阿里云安全获Oracle官方致谢 ｜Weblogic Server远程代码执行漏洞预警(CVE-2021-2109)

原创 漏洞预警 阿里云应急响应 2021年01月20日 01:59

01 漏洞简述

2020年11月19日，阿里云安全向Oracle官方报告了Weblogic Server远程代码执行漏洞，漏洞编号为CVE-2021-2109，漏洞等级：高危。

02 时间轴

- **2020/11/19**

阿里云安全向Oracle官方报告了Weblogic Server远程代码执行漏洞

- **2020/11/19**

阿里云WAF更新防护策略

- **2021/01/20**

Oracle官方发布了漏洞补丁，分配CVE编号为CVE-2021-2109，并向阿里云安全公开致谢

- **2021/01/20**

阿里云安全发布漏洞风险提示

03 风险等级

评定方式	等级
威胁等级	高危
影响范围	较广
利用难度	低

04 影响版本

Weblogic Server

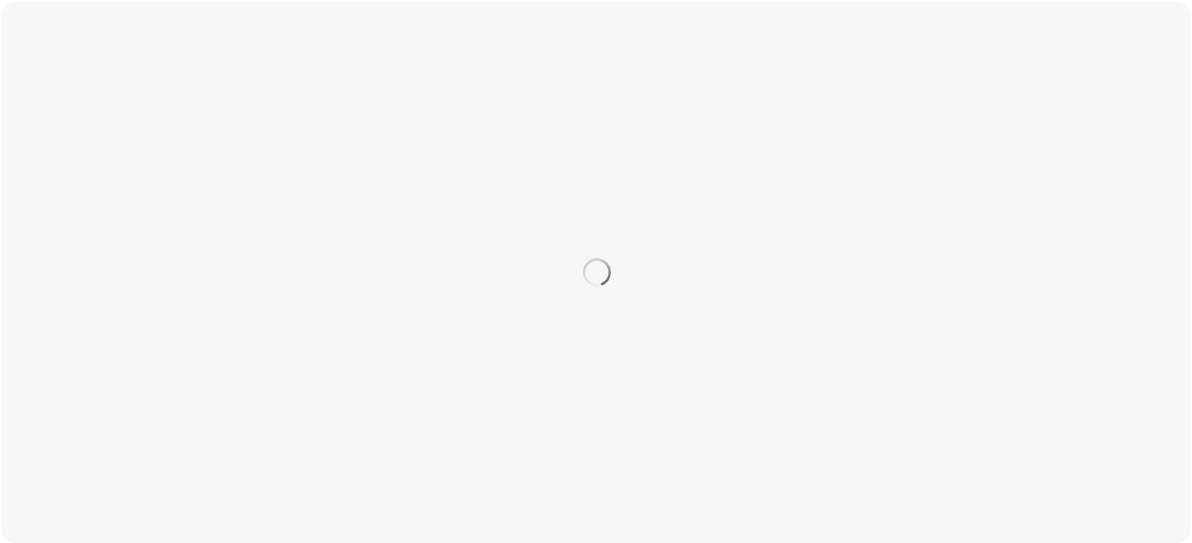
- 10.3.6.0.0
- 12.1.3.0.0
- 12.2.1.3.0
- 12.2.1.4.0
- 14.1.1.0.0

05 漏洞分析

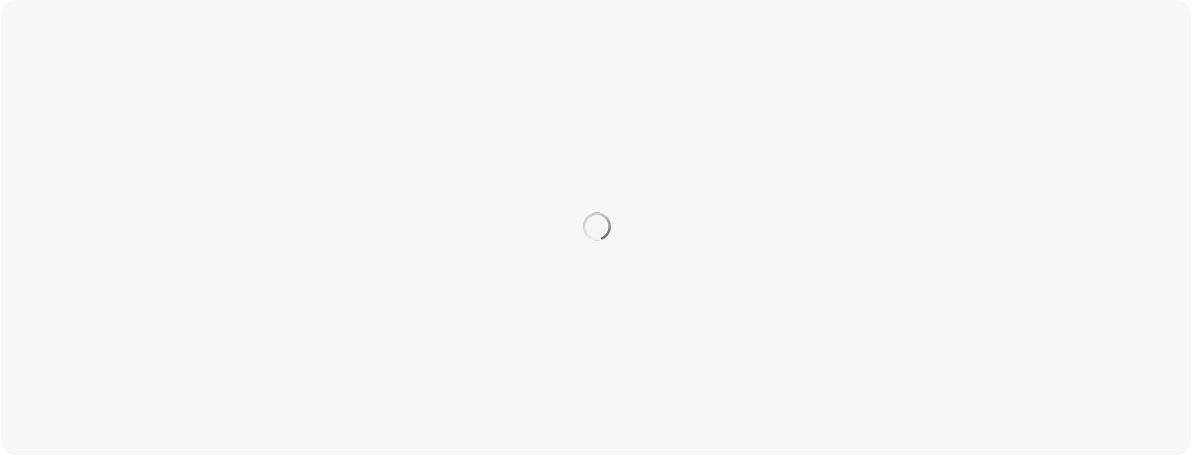
这个漏洞利用的有两个关键类，第一个类是com.bea.console.handles.JndiBindingHandle 跟进这个类看下

可以看到Handle只是用来做对象的实例化，并没有执行功能，理论上Weblogic Server的console的操作大部分是建立在Action的基础上，所以我们还需要去寻找一个Action。

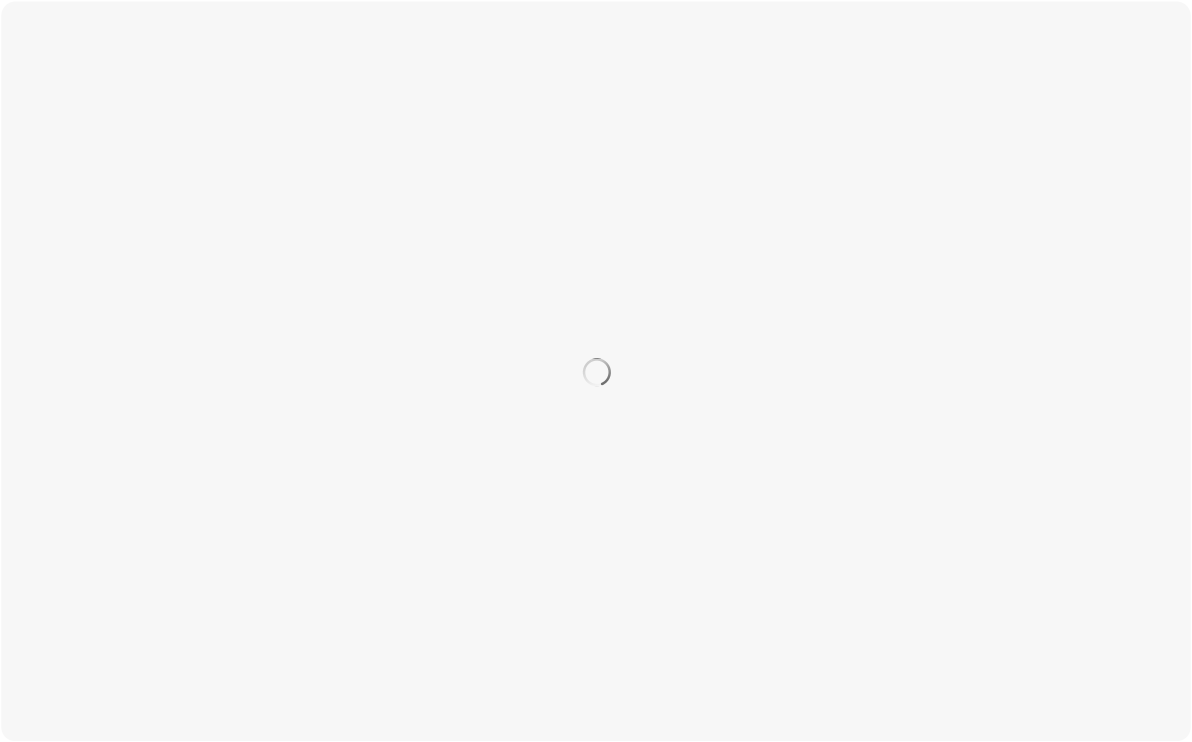
去看一下Weblogic Server的consolejndi.portal文件，以JNDIBindingPageGeneral为关键字，发现路径指向jndibinding.portlet



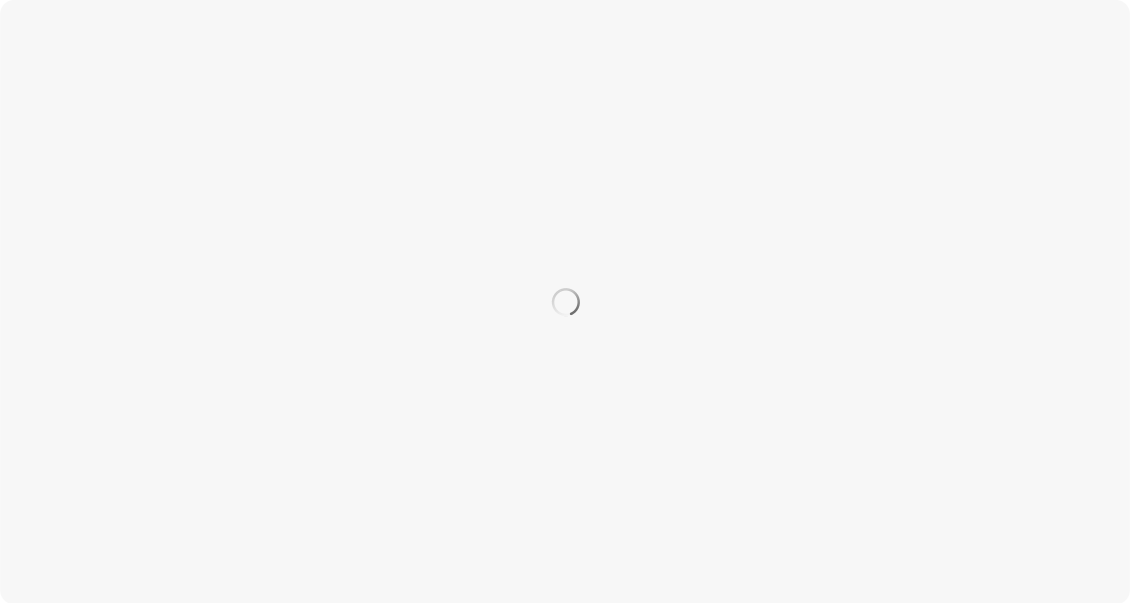
继续跟进jndibinding.portlet可以找到这次利用的另一个关键的类JNDIBindingAction



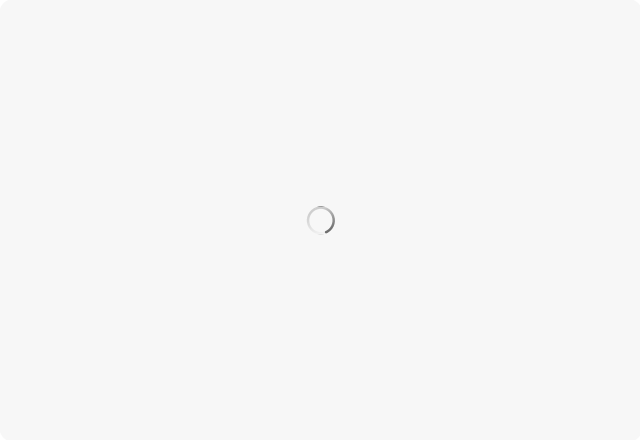
继续跟进JNDIBindingAction.execute的代码



找到了JNDI注入攻击中关键的lookup函数（lookup函数的值由context和bindName决定），但这里有个前提，需要serverMBean不为空，而serverMBean是由DomainMBean.lookupServer来获取，于是在这个函数下断点



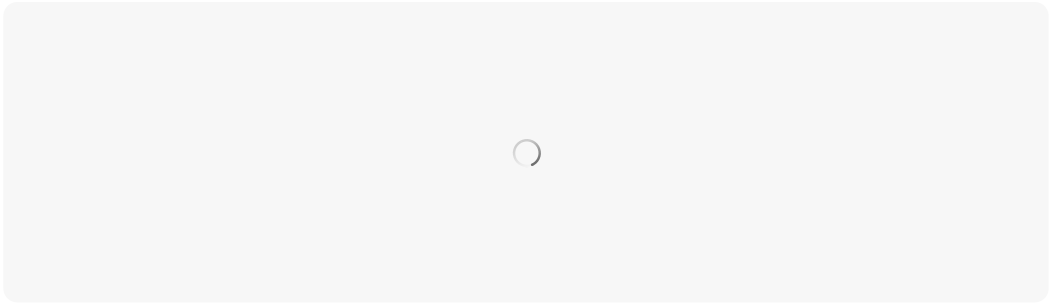
想要返回不为空，则需要传给lookupServer的值等于this._Servers中的name，而this._Servers只有一个值，利用动态调试把name的值取出



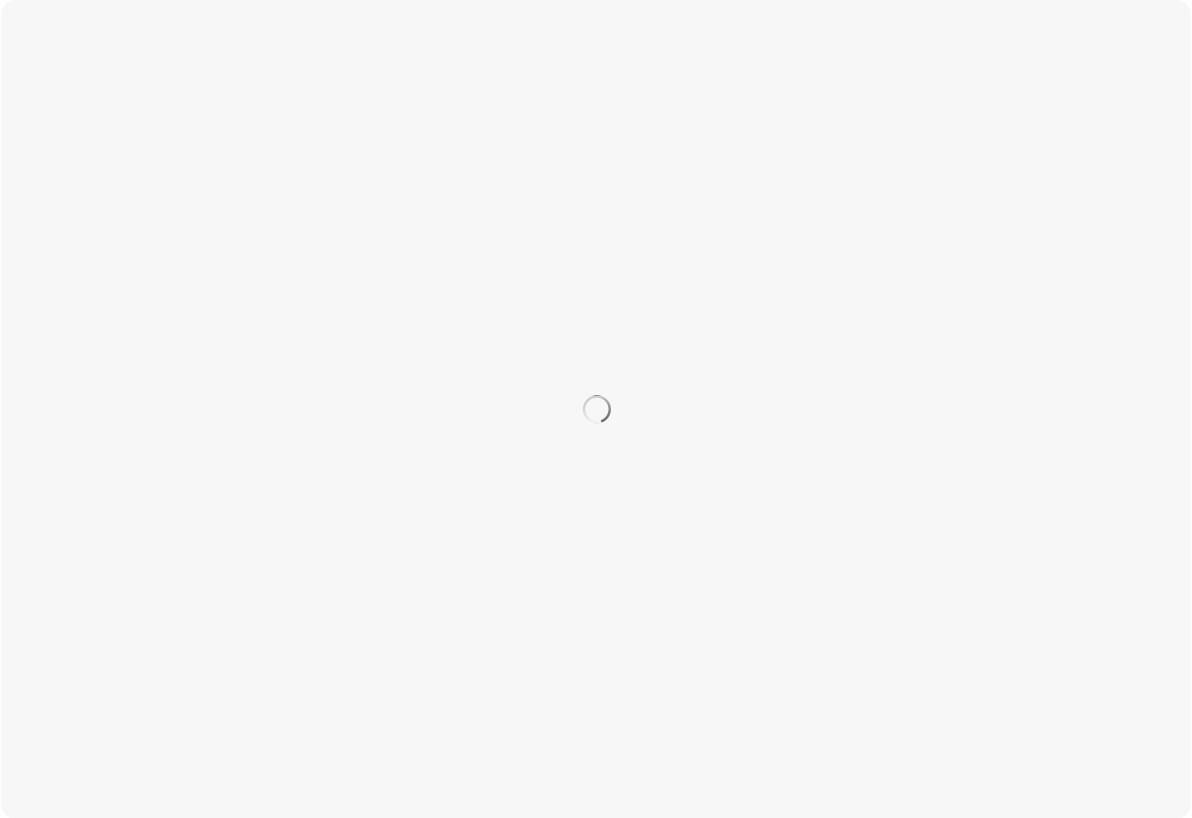
关键流程已经梳理完毕，重新去看下JNDIBindingAction的代码，如果想要实现JNDI注入攻击，我们需要满足2点要求：

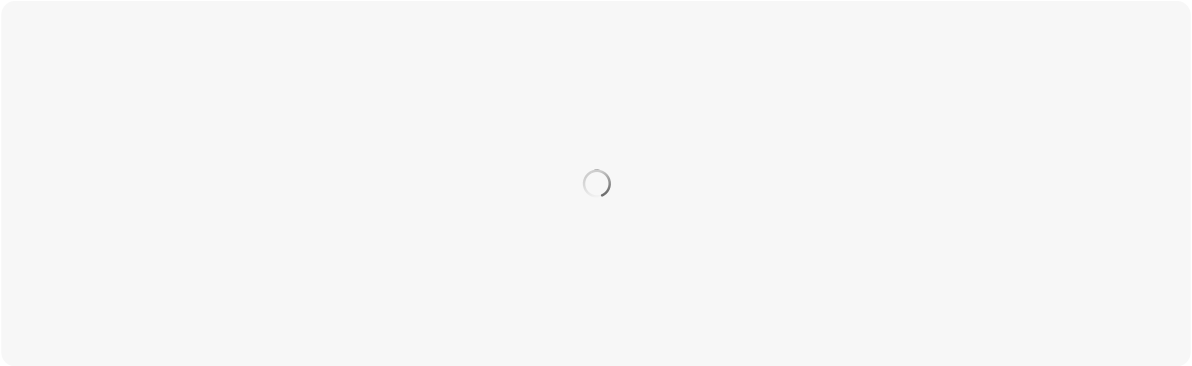
- context + "." +bindName的值要符合合法的JNDI地址格式
- serverName的值为AdminServer

而context、bindName、serverName的值都是从bindingHandle中获取的，正巧我们可以控制JndiBindingHandle实例化的值（objectIdentifier），接着来就需要看下objectIdentifier和以上3个值有什么关系了，看一下3个成员变量的get函数，发现他们都和Component有关，

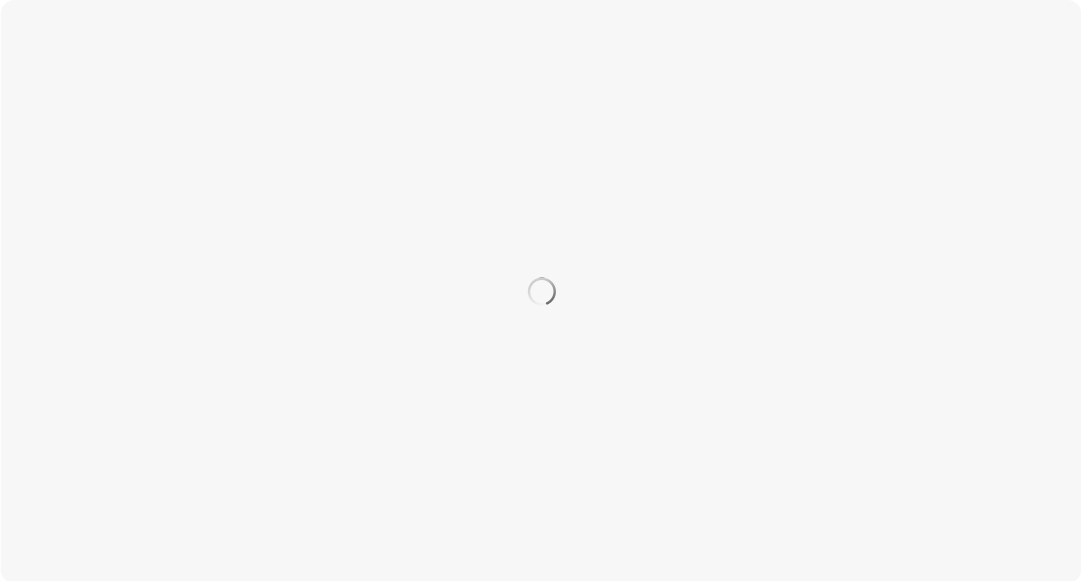


跟进getComponents函数，代码如下：

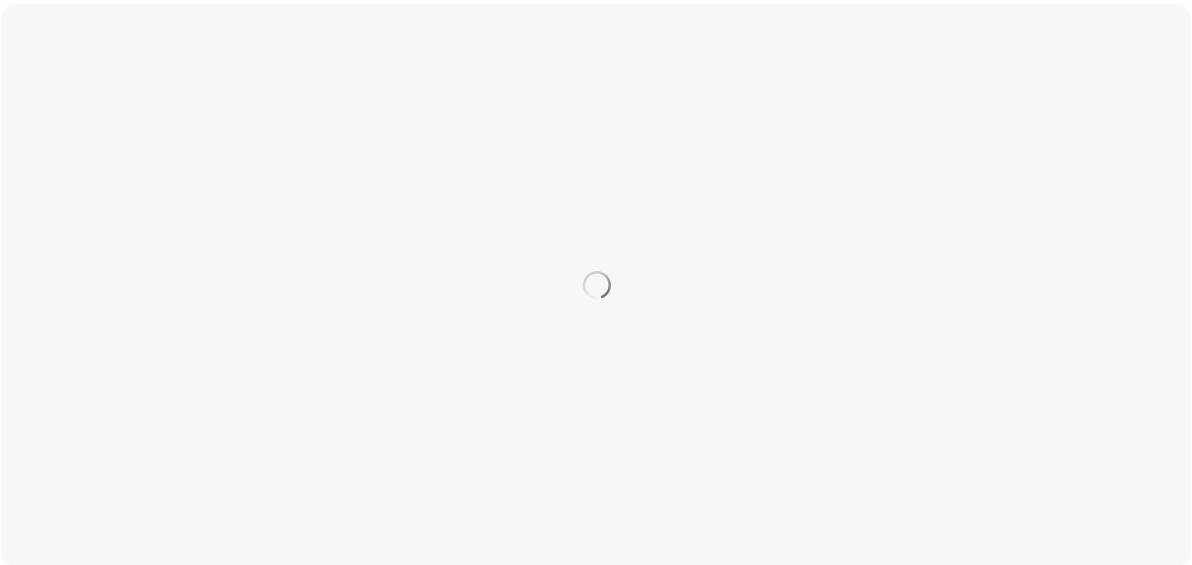




这里结合调用栈信息



可以发现components的值就是把objectIdentifier的值用分号; 分割开来，也就是说我们想要控制的值全都可以通过bjectIdentifier来控制了，PoC的构造也就水到渠成了，我们可以通过LDAP协议方式实现JNDI注入攻击，加载远程CodeBase下的恶意类 ldap://127.0.0;1:1389/EvilObject，由于代码中会自动补全一个.因此可以将context定位为ldap://127.0.0将bindName定位为1:1389/EvilObject，最后的serverName必须为AdminServer，因此构造完整的PoC后，漏洞利用效果如图：：



06 修复建议

- 由于是通过JNDI注入进行远程命令执行，所以受到JDK版本的影响，建议升级Weblogic Server运行环境的JDK版本，具体参考如下：

- 更新最新补丁，参考Oracle官网发布的补丁：

Oracle Critical Patch Update Advisory - January 2021

https://www.oracle.com/security-alerts/cpujan2021.html

- 阿里云WAF支持该漏洞防御，并提供免费应急支持，请钉钉扫描下方二维码加入应急支持群

修改于2021年01月20日