

培训教育
Training Services



中间件安全

主讲：Gnosis

4.1.4 Jboss

知识子域: Jboss 的安全设置

知识子域: Jboss 的漏洞利用与防范

知识子域: Jboss 的日志审计方法

知识子域:JBoss 的安全设置

了解设置 jmx-console/web-console 密码的方法

了解开启日志功能的方法

了解设置通讯协议，开启 HTTPS 访问

了解修改 Web 的访问端口

Jboss 的安全设置——了解设置 jmx-console/web-console 密码的方法

➤ 简介

- JBoss是一个运行EJB的J2EE应用服务器。它是开放源代码的项目，遵循最新的J2EE规范。从JBoss项目开始至今，它已经从一个EJB容器发展成为一个基于的J2EE的一个web 操作系统(operating system for web)，它体现了J2EE规范中最新的技术。无论是学习还是应用，JBoss为我们提供了一个非常优秀的平台。
- JBoss是一个管理EJB的容器和服务器，支持EJB 1.1、EJB 2.0和EJB3.0的规范。但JBoss核心服务不包括支持servlet/JSP的WEB容器，一般与Tomcat或Jetty绑定使用。
- JBoss具有如下优点：
 - 1、JBoss是免费的,开放源代码J2EE的实现,通过LGPL许可证进行发布.但同时也有闭源的,开源和闭源流入流出的不是同一途径。
 - 2、JBoss需要的内存和硬盘空间比较小。
 - 3、安装便捷：解压后,只需配置一些环境变量即可。
 - 4、JBoss支持"热部署"，部署BEAN时,只拷贝BEAN的JAR文件到部署路径下即可自动加载它,如果有改动,也会自动更新
 - 5、JBoss与Web服务器在同一个Java虚拟机中运行,Servlet调用EJB不经过网络,从而大大提高运行效率,提升安全性能
 - 6、用户可以直接实施J2EE-EAR，而不是以前分别实施EJB- JAR和Web-WAR，非常方便。
 - 7、JBoss支持集群

<http://blog.csdn.net/wqiancangq/article/details/48137697>

JBOSS的部署

安装JDK

```
Last login: Tue Aug 29 01:07:21 2017 from 192.168.85.1
[root@localhost ~]# java -version
按住鼠标左键选择截图区域  "1.7.0_79"
鼠标右键或ESC退出截屏
java version "1.7.0_79"
Java(TM) SE Runtime Environment (build 1.7.0_79-b15)
Java HotSpot(TM) 64-Bit Server VM (build 24.79-b02, mixed mode)
[root@localhost ~]#
```

配置环境变量

```
export JAVA_HOME=/data/server/jdk1.7.0_79
export JBOSS_HOME=/data/jboss-as-7.1.1.Final
export PATH=$JAVA_HOME/bin:$PATH
```

JBOSS的部署

配置配置文件

```
[root@localhost configuration]# pwd  
/data/jboss-as-7.1.1.Final/standalone/configuration  
[root@localhost configuration]# vim standalone.xml
```

```
<socket-binding name="ajp" port="8009"/>  
<socket-binding name="http" port="8080"/>  
<socket-binding name="https" port="8443"/>  
<socket-binding name="osgi-http" interface="management" port="8090"/>  
<socket-binding name="remoting" port="4447"/>  
<socket-binding name="txn-recovery-environment" port="4712"/>  
<socket-binding name="txn-status-manager" port="4713"/>  
<outbound-socket-binding name="mail-smtp">  
    <remote-destination host="localhost" port="25"/>
```

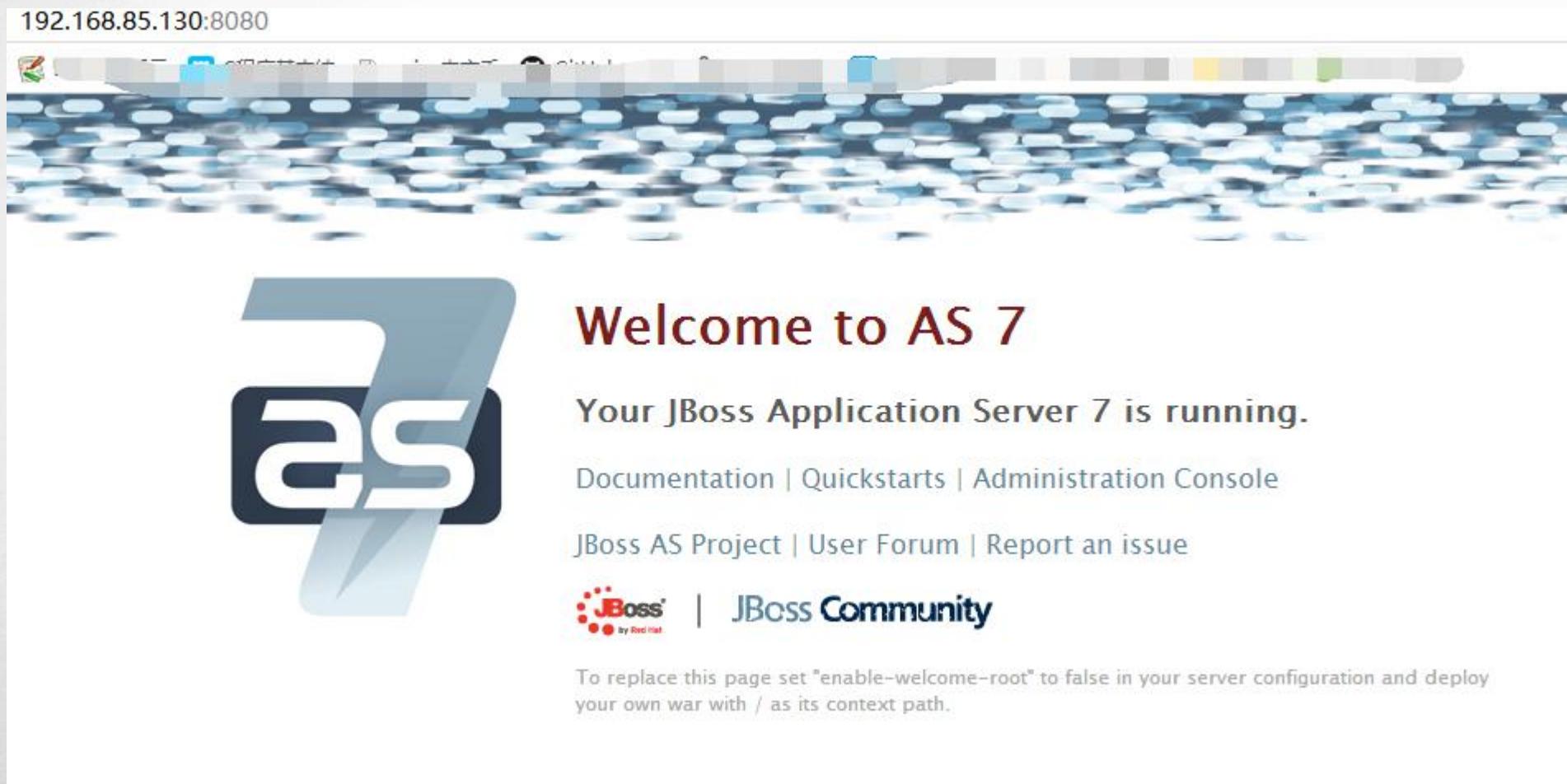
Jboss 的安全设置——了解设置 jmx-console/web-console 密码的方法

JBOSS的部署

```
[root@localhost configuration]# netstat -anlpt | grep java
tcp        0      0 192.168.85.130:9990          0.0.0.0:*              LISTEN      4358/java
tcp        0      0 192.168.85.130:9999          0.0.0.0:*              LISTEN      4358/java
tcp        0      0 192.168.85.130:8080          0.0.0.0:*              LISTEN      4358/java
tcp        0      0 192.168.85.130:4447          0.0.0.0:*              LISTEN      4358/java
```

```
01:27:53,622 INFO  [org.apache.coyote.http11.Http11Protocol] (MSC service thread 1-1) Star
n http--192.168.85.130-8080
01:27:54,090 INFO  [org.jboss.as.remoting] (MSC service thread 1-1) JBAS017100: Listening
9
01:27:54,154 INFO  [org.jboss.as.server.deployment.scanner] (MSC service thread 1-2) JBAS0
temDeploymentService for directory /data/jboss-as-7.1.1.Final/standalone/deployments
01:27:54,204 INFO  [org.jboss.as.remoting] (MSC service thread 1-1) JBAS017100: Listening
7
01:27:54,276 INFO  [org.jboss.as.connector.subsystems.datasources] (MSC service thread 1-2
ta source [java:jboss/datasources/ExampleDS]
01:27:54,328 INFO  [org.jboss.as] (Controller Boot Thread) JBAS015951: Admin console liste
.85.130:9990
01:27:54,337 INFO  [org.jboss.as] (Controller Boot Thread) JBAS015874: JBoss AS 7.1.1.Fina
3353ms - Started 133 of 208 services (74 services are passive or on-demand)
```

Jboss 的安全设置——了解设置 jmx-console/web-console 密码的方法



Jboss 的安全设置——了解设置 jmx-console/web-console 密码的方法

设置 jmx-console/web-console 密码

```
What type of user do you wish to add?  
a) Management User (mgmt-users.properties)  
b) Application User (application-users.properties)  
(a): a  
  
Enter the details of the new user to add.  
Realm (ManagementRealm) : admin  
Username : admin  
Password :  
Re-enter Password :  
The username 'admin' is easy to guess  
Are you sure you want to add user 'admin' yes/no? yes  
About to add user 'admin' for realm 'admin'  
Is this correct yes/no? yes  
Added user 'admin' to file '/data/jboss-as-7.1.1.Final/  
standalone/configuration/mgmt-users.properties'  
Added user 'admin' to file '/data/jboss-as-7.1.1.Final/  
domain/configuration/mgmt-users.properties'
```

欢迎来到AS 7

您的JBoss应用服务器7正在运行。

但是您尚未添加任何用户以访问管理控制台。

要添加新用户在AS 7安装的bin文件夹中执行add-user.sh脚本，并输入所请求的信息。

默认情况下，AS 7使用的领域名称为“ManagementRealm”，默认情况下已经选择。



```
darranl@localhost:~/src/jbossas7/jboss-as/build/target/jboss-as-7.1.0.Alpha2-SNAPSHOT/bin$ ./add-user.sh  
Enter details of new user to add.  
Realm (ManagementRealm) :  
Username : myNewUser  
Password :  
Re-enter Password :  
About to add user 'myNewUser' for realm 'ManagementRealm'  
Is this correct yes/no? yes  
Added user 'myNewUser' to file '/home/darranl/src/jbossas7/jboss-as/build/target/jboss-as-7.1.0.Alpha2-SNAPSHOT/standalone/configuration/mgmt-users.properties'  
Added user 'myNewUser' to file '/home/darranl/src/jbossas7/jboss-as/build/target/jboss-as-7.1.0.Alpha2-SNAPSHOT/domain/configuration/mgmt-users.properties'  
[darranl@localhost bin]$
```

添加用户后，请按照此链接再试一次。

Jboss 的安全设置——了解设置 jmx-console/web-console 密码的方法

设置 jmx-console/web-console 密码

The screenshot shows the JBoss Application Server 7.1 Management Console interface. The left sidebar has a navigation menu with sections like Server Status, Configuration (which is selected), Subsystem Metrics, Runtime Operations, and Deployments. The main content area is titled "Standalone Server" and shows the configuration for "Server: localhost". It indicates that the server configuration seems up-to-date. The "Code Name" is set to "Brontes" and the "Release version" is "7.1.1.Final". The "Server State" is listed as "running". Below this, there are tabs for "Extensions" and "Environment Properties", with "Extensions" currently selected. The "Name" column lists various subsystems: org.jboss.as.clustering.infinispan, org.jboss.as.configadmin, org.jboss.as.connector, org.jboss.as.deployment-scanner, org.jboss.as.ee, org.jboss.as.ejb3, org.jboss.as.jaxrs, and org.jboss.as.jdr. At the bottom right, there are navigation icons and the text "1-8 of 23".

Jboss7配置日志理论知识介绍

- Jboss 7日志可以在XML配置文件和日志管理属性文件内配置。默认日志配置在configuration目录的logging.properties文件内。
- 通常情况下，对于大多数安装，logging.properties内的默认值已经足够了。如要自定义日志类型，建议在xml配置文(standalone.xml或domain.xml文件，logging subsystem)内配置，可以定义7个主要类别：
 - <root-logger />
 - <logger category=" " />
 - >console-handler />
 - <file-handler />
 - <periodic-rotating-file-handler />
 - <size-rotating-file-handler />
 - <async-handler />
- 主要，应该使用XML配置文件，当logging子系统启动后日志管理属性会被忽略

Jboss 的安全设置——了解开启日志功能的方法

下面结合具体的示例解释XML配置文件和日志管理属性是如果记录Jboss 7的日志

- XML配置
- 此处的xml配置指的是standalone.xml或domain.xml文件，他们分别是standalond和domain模式启动的xml配置文件。以standalone.xml为例
- Standalone.xml文件中关于日志的配置信息如下：

```
subsystem xmlns="urn:jboss:domain:logging:1.1">
  <console-handler name="CONSOLE">
    <level name="INFO"/>
    <formatter>
      <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n"/>
    </formatter>
  </console-handler>
  <periodic-rotating-file-handler name="FILE">
    <formatter>
      <pattern-formatter pattern="%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%E%n"/>
    </formatter>
    <file relative-to="jboss.server.log.dir" path="server.log"/>
    <suffix value=".yyyy-MM-dd"/>
    <append value="true"/>
  </periodic-rotating-file-handler>
  <logger category="com.arjuna">
    <level name="WARN"/>
  </logger>
  <logger category="org.apache.tomcat.util.modeler">
```

Jboss 的安全设置——了解开启日志功能的方法

- 如下配置做简单的解释： 日志输出到console和file。 其他file，位于server.log 中。 日志级别是INFO
- 我们可以启动一下看看结果

```
02:16:29,553 INFO [org.jboss.as.connector.subsystems.datasources] (MSC service thread 1-1) JBAS010  
[a source [java:jboss/datasources/ExampleDS]  
02:16:29,989 INFO [org.jboss.as.controller] (Controller Boot Thread) JBAS014774: Service status re  
JBAS014777: Services which failed to start:      service jboss.remoting.server.management: org.jb  
ce.StartException in service jboss.remoting.server.management: JBAS017112: 地址已在使用 /192.168.85  
    service jboss.web.connector.http: org.jboss.msc.service.StartException in service jboss.web.c  
JBAS018007: Error starting web connector  
    service jboss.remoting.server.remoting-connector: org.jboss.msc.service.StartException in ser  
remoting.server.remoting-connector: JBAS017112: 地址已在使用 /192.168.85.130:4447  
    service jboss.serverManagement.controller.management.http: org.jboss.msc.service.StartExcepti  
jboss.serverManagement.controller.management.http: 地址已在使用 /192.168.85.130:9990  
  
02:16:30,022 INFO [org.jboss.as] (Controller Boot Thread) JBAS015954: Admin console is not enabled  
02:16:30,023 ERROR [org.jboss.as] (Controller Boot Thread) JBAS015875: JBoss AS 7.1.1.Final "Bronte  
th errors) in 6232ms - Started 126 of 208 services (7 services failed or missing dependencies, 74  
passive or on-demand)  
02:16:30,214 INFO [org.jboss.as.osgi] (MSC service thread 1-2) JBAS011942: Stopping OSGi Framework
```

➤ Logging-properties(日志管理器)

- 日志管理器，就必须在管理器中
- logging.p

```
[html] view plain copy print ?  
01. # Note this file has been generated and will be overwritten if a  
02. # logging subsystem has been defined in the XML configuration.  
03.  
04.  
05. # Additional loggers to configure (the root logger is always configured)  
06. loggers=jacorb,com.arjuna,org.apache.tomcat.util.modeler,org.jboss.as.config,jacorb.config,sun.rmi  
07.  
08. logger.level=INFO  
09. logger.handlers=CONSOLE,FILE  
10.  
11. logger.jacorb.level=WARN  
12. logger.jacorb.useParentHandlers=true  
13.  
14. logger.com.arjuna.level=WARN  
15. logger.com.arjuna.useParentHandlers=true  
16.  
17. logger.org.apache.tomcat.util.modeler.level=WARN  
18. logger.org.apache.tomcat.util.modeler.useParentHandlers=true  
19.  
20. logger.org.jboss.as.config.level=DEBUG  
21. logger.org.jboss.as.config.useParentHandlers=true  
22.  
23. logger.jacorb.config.level=ERROR  
24. logger.jacorb.config.useParentHandlers=true  
25.  
26. logger.sun.rmi.level=WARN  
27. logger.sun.rmi.useParentHandlers=true  
28.  
29. handlers=CONSOLE,org.jboss.logging.handlers.ConsoleHandler
```

日志管理
，日志

Jboss 的安全设置——了解设置通讯协议，开启 HTTPS 访问

➤ 1. 生成密钥

进入%JAVA_HOME%/bin目录
执行命令

keytool -genkey -alias tomcat -keyalg RSA -keystore F:\tomcat.keystore -validity 36500

参数简要说明：“F:\tomcat.keystore”含义是将证书文件保存在F盘，证书文件名称是tomcat.keystore；“-validity 36500”含义是证书有效期，36500表示100年，默认值是90天

完成上述输入后，直接回车则在你在第二步中定义的位置找到生成的文件

```
<subsystem xmlns="urn:jboss:domain:web:7.0.1.Final" name="default">
    <!-- connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"-->
    <connector name="https" protocol="HTTP/1.1" socket-binding="https" scheme="https" secure="true">
        <ssl name="https" password="123.com" certificate-key-file="../standalone/configuration/tomcat.keystore" />
    </connector>
    <ssl name="https" password="changeit" certificate-key-
        file="../standalone/configuration/tomcat.keystore"/>
</subsystem>
```

Jboss 的安全设置——了解修改 Web 的访问端口

```
<socket-binding name="management-https" interface="management" port="${jboss  
    <socket-binding name="ajp" port="8009"/>  
    <socket-binding name="http" port="8080"/>  
    <socket-binding name="https" port="8443"/>  
    <socket-binding name="osgi-http" interface="management" port="8090"/>  
    <socket-binding name="remoting" port="4447"/>  
    <socket-binding name="txn-recovery-environment" port="4712"/>  
    <socket-binding name="txn-status-manager" port="4713"/>  
    <outbound-socket-binding name="mail-smtp">  
        <remote-destination host="localhost" port="25"/>  
    </outbound-socket-binding>
```

- 漏洞现象：
- 可以操控远程服务器
- 1、上传文件：
- 上传文件：
- 查看一下根目录是否存在test.xml文件
- 2、远程操作服务器
- 在CMD输入命令操控服务器

先来看基本的攻击思路：利用漏洞部署一个war（war是JavaEE里基本部署单位），这个war里一般只有一个jsp的webshell，利用此shell，黑客可做各种猥琐事。

整个过程中，怎么能搞上去一个webshell是核心。这个“搞上去”，在JBoss世界里，就是deploy一个war。JBoss可真是煞费苦心地提供了N多种部署策略，又在无意中为这N多种部署策略提供M种调用方式，于是乎，跟JBoss相关的漏洞也就理所当然出现了N*M种。这也是造成JBoss漏洞纷繁复杂不好理解的一大原因

Jboss 的漏洞利用与防范

JMXInvokerServlet/jmx-console/web-console 漏洞利用与防范

看下图，用这么个url可以方便地部署个“一句话后门”。这里，为了方便阅读方便，对这个长url做了分行处理并加了适当的说明。

```
http://www.example.com:8080/jmx-console/HtmlAdaptor  
    ?action=invokeOpByName  
    &name=jboss.admin:service=DeploymentFileRepository  
    &methodName=store  
    &argType=java.lang.String  
    Folder Name: &arg0=shell.war  
                  &argType=java.lang.String  
    File prefix: &arg1=shell  
                  &argType=java.lang.String  
    File suffix: &arg2=.jsp  
                  &argType=java.lang.String  
    File content: &arg3=<% Runtime.getRuntime().exec(  
                           request.getParameter("c"));  
                  %>  
                  &argType=boolean  
    No hot deploy: &arg4=True
```

- JBoss收到这个请求后，就会部署一个shell.war的文件夹，它里面只有一个文件shell.jsp。随后就可以利用此shell.jsp执行各种命令啦。
- 简短介绍了部署流程后，重点说下上图中红框内标示的三部分。
- 1，先说第二个“DeploymentFileRepository”，它是JBoss运行环境里创建的JMX对象，可通过“jboss.admin:service”名找到。这个对象实质上是一个服务，DeploymentFileRepository名字出卖了它的作用，可以deploy一个file。
- 2，再回过头来看第一个“jmx-console/HtmlAdaptor”，它本质上是JBoss内在服务的调用接口。这里关于JMX多说两句。JMX作为JavaEE体系里一个标准，是为了方便运维（与监控）而设计的，其基本思想是给每一个服务主体创建一个“影子”对象，这个“影子”对象可控制查看主体对象的一切。这里的“jmx-console/HtmlAdaptor”正是控制查看所有影子对象的一个总入口。
- 3，第三部分不用太多地说明，它是黑客行为的最终目标。其内容可以定制，表现形式又随具体的攻击组合不同而异。也就是“蛊”可以随场景和目的不同而定制。

- 借助这个典型事件分析，从而对JBoss攻击所涉及概念有形象了解后，看针对JBoss常见的攻击方式，也就是种蛊方式。
- 1, jmx-console/HtmlAdaptor + DeploymentFileRepository。也就是上面详细分析的那个。之所以这里再列出来，完全是为了向其致敬。可以说，这个种蛊方式最方便。也正是它，才有了当年大名鼎鼎的JBoss蠕虫事件。不过，随着行业对JBoss安全的重视，适合此组合的场景已经很少有了。
- 2, /jmx-console/HtmlAdaptor + BSHDeployer。跟上面的类似，不过，换了另一种部署方式。这里的BSH是一种shell语言，JVM支持，JBoss也可就出于顺便支持了。这个方式是实际是通过HTTP请求给BSHDeployer传了一段beanshell脚本，黑客关心的webshell就内嵌在这个脚本里。
- 3, web-console/Invoker + DeploymentFileRepository。跟第一种类似，只不过是换了调用的入口，背后的服务还是DeploymentFileRepository。

- 4, invoker/JMXInvokerServlet + application/x-java-serialized-object;class=org.jboss.invocation.MarshalledInvocation + BSHDeployer。这个在利用方式上跟前面几个稍稍不同。前面几个上传的payload还是肉眼可以直接读出的文本，这个有变化了，它实质是发送了一个用HTTP请求发送RMI。这个过程在原理上相当于在第2个的基础上套了一层 MarshalledInvocation调用。稍微介绍下 MarshalledInvocation。Java里，可以方便地把一次方法调用用一个Java类描述下来，随后再编译后字节码通过网络发送，服务端收到请求后，再解析这个字节码里的调用信息。这里，最终又把请求转给了 BSHDeployer。
- 5, invoker/JMXInvokerServlet + application/x-java-serialized-object;class=org.jboss.invocation.MarshalledInvocation + MainDeployer。回到这些天爆出的利用方式 (<http://www.exploit-db.com/exploits/28713/>) 了。这种方式跟第4种类似，不同体现在MainDeployer。上面第4种用BSHDeployer时， webshell全部内容可以在请求中传送，而MainDeployer就不行了， 它需要一个指向远程war的url， MainDeployer把这个war下载下来后，再正常地部署。

- JBoss漏洞的利用本质上是创建一个webshell，这样它的危害就转化成webshell的功能和Jboss进程的用户权限。如果放任自流，然后... 就没有然后了。

- 如何防御？从上面的分析，我们知道，webshell的部署过程有两个关键点：找到合适的入口调用合适的服务。顺着这个思路，把用不着的服务统统关掉，尽可能地减少访问入口（或关掉访问入口或加强权限拦截）。

Jboss 的日志审计方法

- **%a** - 远端IP地址
- **%A** - 本地IP地址
- **%b** - 发送的字节数, 不包括HTTP头, 如果为0, 使用" - "
- **%B** - 发送的字节数, 不包括HTTP头
- **%h** - 远端主机名(如果resolveHost=false, 远端的IP地址)
- **%H** - 请求协议
- **%l** - 从identd返回的远端逻辑用户名 (总是返回 '-')
- **%m** - 请求的方法 (GET, POST, 等)
- **%p** - 收到请求的本地端口号
- **%q** - 查询字符串(如果存在, 以 '?'开始)
- **%r** - 请求的第一行, 包含了请求的方法和URI
- **%s** - 响应的状态码
- **%S** - 用户的session ID
- **%t** - 日志和时间, 使用通常的Log格式
- **%u** - 认证以后的远端用户 (如果存在的话, 否则为' - ')
- **%U** - 请求的URI路径
- **%v** - 本地服务器的名称
- **%D** - 处理请求的时间, 以毫秒为单位
- **%T** - 处理请求的时间, 以秒为单位

THANK YOU 感谢观看
FOR YOUR ATTENTION!

北京谷安天下科技有限公司

谷安天下公司主页：www.gooann.com

谷安培训教育网页：<http://px.gooann.com>

安全意识产品网页：<http://sectv.gooann.com>

产品解决方案网页：<http://product.gooann.com>

谷安信息安全商城：<http://gooannpx.taobao.com>