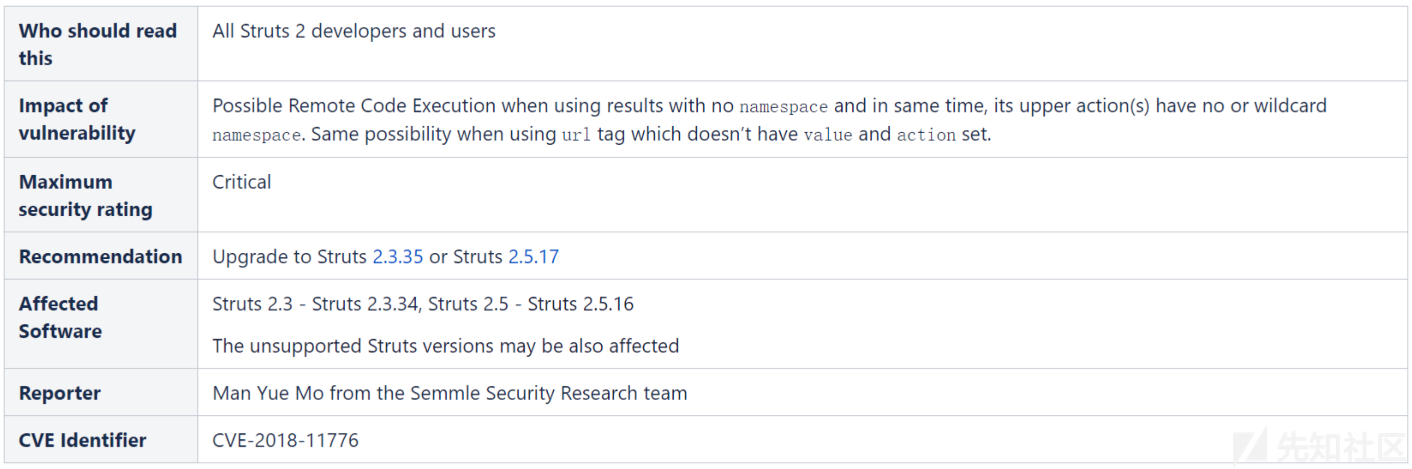
【Struts2-代码执行漏洞分析系列】S2-057

chybeta / 2018-08-23 11:59:23 / 浏览数 9420 安全技术 漏洞分析 顶(0) 踩(0)

漏洞公告

https://cwiki.apache.org/confluence/display/WW/S2-057

问题：
It is possible to perform a RCE attack when namespace value isn't set for a result defined in underlying xml configurations and in same time, its upper action(s) configurations have no or wildcard namespace. Same possibility when using url tag which doesn’t have value and action set and in same time, its upper action(s) configurations have no or wildcard namespace.



Who should read this	All Struts 2 developers and users
Impact of vulnerability	Possible Remote Code Execution when using results with no `namespace` and in same time, its upper action(s) have no or wildcard `namespace`. Same possibility when using `url` tag which doesn’t have `value` and `action` set.
Maximum security rating	Critical
Recommendation	Upgrade to Struts 2.3.35 or Struts 2.5.17
Affected Software	Struts 2.3 - Struts 2.3.34, Struts 2.5 - Struts 2.5.16 The unsupported Struts versions may be also affected
Reporter	Man Yue Mo from the Semmle Security Research team
CVE Identifier	CVE-2018-11776

漏洞发现者的博客： https://lgtm.com/blog/apache_struts_CVE-2018-11776

环境搭建

下载 https://archive.apache.org/dist/struts/2.5.16/struts-2.5.16-all.zip

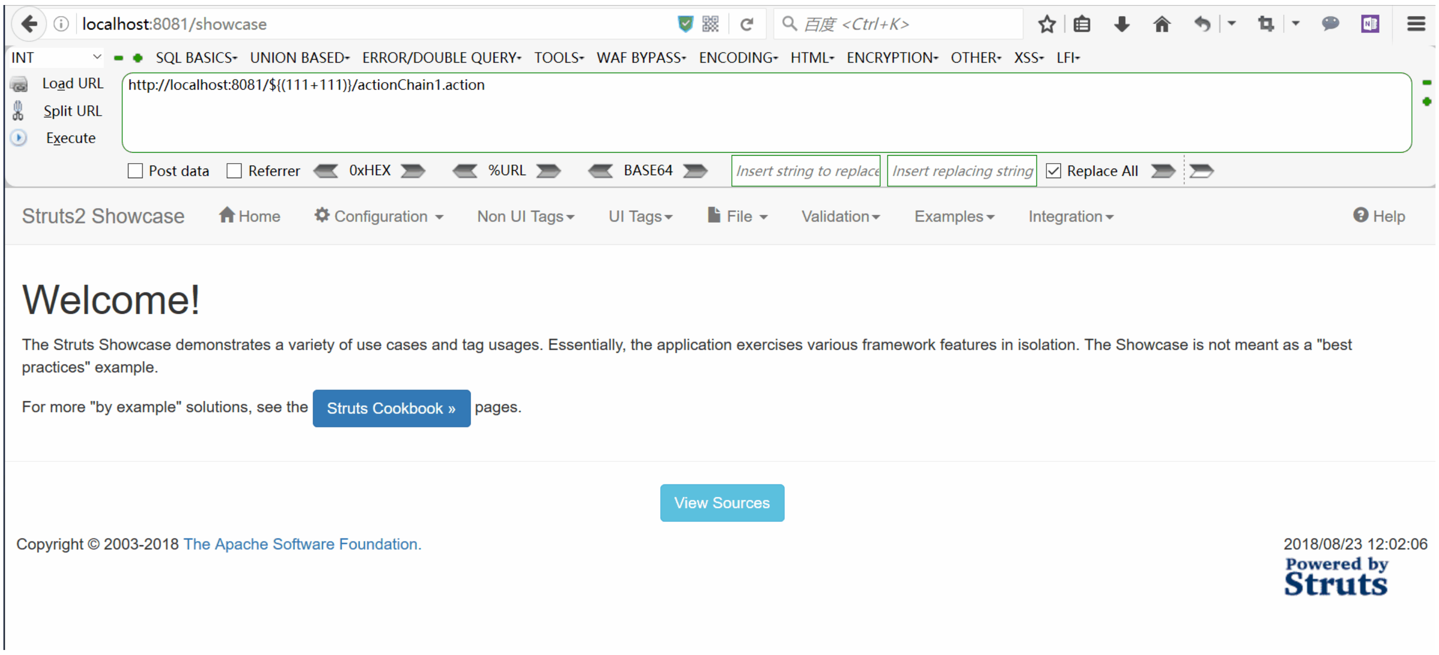
IDEA中打开，修改apps/showcase/src/main/resources/struts-actionchaining.xml 为：

```
<!DOCTYPE struts PUBLIC
   "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
   "http://struts.apache.org/dtds/struts-2.5.dtd">

<struts>
   <package name="actionchaining" extends="struts-default">
       <action name="actionChain1" class="org.apache.struts2.showcase.actionchaining.ActionChain1">
           <result type="redirectAction">
               <param name = "actionName">register2</param>
           </result>
       </action>
   </package>
</struts>
```

同时查看 org/apache/struts2/default.properties:201 ，其值为true

```
### Whether to always select the namespace to be everything before the last slash or not
struts.mapper.alwaysSelectFullNamespace=true
```

访问: http://localhost:8081/${(111+111)}/actionChain1.action


localhost:8081/showcase

http://localhost:8081/${(111+111)}/actionChain1.action

Struts2 Showcase

Welcome!

The Struts Showcase demonstrates a variety of use cases and tag usages. Essentially, the application exercises various framework features in isolation. The Showcase is not meant as a "best practices" example.

For more "by example" solutions, see the Struts Cookbook » pages.

Copyright © 2003-2018 The Apache Software Foundation.

2018/08/23 12:02:06

url变为： http://localhost:8081/222/register2.action

111+111=222 即产生了OGNL注入。

漏洞分析

这次的漏洞可以有多种攻击向量，根据漏洞作者blog有:

1. Redirect action
2. Action chaining
3. Postback result

以上提及的三种都属于Struts2的跳转方式。在 struts-default.xml:190(截取部分)

```
<result-types>
    <result-type name="chain" class="com.opensymphony.xwork2.ActionChainResult"/>
    <result-type name="redirectAction" class="org.apache.struts2.result.ServletActionRedirectResult"/>
    <result-type name="postback" class="org.apache.struts2.result.PostbackResult" />
</result-types>
```

为清楚起见，这里解释一下strut2中对■■result■■的处理过程。这些■■result type都要经过com/opensymphony/xwork2/DefaultActionInvocation.java:367 处理

```
private void executeResult() throws Exception {
    result = createResult();

    String timerKey = "executeResult: " + getResultCode();
    try {
        UtilTimerStack.push(timerKey);
        if (result != null) {
            result.execute(this);
        }
    ...
}
```

首先通过result = createResult()获取到相应的result对象。如果result不为null则执行result.execute(this);。这个execute方法则由具体result对象实现。

有一些具体的result对象比如下面提到的Redirect action和Postback result，会产生一个跳转地址location，并传入org/apache/struts2/result/StrutsResultSupport.java:194:

```
/**
    * Implementation of the <tt>execute</tt> method from the <tt>Result</tt> interface. This will call
    * the abstract method {@link #doExecute(String, ActionInvocation)} after optionally evaluating the
    * location as an OGNL evaluation.
    *
    * @param invocation the execution state of the action.
    * @throws Exception if an error occurs while executing the result.
*/
public void execute(ActionInvocation invocation) throws Exception {
    lastFinalLocation = conditionalParse(location, invocation);
    doExecute(lastFinalLocation, invocation);
}
```

而conditionalParse定义如下，将会执行OGNL表达式。

```
/**
    * Parses the parameter for OGNL expressions against the valuestack
    *
    * @param param The parameter value
    * @param invocation The action invocation instance
    * @return the resulting string
*/
protected String conditionalParse(String param, ActionInvocation invocation) {
    if (parse && param != null && invocation != null) {
        return TextParseUtil.translateVariables(
            param,
            invocation.getStack(),
            new EncodingParsedValueEvaluator());
    } else {
        return param;
    }
}
```

所以可以看到重点是StrutsResultSupport中`conditionalParse(location, invocation)`的location变量。

接下来部分就关注三种result-type的具体实现和具体攻击点。

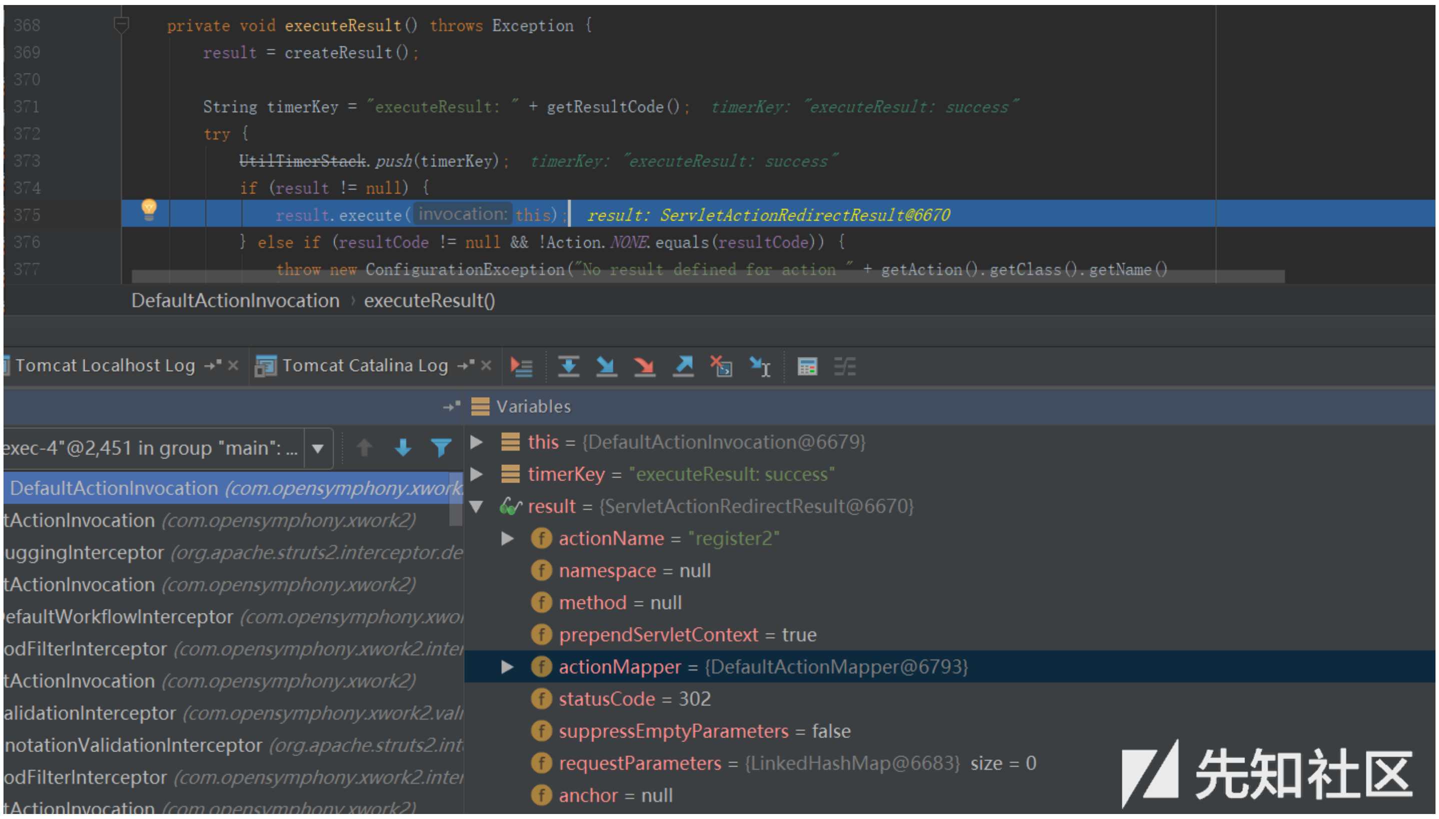
攻击点一：[Redirect action]

apps/showcase/src/main/resources/struts-actionchaining.xml 中注意`<result>`标签中`<type>`为`redirectAction`：

```
<result type="redirectAction">
    <param name = "actionName">register2</param>
</result>
```

`redirectAction`对应的处理类为`org.apache.struts2.result.ServletActionRedirectResult`

在 com/opensymphony/xwork2/DefaultActionInvocation.java:368



```
    private void executeResult() throws Exception {
        result = createResult();

        String timerKey = "executeResult: " + getResultCode();  timerKey: "executeResult: success"
        try {
            ~~UtilTimerStack~~.push(timerKey);  timerKey: "executeResult: success"
            if (result != null) {
                result.execute( invocation: this);  result: ServletActionRedirectResult@6670
            } else if (resultCode != null && !Action.NONE.equals(resultCode)) {
                throw new ConfigurationException("No result defined for action " + getAction().getClass().getName()
```

DefaultActionInvocation › executeResult()

Tomcat Localhost Log   Tomcat Catalina Log

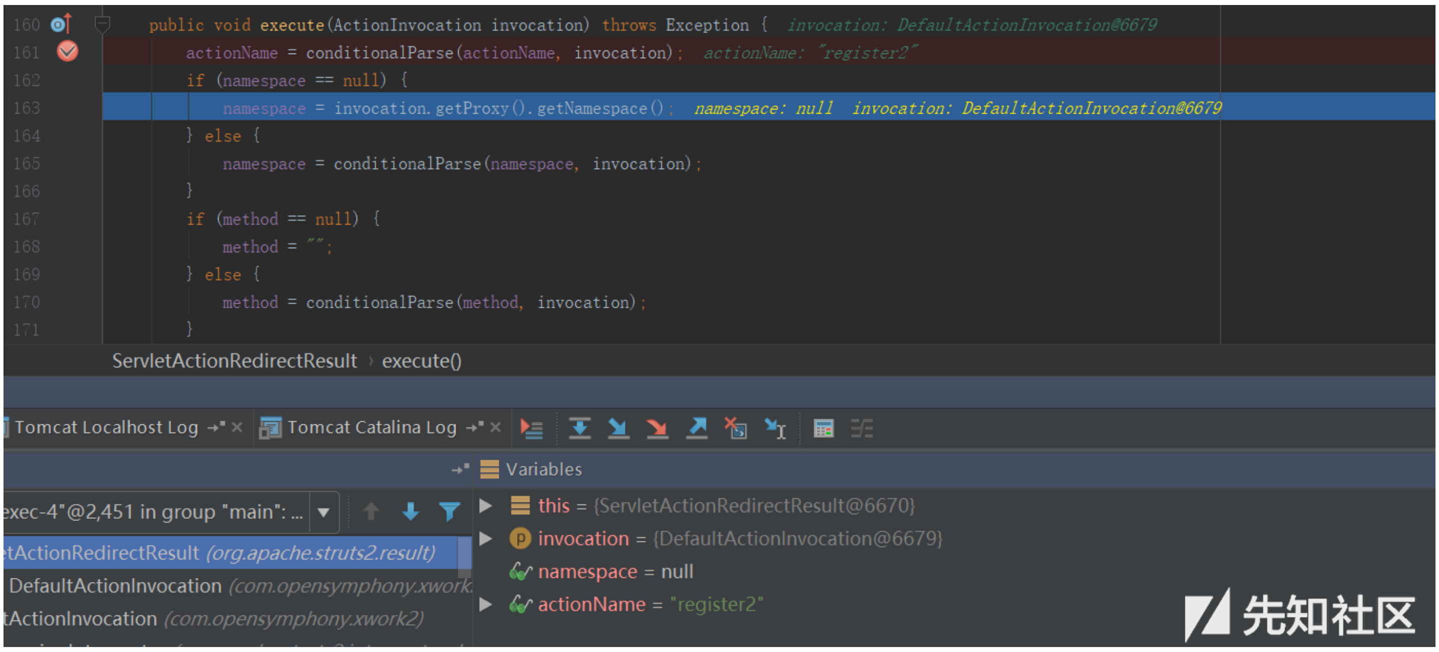
Variables

```
exec-4"@2,451 in group "main": ...
DefaultActionInvocation (com.opensymphony.xwork
tActionInvocation (com.opensymphony.xwork2)
uggingInterceptor (org.apache.struts2.interceptor.de
tActionInvocation (com.opensymphony.xwork2)
efaultWorkflowInterceptor (com.opensymphony.xwo
odFilterInterceptor (com.opensymphony.xwork2.inter
tActionInvocation (com.opensymphony.xwork2)
alidationInterceptor (com.opensymphony.xwork2.vali
notationValidationInterceptor (org.apache.struts2.int
odFilterInterceptor (com.opensymphony.xwork2.inter
tActionInvocation (com.opensymphony.xwork2)
```

```
▶ this = {DefaultActionInvocation@6679}
▶ timerKey = "executeResult: success"
▼ result = {ServletActionRedirectResult@6670}
    ▶ actionName = "register2"
      namespace = null
      method = null
      prependServletContext = true
    ▶ actionMapper = {DefaultActionMapper@6793}
      statusCode = 302
      suppressEmptyParameters = false
      requestParameters = {LinkedHashMap@6683}  size = 0
      anchor = null
```

跟入`redirectAction`的`execute`方法即 org/apache/struts2/result/ServletActionRedirectResult.java:160

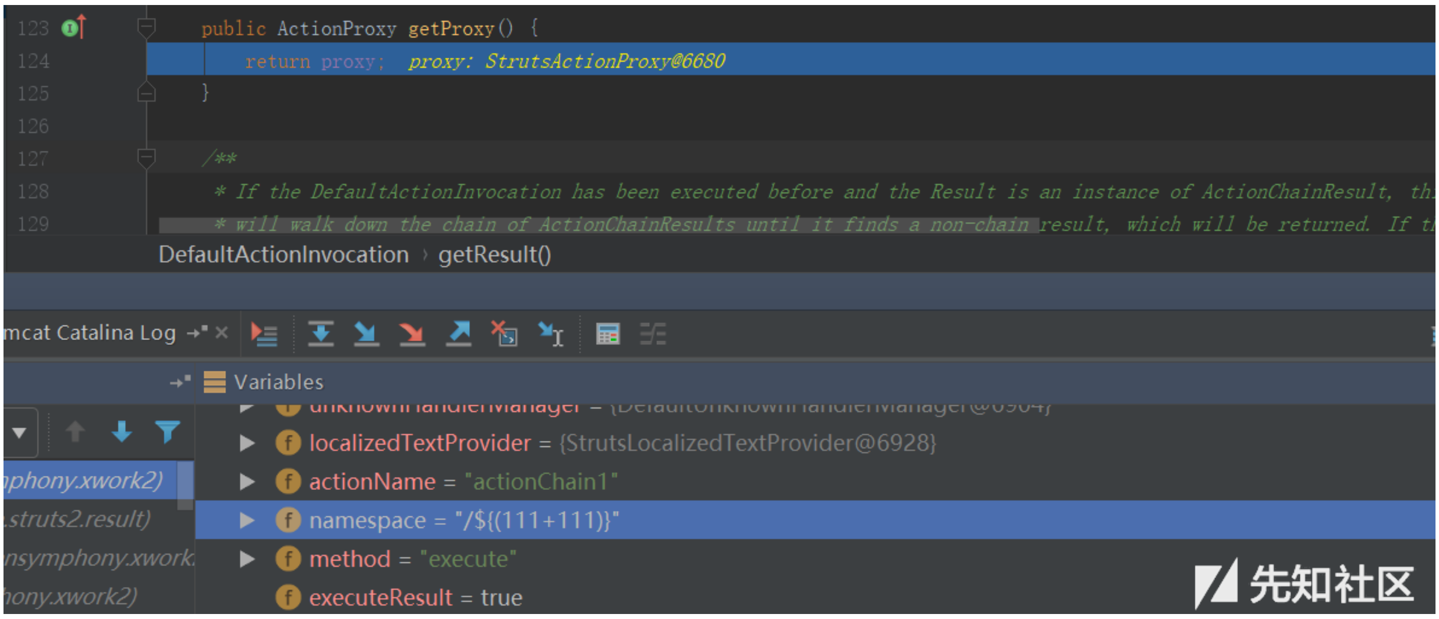
```
public void execute(ActionInvocation invocation) throws Exception {
   actionName = conditionalParse(actionName, invocation);
   if (namespace == null) {
       namespace = invocation.getProxy().getNamespace();
   ...
}
```



```
public void execute(ActionInvocation invocation) throws Exception {   invocation: DefaultActionInvocation@6679
    actionName = conditionalParse(actionName, invocation);   actionName: "register2"
    if (namespace == null) {
        namespace = invocation.getProxy().getNamespace();   namespace: null   invocation: DefaultActionInvocation@6679
    } else {
        namespace = conditionalParse(namespace, invocation);
    }
    if (method == null) {
        method = "";
    } else {
        method = conditionalParse(method, invocation);
    }
ServletActionRedirectResult › execute()
```

```
Tomcat Localhost Log   Tomcat Catalina Log
Variables
exec-4"@2,451 in group "main": ...
tActionRedirectResult (org.apache.struts2.result)
DefaultActionInvocation (com.opensymphony.xwork
tActionInvocation (com.opensymphony.xwork2)

this = {ServletActionRedirectResult@6670}
invocation = {DefaultActionInvocation@6679}
namespace = null
actionName = "register2"
```

由于在配置xml时没有指定naPmespace，所以这里的namespace为null，将会执行`invocation.getProxy().getNamespace();`



```
public ActionProxy getProxy() {
    return proxy;   proxy: StrutsActionProxy@6680
}

/**
 * If the DefaultActionInvocation has been executed before and the Result is an instance of ActionChainResult, th
 * will walk down the chain of ActionChainResults until it finds a non-chain result, which will be returned. If t
DefaultActionInvocation › getResult()
```

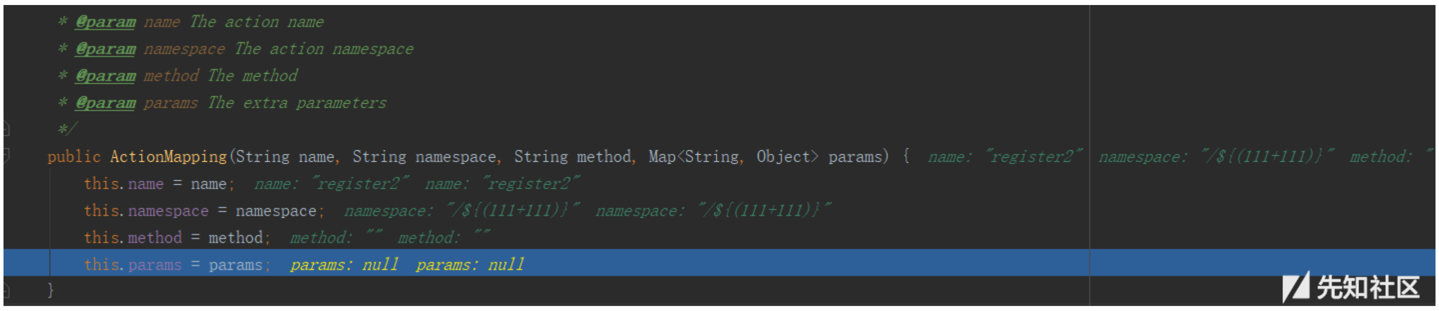
```
mcat Catalina Log
Variables
phony.xwork2)
.struts2.result)
nsymphony.xwork
hony.xwork2)

unknownHandlerManager = {DefaultUnknownHandlerManager@6904}
localizedTextProvider = {StrutsLocalizedTextProvider@6928}
actionName = "actionChain1"
namespace = "/${(111+111)}"
method = "execute"
executeResult = true
```

所以执行后对于result对象的namespace即为`/${(111+111)}`。

同一函数中继续执行 172行

```
public void execute(ActionInvocation invocation) throws Exception {
   ...

   String tmpLocation = actionMapper.getUriFromActionMapping(new ActionMapping(actionName, namespace, method, null));

   setLocation(tmpLocation);

   super.execute(invocation);
}
```

`ActionMapping`生成如下，`this.namespace`值赋为`/${(111+111)}`：

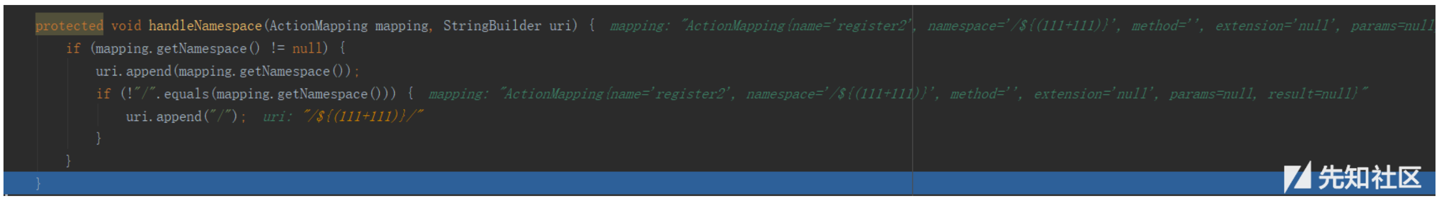


```
     * @param name The action name
     * @param namespace The action namespace
     * @param method The method
     * @param params The extra parameters
     */
    public ActionMapping(String name, String namespace, String method, Map<String, Object> params) {  name: "register2"  namespace: "/${(111+111)}"  method: "
        this.name = name;  name: "register2"  name: "register2"
        this.namespace = namespace;  namespace: "/${(111+111)}"  namespace: "/${(111+111)}"
        this.method = method;  method: ""  method: ""
        this.params = params;  params: null  params: null
    }
```

跟入getUriFromActionMapping:

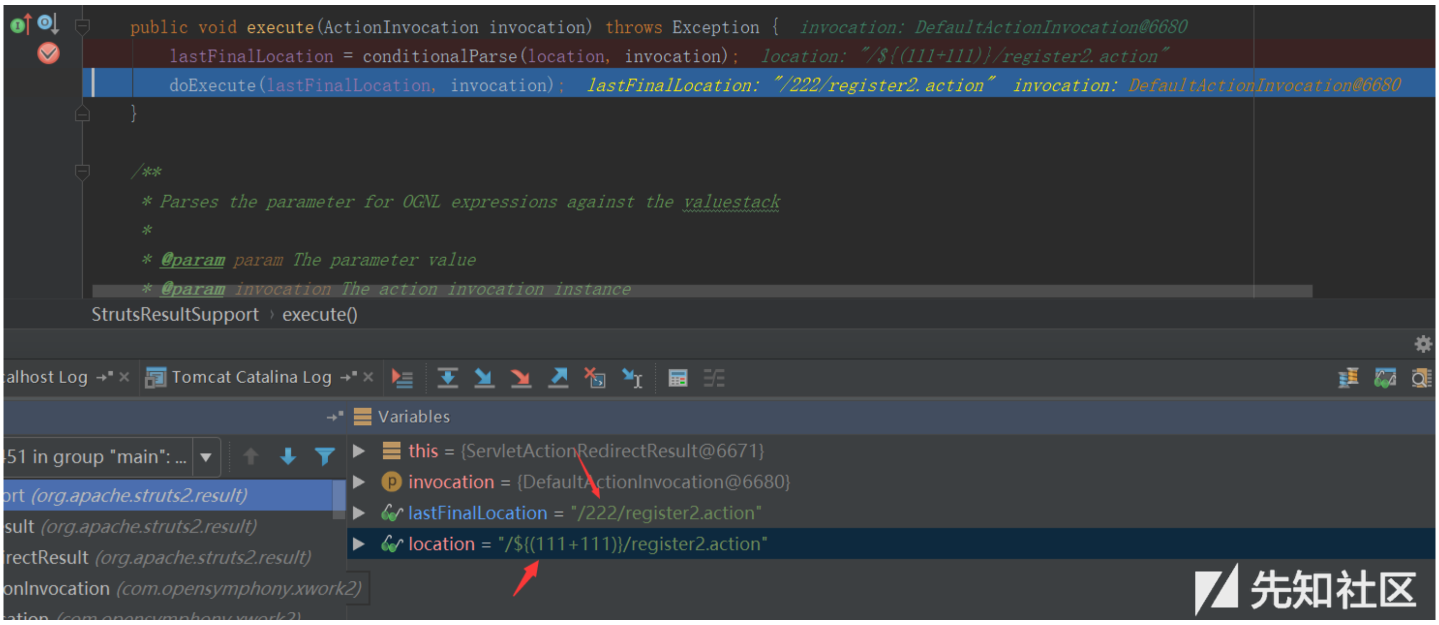
```
public String getUriFromActionMapping(ActionMapping mapping) {
        StringBuilder uri = new StringBuilder();

        handleNamespace(mapping, uri);
        handleName(mapping, uri);
        handleDynamicMethod(mapping, uri);
        handleExtension(mapping, uri);
        handleParams(mapping, uri);

        return uri.toString();
    }
```

handleNamespace处理结果如下：



```
    protected void handleNamespace(ActionMapping mapping, StringBuilder uri) {  mapping: "ActionMapping{name='register2', namespace='/${(111+111)}', method='', extension='null', params=null
        if (mapping.getNamespace() != null) {
            uri.append(mapping.getNamespace());
            if (!"/".equals(mapping.getNamespace())) {  mapping: "ActionMapping{name='register2', namespace='/${(111+111)}', method='', extension='null', params=null, result=null}"
                uri.append("/");  uri: "/${(111+111)}/"
            }
        }
    }
```

当函数返回，tmpLocation值为/${(111+111)}/register2.action，然后通过setLocation(tmpLocation)使得location变量值为/${(111+111)}/register2



```
    public void execute(ActionInvocation invocation) throws Exception {  invocation: DefaultActionInvocation@6680
        lastFinalLocation = conditionalParse(location, invocation);  location: "/${(111+111)}/register2.action"
        doExecute(lastFinalLocation, invocation);  lastFinalLocation: "/222/register2.action"  invocation: DefaultActionInvocation@6680
    }

    /**
     * Parses the parameter for OGNL expressions against the valuestack
     *
     * @param param The parameter value
     * @param invocation The action invocation instance
```

StrutsResultSupport › execute()

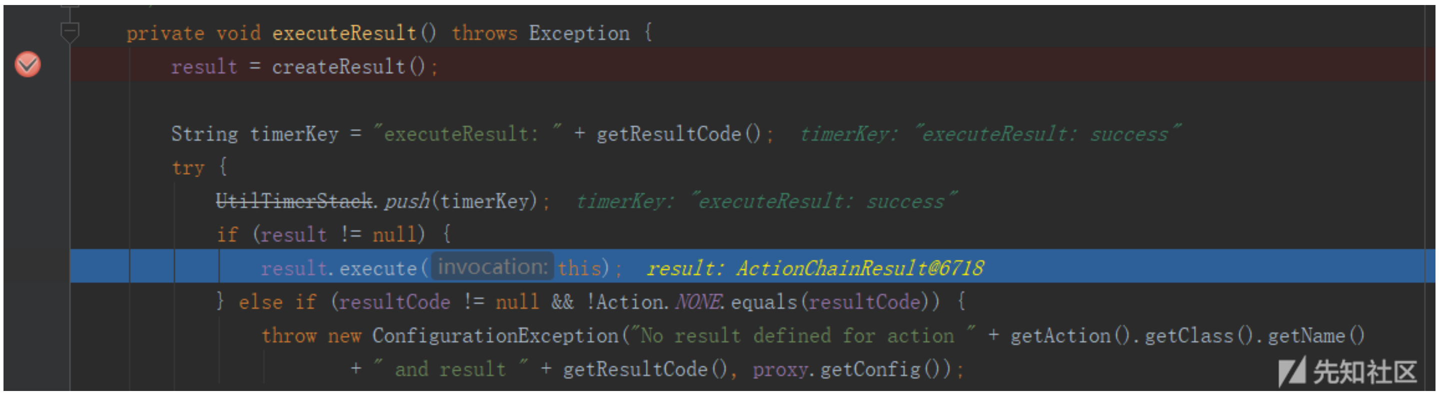
```
this = {ServletActionRedirectResult@6671}
invocation = {DefaultActionInvocation@6680}
lastFinalLocation = "/222/register2.action"
location = "/${(111+111)}/register2.action"
```

alhost Log
Tomcat Catalina Log
Variables
51 in group "main": ...
rt (org.apache.struts2.result)
sult (org.apache.struts2.result)
rectResult (org.apache.struts2.result)
onInvocation (com.opensymphony.xwork2)

攻击点二： Action chaining

apps/showcase/src/main/resources/struts-actionchaining.xml 中注意<result>标签中<type>为chain：

```
<result type="chain">
   <param name = "actionName">register2</param>
</result>
```

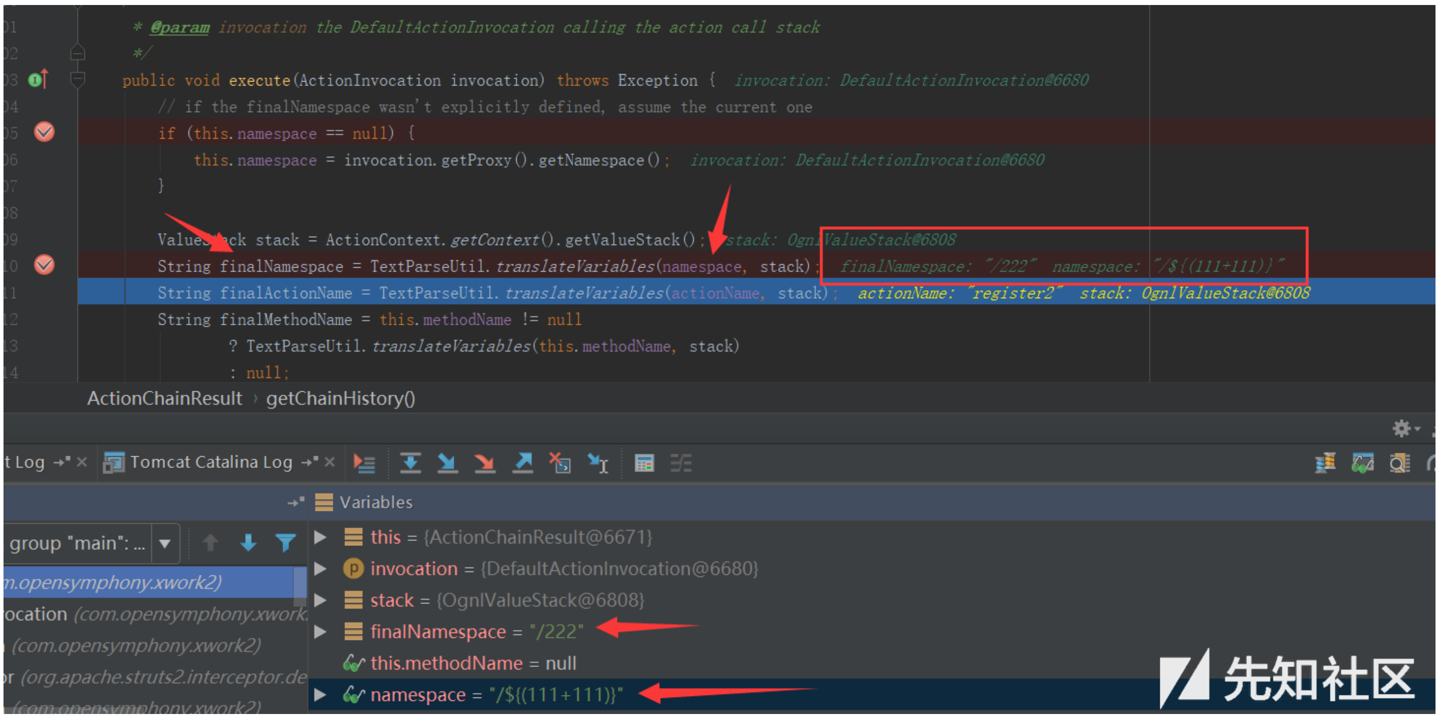


```
private void executeResult() throws Exception {
    result = createResult();

    String timerKey = "executeResult: " + getResultCode();  timerKey: "executeResult: success"
    try {
        ~~UtilTimerStack~~.push(timerKey);  timerKey: "executeResult: success"
        if (result != null) {
            result.execute( invocation: this);  result: ActionChainResult@6718
        } else if (resultCode != null && !Action.NONE.equals(resultCode)) {
            throw new ConfigurationException("No result defined for action " + getAction().getClass().getName()
                + " and result " + getResultCode(), proxy.getConfig());
```

同样会先经过`result = createResult()`，然后调用`result.execute(this);`。这会进入到 com/opensymphony/xwork2/ActionChainResult.java:203

```
public void execute(ActionInvocation invocation) throws Exception {
    // if the finalNamespace wasn't explicitly defined, assume the current one
    if (this.namespace == null) {
        this.namespace = invocation.getProxy().getNamespace();
    }

    ValueStack stack = ActionContext.getContext().getValueStack();
    String finalNamespace = TextParseUtil.translateVariables(namespace, stack);
    String finalActionName = TextParseUtil.translateVariables(actionName, stack);
    ...
}
```

由于没有设定namespace，所以通过`invocation.getProxy().getNamespace()`使得`this.namespace`值为`/${(111+111)}`。然后调用了`String finalNamespace = TextParseUtil.translateVariables(namespace, stack);`对namespace进行OGNL解析。如下



```
     * @param invocation the DefaultActionInvocation calling the action call stack
     */
    public void execute(ActionInvocation invocation) throws Exception {  invocation: DefaultActionInvocation@6680
        // if the finalNamespace wasn't explicitly defined, assume the current one
        if (this.namespace == null) {
            this.namespace = invocation.getProxy().getNamespace();  invocation: DefaultActionInvocation@6680
        }

        ValueStack stack = ActionContext.getContext().getValueStack();  stack: OgnlValueStack@6808
        String finalNamespace = TextParseUtil.translateVariables(namespace, stack);  finalNamespace: "/222"  namespace: "/${(111+111)}"
        String finalActionName = TextParseUtil.translateVariables(actionName, stack);  actionName: "register2"  stack: OgnlValueStack@6808
        String finalMethodName = this.methodName != null
                ? TextParseUtil.translateVariables(this.methodName, stack)
                : null;
ActionChainResult › getChainHistory()
```

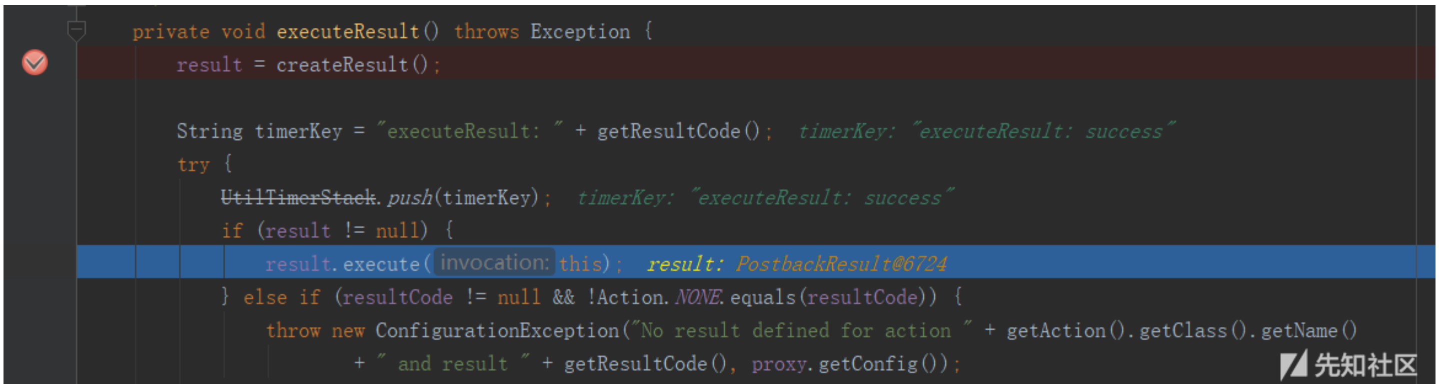
```
Variables
this = {ActionChainResult@6671}
invocation = {DefaultActionInvocation@6680}
stack = {OgnlValueStack@6808}
finalNamespace = "/222"
this.methodName = null
namespace = "/${(111+111)}"
```

t Log
Tomcat Catalina Log
group "main": ...
n.opensymphony.xwork2)
ocation (com.opensymphony.xwork
(com.opensymphony.xwork2)
r (org.apache.struts2.interceptor.de
(com.opensymphony.xwork2)

攻击点三：[Postback result]()

apps/showcase/src/main/resources/struts-actionchaining.xml 中注意`<result>`标签中`<type>`为`postback`：

```
<result type="postback">
    <param name = "actionName">register2</param>
</result>
```



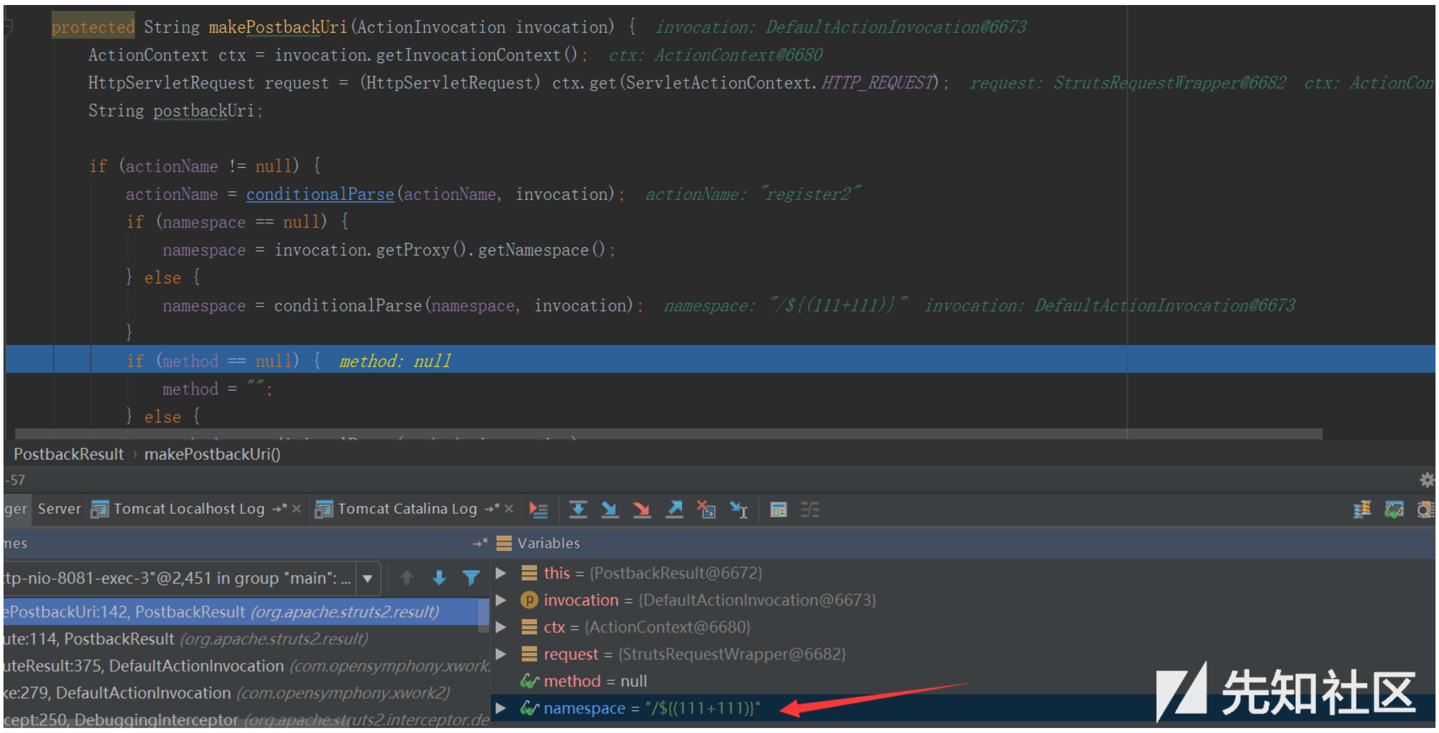
```
private void executeResult() throws Exception {
    result = createResult();

    String timerKey = "executeResult: " + getResultCode();  timerKey: "executeResult: success"
    try {
        ~~UtilTimerStack~~.push(timerKey);  timerKey: "executeResult: success"
        if (result != null) {
            result.execute( invocation: this);  result: PostbackResult@6724
        } else if (resultCode != null && !Action.NONE.equals(resultCode)) {
            throw new ConfigurationException("No result defined for action " + getAction().getClass().getName()
                    + " and result " + getResultCode(), proxy.getConfig());
```

经过`result = createResult()`，跟入定位到`postback`这个result对象的处理方法，在 org/apache/struts2/result/PostbackResult.java:113

```
@Override
public void execute(ActionInvocation invocation) throws Exception {
    String postbackUri = makePostbackUri(invocation);
    setLocation(postbackUri);
    super.execute(invocation);
}
```

跟入`makePostbackUri1`，在org/apache/struts2/result/PostbackResult.java:129

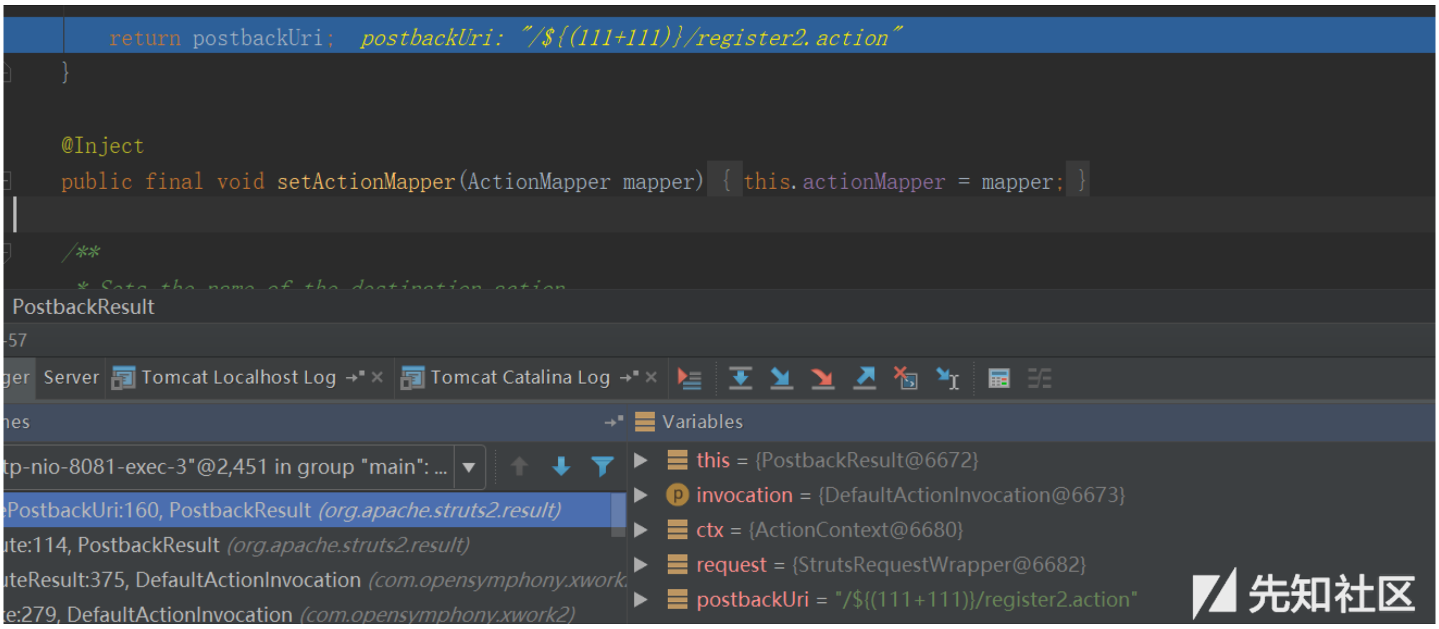
```
protected String makePostbackUri(ActionInvocation invocation) {
    ActionContext ctx = invocation.getInvocationContext();
    HttpServletRequest request = (HttpServletRequest) ctx.get(ServletActionContext.HTTP_REQUEST);
    String postbackUri;

    if (actionName != null) {
        actionName = conditionalParse(actionName, invocation);
        if (namespace == null) {
            namespace = invocation.getProxy().getNamespace();
        } else {
            namespace = conditionalParse(namespace, invocation);
        }
        ...
        postbackUri = request.getContextPath() + actionMapper.getUriFromActionMapping(new ActionMapping(actionName, namespace,
    }
    ...

    return postbackUri;
}
```



```
protected String makePostbackUri(ActionInvocation invocation) {  invocation: DefaultActionInvocation@6673
    ActionContext ctx = invocation.getInvocationContext();  ctx: ActionContext@6680
    HttpServletRequest request = (HttpServletRequest) ctx.get(ServletActionContext.HTTP_REQUEST);  request: StrutsRequestWrapper@6682  ctx: ActionCon
    String postbackUri;

    if (actionName != null) {
        actionName = conditionalParse(actionName, invocation);  actionName: "register2"
        if (namespace == null) {
            namespace = invocation.getProxy().getNamespace();
        } else {
            namespace = conditionalParse(namespace, invocation);  namespace: "/${(111+111)}"  invocation: DefaultActionInvocation@6673
        }
        if (method == null) {  method: null
            method = "";
        } else {
```

```
PostbackResult › makePostbackUri()

tp-nio-8081-exec-3"@2,451 in group "main": ...
ePostbackUri:142, PostbackResult (org.apache.struts2.result)
ute:114, PostbackResult (org.apache.struts2.result)
uteResult:375, DefaultActionInvocation (com.opensymphony.xwork
ke:279, DefaultActionInvocation (com.opensymphony.xwork2)
cept:250, DebuggingInterceptor (org.apache.struts2.interceptor.de

Variables
this = {PostbackResult@6672}
invocation = {DefaultActionInvocation@6673}
ctx = {ActionContext@6680}
request = {StrutsRequestWrapper@6682}
method = null
namespace = "/${(111+111)}"
```

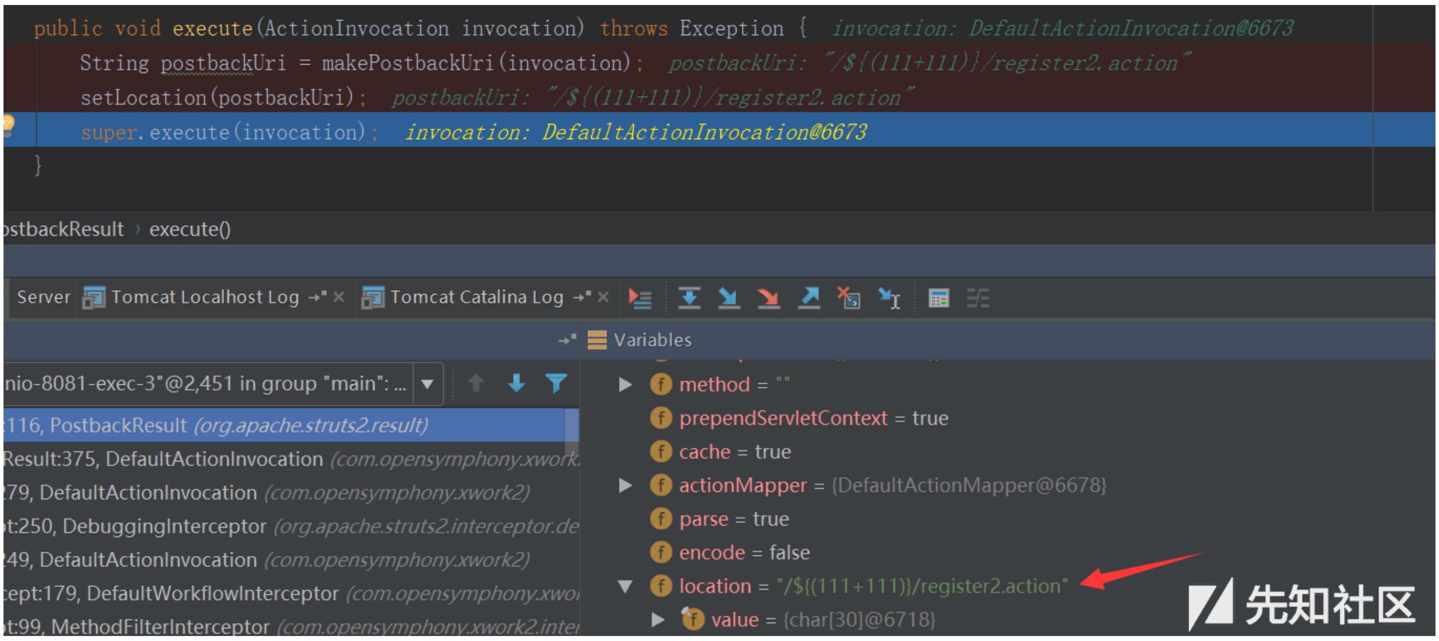
获取到namespace值为`/${(111+111)}`。跟入`actionMapper.getUriFromActionMapping(new ActionMapping(actionName, namespace, method, null))`，其具体执行过程如攻击点一[Redirect action]提到的那样，设置namespace等参数，然后从`getUriFromActionMapping`中返回uri。最后组装的postbackUri为`/${(111+111)}/register2.action`



```
        return postbackUri;   postbackUri: "/${(111+111)}/register2.action"
    }

    @Inject
    public final void setActionMapper(ActionMapper mapper) { this.actionMapper = mapper; }

    /**
     * Sets the name of the destination action
```

```
PostbackResult
-57
ger  Server  Tomcat Localhost Log  Tomcat Catalina Log
es                                                   Variables
tp-nio-8081-exec-3"@2,451 in group "main": ...       ▶ this = {PostbackResult@6672}
PostbackUri:160, PostbackResult (org.apache.struts2.result)      ▶ invocation = {DefaultActionInvocation@6673}
ute:114, PostbackResult (org.apache.struts2.result)             ▶ ctx = {ActionContext@6680}
uteResult:375, DefaultActionInvocation (com.opensymphony.xwork.  ▶ request = {StrutsRequestWrapper@6682}
e:279, DefaultActionInvocation (com.opensymphony.xwork2)        ▶ postbackUri = "/${(111+111)}/register2.action"
```

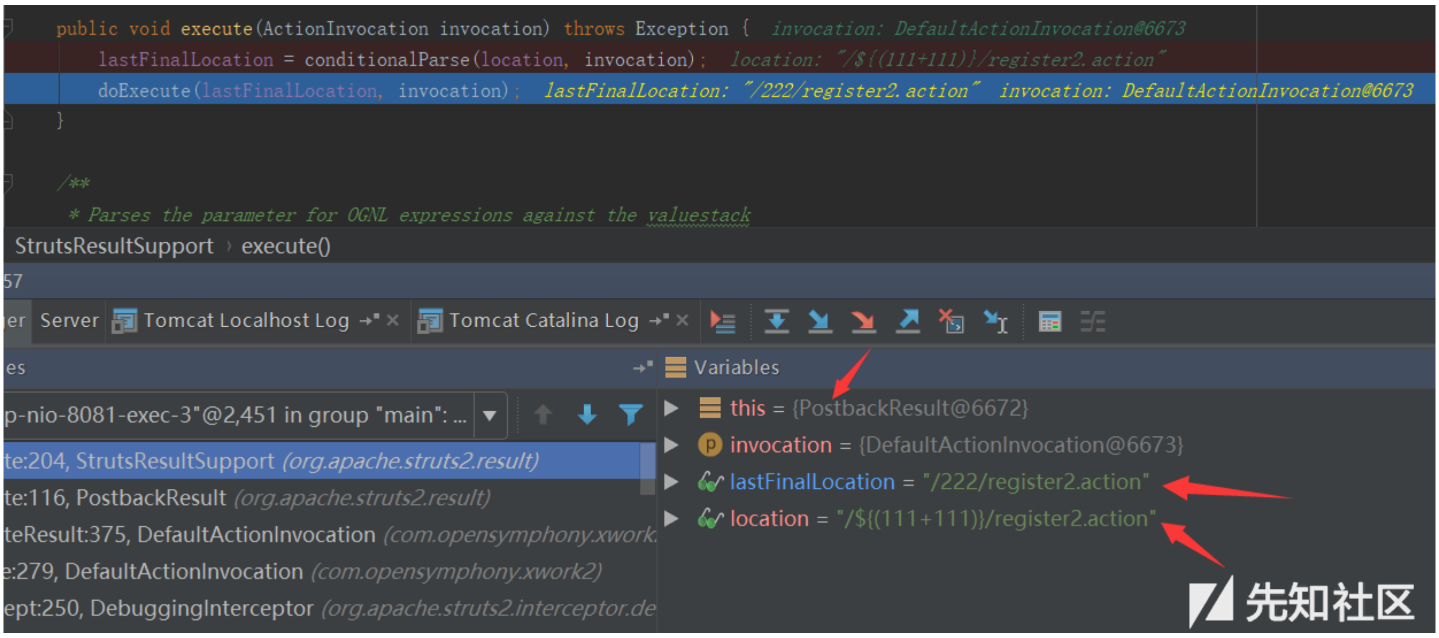
回到前面的execute中通过`setLocation(postbackUri)`设置了location变量：



```
    public void execute(ActionInvocation invocation) throws Exception {   invocation: DefaultActionInvocation@6673
        String postbackUri = makePostbackUri(invocation);   postbackUri: "/${(111+111)}/register2.action"
        setLocation(postbackUri);   postbackUri: "/${(111+111)}/register2.action"
        super.execute(invocation);   invocation: DefaultActionInvocation@6673
    }
```

```
ostbackResult › execute()
Server  Tomcat Localhost Log  Tomcat Catalina Log
                                                     Variables
nio-8081-exec-3"@2,451 in group "main": ...          ▶ method = ""
116, PostbackResult (org.apache.struts2.result)        prependServletContext = true
Result:375, DefaultActionInvocation (com.opensymphony.xwork    cache = true
79, DefaultActionInvocation (com.opensymphony.xwork2)  ▶ actionMapper = {DefaultActionMapper@6678}
t:250, DebuggingInterceptor (org.apache.struts2.interceptor.de   parse = true
49, DefaultActionInvocation (com.opensymphony.xwork2)     encode = false
ept:179, DefaultWorkflowInterceptor (com.opensymphony.xwo ▼ location = "/${(111+111)}/register2.action"
t:99, MethodFilterInterceptor (com.opensymphony.xwork2.inte   ▶ value = {char[30]@6718}
```

此后location变量传入，造成OGNL表达式注入


```
public void execute(ActionInvocation invocation) throws Exception {  invocation: DefaultActionInvocation@6673
    lastFinalLocation = conditionalParse(location, invocation);  location: "/${(111+111)}/register2.action"
    doExecute(lastFinalLocation, invocation);  lastFinalLocation: "/222/register2.action"  invocation: DefaultActionInvocation@6673
}

/**
 * Parses the parameter for OGNL expressions against the valuestack
```

```
p-nio-8081-exec-3"@2,451 in group "main": ...
te:204, StrutsResultSupport (org.apache.struts2.result)
te:116, PostbackResult (org.apache.struts2.result)
teResult:375, DefaultActionInvocation (com.opensymphony.xwork
e:279, DefaultActionInvocation (com.opensymphony.xwork2)
ept:250, DebuggingInterceptor (org.apache.struts2.interceptor.de
```

```
this = {PostbackResult@6672}
invocation = {DefaultActionInvocation@6673}
lastFinalLocation = "/222/register2.action"
location = "/${(111+111)}/register2.action"
```

参考

- https://struts.apache.org/core-developers/namespace-configuration.html