# shiro反序列化漏洞

## Shiro-550反序列化漏洞（CVE-2016-4437）

### 漏洞简介

shiro-550主要是由shiro的rememberMe内容反序列化导致的命令执行漏洞，造成的原因是默认加密密钥是硬编码在shiro源码中，任何有权访问源代码的人都可以知道默认加密密钥。于是攻击者可以创建一个恶意对象，对其进行序列化、编码，然后将其作为cookie的rememberMe字段内容发送，Shiro 将对其解码和反序列化，导致服务器运行一些恶意代码。

特征：cookie中含有rememberMe字段

修复建议：

更新shiro到1.2.4以上的版本。

不使用默认的加密密钥，改为随机生成密钥。

### 漏洞原理

一、Shiro简介
Apache Shiro 是一个强大易用的 Java 安全框架，提供了认证、授权、加密和会话管理等功能，对于任何一个应用程序，Shiro 都可以提供全面的安全管理服务。

在ApacheShiro<=1.2.4版本中AES加密时采用的key是硬编码在代码中的，于是我们就可以构造Remembe Me的值，然后让其反序列化执行。

二、Shiro服务器识别身份加解密处理的流程

（1）加密

1.用户使用账号密码进行登录，并勾选"Remember Me"。

2、Shiro验证用户登录信息，通过后，查看用户是否勾选了”Remember Me“。

3、若勾选，则将用户身份序列化，并将序列化后的内容进行AES加密，再使用base64编码。

4、最后将处理好的内容放于cookie中的rememberMe字段。

（2）解密

1、当服务端收到来自未经身份验证的用户的请求时，会在客户端发送请求中的cookie中获取rememberMe字段内容。

2、将获取到的rememberMe字段进行base64解码，再使用AES解密。

3、最后将解密的内容进行反序列化，获取到用户身份。

三、Key

AES加密的密钥Key被硬编码在代码里

于是可得到Payload的构造流程：

恶意命令-->序列化-->AES加密-->base64编码-->发送Cookie

# Shiro-721反序列化漏洞（CVE-2019-12422）

## Shiro550和Shiro721的区别是什么

Shiro550只需要通过碰撞key，爆破出来密钥，就可以进行利用
Shiro721的ase加密的key一般情况下猜不到，是系统随机生成的，并且当存在有效的用户信息时才会进入下一阶段的流程所以我们需要使用登录后的rememberMe Cookie，才可以进行下一步攻击。

## 漏洞指纹

URL中含有Shiro字段

cookie中含有rememberMe字段

返回包中含有rememberMe

## 漏洞介绍

在Shiro721中，Shiro通过AES-128-CBC对cookie中的rememberMe字段进行加密，所以用户可以通过PaddingOracle加密生成的攻击代码来构造恶意的rememberMe字段，进行反序列化攻击，需要执行的命令越复杂，生成payload需要的时间就越长。

## 漏洞原理

由于Apache Shiro cookie中通过 AES-128-CBC 模式加密的rememberMe字段存在问题，用户可通过 Padding Oracle 加密生成的攻击代码来构造恶意的rememberMe字段，用有效的RememberMe cookie作为Padding Oracle Attack 的前缀，然后制作精心制作的RememberMe来执行Java反序列化攻击

## 攻击流程

登录网站，并从cookie中获取RememberMe。使用RememberMe cookie作为Padding Oracle Attack的前缀。加密syserial的序列化有效负载，以通过Padding Oracle Attack制作精心制作的RememberMe。请求带有新的RememberMe cookie的网站，以执行反序列化攻击。攻击者无需知道RememberMe加密的密码密钥。

加密方式：AES-128-CBC

属于AES加密算法的CBC模式，使用128位数据块为一组进行加密解密，即16字节明文，对应16字节密文，，明文加密时，如果数据不够16字节，则会将数据补全剩余字节

若最后剩余的明文不够16字节，需要进行填充，通常采用PKCS7进行填充。比如最后缺3个字节，则填充3个字节的0x03;若最后缺10个字节，则填充10个字节的0x0a;

若明文正好是16个字节的整数倍，最后要再加入一个16字节0x10的组再进行加密

Padding Oracle Attack原理
Padding Oracle攻击可以在没有密钥的情况下加密或解密密文

Shiro Padding Oracle Attack（Shiro填充Oracle攻击）是一种针对Apache Shiro身份验证框架的安全漏洞攻击。Apache Shiro是Java应用程序中广泛使用的身份验证和授权框架，用于管理用户会话、权限验证等功能。

Padding Oracle Attack（填充Oracle攻击）是一种针对加密算法使用填充的安全漏洞攻击。在加密通信中，填充用于将明文数据扩展到加密算法块大小的倍数。在此攻击中，攻击者利用填充的响应信息来推断出加密算法中的秘密信息。

Shiro Padding Oracle Attack利用了Shiro框架中的身份验证过程中的一个漏洞，该漏洞允许攻击者通过填充信息的不同响应时间来确定身份验证过程中的错误。通过不断尝试不同的填充方式，攻击者可以逐步推断出加密秘钥，并最终获取访问权限。

这种攻击利用了填充错误的身份验证响应来获取关于秘密信息的信息泄漏，然后根据这些信息进行进一步的攻击。为了防止Shiro Padding Oracle Attack，建议及时更新Apache Shiro版本，确保已修复该漏洞，并采取其他安全措施，如使用安全的加密算法和密钥管理策略。

# 漏洞复现：CVE-2016-4437

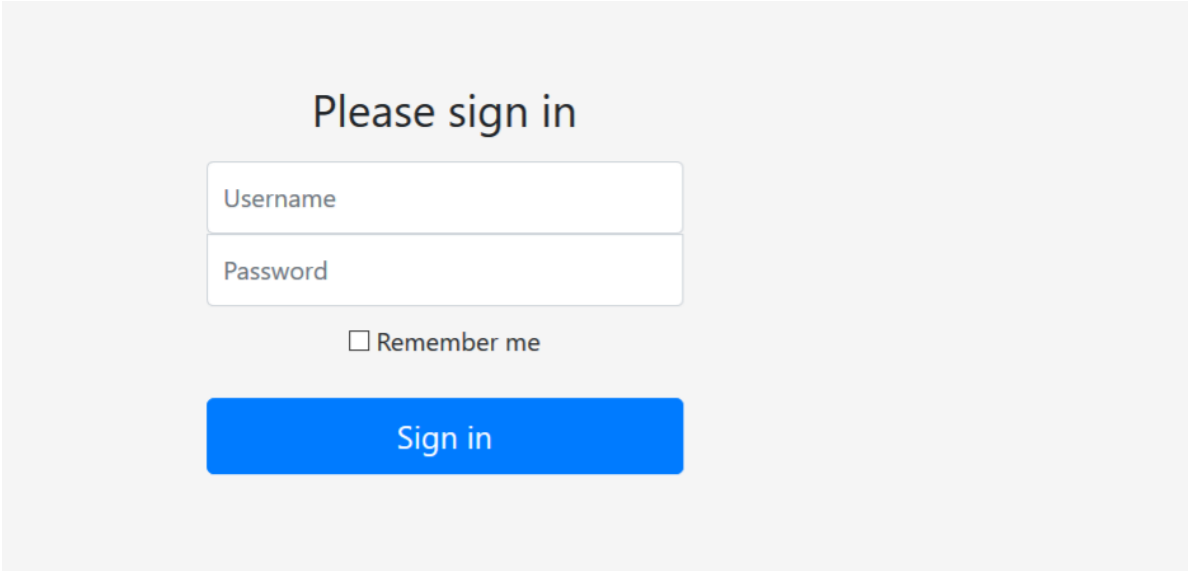
环境：kali linux

靶场使用：vulhub/shiro/CVE-2016-4437

```
┌──(root㉿kali)-[~/桌面/vulhub/shiro/CVE-2016-4437]
└─# docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED              STATUS              PO
RTS                                      NAMES
ac602458520f   vulhub/shiro:1.2.4    "java -jar /shirodem…"   About a minute ago   Up About a minute   0.
0.0.0:8080→8080/tcp, :::8080→8080/tcp    cve-2016-4437_web_1

┌──(root㉿kali)-[~/桌面/vulhub/shiro/CVE-2016-4437]
└─#
```

访问地址：



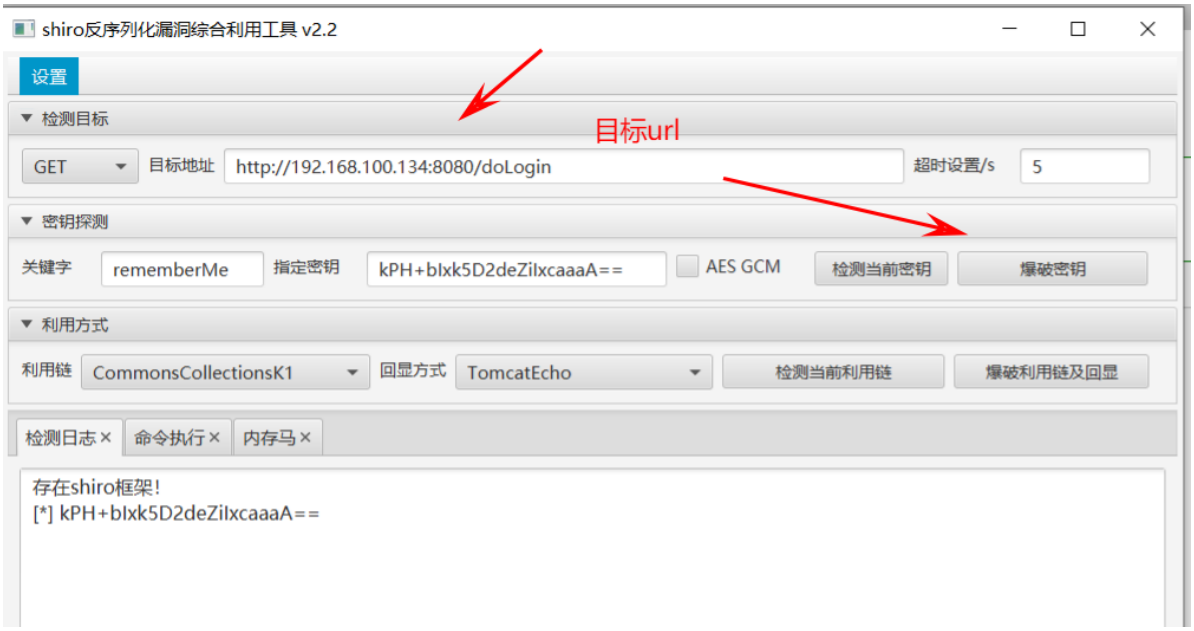

抓包分析一下：


在返回包当中发现存在**rememberMe=deleteMe** 字样，可以大概确定有配置shiro，可以进行下一步。因为shiro本身功能就是一个身份验证管理，所以一般都在登录口可以看到。

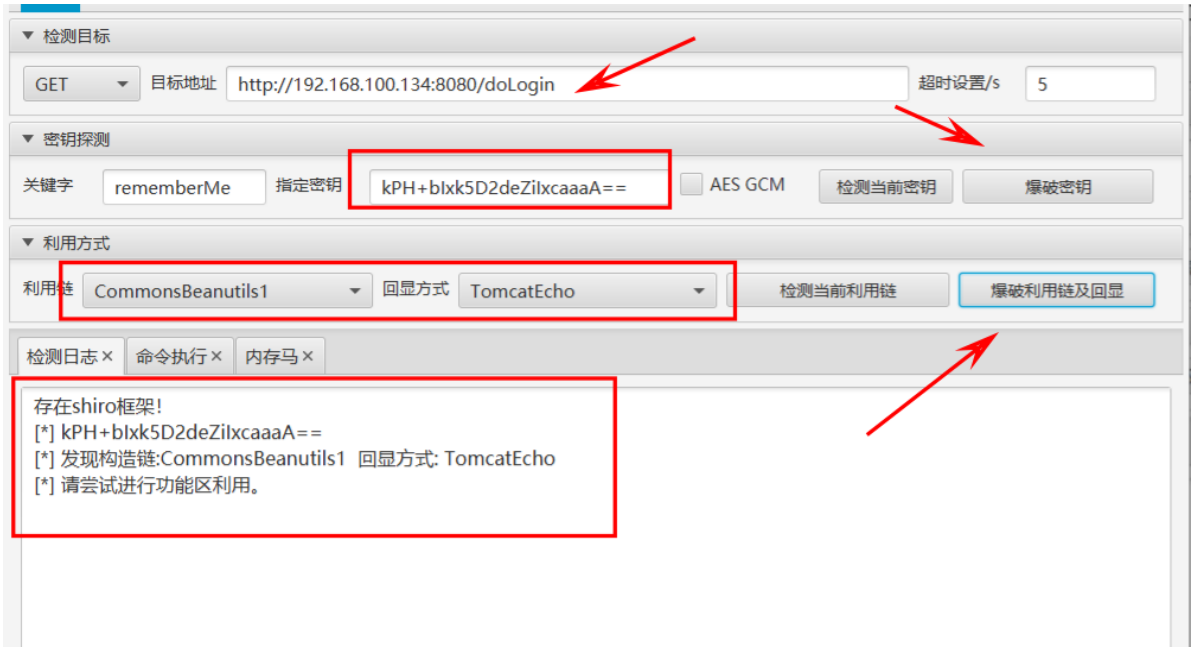
UI一键利用工具

使用工具再进行检测确认：

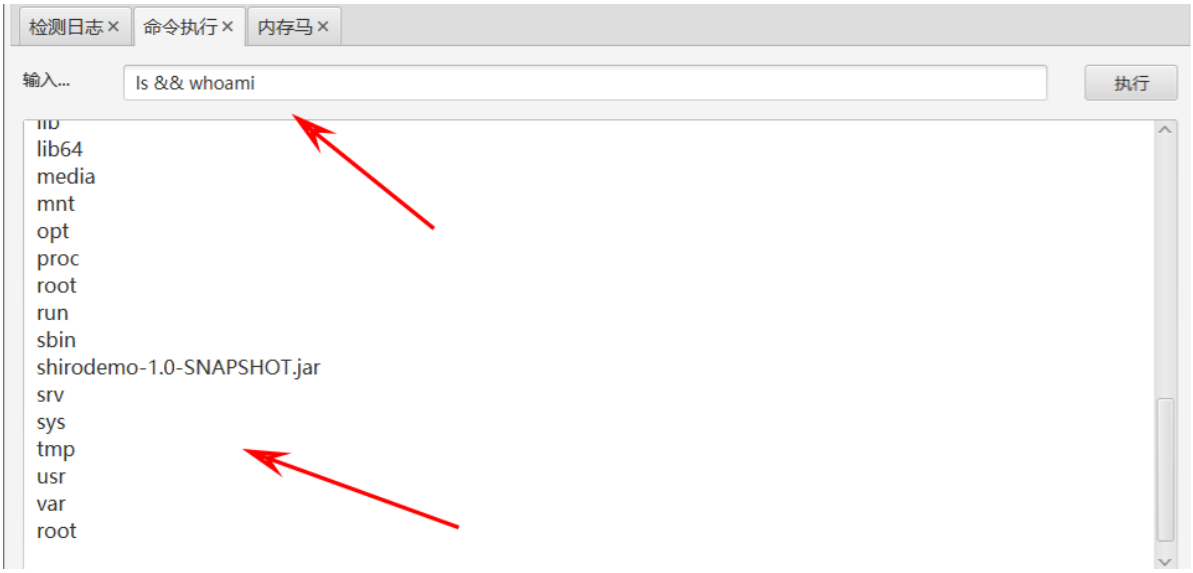
输入目标的url地址，根据关键字进行爆破秘钥




接下来爆破利用链以及回显方式：

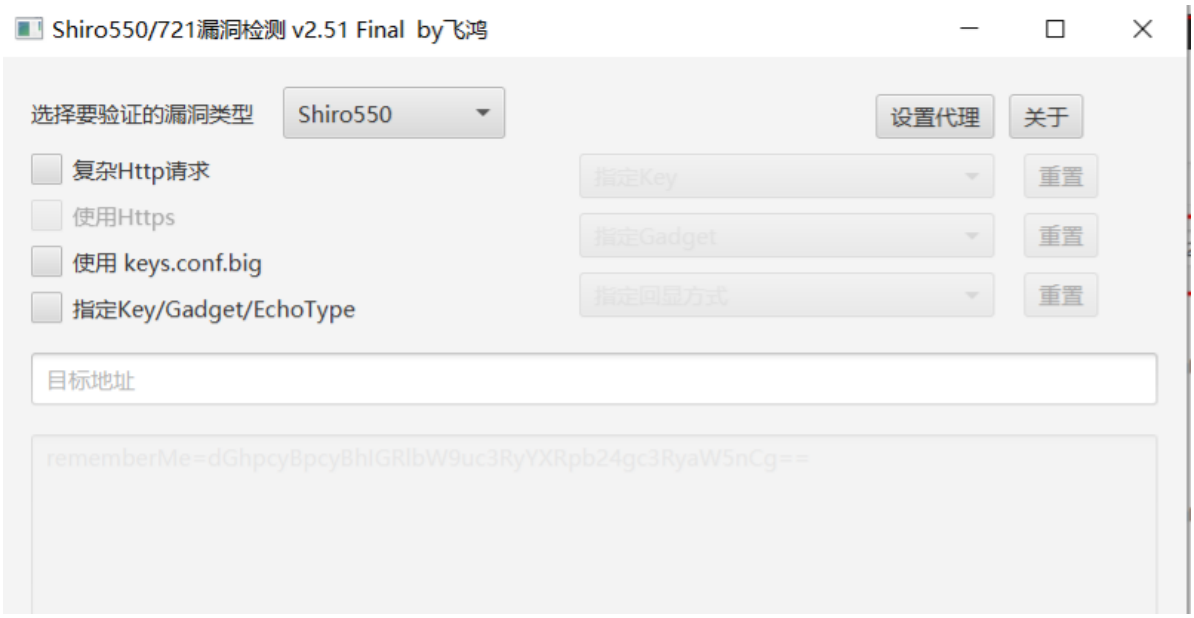



接下来可进行命令执行：

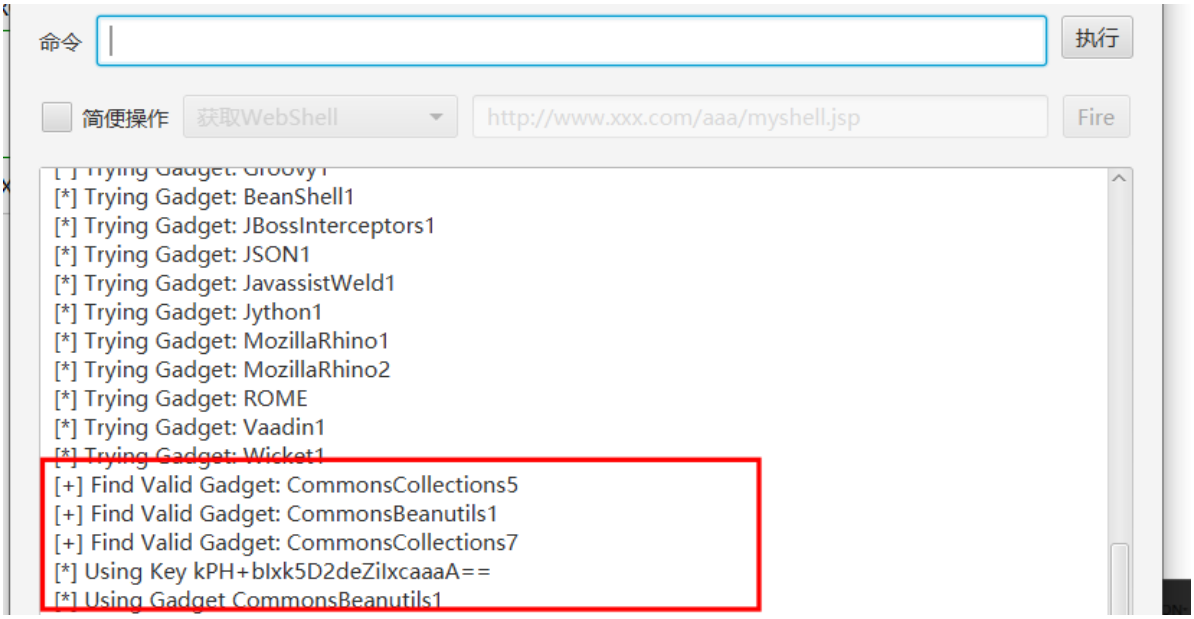



反弹shell：

可使用工具进行检测：



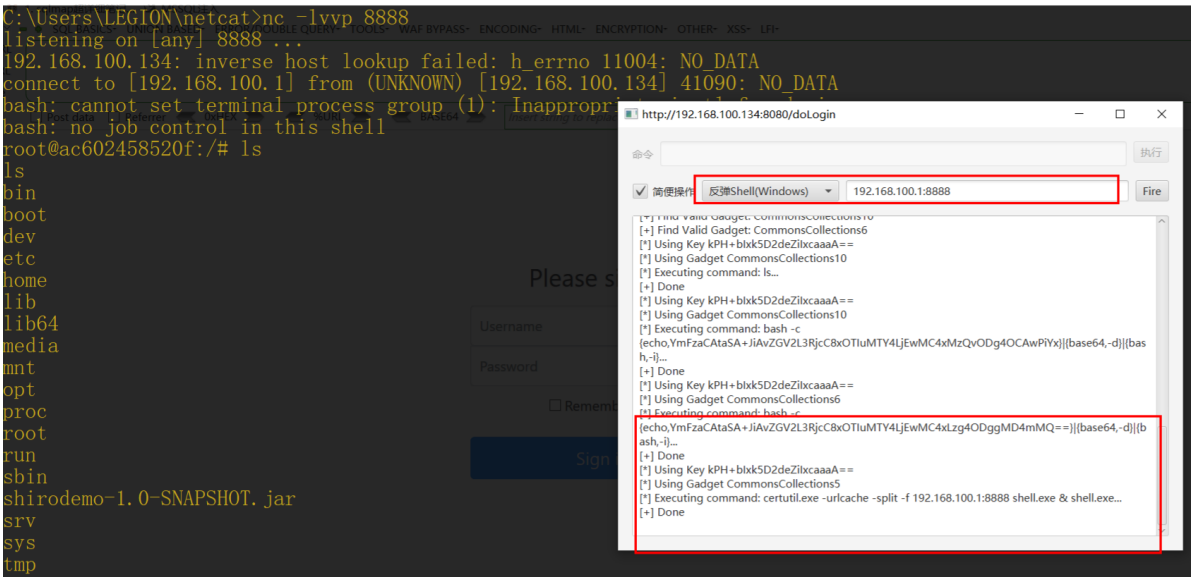

检测完成后可进行命令执行，反弹shell等操作：



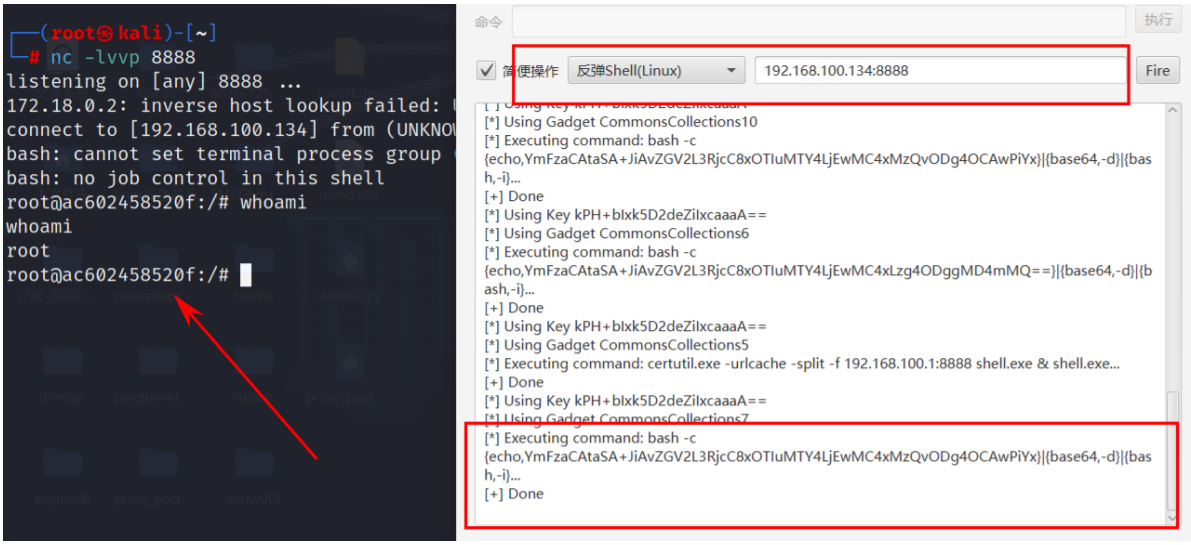

使用工具进行简单的反弹：

设置监听端口

window：




linux：




还有使用ysoserial监听模块JRMP来进行反弹shell，具体可看参考文档，操作都差不多。

正是因为利用简单，所以危害比较大。

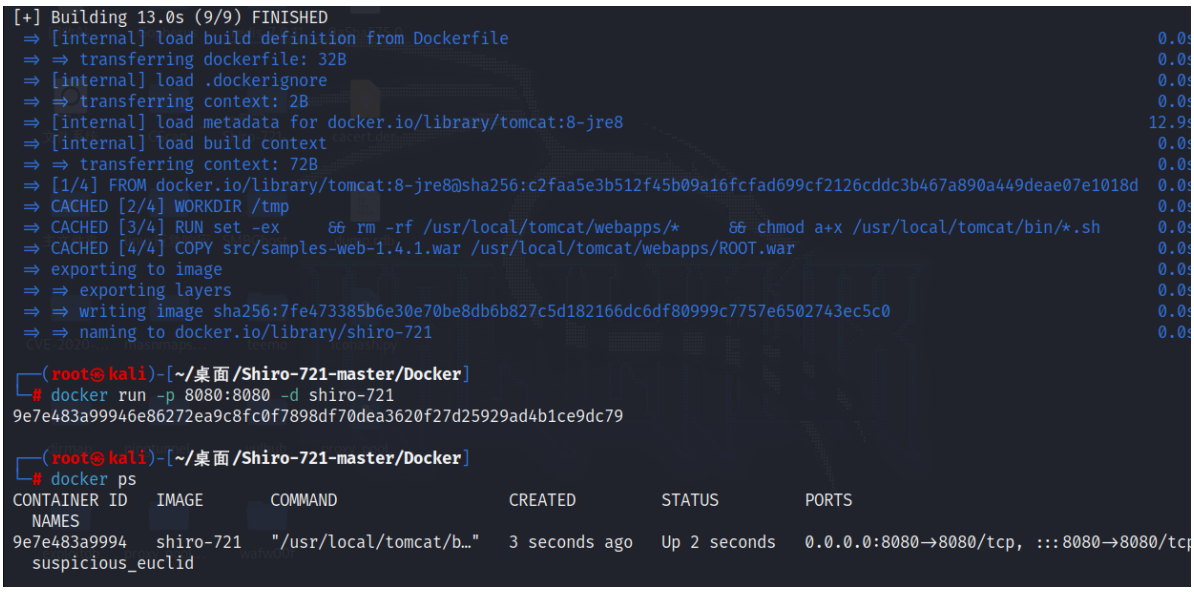
工具地址

# 漏洞复现：CVE-2019-12422

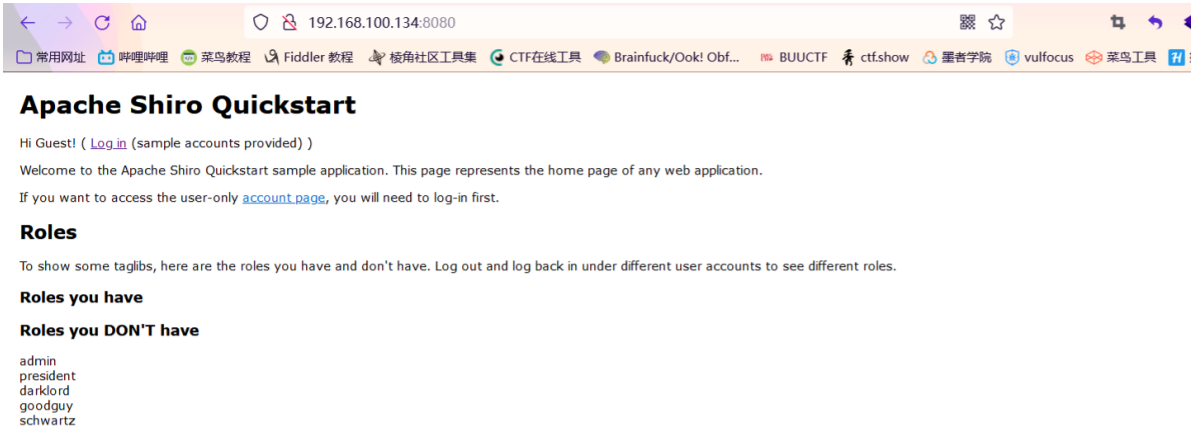
环境：kali linux

docker进行搭建启动

```
git clone https://github.com/3ndz/Shiro-721.git
cd Shiro-721/Docker
docker build -t shiro-721 .
docker run -p 8080:8080 -d shiro-721
```



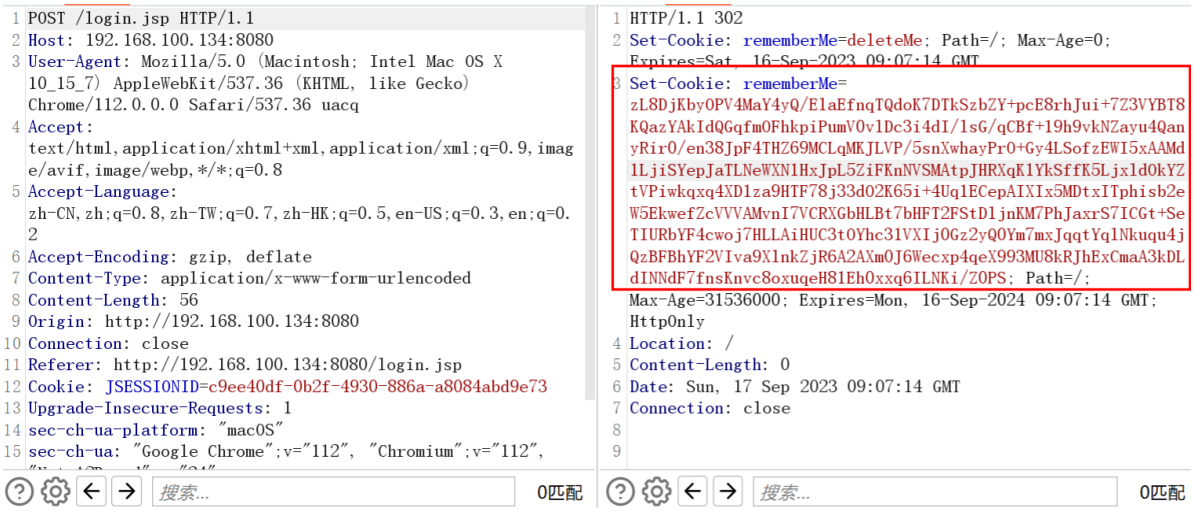

访问：



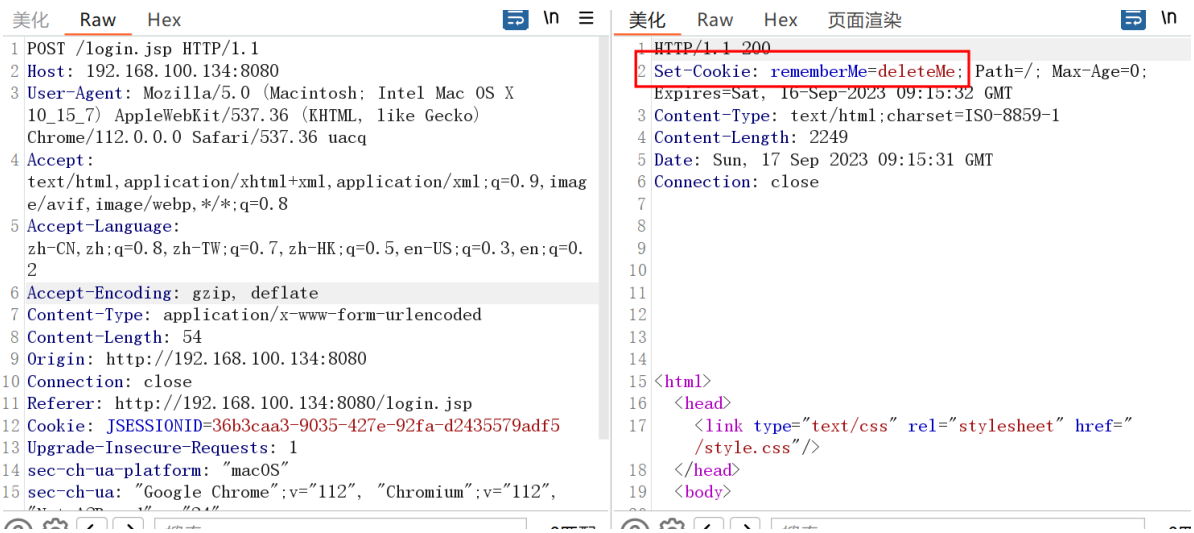

利用过程和shiro550差不多，shiro710需要登录网站，并从cookie中获取RememberMe。

进行登录，使用正确的账号和密码：

先使用正确的账号密码登录后，在抓包获取合法 Cookie（勾选Remember Me）

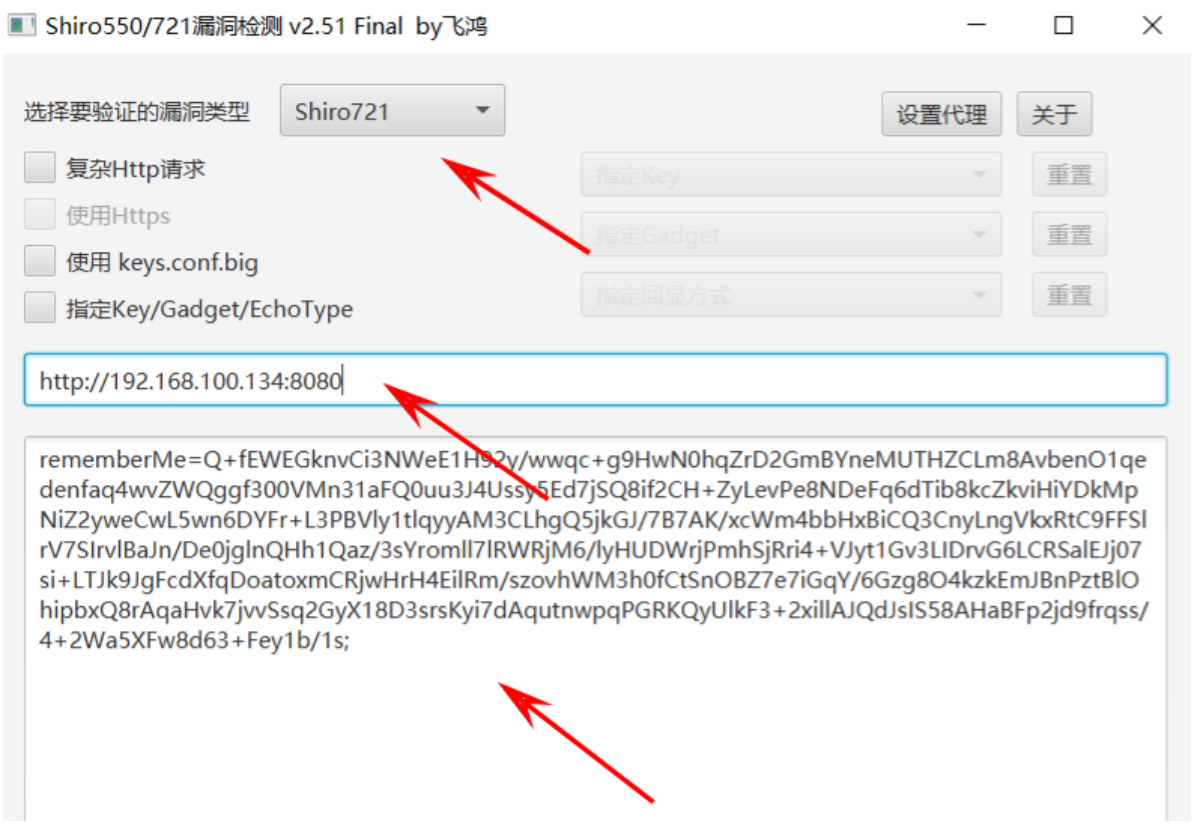



如果认证失败则只能得到 rememberMe=deleteMe




将登录后获取的set-cookie值当中的rememberMe值，将值复制下来，放到工具当中

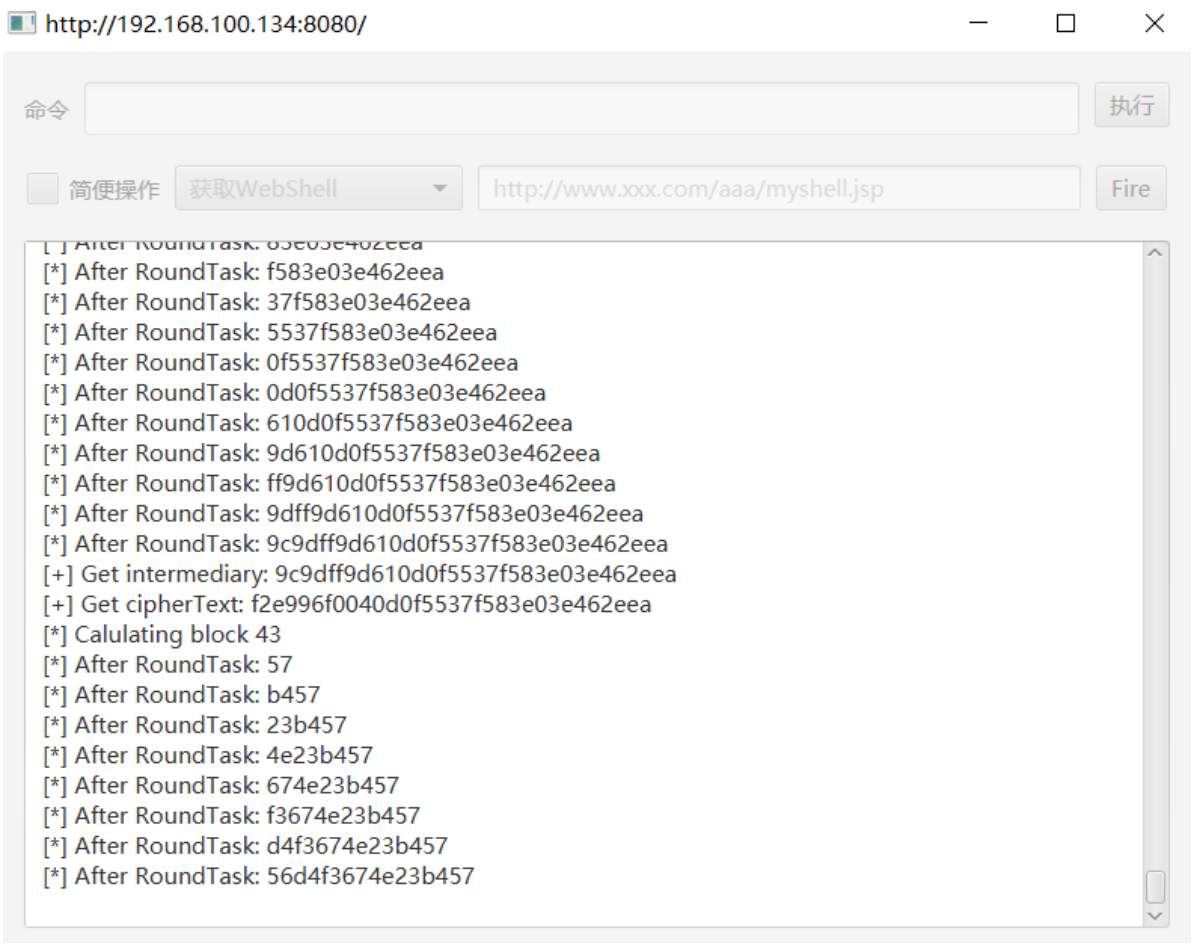
选择shiro721，输入目标url，设置rememberMe值，点击下一步，等待即可


Shiro550/721漏洞检测 v2.51 Final by飞鸿
选择要验证的漏洞类型
Shiro721
设置代理
关于
复杂Http请求
使用Https
使用 keys.conf.big
指定Key/Gadget/EchoType
指定Key
指定Gadget
指定回显方式
重置
重置
重置
http://192.168.100.134:8080
rememberMe=Q+fEWEGknvCi3NWeE1H92y/wwqc+g9HwN0hqZrD2GmBYneMUTHZCLm8AvbenO1qe
denfaq4wvZWQggf300VMn31aFQ0uu3J4Ussy5Ed7jSQ8if2CH+ZyLevPe8NDeFq6dTib8kcZkviHiYDkMp
NiZ2yweCwL5wn6DYFr+L3PBVly1tlqyyAM3CLhgQ5jkGJ/7B7AK/xcWm4bbHxBiCQ3CnyLngVkxRtC9FFSI
rV7SIrvlBaJn/De0jglnQHh1Qaz/3sYromll7lRWRjM6/lyHUDWrjPmhSjRri4+VJyt1Gv3LIDrvG6LCRSalEJj07
si+LTJk9JgFcdXfqDoatoxmCRjwHrH4EilRm/szovhWM3h0fCtSnOBZ7e7iGqY/6Gzg8O4kzkEmJBnPztBlO
hipbxQ8rAqaHvk7jvvSsq2GyX18D3srsKyi7dAqutnwpqPGRKQyUlkF3+2xillAJQdJsIS58AHaBFp2jd9frqss/
4+2Wa5XFw8d63+Fey1b/1s;

选择检测方式：


http://192.168.100.134:8080
使用 ceye.io 进行漏洞检测
使用 dnslog.cn 进行漏洞检测
使用 JRMP + dnslog.cn 进行漏洞检测
IPAddress
HTTPService Port
8080
JRMPListener Port
8088
找到可用Gadget即停止寻找
是
否
使用回显进行漏洞检测
(可选)请输入一个静态资源URL,如 http://www.xxx.com/static/logo.png
上一步
下一步

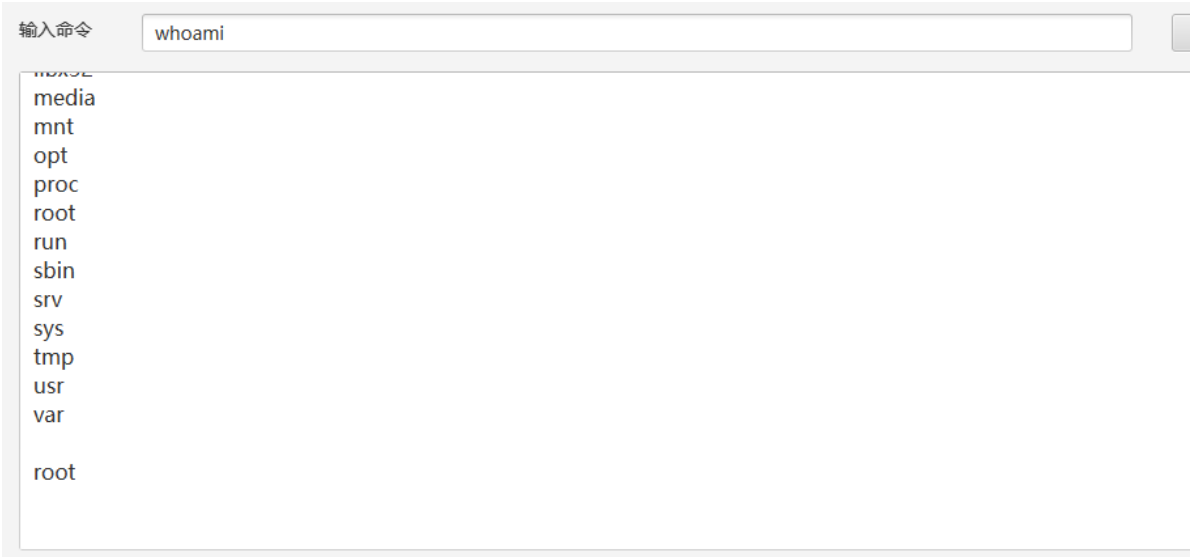
等待即可：




使用shiro综合工具：

输入url后：进行爆破秘钥，爆破利用链以及回显

命令执行:


输入命令
whoami
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
root

工具地址