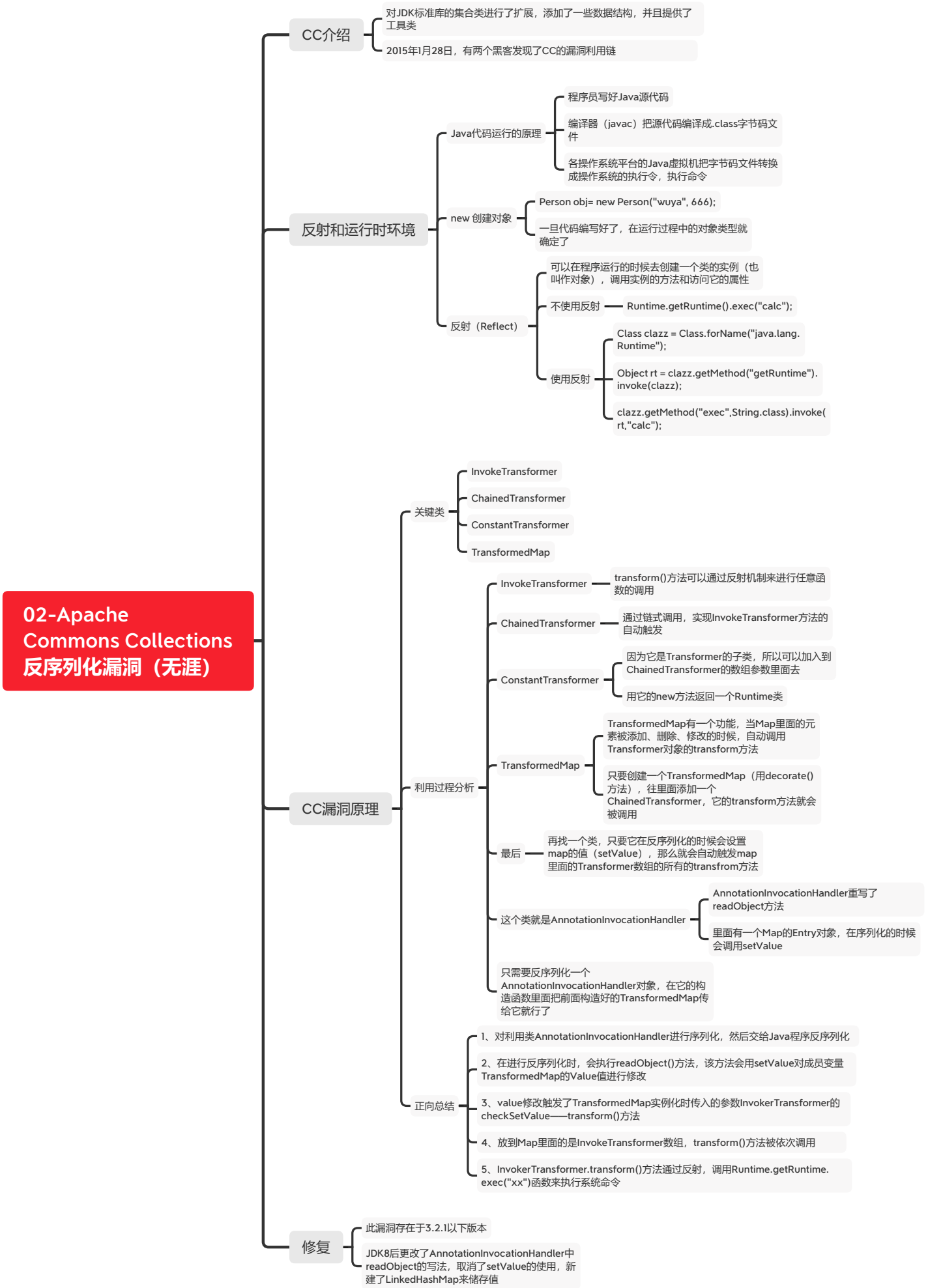
02-Apache Commons Collections 反序列化漏洞（无涯）
CC介绍
对JDK标准库的集合类进行了扩展，添加了一些数据结构，并且提供了工具类
2015年1月28日，有两个黑客发现了CC的漏洞利用链
反射和运行时环境
Java代码运行的原理
程序员写好Java源代码
编译器（javac）把源代码编译成.class字节码文件
各操作系统平台的Java虚拟机把字节码文件转换成操作系统的执行令，执行命令
new 创建对象
Person obj= new Person("wuya", 666);
一旦代码编写好了，在运行过程中的对象类型就确定了
反射（Reflect）
可以在程序运行的时候去创建一个类的实例（也叫作对象），调用实例的方法和访问它的属性
不使用反射
Runtime.getRuntime().exec("calc");
使用反射
Class clazz = Class.forName("java.lang.Runtime");
Object rt = clazz.getMethod("getRuntime").invoke(clazz);
clazz.getMethod("exec",String.class).invoke(rt,"calc");
CC漏洞原理
关键类
InvokeTransformer
ChainedTransformer
ConstantTransformer
TransformedMap
利用过程分析
InvokeTransformer
transform()方法可以通过反射机制来进行任意函数的调用
ChainedTransformer
通过链式调用，实现InvokeTransformer方法的自动触发
ConstantTransformer
因为它是Transformer的子类，所以可以加入到ChainedTransformer的数组参数里面去
用它的new方法返回一个Runtime类
TransformedMap
TransformedMap有一个功能，当Map里面的元素被添加、删除、修改的时候，自动调用Transformer对象的transform方法
只要创建一个TransformedMap（用decorate()方法），往里面添加一个ChainedTransformer，它的transform方法就会被调用
最后
再找一个类，只要它在反序列化的时候会设置map的值（setValue），那么就会自动触发map里面的Transformer数组的所有的transfrom方法
这个类就是AnnotationInvocationHandler
AnnotationInvocationHandler重写了readObject方法
里面有一个Map的Entry对象，在序列化的时候会调用setValue
只需要反序列化一个AnnotationInvocationHandler对象，在它的构造函数里面把前面构造好的TransformedMap传给它就行了
正向总结
1、对利用类AnnotationInvocationHandler进行序列化，然后交给Java程序反序列化
2、在进行反序列化时，会执行readObject()方法，该方法会用setValue对成员变量TransformedMap的Value值进行修改
3、value修改触发了TransformedMap实例化时传入的参数InvokerTransformer的checkSetValue——transform()方法
4、放到Map里面的是InvokeTransformer数组，transform()方法被依次调用
5、InvokerTransformer.transform()方法通过反射，调用Runtime.getRuntime.exec("xx")函数来执行系统命令
修复
此漏洞存在于3.2.1以下版本
JDK8后更改了AnnotationInvocationHandler中readObject的写法，取消了setValue的使用，新建了LinkedHashMap来储存值