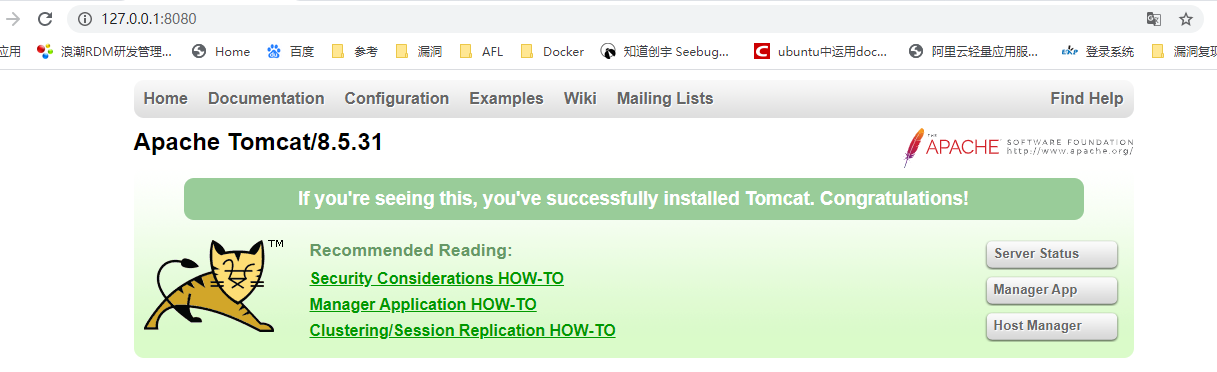
## 复现

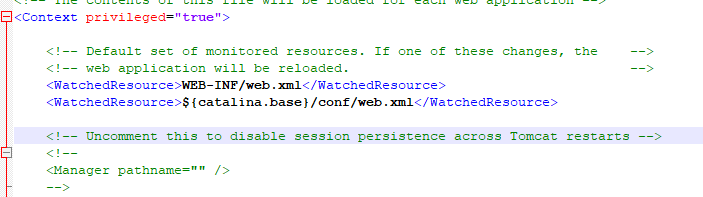
环境：Apache 8.5.39 、JDK 1.8.0-241-b07

先搭建好apache toncat：

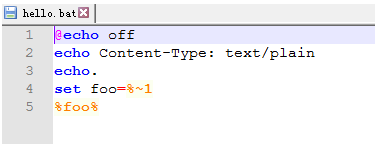


配置Apache Tomcat服务器（修改conf目录配置文件，启用CGI）：

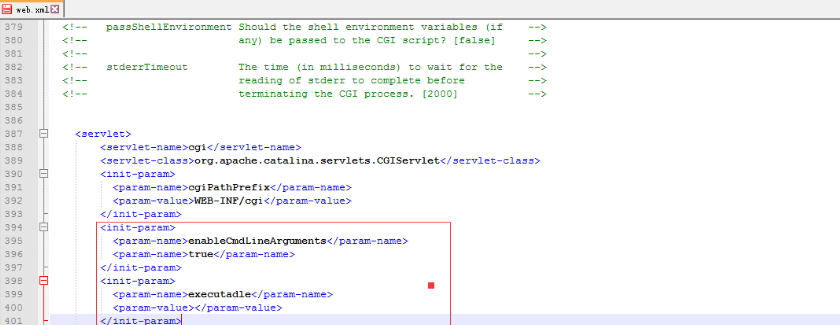
然后修改context.xml：



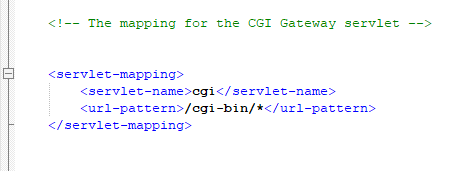
在apache-tomcat-8.5.39\webapps\ROOT\WEB-INF目录新建一个cgi-bin文件夹，创建一个hello.bat的文件，内容如下：



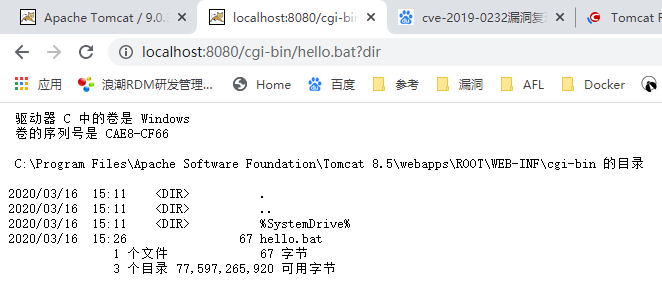
然后在修改：



同时还要修改web.xml以下配置，否则访问cgi目录会提示404:



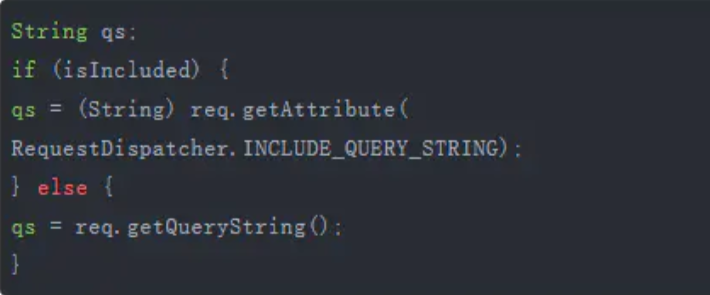
成功执行dir命令;



## 分析

漏洞代码位于tomcat\java\org\apache\catalina\servlets\CGIServlet.java文件中，这里我追踪分析当浏览器访问http://localhost:8080/cgi-bin/hello.bat?&C%3a%5cWindows%5cSystem32%5Cnet+user时，相应代码做了哪些操作。

处理链接的代码在CGIServlet.java中的setupFromRequest函数中：



原始链接字符串

http://localhost:8080/cgi-bin/hello.bat?&C%3a%5cWindows%5cSystem32%5Cnet+user通过经过getQueryString函数，qs得到了初始查询字符串，此时，qs=”&C%3a%5cWindows%5cSystem32%5Cnet+user”：



代码首先判断qs不能为空，且不能含有=符号，因为含有=号代表链接为indexexed query:



然后StringTokneizer类将字符串qs分割为两部分，以+号为分隔符，字符串一分为二，变成了&C%3a%5cWindows%5cSystem32%5Cnet和user：

此时分割的字符串并未直接进入命令行，作者做了一次正则匹配检验，cmdLineArgumentEncodePattern.matcher,目的是检测字符串的合法性，而匹配规则允许字母，数字，以及特殊符号，显然，我们构造的注入链接时满足要求的，没有使用规则之外的字符：



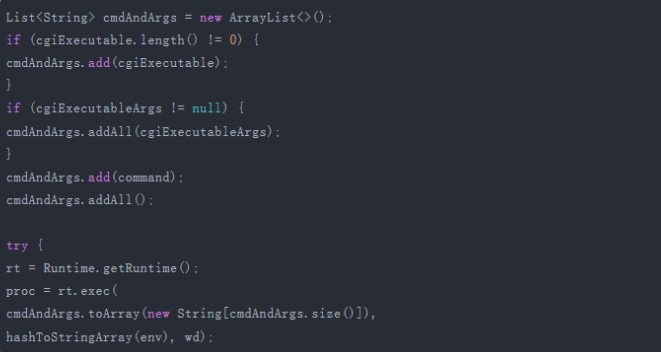
然后下一步，URLDecoder.decode将url编码的字符串转码，&C%3a%5cWindows%5cSystem32%5Cnet解码后为&C:\windows\System32\net,user解码后还是user不变。



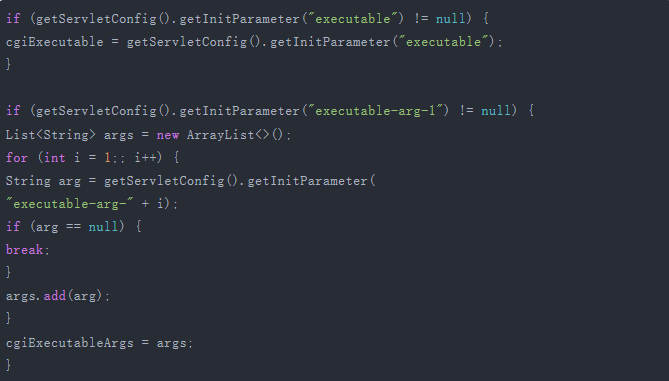
CmdLineParameters是一个ArrayList<String>数组对象，定义如下：



反复回溯跟踪代码流程，就能找到上面处理后的字符串参数是传递到下面的代码做最后的处理：



从最后一句可以看出，Tomcat调用Runtime.getRuntime().exec启动了cmd命令行，参数放在了cmdAndArgs这个ArrayList<String>数组中，cmdAndArgs是由cgiExecutable、cgiExecutableArgs、command与params组合到一起得到的。



还记得开头在 conf/web.xml配置了一个标签executable，配置的值为空，即为默认值，可以推测cgiExecutable与cgiExecutableArgs并没什么太大作用。



Command原子sCGIFullPath,而sCGIFullPath可能是文件或者目录，因为无法调试，但通过观察函数命名可知，command极可能是hello.bat的绝对路径名。很容易就可以分析出，params就是前面解码后的两个参数。总结一下，此时在windows下构造的命令行参数如下：



## 解决方案

升级Tomcat版本或者将web.xml中的enableCmdLineArguments属性初始化为false