1. [How to configure https on Jboss](#configure_https_jboss)
2. [Single sign on](#single_sign_on)
3. [Enable SSL(or https)](#Enable_SSL_JBOSS)
4. [Redirect from http to https on jboss](#redirect_http_https)

JBoss guide: How to enable SSL (HTTPS) on JBoss, as well as other “nice-to-know” configurations

<https://roneiv.wordpress.com/2008/01/03/jboss-tutorial-how-to-enable-ssl-https-on-jboss-as-well-as-other-nice-to-know-configurations/>

JBoss guide: How to enable SSL (HTTPS) on JBoss, as well as other “nice-to-know” configurations

03.01.2008 — roneiv

From time to time you might have the need of running a web application over https, or there can be requests of having Single-Sign-On between multiple web applications running on your server. This small jboss-guide will give you some clues on how to solve tasks like this, with configurations for both jboss-4.0.4.GA and jboss-4.2.2.GA. Since the name of the server instance might differ and it’s also possible to use custom names, I’ll refer to it as jboss/server/<NAME>/, but what I mean here is for example jboss/server/default/.

Changing the port that jboss runs on

For 4.0.4 you should locate the **server.xml** inside jboss/server/<NAME>/deploy/jbossweb-tomcat55.sar/, and then change the port=”8080″ parameter in the HTTP Connector to your wishes, for example port 80 as I have done it here.

In Jboss eap 4.1 it is C:\jboss\jboss-eap\jboss-as\server\default\deploy\jbossweb.sar

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <!-- A HTTP/1.1 Connector on port 8080 -->       <Connector port="80" address="${jboss.bind.address}"          maxThreads="250" strategy="ms" maxHttpHeaderSize="8192"          emptySessionPath="true"          enableLookups="false" redirectPort="8443" acceptCount="100"          connectionTimeout="20000" disableUploadTimeout="true"/> |

For 4.2.2 you do exactly the same, but the server.xml is located inside jboss/server/<NAME>/deploy/jboss-web.deployer/ instead.

Make tomcat able to compile java5 – by default it doesn’t

If you have the need of using java5 (jdk 1.5), you need to set the source-level of the compiler. If you don’t do this and have deployed web-applications with java5 code, you will get exceptions during startup. For 4.0.4 edit the **web.xml** in jboss/server/<NAME>/deploy/jbossweb-tomcat55.sar/conf. Locate the jsp servlet by searching for <servlet-name>jsp</servlet-name>, and uncomment the section that enables jdk1.5 features:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <!-- Uncomment to use jdk1.5 features in jsp pages -->      <init-param>         <param-name>compilerSourceVM</param-name>         <param-value>1.5</param-value>      </init-param> |

For 4.2.2 you find the **web.xml** inside jboss/server/<NAME>/deploy/jbossweb-deployer/conf. Locate the same servlet, and make sure that the parameters both for source & target compiler are set like this:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <servlet>      <servlet-name>jsp</servlet-name>      <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>      <init-param>          <param-name>fork</param-name>          <param-value>false</param-value>      </init-param>      <init-param>          <param-name>xpoweredBy</param-name>          <param-value>false</param-value>      </init-param>      <init-param>       <param-name>compilerSourceVM</param-name>       <param-value>1.5</param-value>      </init-param>  <init-param>       <param-name>compilerTargetVM</param-name>       <param-value>1.5</param-value>      </init-param> |

Activate support for Single Sign-On

For both versions, locate **server.xml** (4.0.4 = jboss/server/<NAME>/deploy/jbossweb-tomcat55.sar/, 4.2.2 = jboss/server/<NAME>/deploy/jboss-web.deployer/). Find the “Host” section, and uncomment the following Valve:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <!-- Uncomment to enable single sign-on across web apps             deployed to this host. Does not provide SSO across a cluster.               If this valve is used, do not use the JBoss ClusteredSingleSignOn             valve shown below.               A new configuration attribute is available beginning with             release 4.0.4:               cookieDomain  configures the domain to which the SSO cookie                           will be scoped (i.e. the set of hosts to                           which the cookie will be presented).  By default                           the cookie is scoped to "/", meaning the host                           that presented it.  Set cookieDomain to a                           wider domain (e.g. "xyz.com") to allow an SSO                           to span more than one hostname.          -->           <Valve className="org.apache.catalina.authenticator.SingleSignOn" /> |

Then in your jboss-web.xml it’s important that all the web applications that are going to “exchange” credentials points to the same security-domain:

|  |  |
| --- | --- |
| 1  2  3  4 | <jboss-web>      <security-domain>java:/jaas/USE\_THE\_SAME\_APPLICATION\_POLICY\_HERE</security-domain>      <context-root>/YOUR\_APPLICATION\_ROOT</context-root>  </jboss-web> |

If you now open and logon to one application, going to another one running on the same server should not prompt you for username/password again. Note that there are alternatives also if you have applications running on different servers/locations – check the other Valves.

Enable SSL on JBoss

In this example I’m only using a self-signed certificate, but the procedure would be more or less the same even if you are going to use a certificate from a Certification Authority.

1. Generate the keystore with the following command

|  |  |
| --- | --- |
| 1 | keytool -genkey -alias tomcat -keyalg RSA -keystore NAME\_OF\_KEYSTORE -validity NUMBER\_OF\_DAYS |

1. Copy the file into the jboss/server/<NAME>/conf/ directory
2. Edit the **server.xml** (4.0.4 = jboss/server/<NAME>/deploy/jbossweb-tomcat55.sar/, 4.2.2 = jboss/server/<NAME>/deploy/jboss-web.deployer/).For 4.0.4 the SSL-connector should be configured like:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <!-- SSL/TLS Connector configuration using the admin devl guide keystore     -->       <Connector port="THE\_PORT\_YOU\_LIKE" address="${jboss.bind.address}"            maxThreads="100" strategy="ms" maxHttpHeaderSize="8192"            emptySessionPath="true"            scheme="https" secure="true" clientAuth="false"            keystoreFile="${jboss.server.home.dir}/conf/THE\_KEYSTORE\_NAME"            keystorePass="PASSWORD\_FOR\_THE\_KEYSTORE" sslProtocol = "TLS" /> |

1. For 4.2.2, configure it like this:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | <Connector port="THE\_PORT\_YOU\_LIKE" protocol="HTTP/1.1" SSLEnabled="true"                maxThreads="150" scheme="https" secure="true"                clientAuth="false"            strategy="ms"                address="${jboss.bind.address}"                keystoreFile="${jboss.server.home.dir}/conf/THE\_KEYSTORE\_NAME"                keystorePass="PASSWORD\_FOR\_THE\_KEYSTORE"                truststoreFile="${jboss.server.home.dir}/conf/THE\_KEYSTORE\_NAME"                truststorePass="PASSWORD\_FOR\_THE\_KEYSTORE"                sslProtocol="TLS"/> |

1. Now you should be able to access your application through https. Remember to use https:// instead of http:// in your browser-url, or else it will fail.
2. Remember that if you want to disable the non-secured port 8080 (or custom), making sure that people can only access through https, comment and disable that connector in the same server.xml.

Tell jboss 4.2.2 to not use the bundled JSF 1.2 implementation

By default this version of jboss comes bundled with the Glassfish JSF 1.2 implementation. If you deploy web applications that use other implementations, like myfaces, you should tell jboss to use the implementations(s) deployed together with the web applications instead. Do this by adding the following to the web.xml of your application(s):

|  |  |
| --- | --- |
| 1  2  3  4 | <context-param>     <param-name>org.jboss.jbossfaces.WAR\_BUNDLES\_JSF\_IMPL</param-name>     <param-value>true</param-value>  </context-param> |

Access jboss-4.2.2GA using ip address instead of localhost – use the “-b” parameter

I’ve been using the 4.0.4 version for some time, and I could start it on my machine (accessing it through localhost:8080), and access it from other machines in my network it by using the ip-address of my machine instead of localhost. With 4.2.2, you can start it the same way and it will work from your machine by going localhost. But trying to start the application remotely from other machines by using the ip-address would fail, giving you a 404.

This is because before 4.2.2.GA, jboss was always bound to the any address “0.0.0.0”. But this was considered a security issue, and this default behavior was removed. It’s now up to the user to explicitly configure this.

What you need to do to solve it is to start the jboss with another parameter, you need to set the bind address for the jboss services. The following command would start a jboss server named “myserver” on ip 192.168.100.100:

|  |  |
| --- | --- |
| 1 | run.bat -c myserver -b 192.168.100.100 |

If you now try to start the application from other machines by using this IP it works! If you use the server named default you can leave out the -c parameter. It’s also possible to revert back to the “old configuration” by using -b 0.0.0.0, but this is not recommended.Ok, I hope that this small guide might be of help to someone! :)

# [Redirecting from non ssl port 8080 to ssl port 8443](http://stackoverflow.com/questions/9526425/redirecting-from-non-ssl-port-8080-to-ssl-port-8443)

<http://stackoverflow.com/questions/9526425/redirecting-from-non-ssl-port-8080-to-ssl-port-8443>

have determined that setting NONE to CONFIDENTIAL in the following will force the redirect from 8080 to 8443

|  |
| --- |
| I am trying to redirect the traffic on non-ssl port 8080 to ssl port 8443( on Jboss 4.2.3.GA version), but its not working. when I access my webapplication on this port it stays on that port and the page gets displayed. Here is my configuration in Server.xml file  <Connector port="8080" address="${jboss.bind.address}"  maxThreads="250" maxHttpHeaderSize="8192"  emptySessionPath="true" protocol="HTTP/1.1"  enableLookups="false" redirectPort="8443" acceptCount="100"  connectionTimeout="20000" disableUploadTimeout="true"/>  <!-- Define a SSL HTTP/1.1 Connector on port 8443  This connector uses the JSSE configuration, when using APR, the  connector should be using the OpenSSL style configuration  described in the APR documentation -->  <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  maxThreads="150" scheme="https" secure="true"  clientAuth="false" sslProtocol="TLS" keystoreFile="conf/sds/keystore"/>  and here is web.xml configuration  <security-constraint>  <web-resource-collection>  <web-resource-name>SUCTR</web-resource-name>  <url-pattern>/\*</url-pattern>  </web-resource-collection>  <user-data-constraint>  <transport-guarantee>CONFIDENTIAL</transport-guarantee>  </user-data-constraint>  I have tried using default port 80 and 443 and also using the specific path in the url-pattern but still its not working. I am not sure what is it i am doing wrong here, can you please point me in the right direction.  thanks.  [redirect](http://stackoverflow.com/questions/tagged/redirect) [ssl](http://stackoverflow.com/questions/tagged/ssl) [jboss](http://stackoverflow.com/questions/tagged/jboss) |
|  | add a comment |

## 2 Answers

[active](http://stackoverflow.com/questions/9526425/redirecting-from-non-ssl-port-8080-to-ssl-port-8443?answertab=active#tab-top)[oldest](http://stackoverflow.com/questions/9526425/redirecting-from-non-ssl-port-8080-to-ssl-port-8443?answertab=oldest#tab-top)[votes](http://stackoverflow.com/questions/9526425/redirecting-from-non-ssl-port-8080-to-ssl-port-8443?answertab=votes#tab-top)

|  |  |
| --- | --- |
| up vote3down vote | edit in web.xml  <security-constraint>  <web-resource-collection>  <web-resource-name>App\_nmae</web-resource-name>  <url-pattern>/\*</url-pattern>  <http-method>GET</http-method>  <http-method>POST</http-method>  </web-resource-collection>  <user-data-constraint>  <transport-guarantee>CONFIDENTIAL</transport-guarantee>  </user-data-constraint>  </security-constraint>  edit in sever.xml  <Connector port="443" protocol="HTTP/1.1" SSLEnabled="true"  maxThreads="150" scheme="https" secure="true"  clientAuth="false" sslProtocol="TLS"  keystoreFile="/opt/apache-tomcat-6.0.13/.keystore"  keystorePass="changeit"/>  it is working for me ..you can try it |