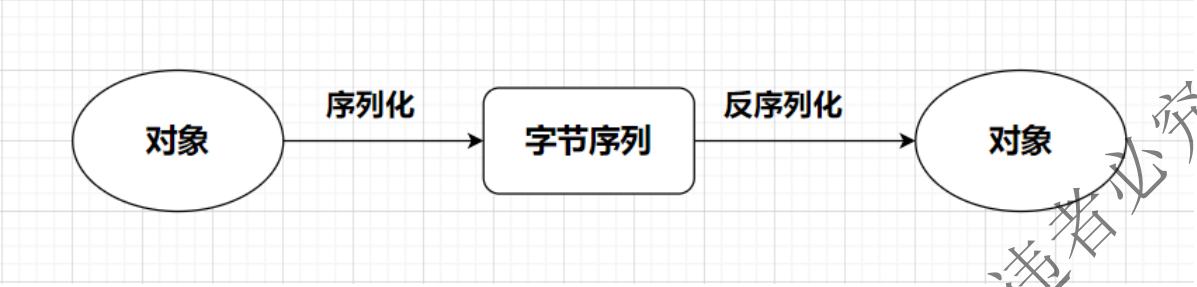
反序列化漏洞在Java代码审计中是非常重要的基础之一。对于基础薄弱的朋友一定要好好学习，刚开始学也不必过多纠结，只需要记住反序列化的方法就好，后面跟着实战一点点领悟其中的奥妙。

一、序列化与反序列化

在了解反序列化之前，一定要先明白序列化。

通过下面一张图，先简单理解序列化与反序列化的关系。



1.1、什么是序列化

序列化是指把 Java 对象转换为字节序列的过程，目的是便于保存在内存、文件、数据库中。

`ObjectOutputStream`类的`writeObject()`方法可以实现序列化。

writeObject()方法：将指定的对象写入ObjectOutputStream中。

```
public final void writeObject(Object obj)
                        throws IOException
```

官方文档详细说明：
`https://docs.oracle.com/javase/8/docs/api/java/io/ObjectOutputStream.html#writeObject-java.lang.Object-`

一个类的对象要想序列化成功，必须满足两个条件：

- 该类必须实现 java.io.Serializable 接口。
- 该类的所有属性必须是可序列化的。如果有一个属性不是可序列化的，则该属性必须注明是短暂的。

1.2、什么是反序列化

反序列化是指把字节序列恢复为 Java 对象的过程。

`ObjectInputStream`类的`readObject()`方法可实现反序列化。

readObject()方法：从ObjectInputStream读取一个对象。

```
public final Object readObject()
                        throws IOException,
                               ClassNotFoundException
```

官方文档详细说明：
`https://docs.oracle.com/javase/8/docs/api/java/io/ObjectInputStream.html#readObject--`

1.3、示例代码

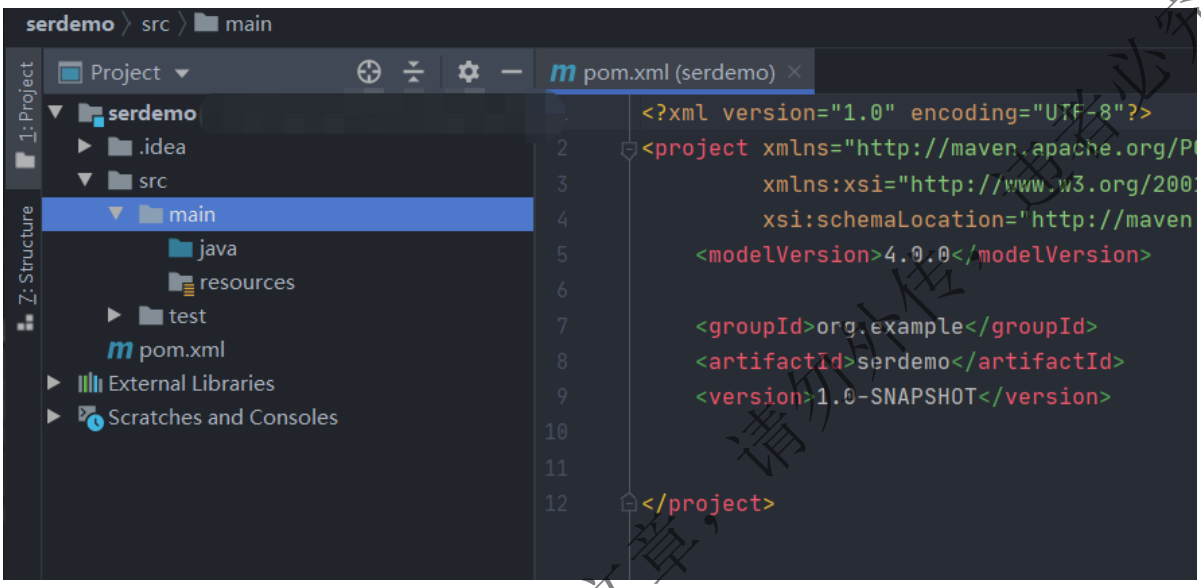
老规矩，先新建一个名为`serdemo`的项目工程，用于下面示例代码的练习。

①、打开IDEA，点击`Create New Project`，创建新的工程。如果打开IDEA后进入之前项目，则需在左上角点击`File -> New -> Porject...`即可。

②、左侧选择Maven，配置默认即可，不用选择任何模板，点击Next。

③、起个项目名称为`serdemo`，其他默认即可，点击Finish。

④、最终目录结构如下图所示：

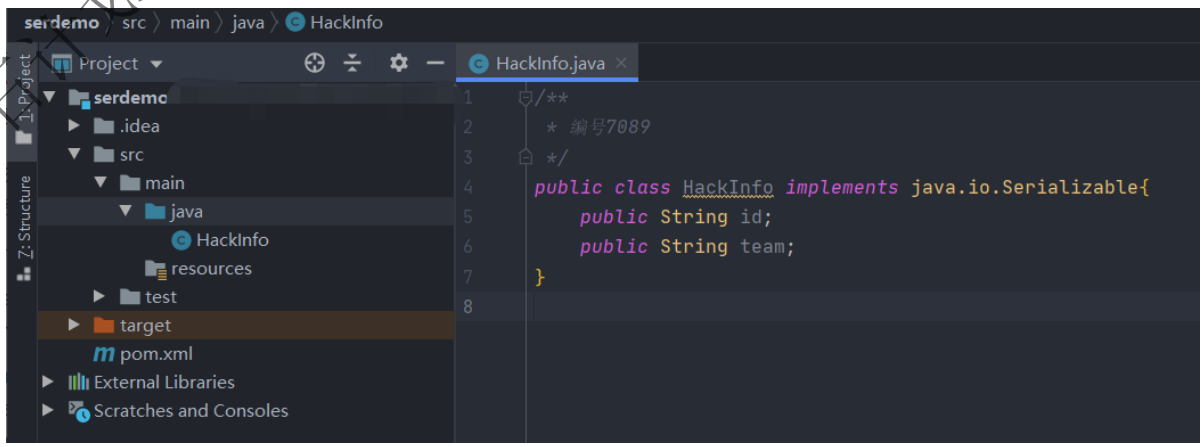


1.3.1、代码Demo

①、在java目录下创建一个名为`HackInfo`的Java Class，并键入以下代码，最终如下图所示：

前面提到，一个类的对象要想序列化成功，必须满足两个条件：1、该类必须实现java.io.Serializable接口。2、该类的所有属性必须是可序列化的。如果有一个属性不是可序列化的，则该属性必须注明是短暂的。

```
public class HackInfo implements java.io.Serializable{
    public String id;
    public String team;
}
```

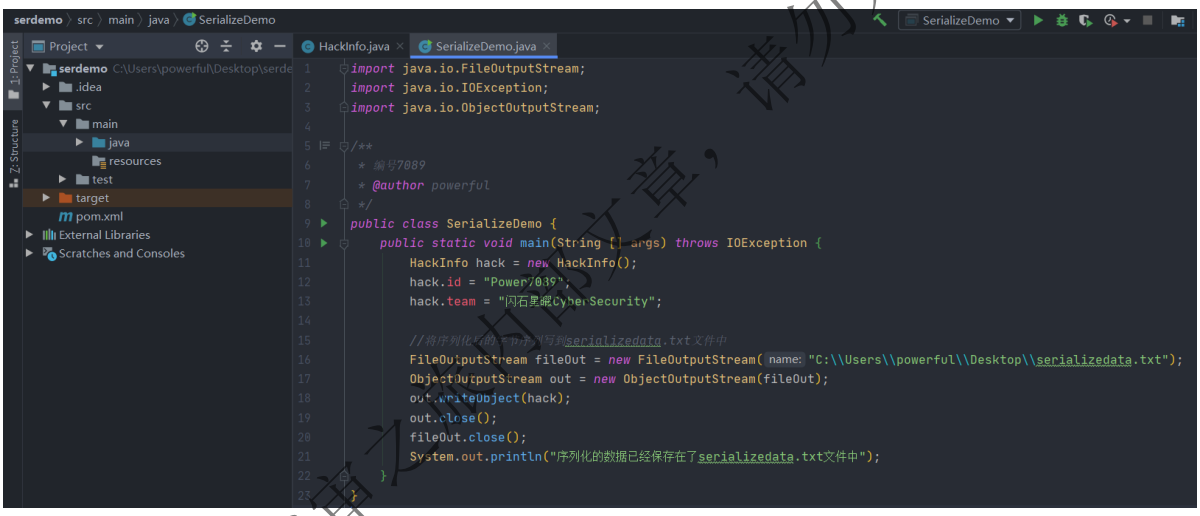


②、在java目录下创建一个名为`SerializeDemo`的Java Class，并键入以下代码，最终如下图所示：

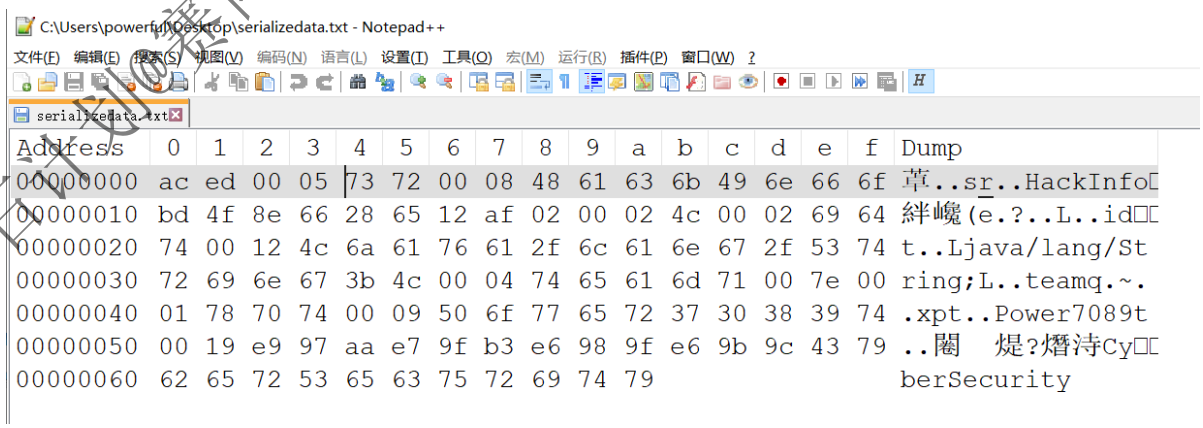
```
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectOutputStream;

public class SerializeDemo {
    public static void main(String [] args) throws IOException {
        HackInfo hack = new HackInfo();
        hack.id = "Power7089";
        hack.team = "闪石星曜CyberSecurity";

        //将序列化后的字节序列写到serializedata.txt文件中
        FileOutputStream fileOut = new
FileOutputStream("C:\\Users\\powerful\\Desktop\\serializedata.txt");
        ObjectOutputStream out = new ObjectOutputStream(fileOut);
        out.writeObject(hack);
        out.close();
        fileOut.close();
        System.out.println("序列化的数据已经保存在了serializedata.txt文件中");
    }
}
```



③、我们可以使用Notepad++打开`serializedata.txt`文件，使用Hex-Editor插件以二进制形式查看数据，如下图所示：



注意：序列化的数据会有明显的特征，都是以`aced 0005 7372`开头的。

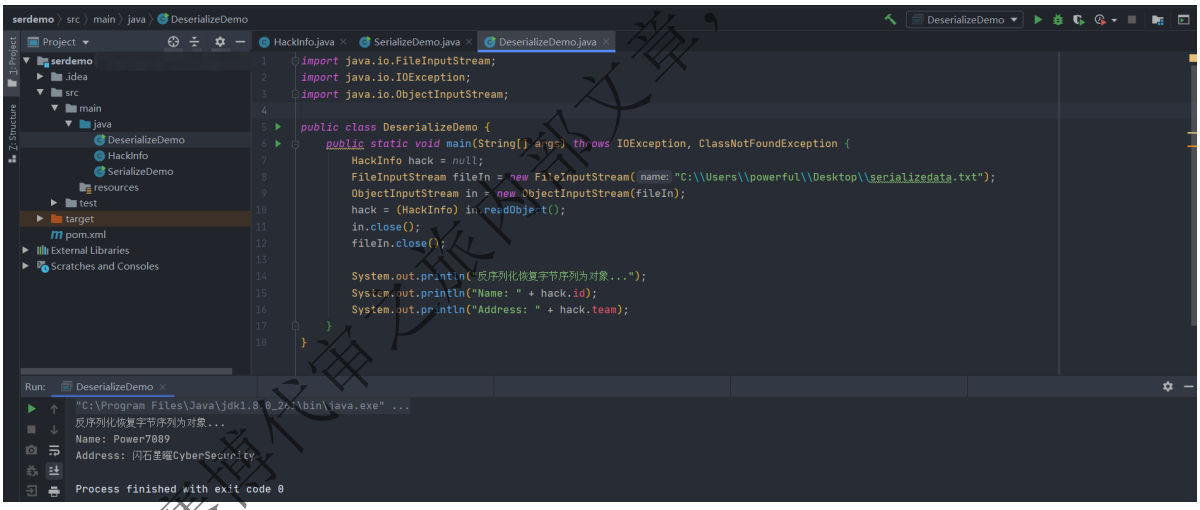
④、下面们通过反序列化操作，将字节序列还原成对象。在java目录下创建一个名为`DeserializeDemo`的Java Class，并键入以下代码，最终如下图所示：

```
import java.io.FileInputStream;
import java.io.IOException;
import java.io.ObjectInputStream;

/**
 * 编号7089
 */
public class DeserializeDemo {
   public static void main(String[] args) throws IOException,
ClassNotFoundException {

       HackInfo hack = null;
       FileInputStream fileIn = new
FileInputStream("C:\\Users\\powerful\\Desktop\\serializedata.txt");
       ObjectInputStream in = new ObjectInputStream(fileIn);
       hack = (HackInfo) in.readObject();
       in.close();
       fileIn.close();

       System.out.println("Deserialized Employee...");
       System.out.println("Name: " + hack.id);
       System.out.println("Address: " + hack.team);
   }
}
```



先从一个简简单单的代码例子理解序列化与反序列化，大家一定要动手调试一下。

在第二章节，我们进一步学习Java反序列化漏洞。