TP3 : CERTIFICATS, TLS/SSL

1. CHIFFREMENT ASYMÉTRIQUE

1. Quelle est la version du paquet OpenSSL de votre système ? Où se trouve les certificats des différents CA (Certification Authority) sur votre environnement ?
2. Chiffrement asymétrique avec RSA :
   a. Générer un couple de clés (publique, privée) pour Alice et sauvegarder-le dans un fichier **`cle.pem`**. Quel est le codage utilisé dans ce fichier ?
   b. Comment extraire la clé publique sauvegardée dans **`cle.pem`** afin de de la stocker dans **`pub.pem`** ?
   c. Bob, qui est en procession du fichier **`pub.pem`**, veut chiffrer un message secret et l'envoyer à Alice en utilisant la clé publique d'Alice. Il envoie le à Alice le message chiffré. Quelle est la commande à utiliser ?
   d. Quelle est la commande qui permet à Alice de déchiffrer le secret en utilisant sa clé privée ?

2. LES CERTIFICATS ÉLECTRONIQUES

1. C'est quoi un certificat électronique et c'est quoi son utilité ?
2. Récupération, visualisation et transcodage de certificats
   a. Saisir le script **`get-cert.sh`** suivant :

```
#!/bin/sh
# usage: get-cert.sh remote.host.name [port]
REMHOST=$1
REMPORT=${2:-443}
echo |\
openssl s_client -connect ${REMHOST}:${REMPORT} 2>&1 |\
sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTIFICATE-/p'
```

   b. Expliquer le rôle de la commande `s_client` de la commande `openssl` et de la commande `sed`.
   c. Exécuter ce script afin de récupérer le certificat d'un site Internet de votre choix, puis visualiser ce certificat (qui est au format PEM).
   d. Convertir ce certificat du format PEM au format DER. C'est quoi la différence entre ces deux formats ? quelles sont les autres formats possibles ?
3. Création d'un certificat x509 auto-signé : créer un couple de clés publique/privée, puis créer une requête de certificat. Utiliser ensuite ces deux informations pour générer un certificat autosigné.

3. TESTER LES LIAISONS TLS/SSL

Le paquet OpenSSL embarque avec lui un serveur SSL. Nous allons l'utiliser pour comprendre le fonctionnement du protocole SSL.

1. Commencer par lancer le serveur SSL d'OpenSSL (commande `s_server`) sur le port `10000` en utilisant le certificat autosignée que vous avez généré dans l'étape précédente ainsi que la clé privée associé.

2. Lancer Wireshark pour capturer le trafic, puis tester l'accès à ce serveur à partir d'un navigateur web graphique ou bien textuel (**lynx**). Analyser tout l'échange SSL et expliquer toutes les étapes opérationnelles du protocole SSL.
3. C'est quoi la notion de suite de chiffrement présente dans le message Client Hello ?
   Est-ce que l'ordre des propositions de ces suites de chiffrement est important ?
   Afficher la liste des suites de chiffrement d'OpenSSL.
4. Afficher la trace d'une session SSL avec le client de test incorporé dans openssl.
   C'est quoi le premier message chiffré ? à quoi sert-il ? (Faire une petite recherche pour répondre à cette question)
5. En utilisant OpenSSL, accéder à un site SSL sur Internet. Quelle est la suite de chiffrement choisie ?
6. Audit d'un site SSL :
   a. Télécharger le script **`testssl.sh`**. De préférence, faites-le en le clonant à partir de **`github`** comme c'est indiqué sur le site de téléchargement.
   b. Tester la robustesse d'un serveur web en utilisant ce script
7. Installer un serveur Apache, Activer le module ssl ainsi que le site virtuel ssl par défaut. Ensuite tester sa robustesse avec le script **testssl.sh**. Faites les changements nécessaires pour durcir la configuration de votre site.