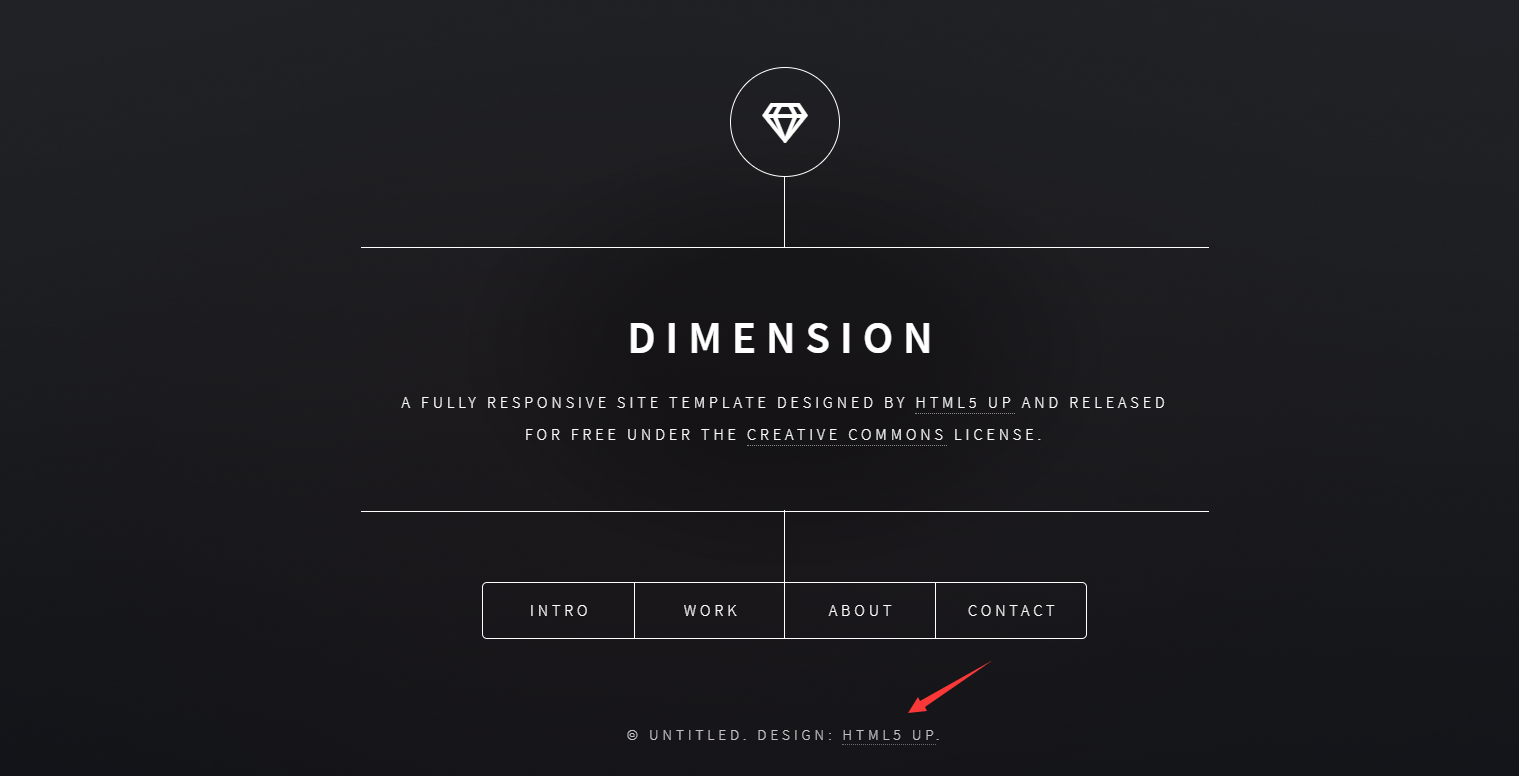
# old

首先是个静态网站，因此第一步信息搜集一下：



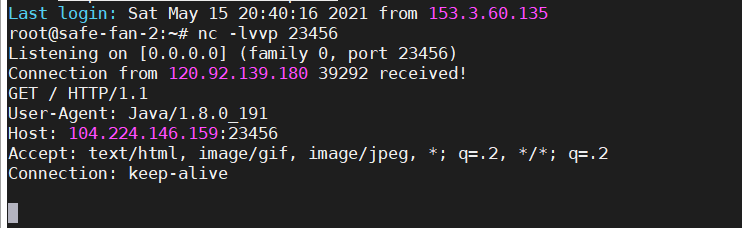
找到一个可能存在ssrf的接口，也可以在主页上点点也能发现，并且看到提示hint.txt，



测试一下是否是ssrf：



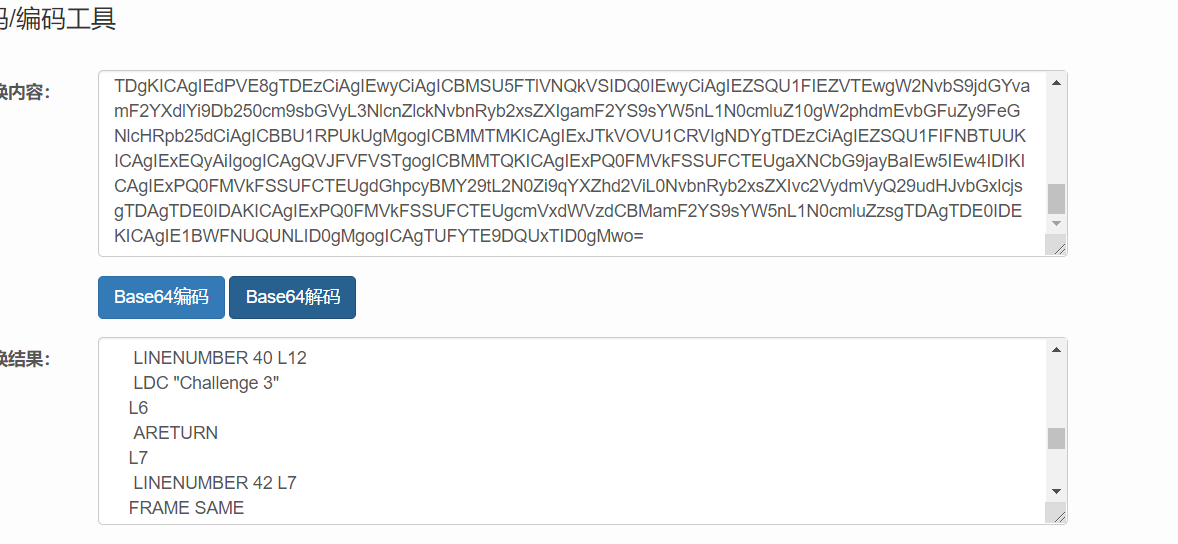
vps可以收到请求



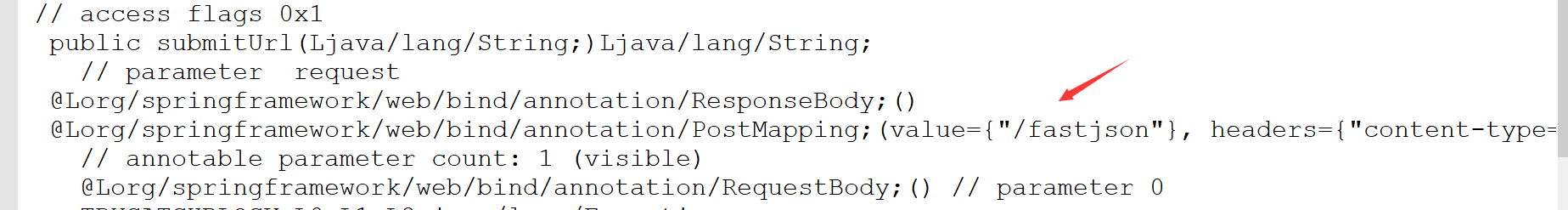
确实存在ssrf，因此尝试file读取文件，根据提示尝试读hint.txt：



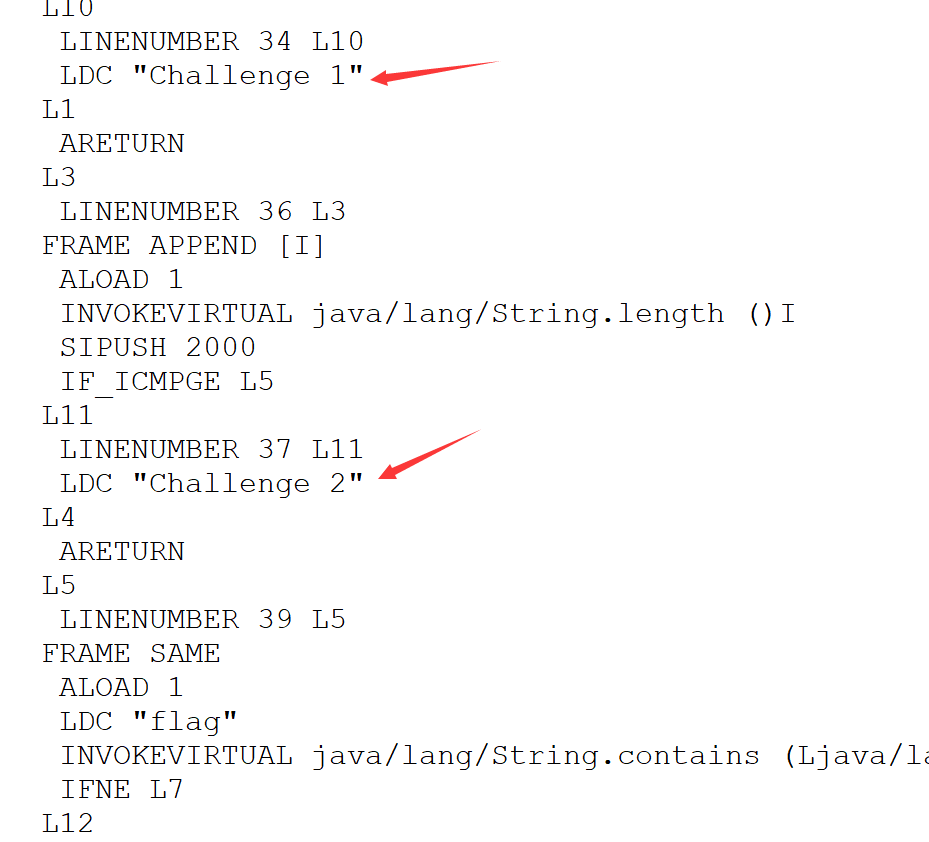
读取到的hint.txt解码后可以发现提示后端用到了FastJson，并且version是1.2.24，并且给了一段字节码指令集



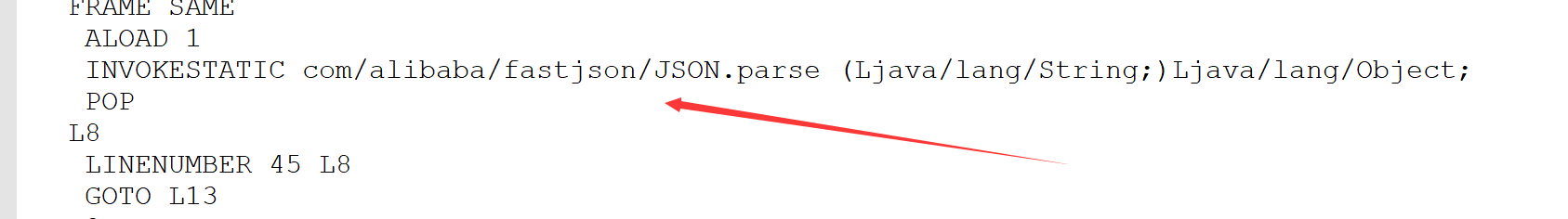
根据Mapping确定FastJson的接口值



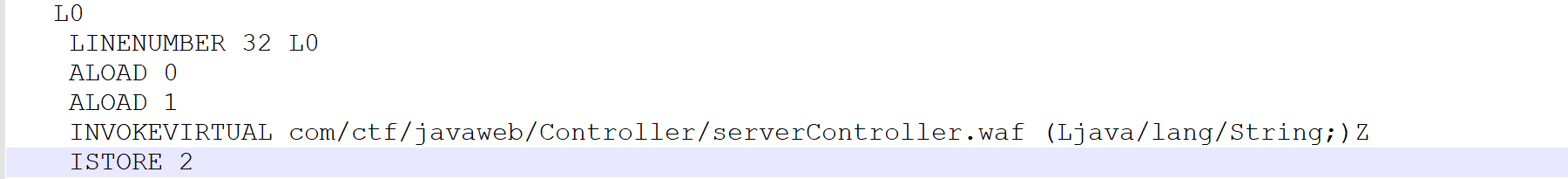
接下来继续向下看：



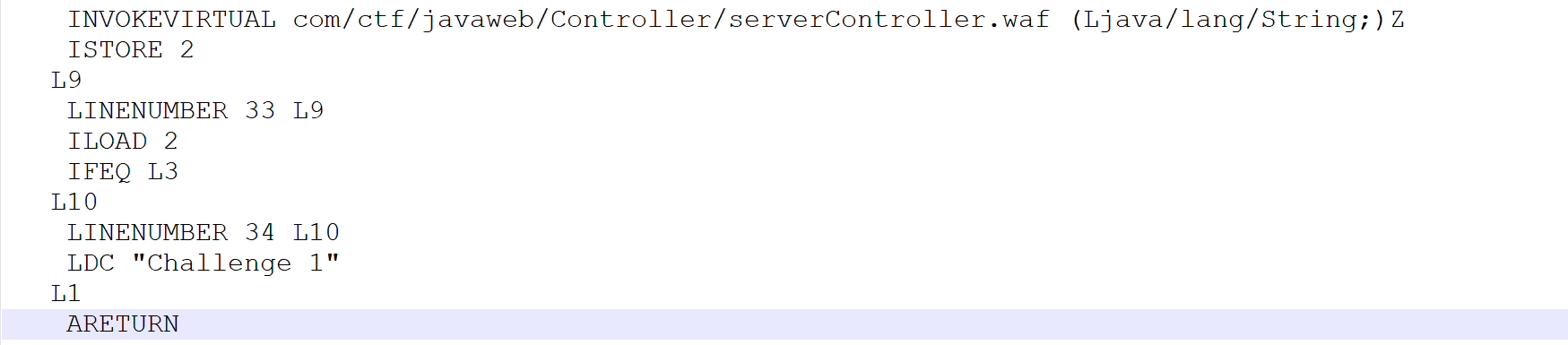
其中有三处challenge，猜到应该要通过这三层挑战，并且在下面可以明显看到这里调用了Json.parse



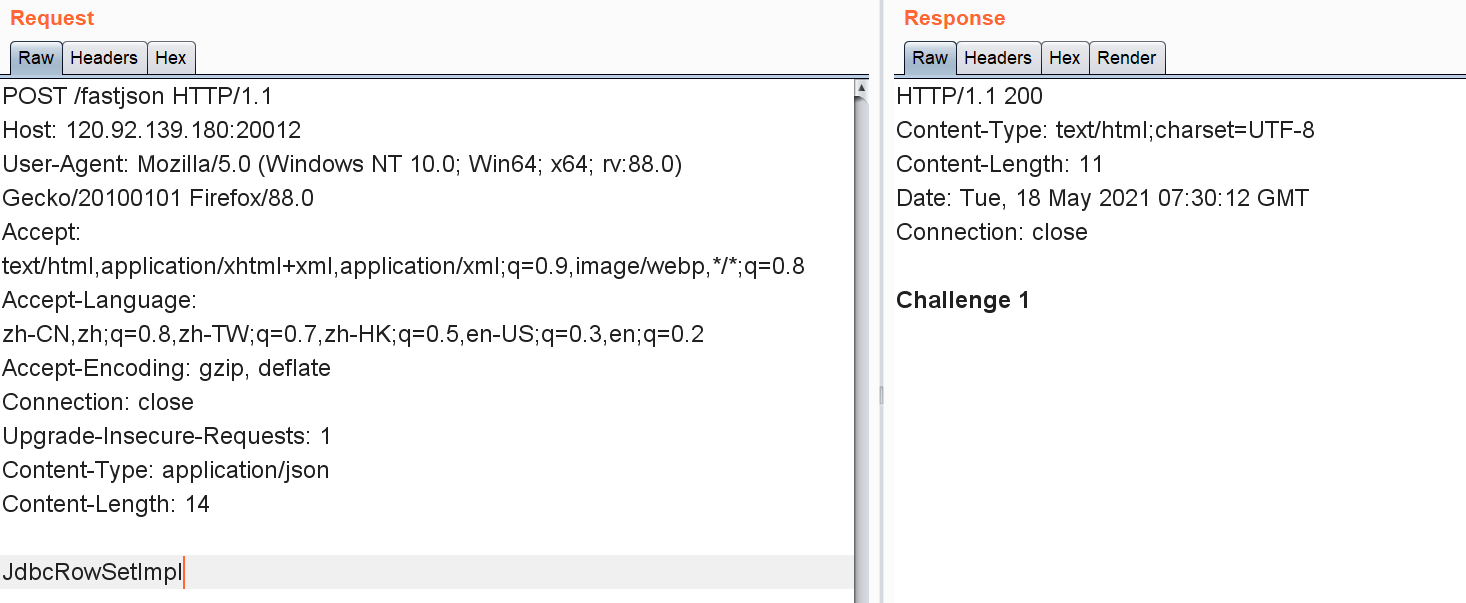
那么接下来就要达到parse的触发

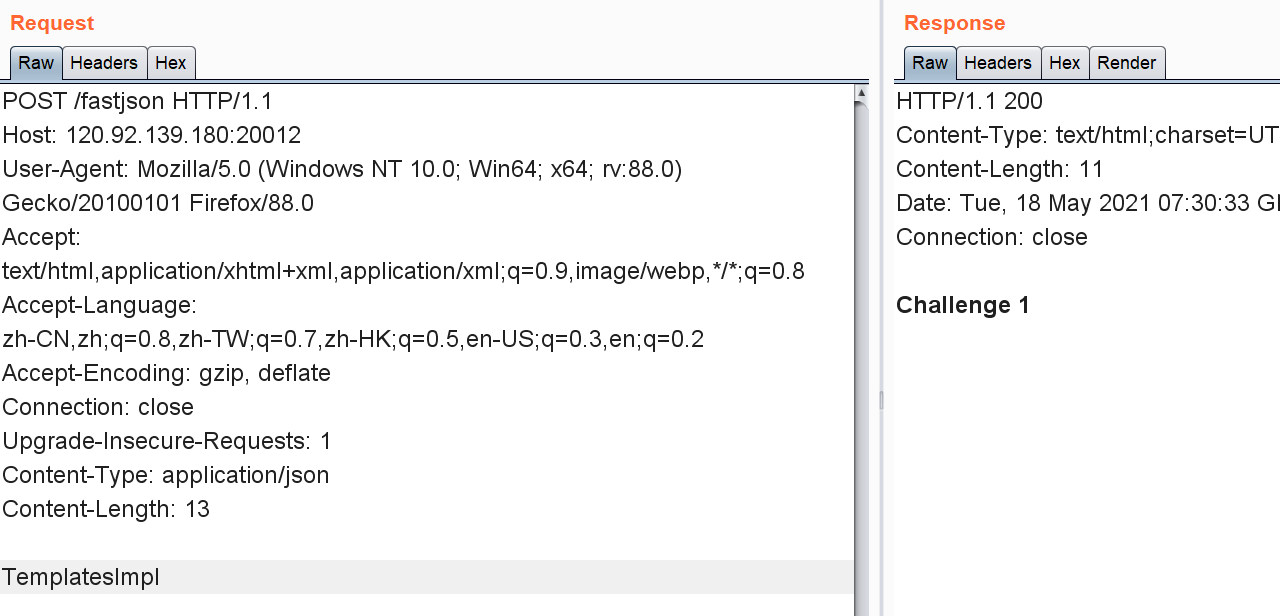


首先从最上面开始看，ALOAD0 ALOAD1,也就是把当前base对象和入口的httpbody的内容放到操作数栈中，然后INVOKEVIRTUAL调用serverController的waf函数，也就是把输入的JSON先经过waf函数过滤，这里返回值是Z类型，也就是布尔值，然后存到局部变量表2处，那么waf的逻辑是不知道的，这里继续往后看可以看到ILOAD2 , IFEQ L3，这里把waf函数返回的结果又压栈和0比，如果等于0，则调到Label3处执行，否则这里直接ARETURN返回，也就是触发waf逻辑无法向后继续执行。

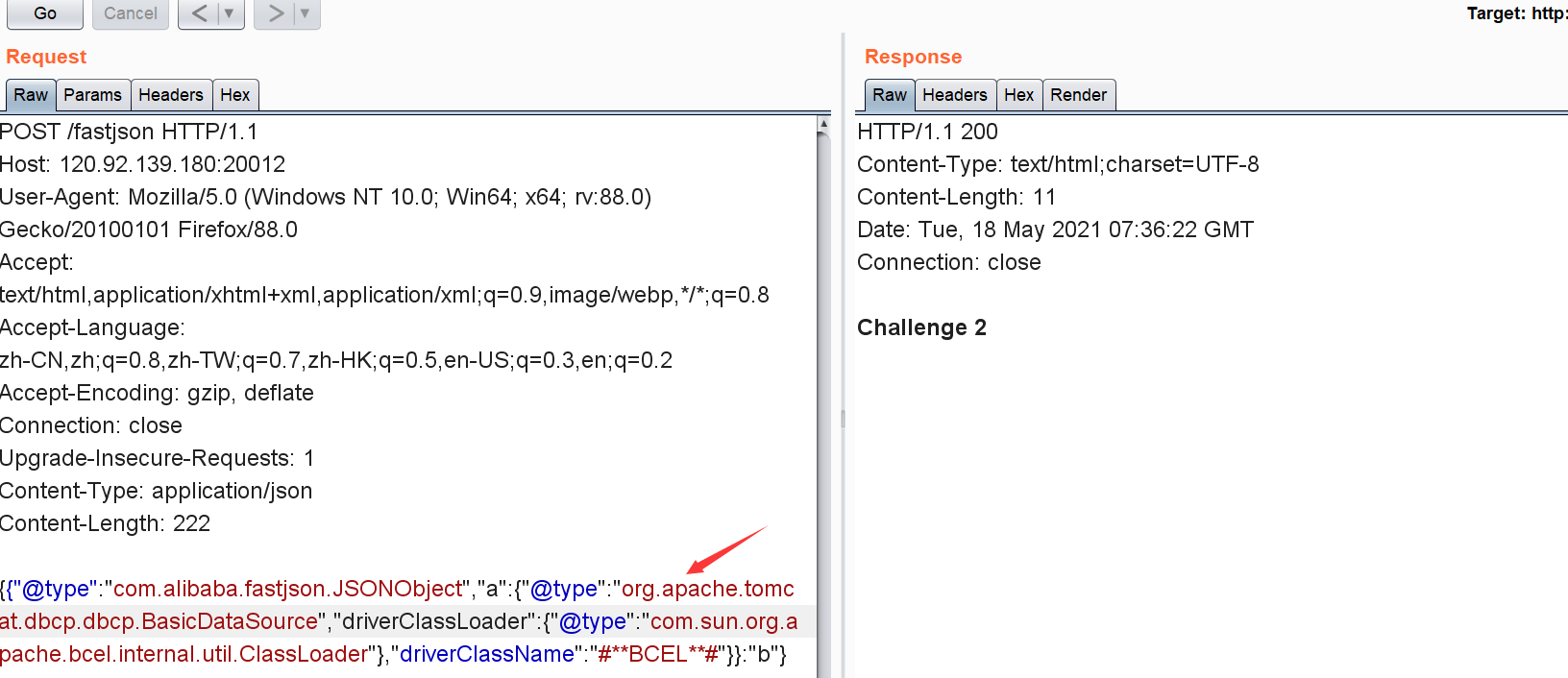


所以接下来要过waf，知道是fj1.2.24，那么常见的就是JdbcRowSetImpl和TemplatesImpl两条链，但是这里只要出现这两条链都会返回challenge1，说明不能用

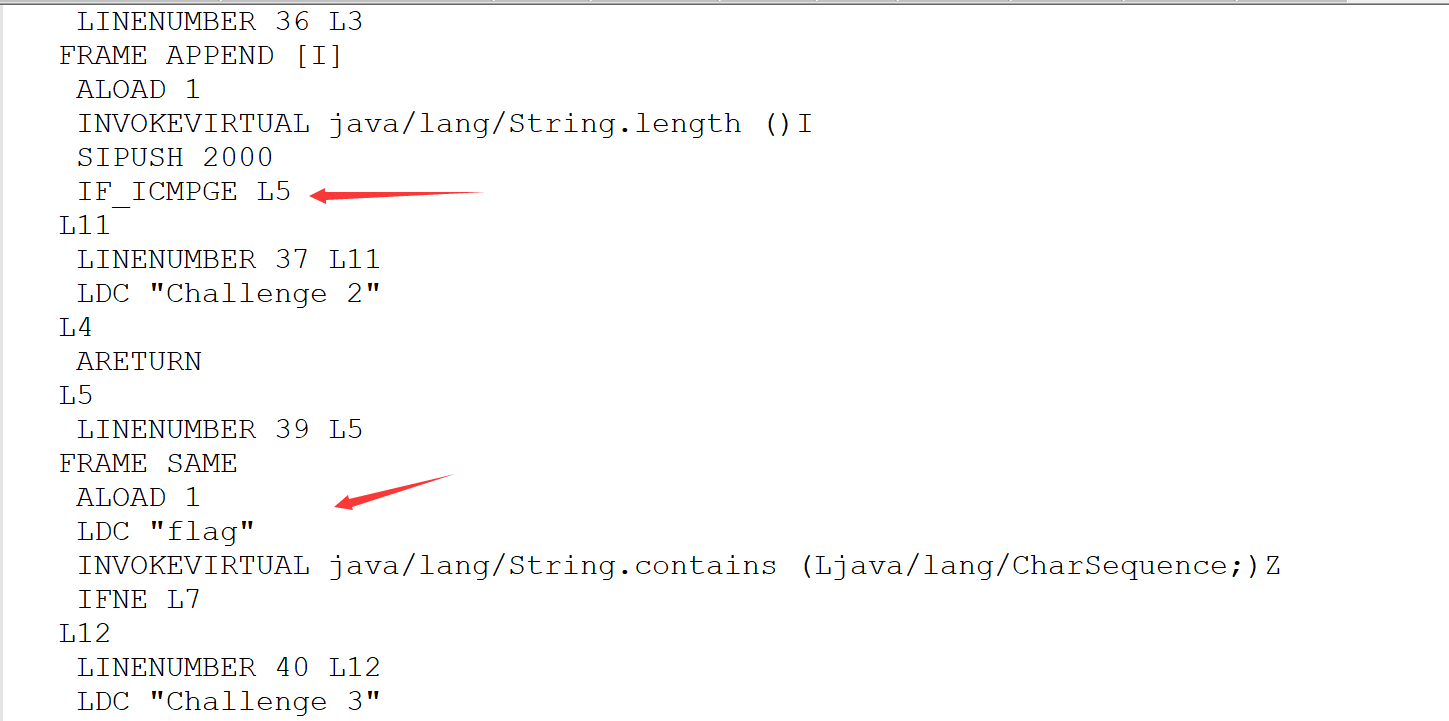




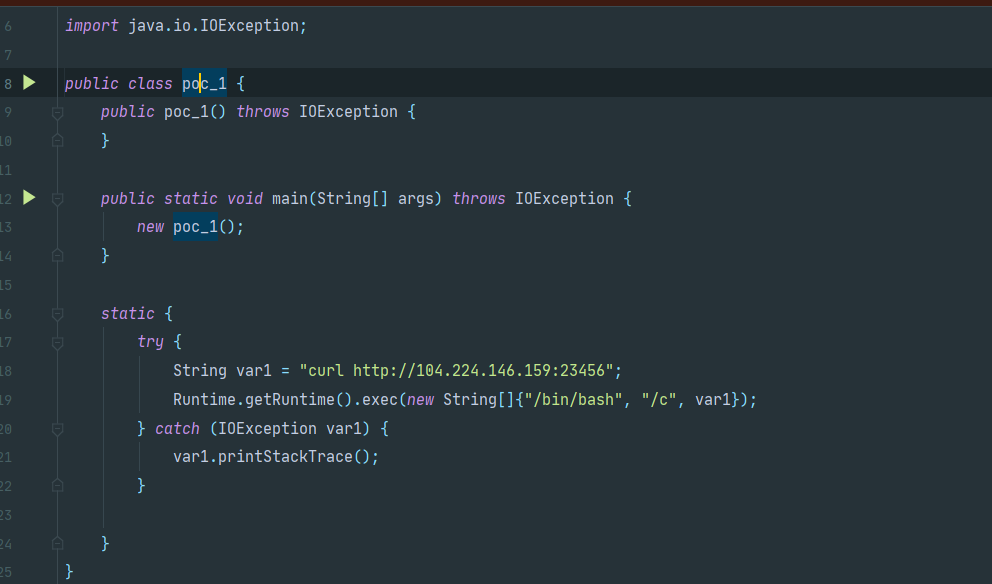
那么肯定想到编码绕过，常见的就是\x和\u，这里出现\x和\u还是会返回challenge1，那么把编码也过滤了，这里1.2.24下实际还有一条链就是basicdatasource，这里虽然看不到后端的pom中的依赖是否满足该链的条件，但是可以进行尝试



当出现这条链时，返回challenge2，说明可以绕过第一层，那么接下来要看下第二层的过滤



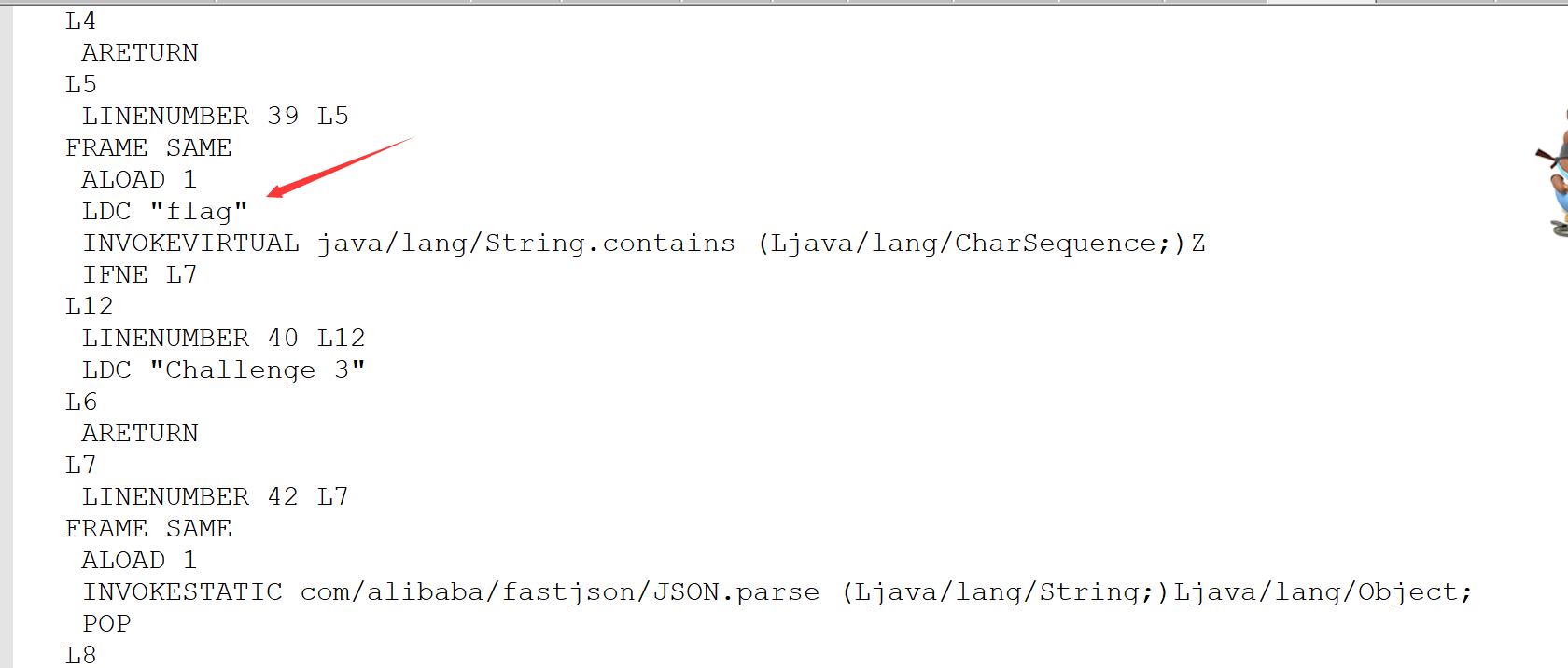
这里ALOAD1 ,然后调用length函数，也就是把http body的json长度压入栈中，然后SIPUSH 2000，压入一个整数，IF\_ICMPGE，也就是判断2000是否大于等于json的长度，如果满足则到Label5，否则ARETURN结束，所以这里第二层就要求我们的JSON长度大于2000，这里先构造一个curl的请求用bcel编码



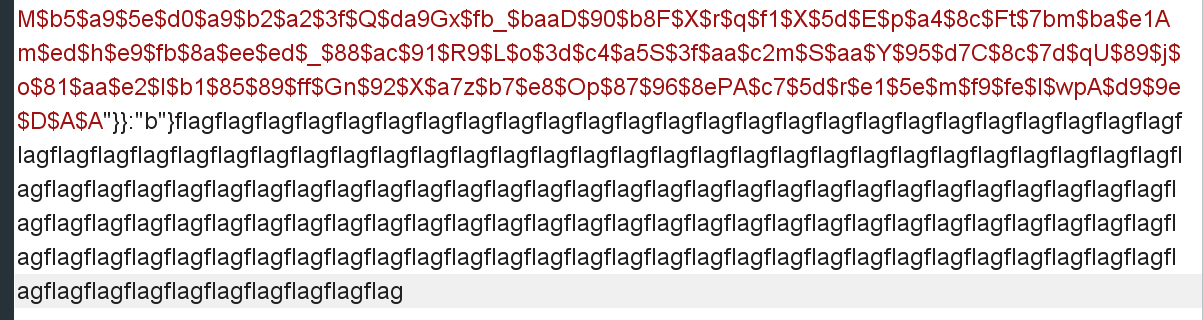
准备一个class文件然后编码得到bcel编码内容填充payload后再发送这里返回的是challenge2，应该是不够长，所以需要填充长度

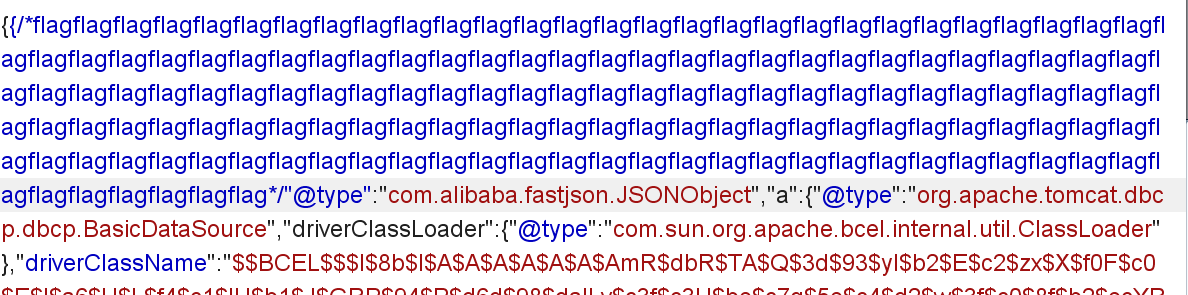


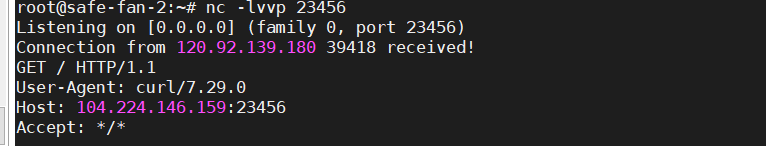
接着往下看，这里ALOD1 LDC flag,然后调用contaions判断JSON中是否包含flag字符串,返回结果压栈用IFNE进行判断，如果包含的话，则到L7，就可以出发parse函数了，如果不包含则ARETURN结束。



所以这里要求JSON长度大于2000并且要包含flag字符串，这里有两种方法，一种是在JSON字符串末尾添加flag,另一种就是添加FastJSON的key注释，/\*\*/的方式：

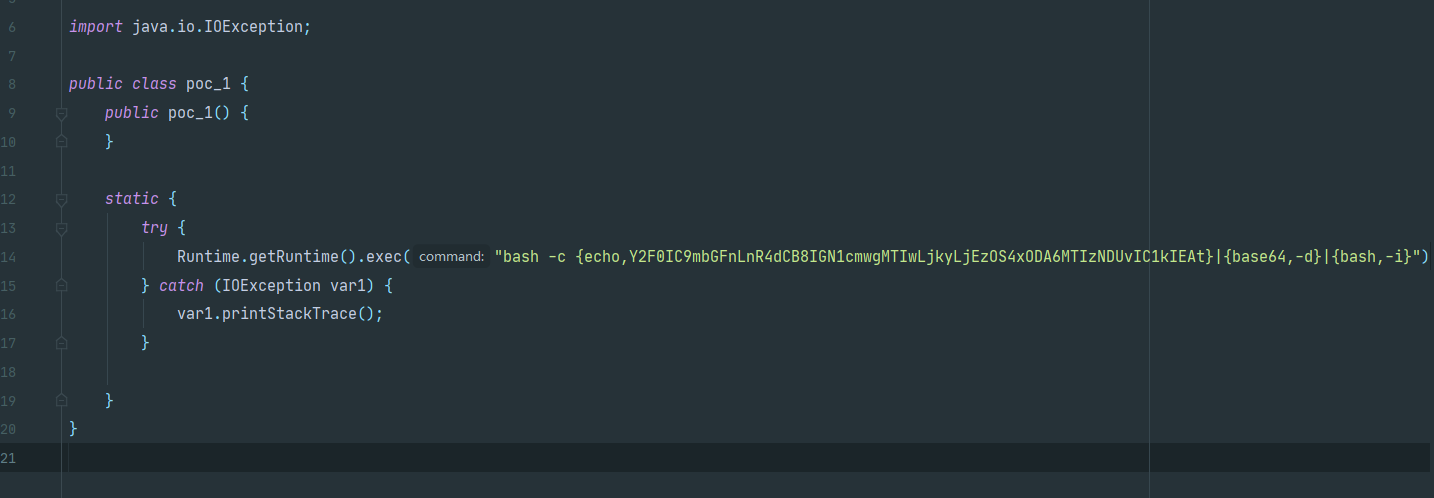






执行后vps都可以收到请求，说明basicdatasource可以用，那么接下来直接拿flag即可

首先整个拿flag的命令，本地编译好class字节码文件



接着生成basicDataSource的payload，发过去就能收到flag了

