Sécurité des Réseaux - TP2

Certificats

Monroe Samuel

30 décembre 2015

1 Préambule

Ce tp a été réalisé sous environnement Mac OS X 10.10, disposant d'OpenSSL v.0.9.8zg 14 July 2015.

2 Création d'une paire de clé publique/privée

2.1 Création de la clé

Le format **PEM** est le format standard pour OpenSSL et d'autres outils SSL, c'est un format destiné à être inclus facilement dans d'autres documents ASCII ou plus riches, ce qui permet le copier-coller de son contenu d'un fichier PEM dans un autre document, et inversément.

Un rapport Européen de 2013 sur les recommandations en termes d'Algorithmes et de Longeurs de Clés stipule qu'une bonne longueur de clé RSA à moyen terme devrait être de **3096 bits**, et à long terme de **15360 bits**.
Le document est disponible via ce lien.

2.2 Visualisation de la clé

— **-in ¡arg¿** : Spécifie le fichier en input
— **-text** : Imprime la clé en texte dans la console
— **-noout** : N'imprime pas la clé en tant que telle dans la console

2.3 Chiffrement de la clé

Le nouveau fichier créé avec -des3 contient la clé chiffrée et donc illisible via un simple **cat**, il est nécéssaire de passer par la commande **rsa** pour que le prompt nous demande la clé privée (mot de passe entré au chiffrement), afin de pouvoir lire cette clé.

2.4 Extraction de la clé publique

Le cat de la clé publique donne une séquence plus petite de caractères au niveau de la console.

3 Chiffrement Déchiffrement avec RSA

La signature numérique générée avec la clé privée est bien identique avec celle obtenue via la clé publique lors de la vérification.

```
srozen@[tp2]:openssl dgst -sha1 -out hash_check message.txt
srozen@[tp2]:openssl rsautl -sign -in hash_check -inkey srozenKey.pem -out sig
srozen@[tp2]:ls
hash_check            report.tex            srozenKeyCrypto.pem
message.txt           sig                   srozenPubKey.pem
message_crypto.txt    signature.txt         tp2-certificats.pdf
message_decrypto.txt  srozenKey.pem
srozen@[tp2]:cat hash_check
SHA1(message.txt)= ea90e2ceceeb59e9271cb17d394509b20b47a5ae
srozen@[tp2]:openssl rsautl -verify -in sig -pubin -inkey srozenPubKey.pem -out
hash_checked
srozen@[tp2]:ls
hash_check            message_decrypto.txt  srozenKey.pem
hash_checked          report.tex            srozenKeyCrypto.pem
message.txt           sig                   srozenPubKey.pem
message_crypto.txt    signature.txt         tp2-certificats.pdf
srozen@[tp2]:cat hash_check
SHA1(message.txt)= ea90e2ceceeb59e9271cb17d394509b20b47a5ae
srozen@[tp2]:cat hash_checked
SHA1(message.txt)= ea90e2ceceeb59e9271cb17d394509b20b47a5ae
```

4 Création d'un Certificats

Le certificat créé comporte les informations demandées lors de sa création, plus une séquence chiffré similaire aux clés.

1. zsh

```
srozen@[certificate]:openssl genrsa -out maCle.pem 2048
Generating RSA private key, 2048 bit long modulus
.....................................................................
.....................................+++
.........................+++
e is 65537 (0x10001)
srozen@[certificate]:openssl rsa -in maCle.pem -des3 -out maCleCrypto.pem
writing RSA key
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
srozen@[certificate]:openssl rsa -in maCle.pem -pubout -out maClePub.pem
writing RSA key
srozen@[certificate]:openssl req -new -key maCle.pem -out maRequete.pem
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:BE
State or Province Name (full name) [Some-State]:Namur
Locality Name (eg, city) []:Sombreffe
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Ephec LLN
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
srozen@[certificate]:ls
maCle.pem        maCleCrypto.pem  maClePub.pem     maRequete.pem
srozen@[certificate]:openssl req -in maRequete.pem -text -noout
Certificate Request:
    Data:
        Version: 0 (0x0)
        Subject: C=BE, ST=Namur, L=Sombreffe, O=Ephec LLN
```

5 Caractéristiques du CA

Le **cat** du Certificat n'affiche qu'une séquence chiffrée de caractère, tandis que la commande **x509** de OpenSSL nous affiche les informations complètes du certificat.

Le certificat est **valide** entre le 5 Mai 2011 à 7h1m10s GMT et le 13 Juin 2015 à 7h1m10s GMT.

La clé fait 2048 bits.

```
srozen@[certificate]:
srozen@[certificate]:openssl x509 -in pereUbuCertif.pem -text -noout
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            96:d1:07:e9:07:3c:2e:e9
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=PL, ST=Province des Palotins, L=UbuPole, O=Royaume de Pologne,
 OU=Departement des Phynances, CN=Pere Ubu/emailAddress=Pere.Ubu@palotin.pl
        Validity
            Not Before: May  5 07:01:10 2011 GMT
            Not After : Jun 13 07:01:10 2015 GMT
        Subject: C=PL, ST=Province des Palotins, L=UbuPole, O=Royaume de Pologne
, OU=Departement des Phynances, CN=Pere Ubu/emailAddress=Pere.Ubu@palotin.pl
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:b4:6f:d4:b3:9e:71:3e:1f:34:fa:fb:a8:eb:18:
                    1a:f3:1f:97:fe:46:68:82:a1:74:44:c9:a6:f9:7b:
```

6 Signature du CA par le certificat

Notre certificat possède bien le CA du pereUbu en plus des options souhaitées, telles la validité 10 jours.

```
maCleCrypto.pem    monCertif.pem      usr.ext
srozen@[certificate]:openssl x509 -in monCertif.pem -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 163 (0xa3)
        Signature Algorithm: sha1WithRSAEncryption
        Issuer: C=PL, ST=Province des Palotins, L=UbuPole, O=Royaume de Pologne,
 OU=Departement des Phynances, CN=Pere Ubu/emailAddress=Pere.Ubu@palotin.pl
        Validity
            Not Before: Sep 26 09:39:17 2015 GMT
            Not After : Oct  6 09:39:17 2015 GMT
        Subject: C=BE, ST=Namur, L=Sombreffe, O=Ephec LLN
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (2048 bit)
                Modulus (2048 bit):
                    00:bc:bd:85:34:a9:e9:95:33:2e:9c:df:65:b2:0f:
                    2f:2b:50:9d:07:ab:c9:65:35:71:98:1a:2b:a7:c1:
                    5e:b3:6c:7f:59:b4:40:52:f9:77:77:ac:9e:b6:28:
                    26:e8:6f:0f:ee:7f:ac:36:52:8f:a1:6f:85:e0:a6:
                    78:53:b7:70:e4:b0:f2:9d:ab:ee:3f:b3:a0:f4:01:
```

7 Vérification du certificat

Un erreur est présente dans la vérification de la validité, la date de validité du CA est dépassée.

```
srozen@[certificate]:openssl verify -CAfile pereUbuCertif.pem monCertif.pem
monCertif.pem: /C=PL/ST=Province des Palotins/L=UbuPole/O=Royaume de Pologne/OU=
Departement des Phynances/CN=Pere Ubu/emailAddress=Pere.Ubu@palotin.pl
error 10 at 1 depth lookup:certificate has expired
OK
```

8 Pour aller plus loin

Le certificat va être utile au site Web pour prouver notre identité et éviter par exemple un risque de phishing au client, ainsi que d'assurer une confidentialité aux données qui transitent via SSL.
La présence de ce certificat établi également une relation de confiance vis-à-vis du client, dans un cadre de vente en ligne, une plus grande confiance du client pourra amener plus d'achats par exemple.

8.1 Mettre en place le certificat sur Apache

Il faut d'abord activer le module SSL sur apache.
Ensuite, faire en sorte qu'Apache écoute sur le port 443 (HTTPS).

Il faut également indiquer au serveur Web où trouver nos certificats.

8.2 Faire accepter le certificat par les cliens Webs

Le certificat doit être reconnu par des autorités de certifications reconnues.