

S2一个有趣的坑

[zhouliu](#) / 2018-04-14 17:23:04 / 浏览数 1405 [技术文章](#) [技术文章](#) [顶\(0\)](#) [踩\(0\)](#)

前言

现在再议Struts2 怕是吸引不了多少看官的眼球，但是这个坑我觉得是对惯性思维的挑战，并不是一点营养都没有。

共识

对于输入的净化，一般我们认为最小化限制会更加安全可靠。比如对于Java这种强类型语言，使用int接受参数比String接受参数更加窄化了输入字符空间，自然在防御XSS、

一句话概括

这里要说的坑就是在Struts2中即便使用int（其他简单类型也相似）接受参数，在视图中仍然可能输出String类型，因此会存在XSS的隐患。

Demo

简单模拟一个根据商品id查询商品信息并将商品信息在页面中输出。

Action

```
public class ProductAction extends ActionSupport{

    private int id; //

    @Override
    public String execute() {

        Product product = null; //■■■■■■■■■■

        ServletActionContext.getRequest().setAttribute("target", product);

        return SUCCESS;
    }

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
}
```

struts-product.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
    "http://struts.apache.org/dtds/struts-2.5.dtd">
<struts>

    <package name="default" extends="struts-default" >
        <interceptors>
            <interceptor-stack name="customizedStack">
                <interceptor-ref name="exception" />
                <interceptor-ref name="alias" />
                <interceptor-ref name="servletConfig" />
                <interceptor-ref name="i18n" />
                <interceptor-ref name="prepare" />
                <interceptor-ref name="chain" />
                <interceptor-ref name="scopedModelDriven" />
                <interceptor-ref name="modelDriven" />
                <interceptor-ref name="fileUpload" />
                <interceptor-ref name="checkbox" />
                <interceptor-ref name="datetime" />
            </interceptor-stack>
        </interceptors>
    </package>
</struts>
```

```

        <interceptor-ref name="multiselect" />
        <interceptor-ref name="staticParams" />
        <interceptor-ref name="actionMappingParams" />
        <interceptor-ref name="params" />
        <interceptor-ref name="conversionError" />
        <interceptor-ref name="validation">
            <param name="excludeMethods">input,back,cancel,browse</param>
        </interceptor-ref>
        <interceptor-ref name="debugging" />
        <interceptor-ref name="deprecation" />
    </interceptor-stack>
</interceptors>
</package>

<package name="product" extends="default" namespace="/">
    <default-interceptor-ref name="customizedStack" />
    <action name="productInfo" class="demo.action.ProductAction">
        <result name="success">/WEB-INF/pages/jsp/productInfo.jsp</result>
    </action>
</package>
</struts>

```

productInfo.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags"%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Product Information</title>
</head>
<body>
    <h1>Product Information</h1>
    <s:if test="%{target==null}">
        Sorry, Product with id:<strong> ${id}</strong> not found!
    </s:if>
    <s:else>
        Product with id: <strong>${id}</strong> found:
        <div>
            ID ■${id}</br>
            Name ■${name}</br>
            Price ■${price}</br>
            Description■${description}</br>
        </div>
    </s:else>
</body>

```

id=1



Product Information

Sorry, Product with id: 1 not found!

id=xianzhi

Product Information

Sorry, Product with id: **xianzhi** not found!

先知社区

id=<svg onload=alert('Oops')>

Product Information

Sorry, Product with id:

not found!

Oops

☐ 阻止此页面创建更多对话框

确定

原因

(仍以上面的Demo为例子简单解释)

1) 大家知道, 在Struts2

中有众多的Interceptor, 其中com.opensymphony.xwork2.interceptor.ParametersInterceptor会找到参数对应的setter。当客户端传递的参数值是String类型 (例如"xianzhi") 不幸的是并没有找到 (只有void setId(int id), 因此会出现错误。

2) 另一个Interceptor--com.opensymphony.xwork2.interceptor.ConversionErrorInterceptor 会将id String类型的参数值保存在Map中。

```
//ConversionErrorInterceptor#intercept
```

```
public String intercept(ActionInvocation invocation) throws Exception {
```

```
    ActionContext invocationContext = invocation.getInvocationContext();
    Map<String, Object> conversionErrors = invocationContext.getConversionErrors();
    ValueStack stack = invocationContext.getValueStack();
```

```
    HashMap<Object, Object> fakie = null;
```

```
    for (Map.Entry<String, Object> entry : conversionErrors.entrySet()) {
        String propertyName = entry.getKey();
        Object value = entry.getValue();
```

```
        if (shouldAddError(propertyName, value)) {
            String message = XWorkConverter.getConversionErrorMessage(propertyName, stack);
```

```
            Object action = invocation.getAction();
            if (action instanceof ValidationAware) {
                ValidationAware va = (ValidationAware) action;
                va.addFieldError(propertyName, message);
            }
        }
    }
```

```
    if (fakie == null) {
```

```

        fakie = new HashMap<Object, Object>();
    }

    fakie.put(propertyName, getOverrideExpr(invocation, value));
}

if (fakie != null) {
    // if there were some errors, put the original (fake) values in place right before the result
    stack.getContext().put(ORIGINAL_PROPERTY_OVERRIDE, fakie);
    invocation.addPreResultListener(new PreResultListener() {
        public void beforeResult(ActionInvocation invocation, String resultCode) {
            Map<Object, Object> fakie = (Map<Object, Object>) invocation.getInvocationContext().get(ORIGINAL_PROPERTY_OVERRIDE);

            if (fakie != null) {
                invocation.getStack().setExprOverrides(fakie); // 这里"xianzhi" Map
            }
        }
    });
}

return invocation.invoke();
}

//OgnlValueStack#setExprOverrides

**
* @see com.opensymphony.xwork2.util.ValueStack#setExprOverrides(java.util.Map)
*/
public void setExprOverrides(Map<Object, Object> overrides) {
    if (this.overrides == null) {
        this.overrides = overrides;
    } else {
        this.overrides.putAll(overrides); // 这里"xianzhi" Map overrides
    }
}

```

3) Action执行完之后渲染页面 (这里是productInfo.jsp) , 页面的\${id}怎么解析呢? 通过ognl一番折腾, 进入下面的方法。

```

//OgnlValueStack

private Object tryFindValue(String expr) throws OgnlException {
    Object value;
    expr = lookupForOverrides(expr);
    if (defaultType != null) {
        value = findValue(expr, defaultType);
    } else {
        value = getValueUsingOgnl(expr);
        if (value == null) {
            value = findInContext(expr);
        }
    }
    return value;
}

private String lookupForOverrides(String expr) {
    if ((overrides != null) && overrides.containsKey(expr)) {
        expr = (String) overrides.get(expr); // overrides 这里"xiaozhi" Map
    }
    return expr;
}

```

谁背锅

如果将struts-product.xml简化为如下:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
    "http://struts.apache.org/dtds/struts-2.5.dtd">

```

```
<struts>


    <package name="product" extends="struts-default" namespace="/">
        <action name="productInfo" class="demo.action.ProductAction">
            <result name="success">/WEB-INF/pages/jsp/productInfo.jsp</result>
        </action>
    </package>

</struts>
```

浏览器访问/productInfo?id=xianzhi

→ ↺ ⓘ localhost:8080/struts2XSS/productInfo?id=xianzhi

TTP Status 404 - No result defined for action demo.action.ProductAction and result input

 Status report

Message No result defined for action demo.action.ProductAction and result input

Description The requested resource is not available.

Apache Tomcat/7.0.81

先知社区

噢，404了！

如果再将struts-product.xml修改为如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
    "http://struts.apache.org/dtds/struts-2.5.dtd">
<struts>

    <package name="product" extends="struts-default" namespace="/">
        <action name="productInfo" class="demo.action.ProductAction">
            <result name="input">/WEB-INF/pages/jsp/productInfo.jsp</result>
        </action>
    </package>

</struts>
```

噢，又回来了！

← → ↺ ⓘ localhost:8080/struts2XSS/productInfo?id=xianzhi

Product Information

Sorry, Product with id: xianzhi not found!

先知社区

看404报错信息“No result defined for action xxx and result input”，为什么result是input呢？我们最初只定义了success！原来是拦截器com.opensymphony.xwork2.interceptor.DefaultWorkflowInterceptor改变了result：

```
//DefaultWorkflowInterceptor#doIntercept

* Intercept {@link ActionInvocation} and returns a <code>inputResultName</code>
* when action / field errors is found registered.
*
* @return String result name
*/
@Override
protected String doIntercept(ActionInvocation invocation) throws Exception {
    Object action = invocation.getAction();
```

```

if (action instanceof ValidationAware) {
    ValidationAware validationAwareAction = (ValidationAware) action;

    if (validationAwareAction.hasErrors()) {
        if (LOG.isDebugEnabled()) {
            LOG.debug("Errors on action [{#0}], returning result name [{#1}]", validationAwareAction, inputResultName);
        }

        String resultName = inputResultName; //inputResultName 为"input"
        resultName = processValidationWorkflowAware(action, resultName);
        resultName = processInputConfig(action, invocation.getProxy().getMethod(), resultName);
        resultName = processValidationErrorAware(action, resultName);

        return resultName;
    }
}

return invocation.invoke();
}

```

回头看我们的Demo配置，并没有DefaultWorkflowInterceptor，但是在struts-default package中定义了

```

<interceptor-stack name="defaultStack">
    <interceptor-ref name="exception"/>
    <interceptor-ref name="alias"/>
    <interceptor-ref name="servletConfig"/>
    <interceptor-ref name="i18n"/>
    <interceptor-ref name="prepare"/>
    <interceptor-ref name="chain"/>
    <interceptor-ref name="scopedModelDriven"/>
    <interceptor-ref name="modelDriven"/>
    <interceptor-ref name="fileUpload"/>
    <interceptor-ref name="checkbox"/>
    <interceptor-ref name="datetime"/>
    <interceptor-ref name="multiselect"/>
    <interceptor-ref name="staticParams"/>
    <interceptor-ref name="actionMappingParams"/>
    <interceptor-ref name="params"/>
    <interceptor-ref name="conversionError"/>
    <interceptor-ref name="validation">
        <param name="excludeMethods">input,back,cancel,browse</param>
    </interceptor-ref>
    <interceptor-ref name="workflow"> <!-- 这里 -->
        <param name="excludeMethods">input,back,cancel,browse</param>
    </interceptor-ref>
    <interceptor-ref name="debugging"/>
    <interceptor-ref name="deprecation"/>
</interceptor-stack>

```

两种情况可能踩坑

- 1) 自定义拦截器配置
- 2) 定义了input "resultName"

笔者曾经跟官方讨论过是否应该杜绝这种坑，但是断断续续两个月的邮件通信后，结论是这应该交给开发者去处理。

点击收藏 | 0 关注 | 1

[上一篇：用侧信道读取特权内存（下）](#) [下一篇：科威盒子导航系统代码审计过程总结](#)

1. 0 条回复

- 动动手指，沙发就是你的了！

[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](#)