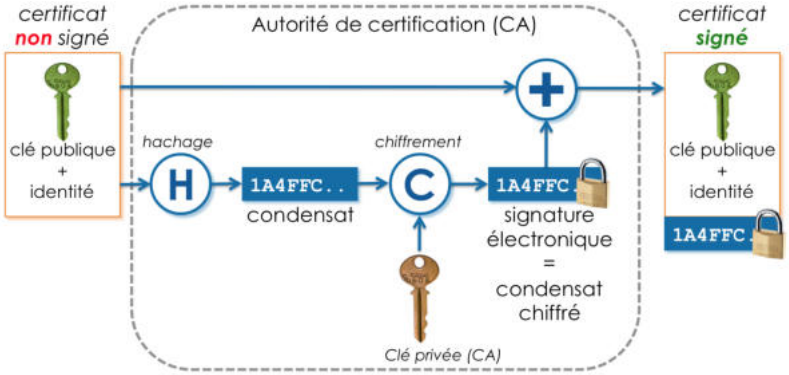

FIGURE 1 – signature électronique

Comme le montre la figure 1, une signature électronique est une empreinte chiffrée par une autorité de certification (CA) avec la clé privée de cette autorité. L'empreinte est le résultat d'un calcul (à l'aide d'une fonction de hachage comme MD5 ou SHA-256) appliqué sur les blocs de données : clé publique + informations.

Lorsqu'un navigateur web se connecte en SSL sur un site web (https ://), il récupère sur le site visité le certificat signé du serveur web et vérifie sa validité. Pour cela, le navigateur doit déchiffrer la signature électronique avec la clé publique de l'une des autorités de certification (CA) qu'il possède dans sa base de données. Si la clé publique du CA n'est pas dans la base, le navigateur émet une alerte, car il n'est pas en mesure de faire confiance au site web.

1.2 Analyse d'un certificat

Dans cette partie, il s'agit d'étudier le certificat associé au site web d'authentification de l'UGA.

FIGURE 2 – site d'authentification de l'UGA

Anatomie d'un certificat

1. Ouvrir le navigateur web Firefox (s'il est déjà ouvert, le fermer puis relancer l'application). Si ce n'est pas la page par défaut du navigateur, se connecter au serveur web `chamilo.iut2.univ-grenoble-alpes.fr`. Normalement, la page doit être redirigée vers le site d'authentification de l'UGA à l'adresse `https://authentification.univ-grenoble-alpes.fr` (voir figure 2).
   Il n'est pas nécessaire de s'authentifier !
2. Dans la barre de saisie de l'URL, cliquer sur le symbole en forme de cadenas (voir image), puis la flèche « Connexion sécurisée », puis « Plus d'information » : une nouvelle fenêtre apparaît, donnant des détails sur le certificat associé au site web. Dans l'onglet « authentification.univ-grenoble-alpes.fr » Quelle est l'adresse du site web couvert par le certificat ? Quelle autorité a vérifié le certificat (champ « Nom de l'émetteur ») ? Pourquoi la vérification est-elle importante ?
3. Dans la section « clé publique », quel est l'algorithme de chiffrement symétrique utilisé ? En quoi la taille d'une clé de chiffrement est-elle importante ?
4. Quand expire le certificat du site ? Pourquoi une durée est-elle nécessaire ?
5. Dans la section "Divers". Il est indiqué « SHA-384 avec chiffrement RSA ». Que nous indique cette information ?

Chaîne de certification Nous nous intéressons maintenant à la chaîne de certification. Sélectionnez successivement les certificats avec les trois onglets ("authentification.univ-grenoble-alpes.fr", "GEANT OV RSA CA 4", et "USERTrust RSA Certification Authority") et notez sur un schéma la valeur du champ « Nom de l'émetteur » en indiquant par une flèche qui est vérifié par qui.

Nous obtenons ainsi la chaîne de certification :

Quelle est la particularité du certificat "USERTrust RSA Certification Authority" ? Pourquoi est-ce nécessaire qu'il soit signé ainsi ?

Fermer toutes les fenêtres sauf la fenêtre principale du navigateur. À partir du menu "Edition", choisir "Préférences". Une nouvelle fenêtre s'ouvre. Choisir "Privacy & Security" puis aller à la section "Security". Cliquer sur le bouton "View Certificates". Une autre fenêtre s'ouvre. Cliquer sur l'onglet "Autorithies".

Pourquoi y a-t-il autant de certificats d'autorité ?

2 Création de certificats

Certificat autosigné (en anglais, self-signed certificate) est un certificat de clé publique qu'un utilisateur émet en son propre nom, par opposition à un certificat émis par une autorité de certification. Un tel certificat est facile à produire et ne coûte rien. Cependant, il ne fournit aucune valeur de confiance.

Parmi les certificats de la chaîne de certification vue plus haut. Quel certificat est en quelque sorte autosigné ?

2.1 Création de la requête de certification

Démarrez la machine virtuelle serveur.

`R3.06-lancement-VM serveur`

S'identifier avec le compte etu puis passer root avec la commande

`su -`

1. Générez une clé identifiant le serveur avec

```
openssl genrsa -out /etc/ssl/serveur_site1.key 4096
```

2. Générez une CSR (*Certificate Signing Request*) en utilisant la commande :

```
openssl req -new -key /etc/ssl/serveur_site1.key -out serveur_site1.req
```

Donnez les détails concernant la ville de Grenoble et choisissez iut2 comme organisation. Laissez les attributs extra vides

3. Que contient cette CSR ? A qui cette requete devrait être envoyée ? Vous pouvez vous aider de cette page digicert-easy-csr.

4. Combien coûte une certification ?
Vous pouvez vous aider de cette page digicert.

5. Comparez le coût trouvé à une solution basée sur *Let's Encrypt*. Combien de temps le certificat de *Let's Encrypt* est-il valide ?

2.2 Autocertification

Comme nous n'avons pas de budget ni de temps pour la certification, nous allons auto certifier le certificat.

1. Auto-certifiez la requete et générez le certificat avec

```
openssl x509 -req -days 30 -in serveur_site1.req
      -signkey /etc/ssl/serveur_site1.key -out /etc/ssl/serveur_site1.crt
```

2. Affichez le certificat avec

```
openssl x509 -text -noout -in /etc/ssl/serveur_site1.crt
```

3. Quelle est la période de validité du certificat ? Quel est l'agorithme employé pour signer le certificat ?

4. Etant donné que le certificat est auto-signé que risque t il de se passer avec les navigateurs voulant se connecter avec un tel certificat ?

3 Utilisation des certificats pour un site https

3.1 Paramétrage du serveur

Sur la machine virtuelle serveur.

1. activez le ssl dans apache

```
a2enmod ssl
```

2. Éditez le fichier **/etc/apache2/sites-available/site1.conf** et ajoutez entre les balises `<VirtualHost>` et `</VirtualHost>` :

```
SSLEngine On

SSLCertificateFile /etc/ssl/serveur_site1.crt

SSLCertificateKeyFile /etc/ssl/serveur_site1.key
```

3. Modifiez `<VirtualHost *:80>` par `<VirtualHost *:443>`. À quoi peut bien servir ce changement ?

4. Activez site1

```
a2ensite site1
```

puis redémarrez apache

```
systemctl restart apache2
```

3.2 Paramétrage du client

Sur la machine client (celle-ci doit être configurée en dhcp, cf TP01) .

1. avec Firefox connectez vous à

```
https://site1.jaune.fr/
```

Pourquoi Firefox vous prévient-il que le site est dangereux ?

2. Acceptez le risque et poursuivez la navigation. Puis trouvez le certificat dans la liste des certificats de Firefox. Celui-ci est un certificat de serveur (onglet "serveurs"). Affichez-le. Comment pouvez-vous constater que c'est un certificat auto signé ?
3. Quelle devrait être la démarche pour faire en sorte que Firefox accepte d'accéder au site web sans nécessité d'exception ?

4 *Handshake* TLS

Sur la VM serveur, lancer Wireshark pour analyser la connexion client-serveur.

Sur la VM client, connectez-vous de nouveau à

`https://site1.jaune.fr/`

Sur la trace Wireshark,

1. Repérez les messages HelloClient et HelloServer de la communication TLS. Quelle suite cryptographique a été négociée ?
2. Est-ce que les autres messages échangés sont lisibles ?

5 Extensions possibles

1. Paramétrez également site2 en https en utilisant la même méthode