LECTRA FASHION PLM

HTTPS mode configuration guide

Date: April 2017

1 GENERAL OVERVIEW

1.1 Scope of this document

This document is for the Lectra Fashion PLM administrator and describes the procedure to follow to configure the PLM in secure mode.

This document does not describe the solution installation process.

1.2 Target versions

This document only applies to Lectra Fashion PLM V5R1 onwards.

It is from this version that all the components connected to the server are validated to work in HTTPS.

1.3 Definition of the terms used

The following terms are used in this document:

HTTP: HyperText Transfer Protocol

HTTPS: HyperText Transfer Protocol Secure

AJP: Apache JServ Protocol

Certificate: data file used to identify, authenticate and cost the exchanges.

2 INTRODUCTION

As a reminder, the PLM entry point is the HTTP Apache server installed at the front of the Widfly application servers (previously called JBoss).

Access to the PLM can be secured by activating the HTTPS protocol on the Apache server.

The confidentiality and integrity of the data sent by the user and received by the server can be guaranteed with a certificate.

The HTTPS protocol is activated by obtaining a certificate and by the configuration of the Apache server.

3 INSTALLING A CERTIFICATE

By default a Lectra self-signed certificate is generated with the installation of the PLM.

It is strongly recommended to use a a certificate delivered by a trusted authority (certificate purchased from an external supplier).

In the Apache server installation directory, modify the Apache24\conf\httpd-ssl.conf file to take the trusted certificate and private key into account and deactivate the generated certificate:

- add the following configuration directive:
  `SSLCertificateFile "${SRVROOT}/conf/<certificate_filename>.cer"`
- comment the following configuration directive:
  `SSLCertificateFile "${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"
```

- add the following configuration directive:
  `SSLCertificateKeyFile "${SRVROOT}/conf/<certificate_key_filename>.key"`
- comment the following configuration directive:
  `SSLCertificateFile "${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"
```

In the case of a certificate chain file, enter the line:

- `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 PROTOCOL MANAGEMENT

To maintain the security of the transmitted data, it is recommended to deactivate the HTTP protocol and to only let the HTTPS protocol activate. The two protocols are active by default at installation.

4.1 Deactivation of the HTTP protocol

By default, the Apache installation is configured to listen on port 80 and to communicate via the HTTP protocol.

This configuration must be deactivated to only leave the communication via HTTPS protocol active.

In the Apache server installation directory, modify the Apache24\conf\httpd.conf and comment the following configuration directive: « 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 HTTPS protocol activation

By default, the Apache installation is configured to listen on port 443 and communicate via the HTTPS protocol.

In the Apache server installation directory, check

- that the 'Listen 443' configuration directive is present in the Apache24\conf\extra\httpd-ssl.conf file.

```
#
# 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
```

- that the 'LoadModule ssl_module modules/mod_ssl.so' configuration directory is present in the Apache24\conf\httpd.conf file.

```
#
# 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 ADDITIONAL CONFIGURATIONS

5.1 Widfly (previously called JBoss)

The communication between the Apache front web server and the Widfly application servers is made via the Ajp protocol. To guarantee that the Widfly server(s) is not directly accessible in HTTP, it is strongly recommended to block external connections via the Windows firewall listening port (generally the 8080 port).

Start the Windows firewall (if not already started) from the server configuration panel. Then launch the Windows firewall with advanced security functions and create a new traffic rule entering the following information.

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