Apache Commons Collections 反序列化漏洞

漏洞爆出

- 2015.01.28 Gabriel Lawrence和Chris Frohoff

https://speakerdeck.com/frohoff/appseccali-2015-marshalling-pickles-how-deserializing-objects-can-ruin-your-day
https://github.com/frohoff/ysoserial

- 2015.11.06 FoxGlove Security @breenmachine

https://commons.apache.org/proper/commons-collections/release_3_2_2.html
https://issues.apache.org/jira/browse/COLLECTIONS-580

本地复现环境

- jdk 1.7.0_80
- IDEA Project Structrure、Settings——Java compile等设置成java7
- Apache Commons Collections ≤ 3.2.1

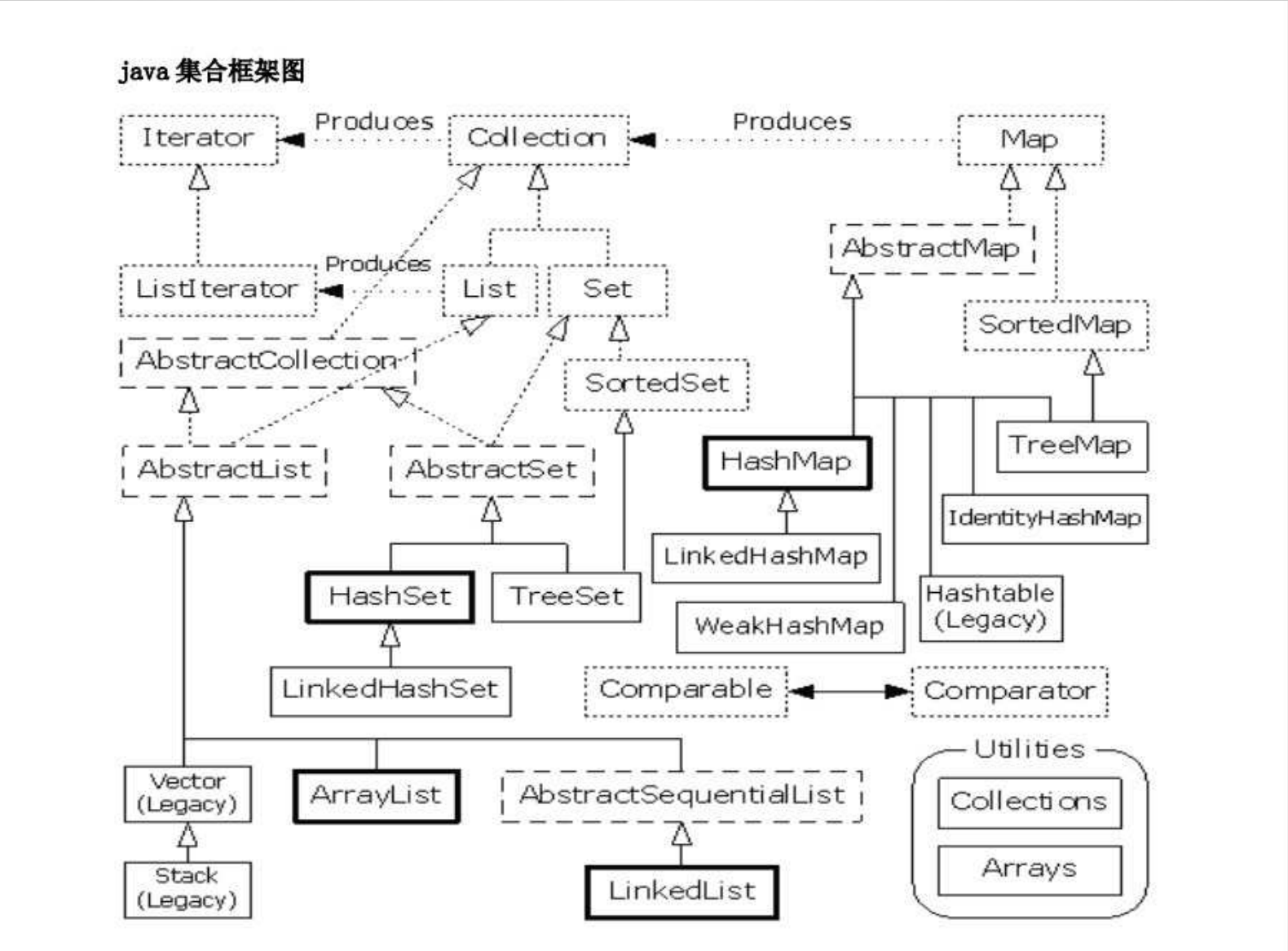
课程目录

1、Apache Commons Collections介绍
2、Java反射机制
3、Apache Commons Collections漏洞原理
4、漏洞修复

01 Apache Commons Collections介绍

Java集合

List
Map
Set



Commons Collections

https://commons.apache.org/proper/commons-collections/

- Bag interface for collections that have a number of copies of each object
- BidiMap interface for maps that can be looked up from value to key as well and key to value
- MapIterator interface to provide simple and quick iteration over maps
- Transforming decorators that alter each object as it is added to the collection
- Composite collections that make multiple collections look like one
- Ordered maps and sets that retain the order elements are added in, including an LRU based map
- Reference map that allows keys and/or values to be garbage collected under close control
- Many comparator implementations
- Many iterator implementations
- Adapter classes from array and enumerations to collections
- Utilities to test or create typical set-theory properties of collections such as union, intersection, and closure

使用


```
<dependency>
    <groupId>commons-collections</groupId>
    <artifactId>commons-collections</artifactId>
    <version>3.1</version>
</dependency>
```

问题

1、哪里出现了可以执行任意代码的问题?
2、序列化的payload怎么构造?

02

Java反射机制

JVM

Java代码运行原理：
1、源码
2、编译器（javac）编译为字节码.class文件
3、各平台JVM解释器把字节码文件转换成操作系统指令

创建对象


```
Person obj= new Person("wuya", 666);
```

反射

在程序运行的时候动态创建一个类的实例，
调用实例的方法和访问它的属性

Class —— Instance
Person —— new Person("无涯")

03

Apache Commons Collections 漏洞原理 ≤ 3.2.1

CC漏洞

- 2015年黑客Gabriel Lawrence和Chris Frohoff发现
- 影响WebLogic、WebSphere、JBoss、Jenkins、OpenNMS等大型框架

CC关键类

- InvokeTransformer

利用Java反射机制来创建类实例

- ChainedTransformer

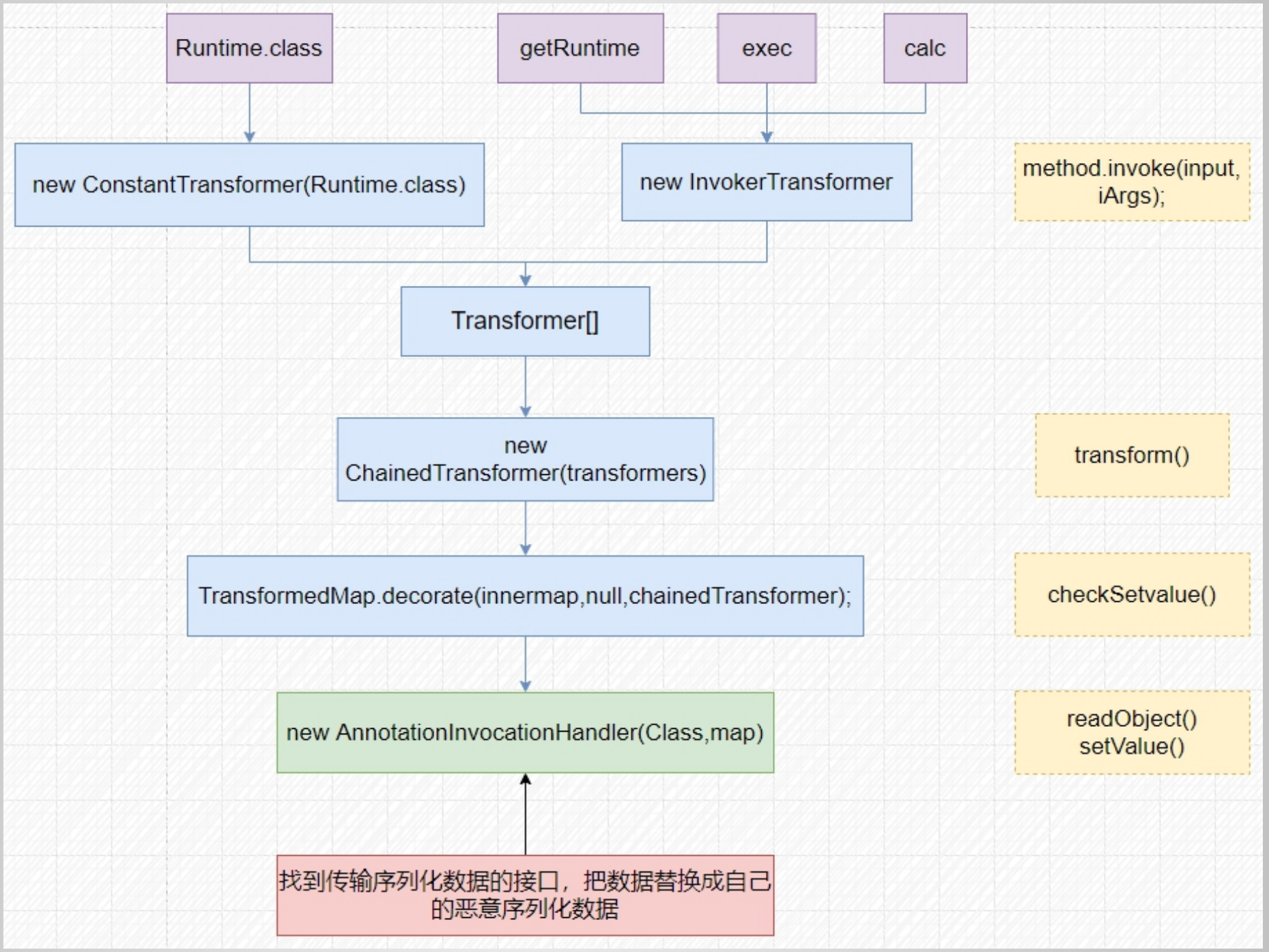
实现了Transformer链式调用，我们只需要传入一个Transformer数组ChainedTransformer就可以实现依次的去调用每一个Transformer的transform()方法

- ConstantTransformer

transform()返回构造函数的对象

- TransformedMap

调用链路


Runtime.class
getRuntime
exec
calc
new ConstantTransformer(Runtime.class)
new InvokerTransformer
method.invoke(input, iArgs);
Transformer[]
new ChainedTransformer(transformers)
transform()
TransformedMap.decorate(innermap,null,chainedTransformer);
checkSetvalue()
new AnnotationInvocationHandler(Class,map)
readObject()
setValue()
找到传输序列化数据的接口，把数据替换成自己的恶意序列化数据

poc构造思路

1、InvokeTransformer

反射执行代码

2、ChainedTransformer

链式调用，自动触发

3、ConstantTransformer

获得对象

4、TransformedMap

元素变化执行transform，setValue——checkSetValue

5、AnnotationInvocationHandler

readObject 调用Map的setValue

调用流程

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

04 漏洞修复

漏洞修复

1、升级Apache Commons Collections到最新版
2、升级JDK版本

Thank you for watching

无涯老师