Jdk7u21 反序列化漏洞Gadget原理

exploitcat / 2017-11-27 14:41:00 / 浏览数 2494

0x00

昨天看到n1nty发了文章分析深度 - Java 反序列化 Payload 之 JRE8u20，分析了Jre8u20这个Gadgets，读了之后顶礼膜拜，Jre
8u20我并没有研究过，不过看到文章开始提到了7u21中的这个Gadget，可能很多同学对这个原理不是很清楚，所以这里翻出一篇当初分析的原理文分享出来，原文如下：

0x01
对这个Gadget的理解需要一定的java动态代理机制的知识。

上面是这个Gadget的主要代码。
主要是创建了一个LinkedHashSet，在这个LinkedHashSet中有两个对象。
（1）一个是包含了恶意代码的TemplatesImpl对象，这个没啥好说的，创建的过程就是assist这个工具生成字节码。
（2）另外一个是一个代理对象，这个代理对象通过Gadgets.createProxy方法获取，对应的代理类的handler是AnnocationInvocationHandler类的一个实例，并且实现了
（3）并且将这个AnnocationInvocationHandler的实例的memberValues设置成了一个特殊的map。这个map是(key="f5a5a608",
value=templates恶意类），为什么这么设置，待会儿再说。
0x02
在执行反序列化操作时，会调用LinkedHashSet的readObject方法，实际上调用的是其父类HashSet的readObject方法：

在开始是对序列化数据进行了一些校验，校验成功之后，就创建了HashMap装Set中的对象，因此这里是调用了map.put方法。
所以我们来看看java 7中的HashMap.put方法：

这里首先是要计算key的hash，然后通过hash找到对应的索引位置，之后遍历链表中的index位置的元素，然后比较这个元素的hash和要设置的key的hash，来判断值是否已

如果存在则覆盖并返回旧的value，否则就创建一个新的位置插入元素。

POC其实是通过这里的key.equals完成TemplatesImpl中恶意代码触发的，为什么呢？

由于PoC中设置set的顺序是先templates后proxy，所以这里希望调用的是proxy对象的equals，熟悉动态代理的同学都知道，proxy对象中方法的调用是首先要交给handle

这里实际调用了AnnocationInvocationHandler的equalsImpl方法：

通过源码我们就可以看到，这里就是把Object的方法遍历出来，然后挨个儿地调用，所以这样自然会触发TemplatesImpl的getOutputProperties方法，从而触发RCE。

当然，这里就有个问题，我们回到if条件的代码部分：

由于Java的短路机制（具体可看Thinking in java的具体章节），我们必须要满足前一个与运算的条件，即满足e.hash ==
hash这个条件。注意到，当我们将proxy设置到Set时，Set里已经有一个TemplatesImpl的对象在。所以这里实际上就是要满足templates== proxy。

两个对象进行对比，实际上是分别调用两者的hashCode，然后比较其返回值是否相同来判断对象是否相同，实际上需要满足templates.hashCode() ==
proxy.hashCode()。

当我们调用Proxy.hashCode方法的时候，首先调用AnnocationInvocationHandler的invoke方法：

这里是调用的hashCodeImpl方法：

这里从memberValues中进行遍历，还记得PoC中创建Proxy的时候，将AnnocationInvocationHandler的memberValues设置为了(key="f5a5a608",
value=templates恶意类），这里获取出map中的key，也就是f5a5a608这个字符串，调用其hashCode方法，我们可以测试一下，运行这句代码：

```
System.out.println("f5a5a608".hashCode());
```

可以看到返回的值是0，具体原理就不赘述了，有兴趣的同学可以研究一下。

那么这里的hashCodeImpl方法返回的实际上就是memberValueHashCode(e.getValue())的hash值，实际上就是templtes对象的hashCode，所以e.hash ==
hash是成立的，我们的key.equals就触发了，从而使得TemplatesImpl.getOutProperties()方法调用成功，执行了恶意代码。

0x03

Jdk7u21这个Gadgets用到了很多Java内部的类和机制，其中最重要的当然是TemplatesImpl和AnnocationInvocationHandler这两个类，而AnnocationInvocationHand

上一篇：安全工具系列 -- Burpsui... 下一篇：域渗透——利用SYSVOL还原组策...

1. 0 条回复
   - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板