heig-vd
Haute Ecole d'Ingénierie et de Gestion
du Canton de Vaud

Applications et Services Internet

SSL - JSSE (Java Secure Socket Extension)

Rapport de laboratoire

Romain de Wolff

IL2008

19 janvier 2008

Table des matières

1 Introduction

Le but de ce laboratoire est de modifier le serveur web créé en Java durant un laboratoire précédant et d'y implémenter le protocole SSL/TLS. Notre serveur web doit se lancer dans deux mode différent : un mode avec authentification du client nécessaire et un autre sans.

2 Utilisation du logiciel

Le lancement du serveur s'effectue à l'aide de la ligne de commande. Pour lancer le serveur avec authentification du client sur le port 443 (port par défaut de SSL) il faut utiliser la commande suivante :

```
java WebServer 1 443
```

Le "1" permet de dire que l'on activer l'authentification du client. Pour le rendre facultatif, on utilisera la commande suivante :

```
java WebServer 0 443
```

Notons que pour lancer le serveur sur le port 443 comme montré ci dessus il faut exécuter la commande en administrateur.

3 Clé publique générée

Voici la clé publique qui nous a été generée par Keytool :

```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBujCCASMCAQAwejELMAkGA1UEBhMCQ0gxCzAJBgNVBAgTA1ZEMREwDwYDVQQHEwhMYXVzYW5u
ZTEZMBcGA1UEChMQd3d3L1lRBR0FEQVJULmNvbTEcMBoGA1UECxMTUGVyc29uYWwgV2ViIFNlcnZl
cjESMBAGA1UEAxMJMTI3LjAuMC4xMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCjcLrCVl/h
50CuSHjNevhTrRS0bCQ1oCN27c3hTLdDbLVjDNqUJqziTXpowFTUXmM/hrbKwVzM5+I4krwx/6dW
oVVhaGywxkQwN4mQ2rgFvkdm8xIpPKfyVMTLYRQLfd89qLYC8C0SUR3MqzuNRpT7lnla1RB9A6Mg
IIx53mf8UQIDAQABoAAwDQYJKoZIhvcNAQEEBQADgYEAAht/hvwHdTlQb5ZPe6EmBbMJe6VozqQT
yzaA2q6+4Y+FzuQ0PT7oePyg22e6HTiEtxRIhNGiCXVlceeNxKYFBGoBGVSGOHvauWFGRntErntQ
X7vYKW5XCjHfEpsMwKsj42b4zMFn743IT/LmiC/NsghW3q+UD7AUslld4+XaX68=
-----END NEW CERTIFICATE REQUEST-----
```

4 Réponses aux questions

Lors de la connexion sur notre serveur HTTPS à l'aide du navigateur Firefox, le serveur nous affiche une alerte comme le montre la figure 1

Nous acceptons ce certificat et nous allons voir le site s'afficher. On remarque que le site est sécurisé grâce à l'icone représentant un cadenas (an bas à droite dans Firefox) que l'on peut voire sur la figure 2.

En cliquant sur le cadenas on peut afficher les informations relatives à la sécurité et donc du certificat que l'on a accepté. La figure 3 nous montre à quoi ressemble cette fenêtre.

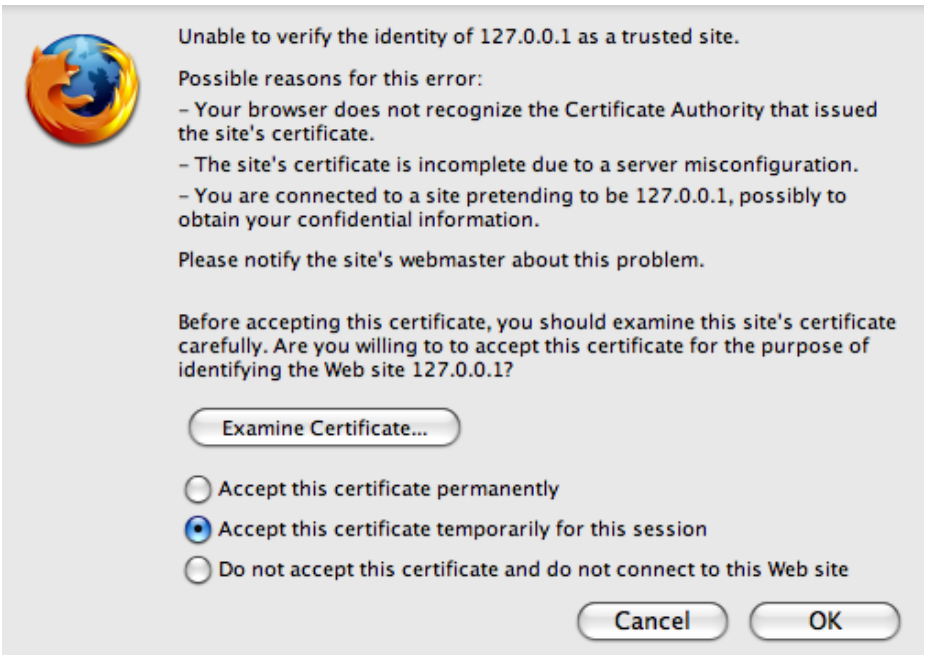

FIG. 1 – Alerte affichée lors de la connexion sur le site sécurisé.

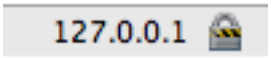

FIG. 2 – Icone dans la barre des tâches du navigateur Firefox.

Fig. 3 – Information sur la sécurité du site.

La figure 3 nous montre les informations sur le certificat et nous dit que nous faisons confiance au CA mentionné. En cliquant sur le bouton “View” on obtient les informations sur le certificat, exactement les information que nous avons introduites lors de la création à l’aide de *Keytool*. La figure 4 nous montre cette fenêtre.

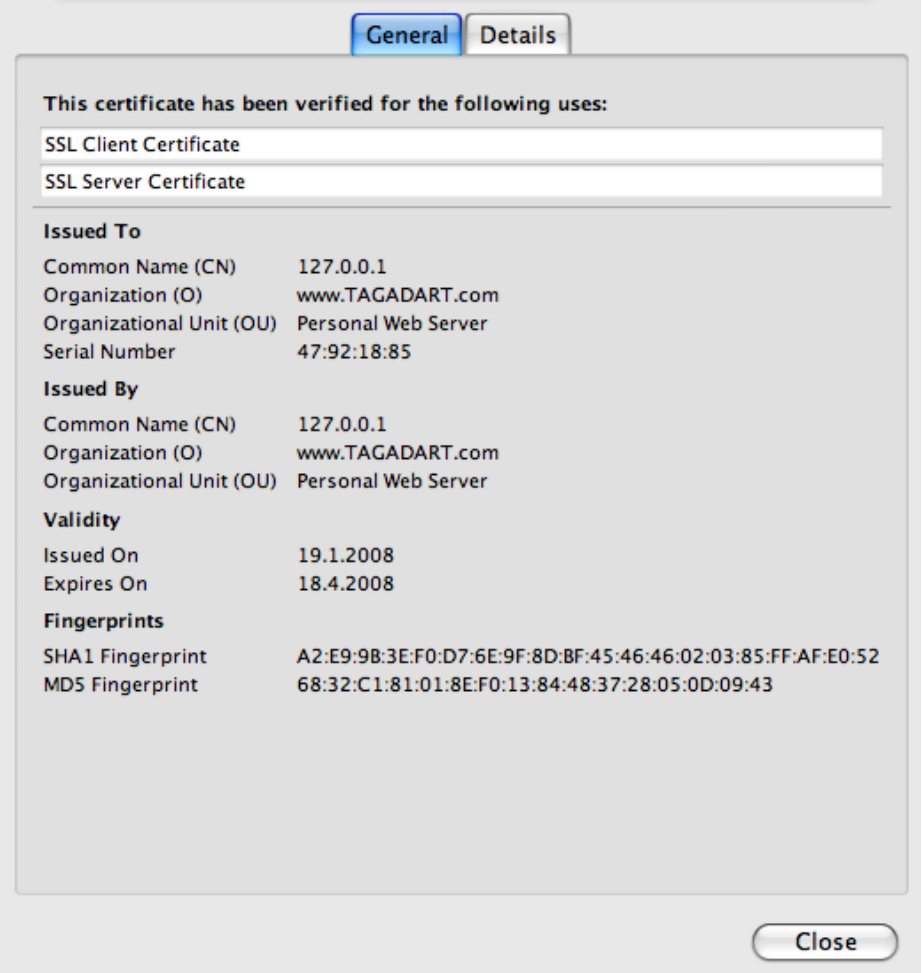

Fig. 4 – Affichage détaillée des informations sur le certificats que nous avons créé.

4.1 En examinant le certificat du serveur, comment pouvez-vous en déduire que celui-ci est auto-signé ?

On le voit bien sur la figure 4 : les champs “Issued To” et “Issued By” sont identiques. On sait dès lors que le certificat est auto-signé.

4.2 Pourquoi serait-il intéressant de faire signer le certificat par une entité comme Verisign ?

Verisign est une entreprise de type CA : elle émet des certificats qu'elle vends. Ces certificats sont réputés fiable. L'avantage d'avoir un certificat d'un CA reconnue est que les utilisateurs qui se connecte sur le site peuvent, grâce à la renomée de Verisign, savoir que le site utilise une encryption de qualité. C'est donc pour des questions de sécurité et de véracité que nous avons avantage à utiliser les services offert par une société comme Verisign.

L'utilisateur sera donc mis en confiance et n'aura plus d'avertissement du navigateur comme quoi le certificat est douteux.

5 Conclusion

6 Références

http://java.sun.com/javase/6/docs/technotes/guides/security/jsse/JSSERefGuide.html Java Secure Socket Extension (JSSE) - Reference Guide, Sun.com

http://java.sun.com/javase/technologies/security/ Java SE Security, Sun.com

http://fr.wikipedia.org/wiki/Transport_Layer_Security TLS et SSL, Wikipedia.org

http://www.javaworld.com/javatips/jw-javatip115.html Secure JavaMail with SSL, JavaWorld.com