登录

Struts漏洞 S2-045 调试学习

CoColi / 2019-04-02 09:51:00 / 浏览数 3694 安全技术 漏洞分析 顶(0) 踩(0)

S2-045 调试学习

漏洞版本

2.3.31-2.3.5 2.5-2.5.10

漏洞成因

content-type里有multipart/form-data就会走JakartaMultiPartRequest,捕捉了异常信息(里面带有payload),后又OGNL解析了

payload

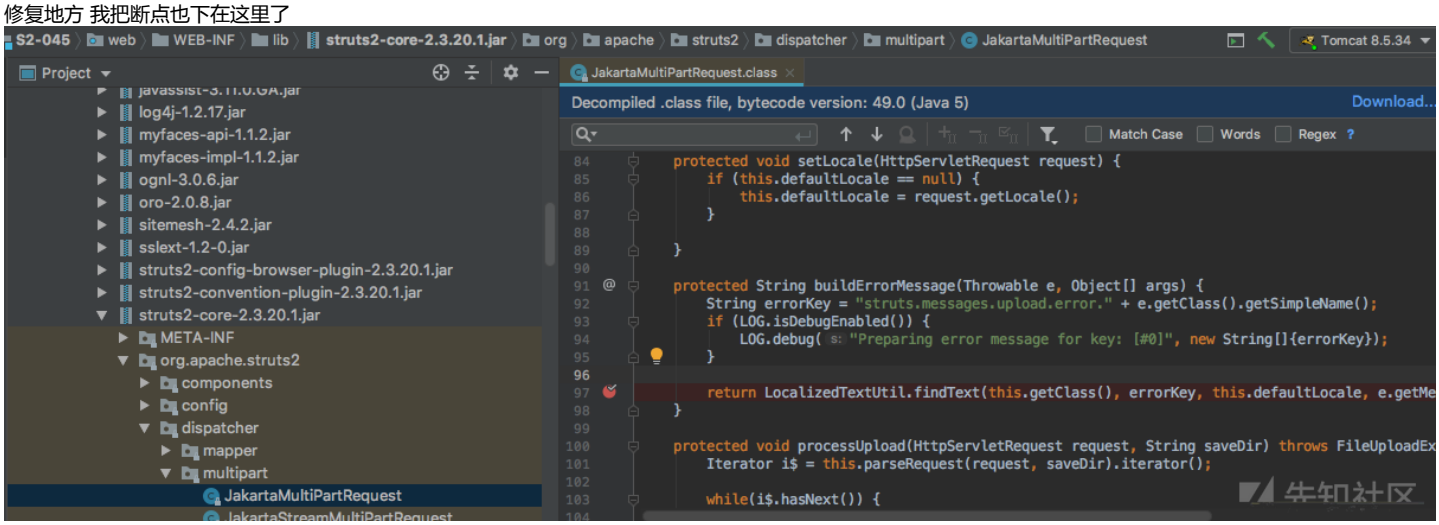
```
%{(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.x
```

从burp里面导的curl command

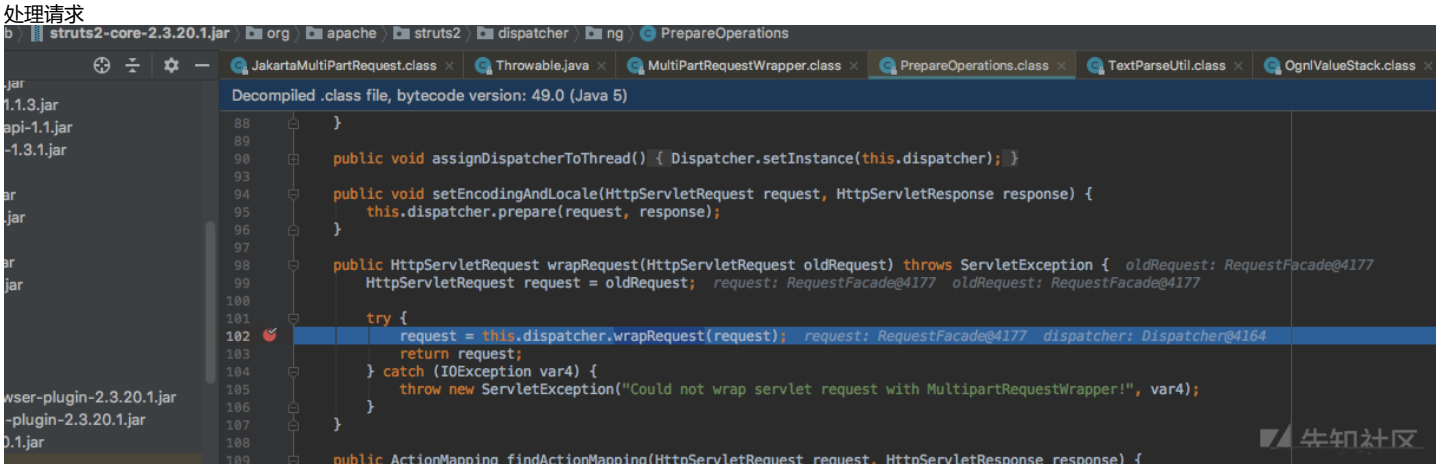
```
curl -i -s -k  -X $'POST' \
   -H $'Host: 192.168.95.1:8081' -H $'User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gec
   $'http://192.168.95.1:8081/S2_045_war_exploded/'
```

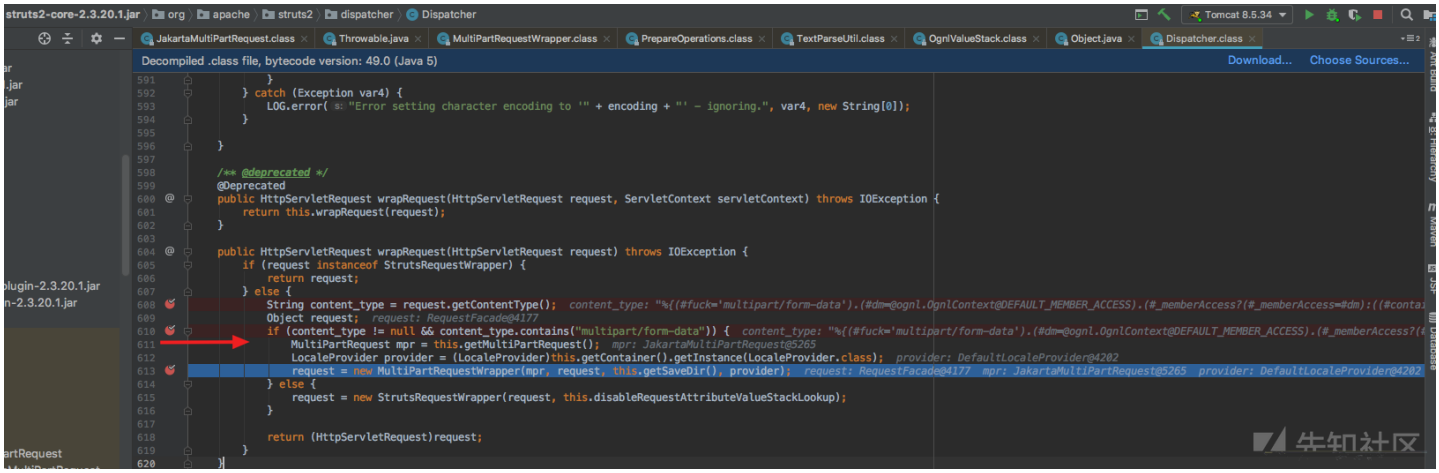
调试

修复地方 我把断点也下在这里了



都先经过web.xml 拦截 Struts2PrepareAndExecutefilter
处理请求



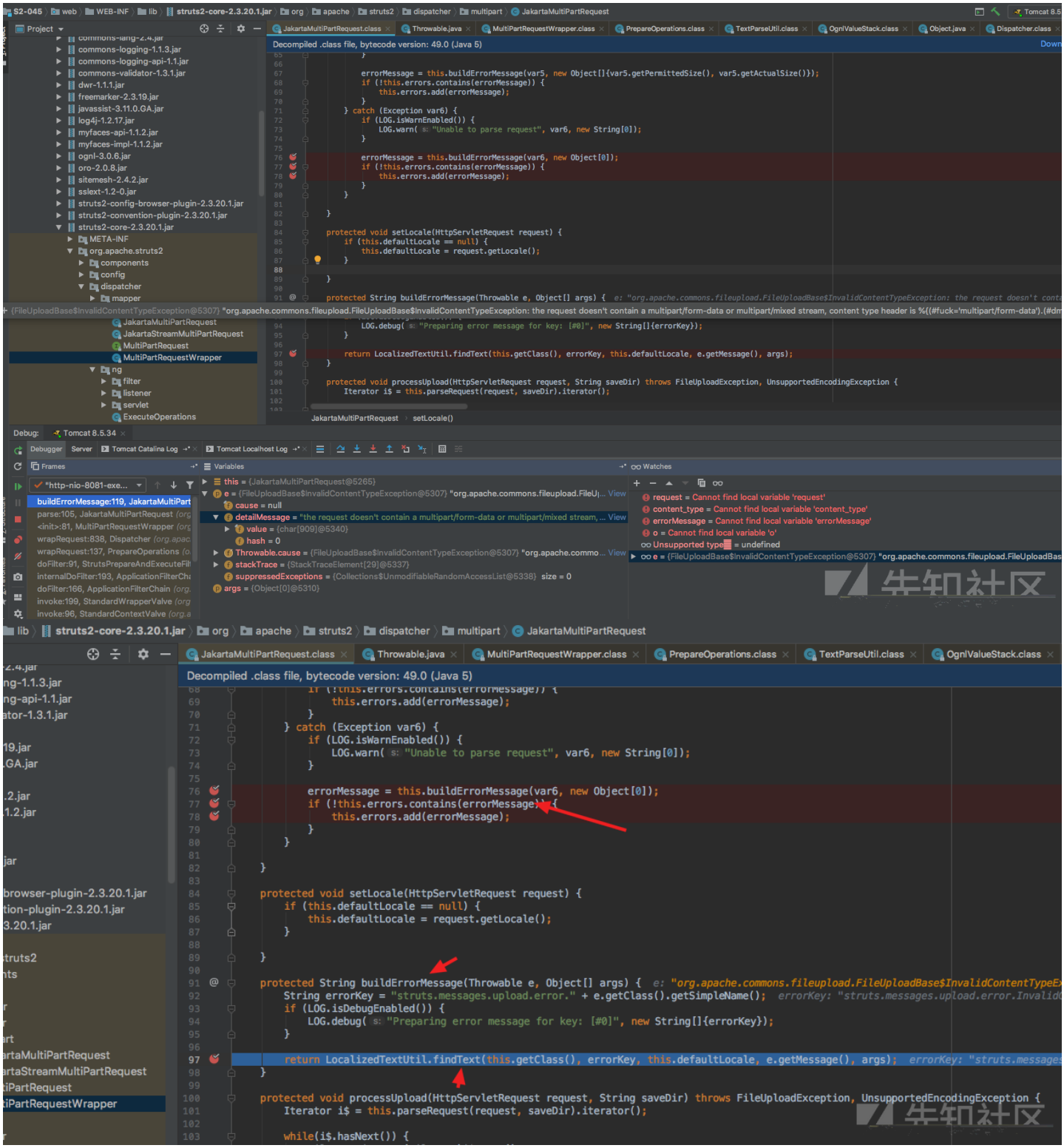


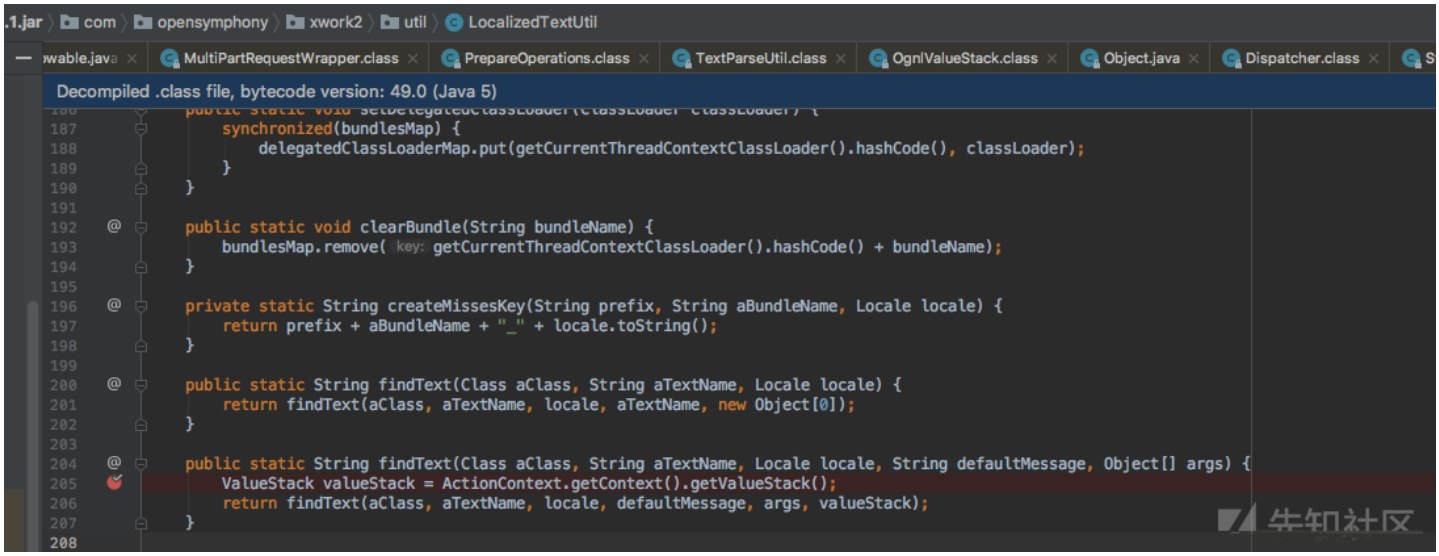
这个地方 获取ContentType

2-045的POC一般都有(#nike='multipart/form-data')这样一句，就是使content_type.contains("multipart/form-data")判断为true

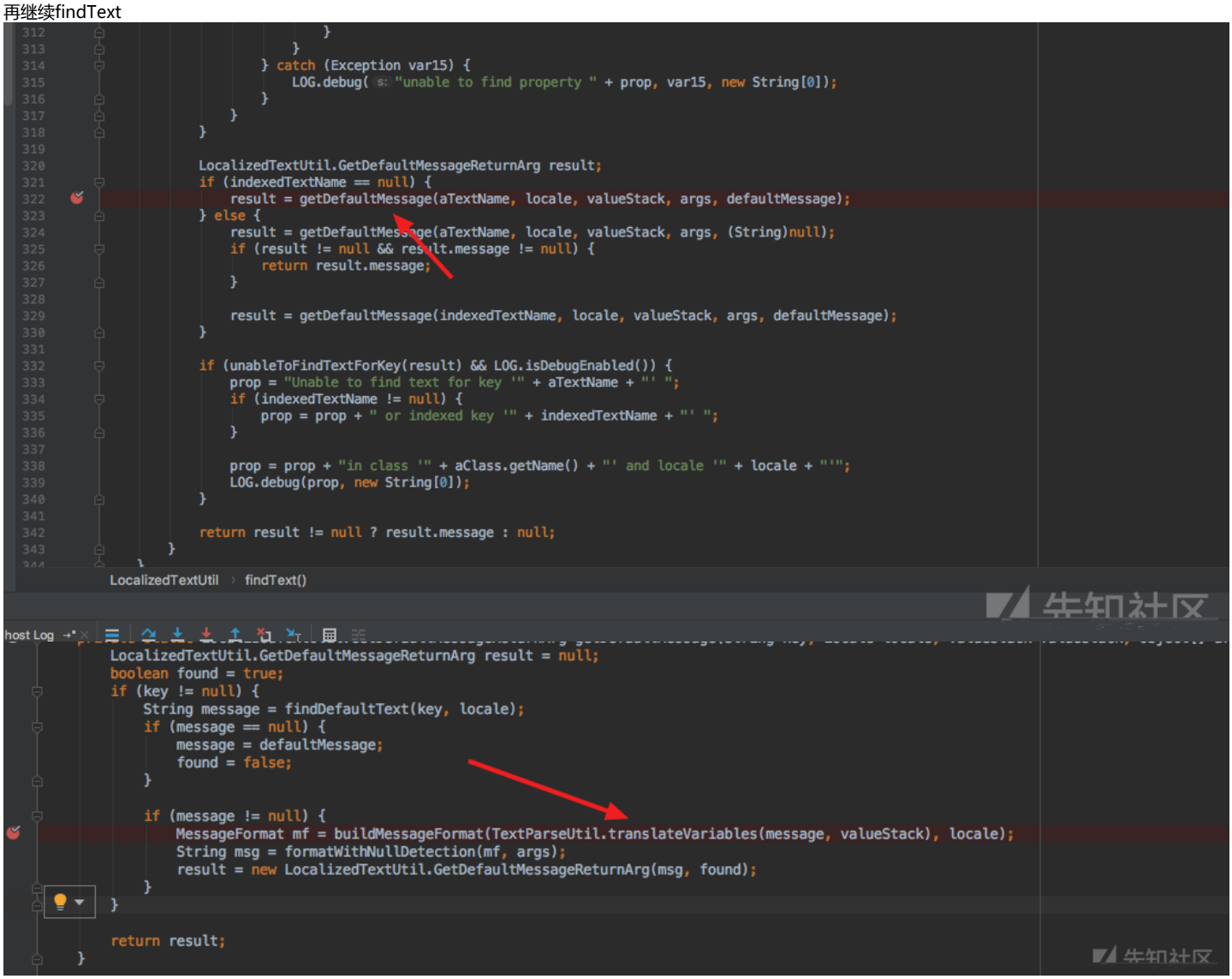
继续追踪getMultiPartRequest方法。通过配置struts.multipart.parser属性，可以指定不同的解析类，而默认就是org.apache.struts2.dispatcher.multipart.JakartaMultiP

这里已经可以看到是用JakartaMultiPartReques解析了

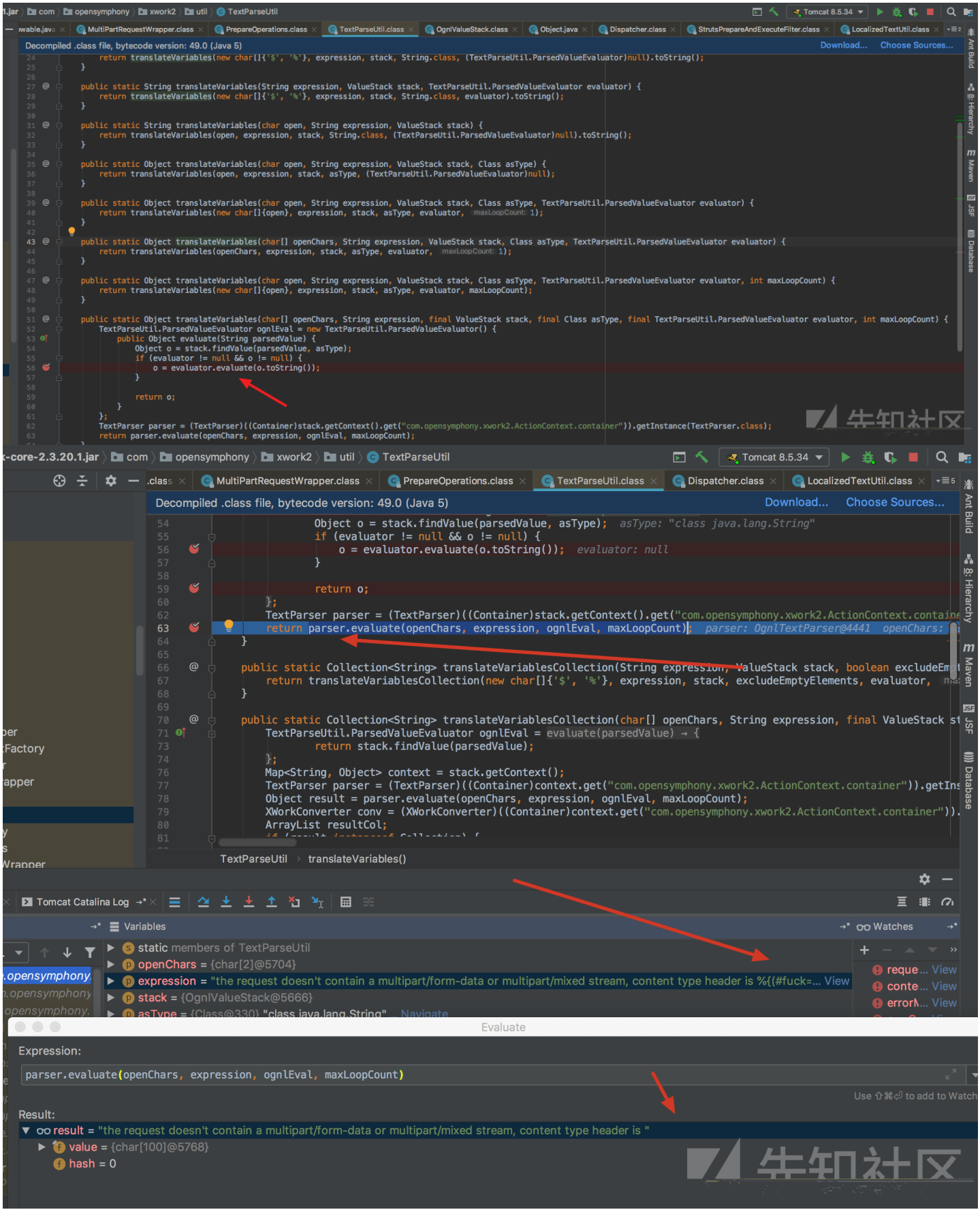

S2-045 web WEB-INF lib struts2-core-2.3.20.1.jar org apache struts2 dispatcher multipart JakartaMultiPartRequest
Project
JakartaMultiPartRequest.class
Throwable.java
MultiPartRequestWrapper.class
PrepareOperations.class
TextParseUtil.class
OgnlValueStack.class
Object.java
Dispatcher.class
Decompiled .class file, bytecode version: 49.0 (Java 5)
commons-lang-2.4.jar
commons-logging-1.1.3.jar
commons-logging-api-1.1.jar
commons-validator-1.3.1.jar
dwr-1.1.1.jar
freemarker-2.3.19.jar
javassist-3.11.0.GA.jar
log4j-1.2.17.jar
myfaces-api-1.1.2.jar
myfaces-impl-1.1.2.jar
ognl-3.0.6.jar
oro-2.0.8.jar
sitemesh-2.4.2.jar
sslext-1.2-0.jar
struts2-config-browser-plugin-2.3.20.1.jar
struts2-convention-plugin-2.3.20.1.jar
struts2-core-2.3.20.1.jar
META-INF
org.apache.struts2
components
config
dispatcher
mapper
JakartaMultiPartRequest
JakartaStreamMultiPartRequest
MultiPartRequest
MultiPartRequestWrapper
ng
filter
listener
servlet
ExecuteOperations
errorMessage = this.buildErrorMessage(var5, new Object[]{var5.getPermittedSize(), var5.getActualSize()});
if (!this.errors.contains(errorMessage)) {
this.errors.add(errorMessage);
} catch (Exception var6) {
if (LOG.isWarnEnabled()) {
LOG.warn(s: "Unable to parse request", var6, new String[0]);
errorMessage = this.buildErrorMessage(var6, new Object[0]);
protected void setLocale(HttpServletRequest request) {
if (this.defaultLocale == null) {
this.defaultLocale = request.getLocale();
protected String buildErrorMessage(Throwable e, Object[] args) {
LOG.debug(s: "Preparing error message for key: [#0]", new String[]{errorKey});
return LocalizedTextUtil.findText(this.getClass(), errorKey, this.defaultLocale, e.getMessage(), args);
protected void processUpload(HttpServletRequest request, String saveDir) throws FileUploadException, UnsupportedEncodingException {
Iterator i$ = this.parseRequest(request, saveDir).iterator();
JakartaMultiPartRequest › setLocale()
Debug: Tomcat 8.5.34
Debugger
Server
Tomcat Catalina Log
Tomcat Localhost Log
Frames
Variables
Watches
"http-nio-8081-exe...
buildErrorMessage:119, JakartaMultiPart
parse:105, JakartaMultiPartRequest
<init>:81, MultiPartRequestWrapper
wrapRequest:838, Dispatcher
wrapRequest:137, PrepareOperations
doFilter:91, StrutsPrepareAndExecuteFil
internalDoFilter:193, ApplicationFilterCha
doFilter:166, ApplicationFilterChain
invoke:199, StandardWrapperValve
invoke:96, StandardContextValve
this = {JakartaMultiPartRequest@5265}
e = {FileUploadBase$InvalidContentTypeException@5307} "org.apache.commons.fileupload.FileU... View
cause = null
detailMessage = "the request doesn't contain a multipart/form-data or multipart/mixed stream, ... View
value = {char[909]@5340}
hash = 0
Throwable.cause = {FileUploadBase$InvalidContentTypeException@5307} "org.apache.commo... View
stackTrace = {StackTraceElement[29]@5337}
suppressedExceptions = {Collections$UnmodifiableRandomAccessList@5338} size = 0
args = {Object[0]@5310}
request = Cannot find local variable 'request'
content_type = Cannot find local variable 'content_type'
errorMessage = Cannot find local variable 'errorMessage'
o = Cannot find local variable 'o'
Unsupported type = undefined
e = {FileUploadBase$InvalidContentTypeException@5307} "org.apache.commons.fileupload.FileUploadBas
lib struts2-core-2.3.20.1.jar org apache struts2 dispatcher multipart JakartaMultiPartRequest
JakartaMultiPartRequest.class
Throwable.java
MultiPartRequestWrapper.class
PrepareOperations.class
TextParseUtil.class
OgnlValueStack.class
Decompiled .class file, bytecode version: 49.0 (Java 5)
ng-1.1.3.jar
ng-api-1.1.jar
ator-1.3.1.jar
19.jar
GA.jar
.2.jar
1.2.jar
jar
browser-plugin-2.3.20.1.jar
tion-plugin-2.3.20.1.jar
3.20.1.jar
truts2
nts
rt
artaMultiPartRequest
artaStreamMultiPartRequest
iPartRequest
iPartRequestWrapper
if (!this.errors.contains(errorMessage)) {
this.errors.add(errorMessage);
} catch (Exception var6) {
if (LOG.isWarnEnabled()) {
LOG.warn(s: "Unable to parse request", var6, new String[0]);
errorMessage = this.buildErrorMessage(var6, new Object[0]);
if (!this.errors.contains(errorMessage)) {
this.errors.add(errorMessage);
protected void setLocale(HttpServletRequest request) {
if (this.defaultLocale == null) {
this.defaultLocale = request.getLocale();
protected String buildErrorMessage(Throwable e, Object[] args) {
String errorKey = "struts.messages.upload.error." + e.getClass().getSimpleName();
if (LOG.isDebugEnabled()) {
LOG.debug(s: "Preparing error message for key: [#0]", new String[]{errorKey});
return LocalizedTextUtil.findText(this.getClass(), errorKey, this.defaultLocale, e.getMessage(), args);
protected void processUpload(HttpServletRequest request, String saveDir) throws FileUploadException, UnsupportedEncodingException {
Iterator i$ = this.parseRequest(request, saveDir).iterator();
while(i$.hasNext()) {



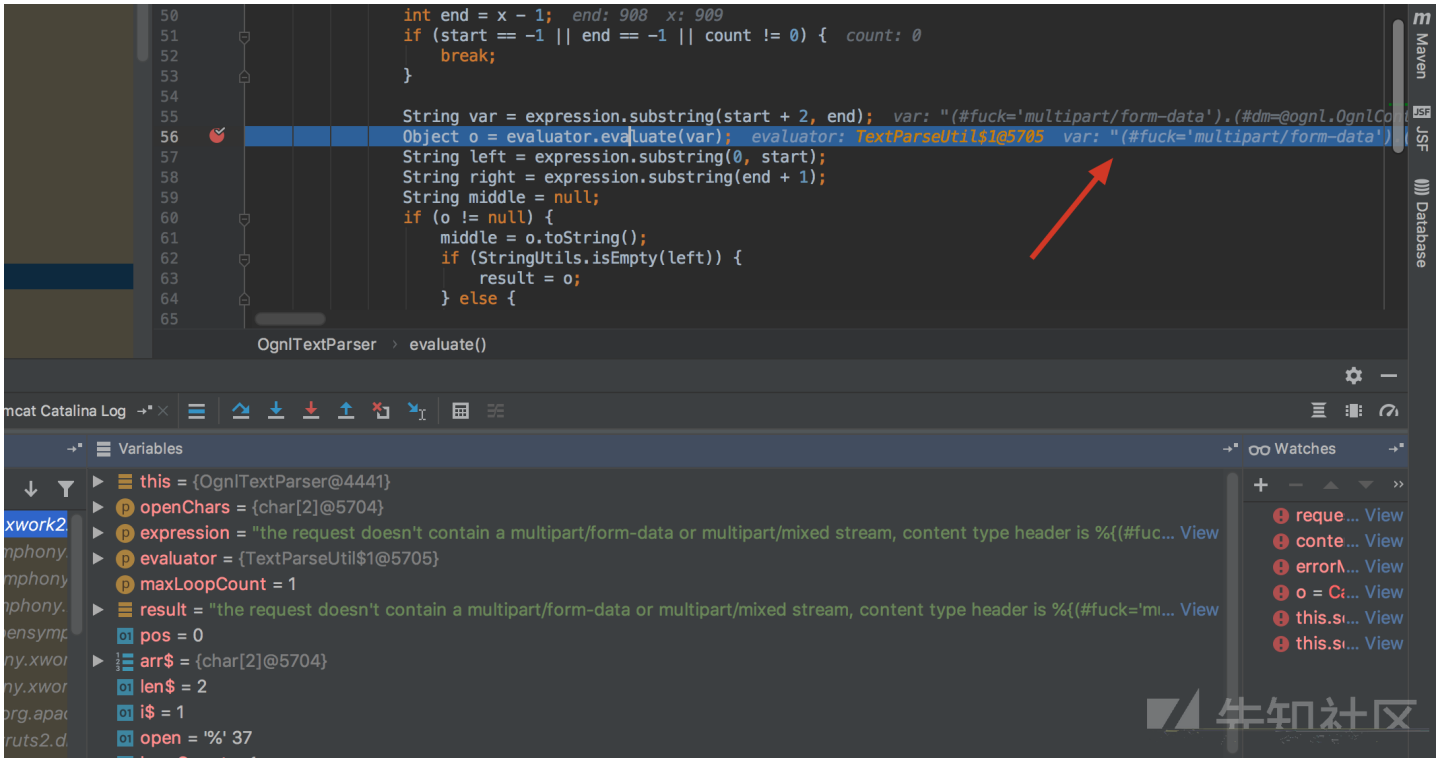
再继续findText



com.opensymphony.xwork2.util.TextParseUtil.translateVariables(String, ValueStack) 方法主要用于扩展字符串中由 ${} 或 %{} 包裹的 OGNL 表达式，这里也就是 OGNL 的入口，随后 action message 将进入 OGNL 的处理流程，漏洞被触发。

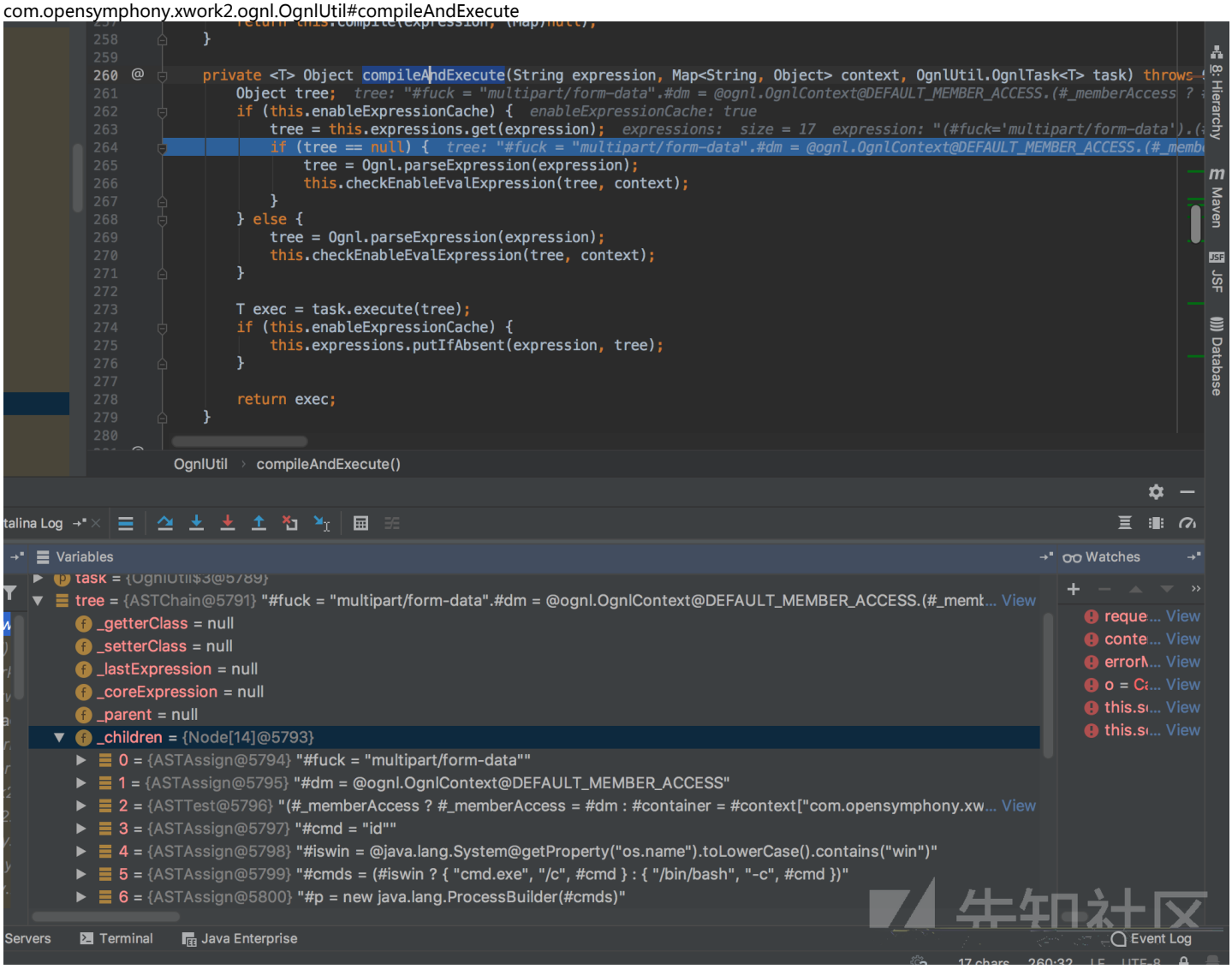


下一步 又继续 解析
com.opensymphony.xwork2.util.TextParseUtil.ParsedValueEvaluator#evaluate

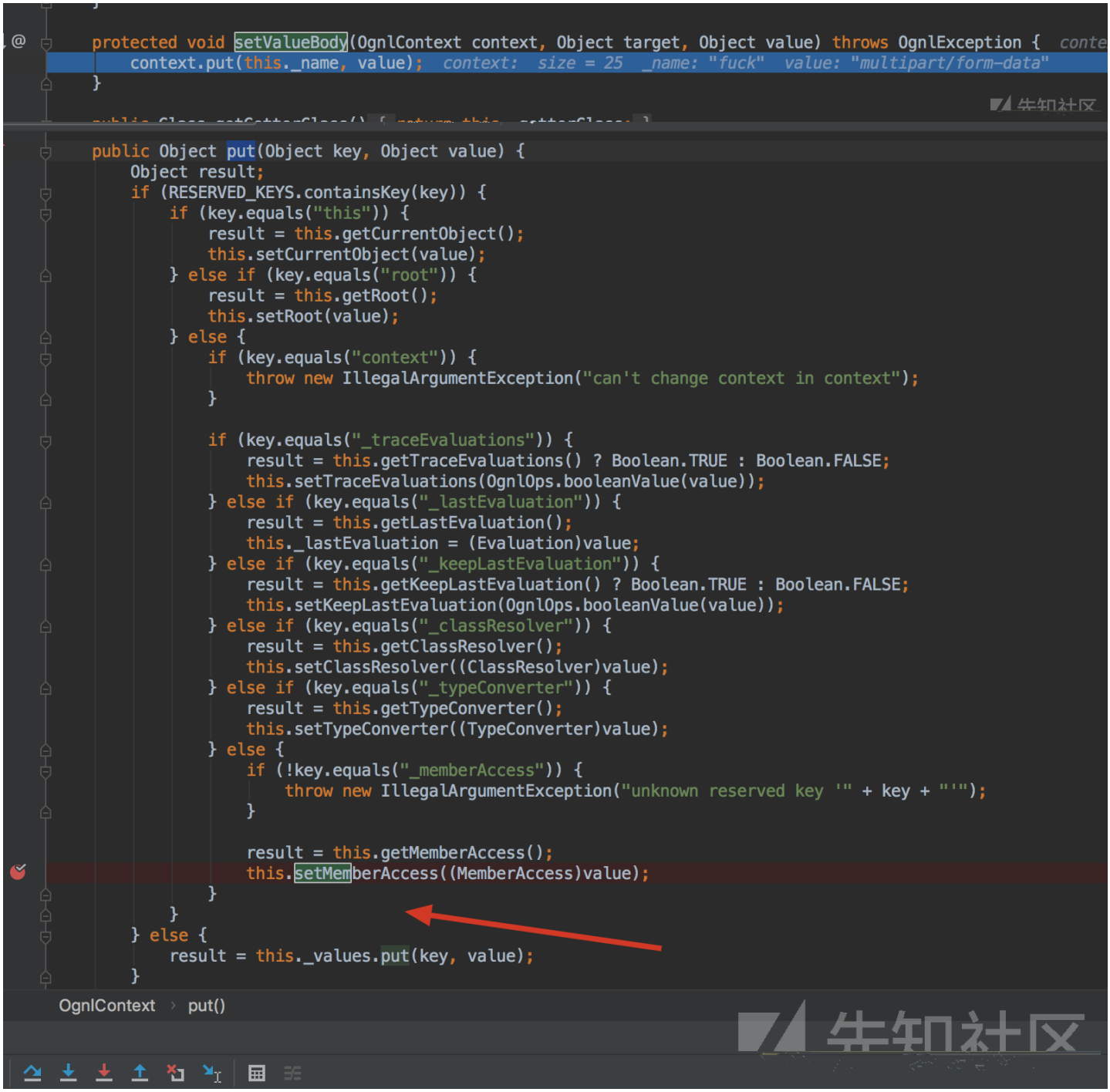


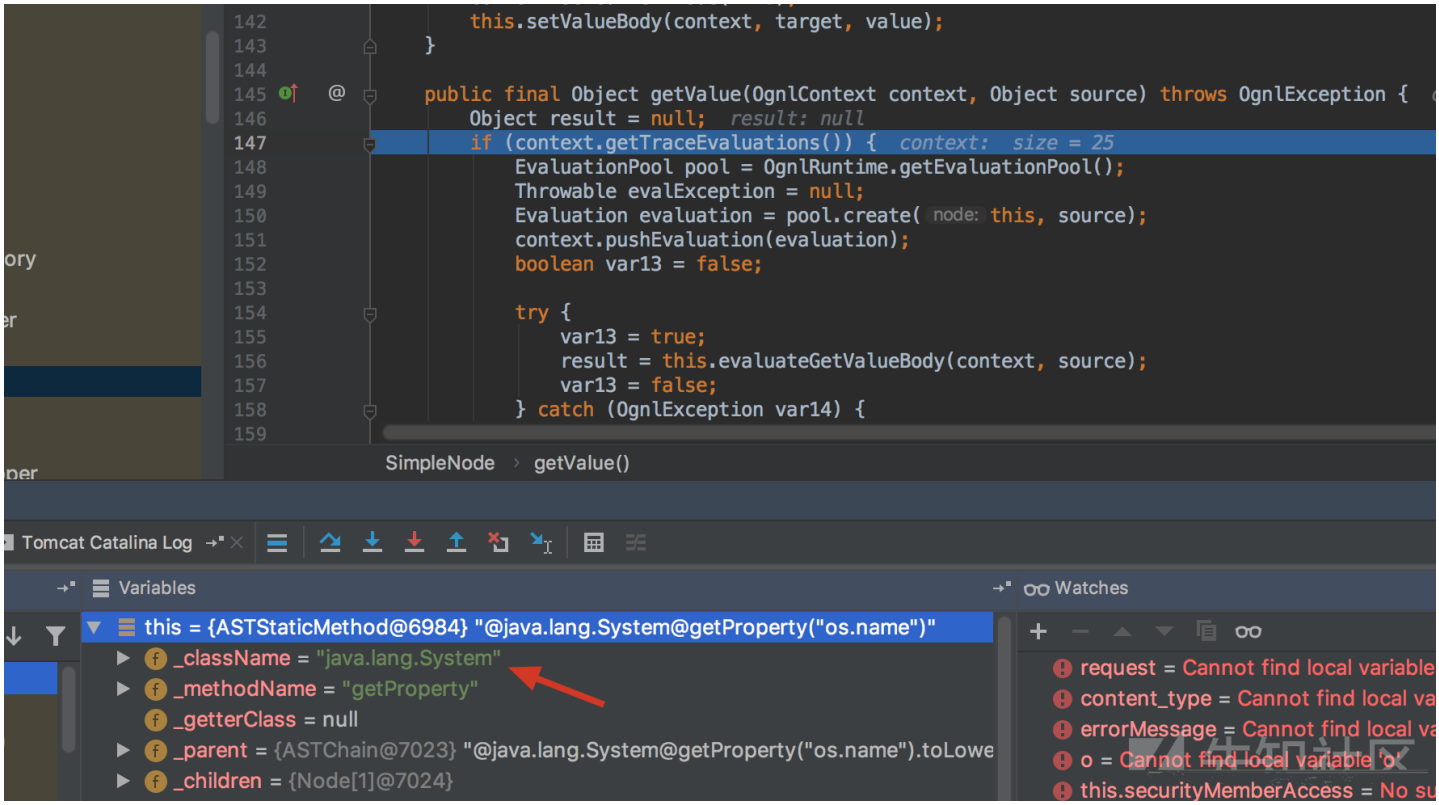
继续跟 发现进了语法树

com.opensymphony.xwork2.ognl.OgnlUtil#compileAndExecute

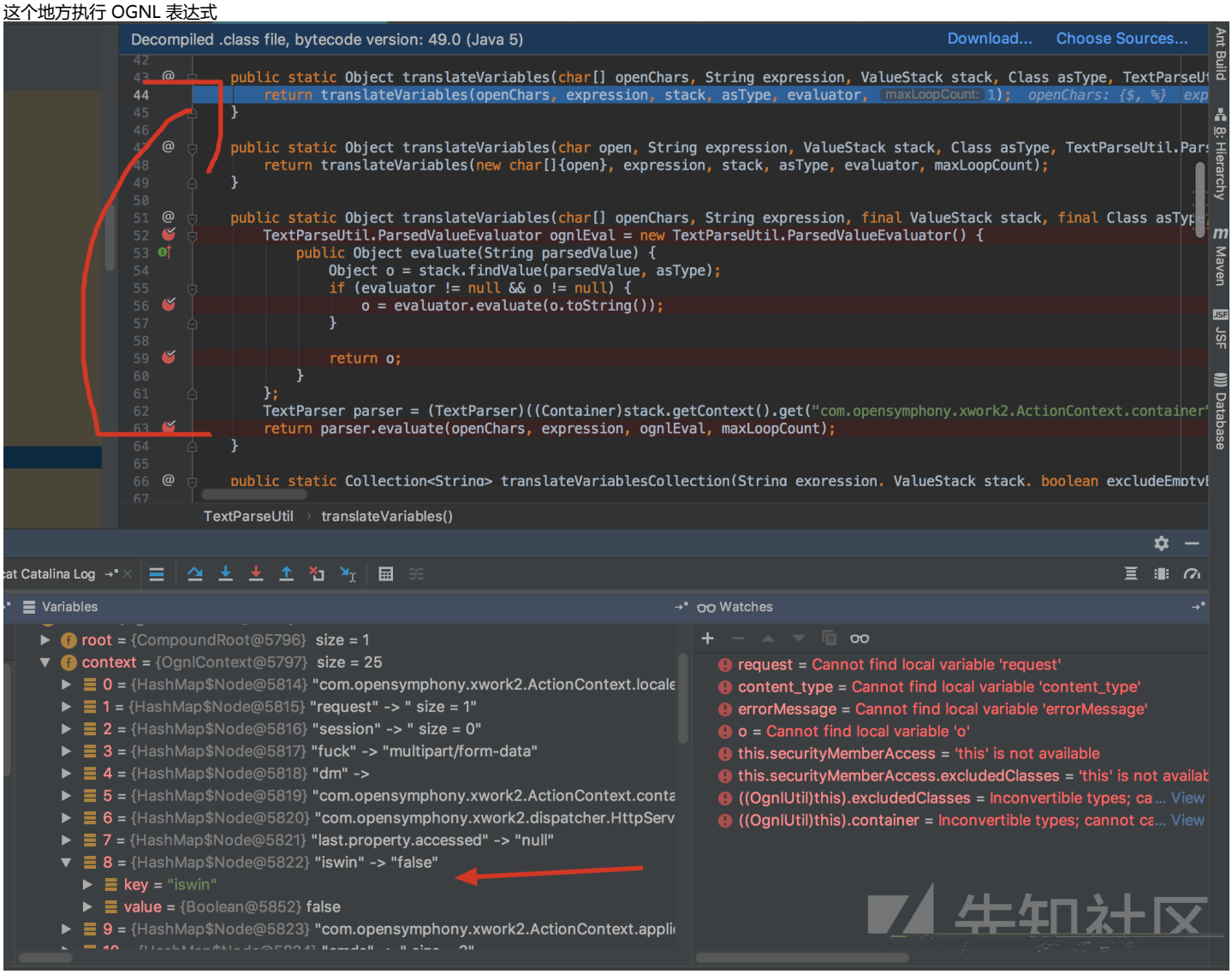


ognl.ASTVarRef#setValueBody


protected void setValueBody(OgnlContext context, Object target, Object value) throws OgnlException {
context.put(this._name, value); context: size = 25 _name: "fuck" value: "multipart/form-data"
}
public Object put(Object key, Object value) {
Object result;
if (RESERVED_KEYS.containsKey(key)) {
if (key.equals("this")) {
result = this.getCurrentObject();
this.setCurrentObject(value);
} else if (key.equals("root")) {
result = this.getRoot();
this.setRoot(value);
} else {
if (key.equals("context")) {
throw new IllegalArgumentException("can't change context in context");
}
if (key.equals("_traceEvaluations")) {
result = this.getTraceEvaluations() ? Boolean.TRUE : Boolean.FALSE;
this.setTraceEvaluations(OgnlOps.booleanValue(value));
} else if (key.equals("_lastEvaluation")) {
result = this.getLastEvaluation();
this._lastEvaluation = (Evaluation)value;
} else if (key.equals("_keepLastEvaluation")) {
result = this.getKeepLastEvaluation() ? Boolean.TRUE : Boolean.FALSE;
this.setKeepLastEvaluation(OgnlOps.booleanValue(value));
} else if (key.equals("_classResolver")) {
result = this.getClassResolver();
this.setClassResolver((ClassResolver)value);
} else if (key.equals("_typeConverter")) {
result = this.getTypeConverter();
this.setTypeConverter((TypeConverter)value);
} else {
if (!key.equals("_memberAccess")) {
throw new IllegalArgumentException("unknown reserved key '" + key + "'");
}
result = this.getMemberAccess();
this.setMemberAccess((MemberAccess)value);
}
}
} else {
result = this._values.put(key, value);
}
OgnlContext › put()



这个地方执行 OGNL 表达式



在mac上调试的

payload里

```
#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains('win'))
```

执行之后就是false

com.opensymphony.xwork2.ognl.OgnlValueStack#setOgnlUtil

```
    @Inject
    public void setOgnlUtil(OgnlUtil ognlUtil) {
        this.ognlUtil = ognlUtil;
        this.securityMemberAccess.setExcludedClasses(ognlUtil.getExcludedClasses());
        this.securityMemberAccess.setExcludedPackageNamePatterns(ognlUtil.getExcludedPackageNamePatterns());
    }
```


当请求到来的时候，一个ActionContext对象被createActionContext方法创建。
OgnlValueStack 的setOgnlUtil函数被调用，以用来初始化OgnlValueStack 的securityMemberAccess ，这样就获得OgnlUtil的全局实例
这就意味着全局OgnlUtil 实例都共享相同的SET：excludedClasses, excludedPackageNames 和 excludedPackageNamePatterns作为_memberAccess，所以清除这些之后也会清除与_memberAccess相匹配的SET。
在那之后，OGNL 就可以自由的访问DEFAULT_MEMBER_ACCESS对象并且 OgnlContext 的 setMemberAccess 代替了 _memberAccess和DEFAULT_MEMBER_ACCESS，这样就可以执行任意代码了

参考文章

参考文章
作为武器的CVE-2018-11776：绕过Apache Struts 2.5.16 OGNL 沙箱
Struts2 架构图
可能有的地方说的不对,希望师傅们指正(萌新瑟瑟发抖)

上一篇：OSINT Primer：组织（第... 下一篇：详解变形金刚

1. 0 条回复
   - 动动手指，沙发就是你的了！