LECTRA FASHION PLM

Guide de configuration du mode HTTPS

Date : Avril 2017

1 APERÇU GENERAL

1.1 Portée de ce document

Ce document s'adresse à l'administrateur du Lectra Fashion PLM et décrit la procédure à suivre pour configurer le PLM en mode sécurisé.

Ce document ne décrit pas le processus d'installation de la solution.

1.2 Versions cibles

Ce document est uniquement applicable à partir de la version V5R1 du Lectra Fashion PLM.

C'est à partir de cette version que l'ensemble des composants se connectant au serveur ont été validés pour fonctionner en https.

1.3 Définition des termes utilisés

Les termes suivants sont utilisés dans ce document :

HTTP : HyperText Transfer Protocol

HTTPS : HyperText Transfer Protocol Secure

AJP : Apache JServ Protocol

Certificat : fichier de données utilisé pour identifier, authentifier et chiffrer des échanges.

2 INTRODUCTION

Pour rappel, le point d'entrée du PLM est le serveur HTTP Apache installé en frontal de serveurs d'applications Widfly (anciennement appelé JBoss).

Sécuriser l'accès au PLM correspond à activer le protocole https sur le serveur Apache.

C'est grâce à un certificat que la confidentialité et l'intégrité des données envoyées par l'utilisateur et reçues du serveur peuvent être garanties.

L'activation du protocole https passe donc par l'obtention d'un certificat et par la configuration du serveur Apache.

3 MISE EN PLACE D'UN CERTIFICAT

Par défaut un certificat auto-signé par Lectra est généré lors de l'installation du PLM.

L'utilisation d'un certificat délivré par une autorité de confiance est fortement recommandée (certificat acheté auprès d'un fournisseur externe).

Dans le répertoire d'installation du serveur Apache, modifier le fichier Apache24\conf\httpd-ssl.conf pour prendre en compte le certificat de confiance et sa clé privée et désactiver le certificat généré:

- ajouter la directive de configuration suivante:
  `SSLCertificateFile "${SRVROOT}/conf/<certificate_filename>.cer"`
- commenter la directive de configuration suivante:
  `SSLCertificateFile "${SRVROOT}/conf/autosigned-certificate.crt"`

```
#   Server Certificate:
#   Point SSLCertificateFile at a PEM encoded certificate.  If
#   the certificate is encrypted, then you will be prompted for a
#   pass phrase.  Note that a kill -HUP will prompt again.  Keep
#   in mind that if you have both an RSA and a DSA certificate you
#   can configure both in parallel (to also allow the use of DSA
#   ciphers, etc.)
#   Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
#   require an ECC certificate which can also be configured in
#   parallel.
SSLCertificateFile "${SRVROOT}/conf/official.cer"
#SSLCertificateFile "${SRVROOT}/conf/autosigned-certificate.crt"
```

- ajouter la directive de configuration suivante:
  `SSLCertificateKeyFile "${SRVROOT}/conf/<certificate_key_filename>.key"`
- commenter la directive de configuration suivante:
  `SSLCertificateFile "${SRVROOT}/conf/autogenerated-key.key"`

```
#   Server Private Key:
#   If the key is not combined with the certificate, use this
#   directive to point at the key file.  Keep in mind that if
#   you've both a RSA and a DSA private key you can configure
#   both in parallel (to also allow the use of DSA ciphers, etc.)
#   ECC keys, when in use, can also be configured in parallel
SSLCertificateKeyFile "${SRVROOT}/conf/official.key"
# SSLCertificateKeyFile "${SRVROOT}/conf/autogenerated-key.key"
```

Dans le cas d'une chaine de certificats, renseigner la ligne :
- `SSLCertificateChainFile "${SRVROOT}/conf/<chaincertificate_filename>.crt"`

```
#   Server Certificate Chain:
#   Point SSLCertificateChainFile at a file containing the
#   concatenation of PEM encoded CA certificates which form the
#   certificate chain for the server certificate. Alternatively
#   the referenced file can be the same as SSLCertificateFile
#   when the CA certificates are directly appended to the server
#   certificate for convenience.
```

```
SSLCertificateChainFile "${SRVROOT}/conf/officiel-chain.crt"
```

4 GESTION DU PROTOCOLE

Afin de garantir la sécurité des données transmises, il est recommandé de désactiver le protocole http et de ne laisser activer que le protocole https. Par défaut à l'installation, les deux protocoles sont actifs.

4.1 Désactivation du protocole http

Par défaut, l'installation d'Apache est configurée pour écouter sur le port 80 et communiquer via le protocole http.

Cette configuration doit être désactivée pour ne laisser active que la communication via le protocole https.

Dans le répertoire d'installation du serveur Apache, modifier le fichier Apache24\conf\httpd.conf en commentant la directive de configuration suivante: « Listen 80 ».

```
# Change this to Listen on specific IP addresses as shown below to
# prevent Apache from glomming onto all bound IP addresses.
#
#Listen 80
```

4.2 Activation du protocole https

Par défaut, l'installation d'Apache est configurée pour écouter sur le port 443 et communiquer via le protocole https.

Dans le répertoire d'installation du serveur Apache, vérifiez

- que la directive de configuration suivante « Listen 443 » est bien présente dans le fichier Apache24\conf\extra\httpd-ssl.conf.

```
#
# When we also provide SSL we have to listen to the
# standard HTTP port (see above) and to the HTTPS port
# Note: Configurations that use IPv6 but not IPv4-mapped addresses need
two
# Listen directives: "Listen [::]:443" and "Listen 0.0.0.0:443"
#
Listen 443
```

- que la directive de configuration suivante « LoadModule ssl_module modules/mod_ssl.so » est bien présente dans le fichier Apache24\conf\httpd.conf.

```
#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a
DSO you
# have to place corresponding `LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are
used.
# Statically compiled modules (those listed by `httpd -l') do not need
```

```
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
#LoadModule access_compat_module modules/mod_access_compat.so
LoadModule ssl_module modules/mod_ssl.so
```

5 CONFIGURATIONS SUPPLEMENTAIRES

5.1 Widfly (anciennement appelé JBoss)

La communication entre le serveur web frontal Apache et les serveurs applicatifs Widfly se fait via le protocole Ajp. Pour garantir que le(s) serveur(s) Widfly ne soit pas directement accessible en http, il est fortement recommandé de bloquer aux connections externes via le firewall Windows leur port d'écoute (généralement le port 8080).

Démarrer, s'il ne l'est déjà pas, le firewall Windows depuis le panneau de configuration du serveur. Puis lancer le firewall Windows avec fonctions avancées de sécurité et créer une nouvelle règle de trafic entrant avec les renseignements suivants :

General	
Action	Block the connection
Protocols and Ports	
Protocol type	TCP
Local port	Specific Ports
	8080
Remote port	All Ports
Scope	
Local IP address	Any IP address
Remote IP address	Any IP address
Advanced	
Profiles	Domain, Private, Public