# Technical White Paper - JBoss Security

## Securing JMX

1.0

# Table of Contents

# Target Audience

This technical white paper on Securing JMX is intended for system administrators and developers who install JBoss in production environments.

# Preface

The Technical White Papers from JBoss Security are important sources of information for secure operation of JBoss Products as well as applications running on them.

If you have questions, please feel free to contact the JBoss Security team at the afore mentioned URL.

# 1

# Secure JMX Console (Authentication Only)

## 1.1. About the JMX Console

The jmx-console is the default console that is available with the JBoss Application Server. It displays the various MBean Services that are running in a JBoss Application Server instance. A user is able to get and set attributes and invoke operations on the various services. For more information, please refer to the 'JBoss Application Server User or Administrator's Guide'. The following wiki page has a good description of the JMX Console. `http://wiki.jboss.org/wiki/Wiki.jsp?page=JMXConsole`

## 1.2. Simple Security for the JMX Console

If you want to have simple secured jmx console where the user/password and user/roles come from properties files, then you can follow the following steps:

1. Locate the jmx-console.war directory in the deploy directory of your server configuration. If you are just using the default configuration, then it will be under JBOSS_DIR/server/default/deploy directory and you are using the clustered configuration, then it will be under JBOSS_DIR/server/all/deploy directory.

2. Now edit the web.xml file under jmx-console.war/WEB-INF directory and uncomment the security constraint block as shown below:

```
   <!-- A security constraint that restricts access to the HTML JMX console
   to users with the role JBossAdmin. Edit the roles to what you want and
   uncomment the WEB-INF/jboss-web.xml/security-domain element to enable
   secured access to the HTML JMX console. -->
   <security-constraint>
     <web-resource-collection>
       <web-resource-name>HtmlAdaptor</web-resource-name>
       <description>An example security config that only allows users with the
         role JBossAdmin to access the HTML JMX console web application
       </description>
       <url-pattern>/*</url-pattern>
       <http-method>GET</http-method>
       <http-method>POST</http-method>
     </web-resource-collection>
     <auth-constraint>
       <role-name>JBossAdmin</role-name>
     </auth-constraint>
   </security-constraint>
```

   Edit the jboss-web.xml file also and uncomment the security-domain element as shown below:

```
<jboss-web>
```

```
   <!-- Uncomment the security-domain to enable security. You will
      need to edit the htmladaptor login configuration to setup the
      login modules used to authentication users.  -->
      <security-domain>java:/jaas/jmx-console</security-domain>
</jboss-web>
```

3. Now locate the two properties files called as jmx-console-users.properties and jmx-console-roles.properties that will be in the conf directory of your server configuration under the props sub-directory('default', 'all' or 'custom'). An example of the location will be /server/default/conf/props.

4. In the jmx-console-users.properties, you can add/change the user/password combination.

5. In the jmx-console-roles.properties, you will need to assign roles to the users you added or changed in step.4. Just remember to add JBossAdmin role to the users who will be using the jmx-console.

6. Now when you start JBoss and try to access the jmx-console, you should see a pop up appear that will ask you to enter the username and password. You can use one of the users/password combination that you configured in steps 4 and 5.

# 2

# Secure JMX Console (Access Control)

## 2.1. Need for Access Control on the JMX Console

The previous chapter talked about securing the jmx console. The security provided there applied to the entire console with no controls on what an user can do with reference to the various JMX operations possible on the console.

## 2.2. Details

You will need to follow the following steps to enable access control on the jmx console.

1. Perform all the steps outlined in the earlier chapter to secure the jmx-console.

   Edit the web.xml file of deploy/jmx-console.war/WEB-INF in the server configuration you are using (default, all, custom etc). You will need to uncomment the filter settings as shown here:

```
<!-- -->
       <filter>
         <filter-name>JmxOpsAccessControlFilter</filter-name>
         <filter-class>org.jboss.jmx.adaptor.html.JMXOpsAccessControlFilter</filter-class>
         <init-param>
           <param-name>updateAttributes</param-name>
           <param-value>UpdateAttributeRole</param-value>
           <description>Comma-delimited Roles that define the JMX Operation denoting updation of Attribut
         </init-param>
         <init-param>
           <param-name>invokeOp</param-name>
           <param-value>InvokeOpRole</param-value>
           <description>Comma-delimited Roles that define the JMX Operation denoting Invocation of Operat
         </init-param>
      </filter>
      <filter-mapping>
         <filter-name>JmxOpsAccessControlFilter</filter-name>
         <servlet-name>HtmlAdaptor</servlet-name>
      </filter-mapping>
```

2. Now if an user is allowed to click the 'invoke' buttons on the various MBean services in the jmx console (action will invoke operations), then the user needs to have 'InvokeOpRole'. If the user is allowed to click the 'Apply Changes' button(action will update the jmx attributes of the service), then the user needs to have 'updateAttributeRole'.

   For this to apply, you will need to update the jmx-console-roles.properties file. An example is shown below:

```
# A sample roles.properties file for use with the UsersRolesLoginModule
 admin=JBossAdmin,HttpInvoker,UpdateAttributeRole
```

```
admin2=JBossAdmin,HttpInvoker,InvokeOpRole
```

## 2.3. Reference

http://wiki.jboss.org/wiki/Wiki.jsp?page=AccessControlForJMXConsole

# 3

# Secure the JMX Invokers (Authentication Only)

## 3.1. Introduction

The JMX invokers are the entry points to the MBean Server. If there is a need to restrict external access to the MBean Server, then there is a need to secure the invokers.

## 3.2. Enable Authentication for the JMX Invoker

The JMX invoker can be made to authenticate against the JBoss Security JAAS framework like the web or ejb layers.

### 3.2.1. Modifications Required

You will need to edit the jmx-invoker-service.xml file under the deploy directory of your server configuration (eg: default configuration). Please uncomment the following section.

```
<operation>
   <description>The detached invoker entry point</description>
   <name>invoke</name>
   <parameter>
       <description>The method invocation context</description>
       <name>invocation</name>
       <type>org.jboss.invocation.Invocation</type>
   </parameter>
   <return-type>java.lang.Object</return-type>
   <!-- Uncomment to require authenticated users -->
   <descriptors>
     <interceptors>
         <interceptor code="org.jboss.jmx.connector.invoker.AuthenticationInterceptor"
             securityDomain="java:/jaas/jmx-console"/>
         </interceptors>
   </descriptors>
</operation>
```

The value of the security domain needs to be defined in the conf/login-config.xml. You can resuse the security domain used to restrict access to the jmx-console.

### 3.2.2. Troubleshooting

When you use JDK5+, you may see the following error:

```
org.jboss.deployment.DeploymentException?: No PropertyDescriptor? for attribute:securityDomain; -
nested throwable: (java.beans.IntrospectionException?: No PropertyDescriptor? for attribute:securityDomai
```

This is due to a change in how the jmx descriptor names are stored with case preserved. To work around this isssue simply use all lower case attribute names.

```
<interceptors>
    <interceptor code="org.jboss.jmx.connector.invoker.AuthenticationInterceptor"
                 securitydomain="java:/jaas/jmx-console"/>
</interceptors>
```

# 4

# Secure the JMX Invokers (Authorization/Access Control)

## 4.1. Introduction

In the last chapter, you enabled authentication on the JMX invoker. This would basically restrict access to just the actors that have identified themselves. There may need for fine-grained access control on the JMX invoker. You may want to restrict access to users who have certain roles.

## 4.2. Enable Access Control or Authorization for the JMX Invoker

The JMX invoker can be made to authorize all requests.

### 4.2.1. Modifications Required (Use Case 1)

You will need to edit the jmx-invoker-service.xml file under the deploy directory of your server configuration (eg: default configuration). Please add an additional interceptor to look as follows:

```
<operation>
   <description>The detached invoker entry point</description>
   <name>invoke</name>
   <parameter>
       <description>The method invocation context</description>
       <name>invocation</name>
       <type>org.jboss.invocation.Invocation</type>
   </parameter>
   <return-type>java.lang.Object</return-type>
   <!-- Uncomment to require authenticated users -->
   <descriptors>
     <interceptors>
         <interceptor code="org.jboss.jmx.connector.invoker.AuthenticationInterceptor"
             securityDomain="java:/jaas/jmx-console"/>
         <interceptor code="org.jboss.jmx.connector.invoker.AuthorizationInterceptor"
                     authorizingClass="org.jboss.jmx.connector.invoker.RolesAuthorization"/>
     </interceptors>
   </descriptors>
</operation>
```

The AuthorizationInterceptor will use the JAAS authenticated subject that has been created by the AuthenticationInterceptor and will allow access to users who have a role called as "JBossAdmin".

### 4.2.2. Modifications Required (Use Case 2)

You will need to edit the jmx-invoker-service.xml file under the deploy directory of your server configuration (eg: default configuration). Please add an additional interceptor to look as follows:

```
<operation>
   <description>The detached invoker entry point</description>
   <name>invoke</name>
   <parameter>
       <description>The method invocation context</description>
       <name>invocation</name>
       <type>org.jboss.invocation.Invocation</type>
   </parameter>
   <return-type>java.lang.Object</return-type>
   <!-- Uncomment to require authenticated users -->
   <descriptors>
     <interceptors>
         <interceptor code="org.jboss.jmx.connector.invoker.AuthenticationInterceptor"
             securityDomain="java:/jaas/jmx-console"/>
         <interceptor code="org.jboss.jmx.connector.invoker.AuthorizationInterceptor"
                     authorizingClass="org.jboss.jmx.connector.invoker.ExternalizableRolesAuthorization"/
     </interceptors>
   </descriptors>
</operation>
```

This use case handles cases where you can configure the various roles that an user can have to gain access. The previous use case just handled the case when the users had a role called as "JBossAdmin".

In this case, you will need to provide a properties file called as "jmx-invoker-roles.properties" in a jar file or place it in the conf directory. The format of this file should be:

```
#Specify the roles that are authorized to access the jmx invoker delimited by comma
roles=testRole,testRole1
```

## 4.3. Reference

http://wiki.jboss.org/wiki/Wiki.jsp?page=SecureTheInvokers