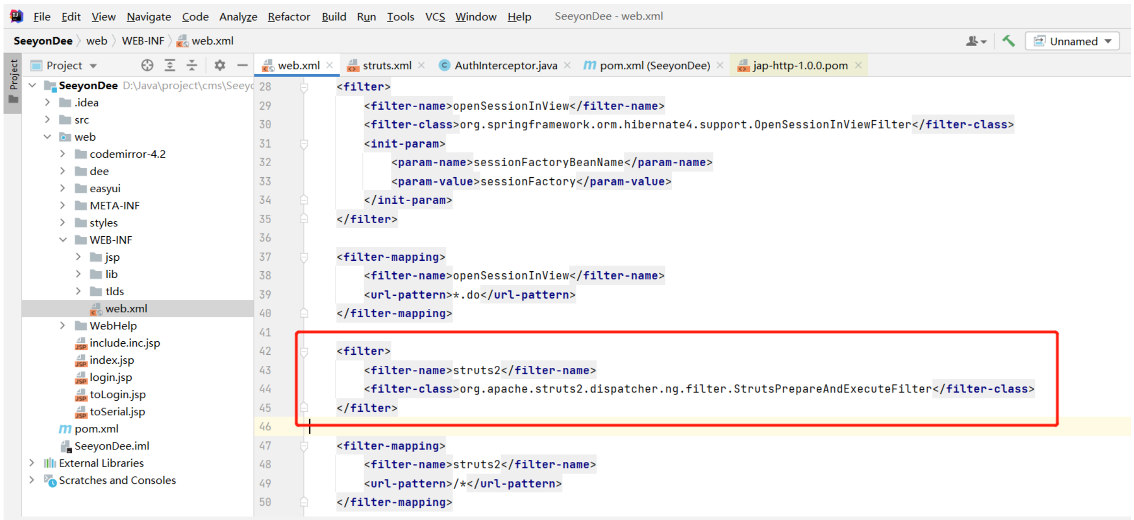
DEE可视化工具

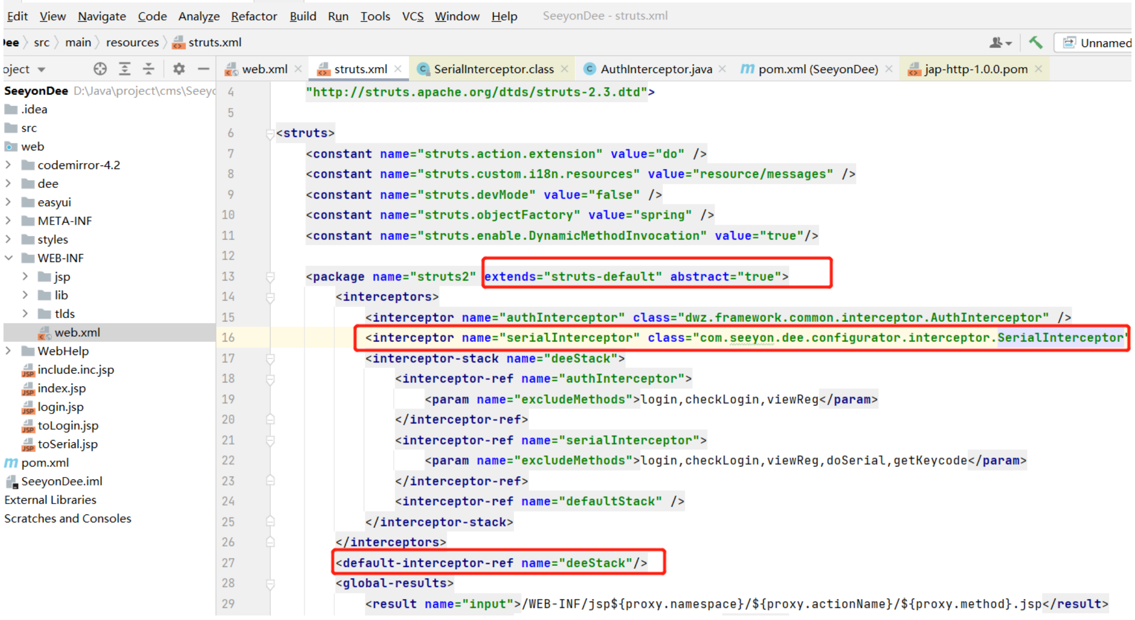
软件破解

认证过程

- 系统使用`struts2`框架开发，通过在`struts2.xml`配置文件当中找到过滤器获取到认证程序。



```xml
<filter>
    <filter-name>openSessionInView</filter-name>
    <filter-class>org.springframework.orm.hibernate4.support.OpenSessionInViewFilter</filter-class>
    <init-param>
        <param-name>sessionFactoryBeanName</param-name>
        <param-value>sessionFactory</param-value>
    </init-param>
</filter>

<filter-mapping>
    <filter-name>openSessionInView</filter-name>
    <url-pattern>*.do</url-pattern>
</filter-mapping>

<filter>
    <filter-name>struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```



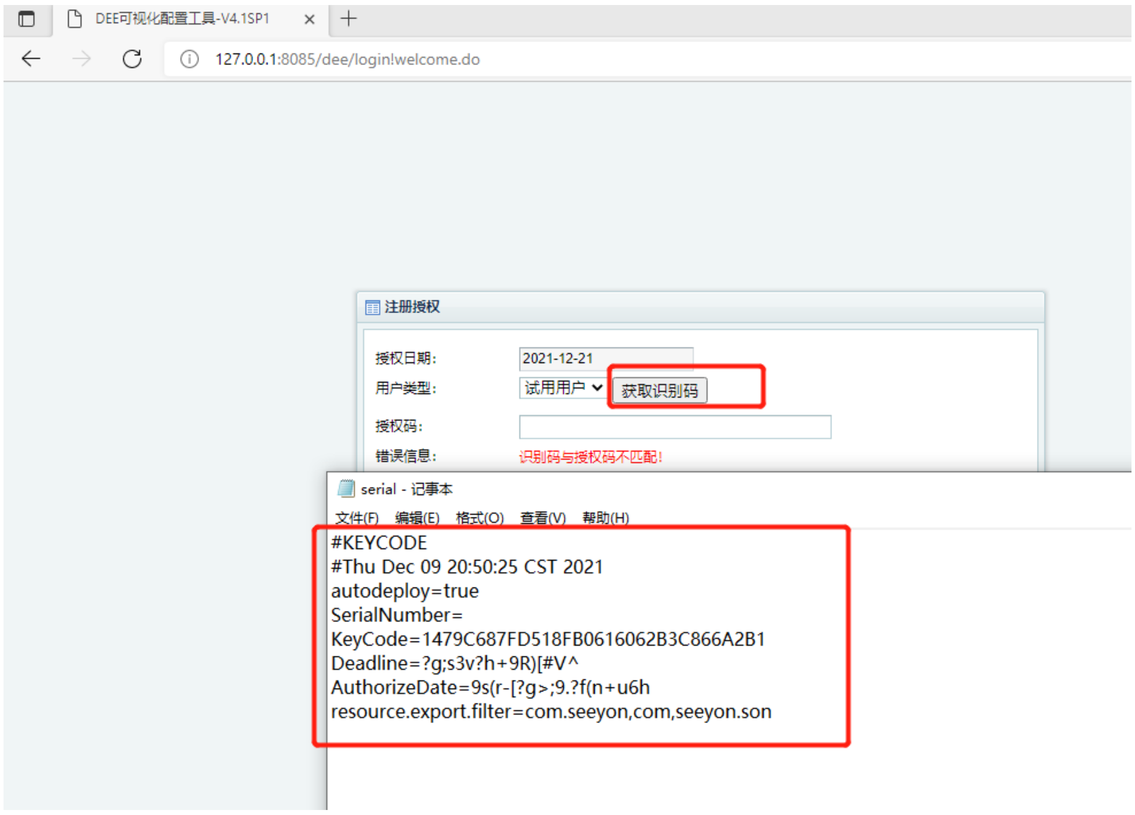
```xml
    "http://struts.apache.org/dtds/struts-2.3.dtd">

<struts>
    <constant name="struts.action.extension" value="do" />
    <constant name="struts.custom.i18n.resources" value="resource/messages" />
    <constant name="struts.devMode" value="false" />
    <constant name="struts.objectFactory" value="spring" />
    <constant name="struts.enable.DynamicMethodInvocation" value="true"/>

    <package name="struts2" extends="struts-default" abstract="true">
        <interceptors>
            <interceptor name="authInterceptor" class="dwz.framework.common.interceptor.AuthInterceptor" />
            <interceptor name="serialInterceptor" class="com.seeyon.dee.configurator.interceptor.SerialInterceptor"
            <interceptor-stack name="deeStack">
                <interceptor-ref name="authInterceptor">
                    <param name="excludeMethods">login,checkLogin,viewReg</param>
                </interceptor-ref>
                <interceptor-ref name="serialInterceptor">
                    <param name="excludeMethods">login,checkLogin,viewReg,doSerial,getKeycode</param>
                </interceptor-ref>
                <interceptor-ref name="defaultStack" />
            </interceptor-stack>
        </interceptors>
        <default-interceptor-ref name="deeStack"/>
        <global-results>
            <result name="input">/WEB-INF/jsp${proxy.namespace}/${proxy.actionName}/${proxy.method}.jsp</result>
```

- `com.seeyon.dee.configurator.interceptor.SerialInterceptor`

```java
//
// Source code recreated from a .class file by IntelliJ IDEA
// (powered by FernFlower decompiler)
//

package com.seeyon.dee.configurator.interceptor;

import com.opensymphony.xwork2.ActionContext;
import com.opensymphony.xwork2.ActionInvocation;
import com.opensymphony.xwork2.interceptor.MethodFilterInterceptor;
import com.seeyon.v3x.dee.common.base.util.AuthorizeUtil;
import com.seeyon.v3x.dee.common.base.util.Encipher;
import com.seeyon.v3x.dee.util.DataChangeUtil;
import java.io.File;
import java.io.FileInputStream;
import java.io.InputStream;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.Properties;
import javax.servlet.http.HttpServletRequest;

public class SerialInterceptor extends MethodFilterInterceptor {
    private static final long serialVersionUID = 4085963428017959650L;
    private static final String filePath = DataChangeUtil.getRealPath("/") +
"config" + "/serial.properties";

    public SerialInterceptor() {
    }

    protected String doIntercept(ActionInvocation invocation) throws Exception {
        ActionContext actionContext = invocation.getInvocationContext();
        HttpServletRequest request =
(HttpServletRequest)actionContext.get("com.opensymphony.xwork2.dispatcher.HttpSe
rvletRequest");
        String errorMsg = "";
        InputStream in = new FileInputStream(filePath);
        Properties property = new Properties();
        property.load(in);
        in.close();
        String keycode = property.getProperty("KeyCode");
        String SerialNumber = property.getProperty("SerialNumber");
        String encode_authorizeDate = property.getProperty("AuthorizeDate");
        String authorizeDate = Encipher.Decode(encode_authorizeDate);
        String encode_deadline = property.getProperty("Deadline");
        String deadline = Encipher.Decode(encode_deadline);
        String logfilesPath = DataChangeUtil.getRealPath("/") +
"../../../../logs";
        File logsPath = new File(logfilesPath);
        Date lastDate = null;
        if (logsPath.exists() && logsPath.isDirectory()) {
            File[] logs = logsPath.listFiles();
            File[] var20 = logs;
            int var19 = logs.length;

            for(int var18 = 0; var18 < var19; ++var18) {
                File file = var20[var18];
                if (lastDate == null) {
                    lastDate = new Date(file.lastModified());
                } else {
                    lastDate = lastDate.getTime() > file.lastModified() ?
lastDate : new Date(file.lastModified());
                }
            }
        }

        if (lastDate != null && (new Date()).before(lastDate)) {
            errorMsg = errorMsg + "请勿修改系统时间！";
        } else {
            if (keycode == null || !AuthorizeUtil.getK(encode_authorizeDate,
encode_deadline).equals(keycode) || !AuthorizeUtil.getS(keycode,
"250").equals(SerialNumber)) {
                errorMsg = errorMsg + "识别码与授权码不匹配！";
            }

            if (authorizeDate != null && !"".equals(authorizeDate.trim()) &&
deadline != null && !"".equals(deadline.trim())) {
                SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");

                try {
                    Date begin = df.parse(authorizeDate);
                    Calendar calBegin = Calendar.getInstance();
                    calBegin.setTime(begin);
                    Calendar calNow = Calendar.getInstance();
                    if (calNow.before(calBegin)) {
                        errorMsg = errorMsg + "未进入授权期限！";
                    }

                    calBegin.add(2, Integer.parseInt(deadline));
                    if (calNow.after(calBegin)) {
                        errorMsg = errorMsg + "授权已过期，请重新注册！";
                        request.setAttribute("Expired", true);
                    }
                } catch (ParseException var21) {
                    errorMsg = errorMsg + "授权日期错误！";
                }
            } else {
                errorMsg = errorMsg + "授权日期和期限为空！";
                request.setAttribute("Expired", true);
            }
        }

        if (errorMsg.trim().equals("")) {
            return invocation.invoke();
        } else {
            request.setAttribute("errorMsg", errorMsg);
            request.setAttribute("authorizeDate", encode_authorizeDate != null
&& !"".equals(encode_authorizeDate.trim()) ?
Encipher.Decode(encode_authorizeDate) : (new SimpleDateFormat("yyyy-MM-
dd")).format(new Date(System.currentTimeMillis())));
            request.setAttribute("deadline", deadline);
            request.setAttribute("keyCode", keycode);
            request.setAttribute("serialNumber", SerialNumber);
            return "serial";
        }
    }
}
```

算法逆向

- 在系统安装后会在`webapps\ROOT\WEB-INF\classes\config`目录生成一个`serial.properties`认证配置文件，也可以通过注册授权界面的获取识别码更新该配置文件。


DEE可视化配置工具-V4.1SP1
127.0.0.1:8085/dee/login!welcome.do
注册授权
获取识别码
serial - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

授权日期: 2021-12-21
用户类型: 试用用户
授权码:
错误信息: 识别码与授权码不匹配!

```
#KEYCODE
#Thu Dec 09 20:50:25 CST 2021
autodeploy=true
SerialNumber=
KeyCode=1479C687FD518FB0616062B3C866A2B1
Deadline=?g;s3v?h+9R)[#V^
AuthorizeDate=9s(r-[?g>;9.?f(n+u6h
resource.export.filter=com.seeyon,com,seeyon.son
```

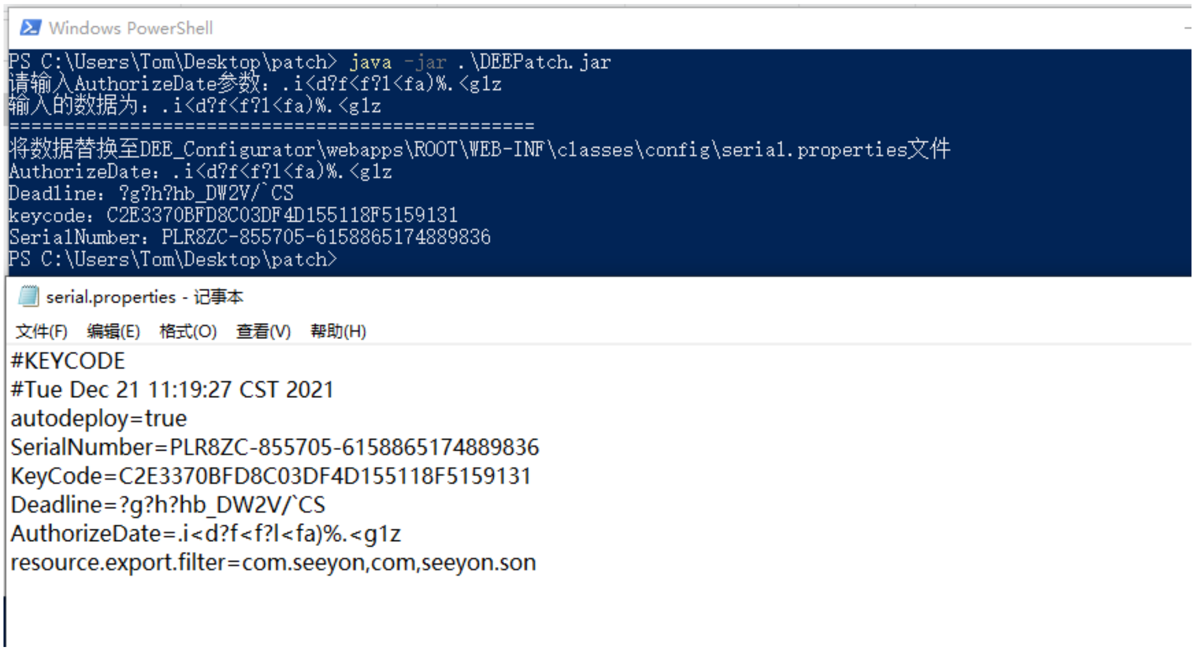
- 之后根据认证代码，其中的`dedeline`指的应该是授权天数，然后关键代码如下

```
if (keycode == null || !AuthorizeUtil.getK(encode_authorizeDate,
encode_deadline).equals(keycode) || !AuthorizeUtil.getS(keycode,
"250").equals(SerialNumber)) {
                errorMsg = errorMsg + "识别码与授权码不匹配！";
            }
```

> 只需要我们的`SerialNumber`和`KeyCode`正确，就可以完成认证，代码逻辑还是比较简单的，所有直接上代码。用于加解密的代码可以直接用他的，都不需要逆向。

```
package com.seeyon.dee.configurator.interceptor;

import com.seeyon.v3x.dee.common.base.util.AuthorizeUtil;
import com.seeyon.v3x.dee.common.base.util.Encipher;

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String str2=null;
        System.out.print("请输入AuthorizeDate参数：");
        if (scanner.hasNextLine()) {
            str2 = scanner.nextLine();
            System.out.println("输入的数据为：" + str2);
        }
        if(str2==null){
            str2="9s(r-[?g>;9.?f(n+u6h";
        }
        scanner.close();
        /*生成授权天数*/
        String Deadline = Encipher.Encode("1000");
```

```java
        /*生成keycode*/
        String keycode = AuthorizeUtil.getK(str2, Deadline);

        /*生成注册序列号*/
        String serialNumber = AuthorizeUtil.getS(keycode, "250");
        System.out.println("================================================");
        System.out.println("将数据替换至DEE_Configurator\\webapps\\ROOT\\WEB-INF\\classes\\config\\serial.properties文件");
        System.out.println("AuthorizeDate: "+str2);
        System.out.println("Deadline: "+Deadline);
        System.out.println("keycode: "+keycode);
        System.out.println("SerialNumber: "+serialNumber);

    }
}
```

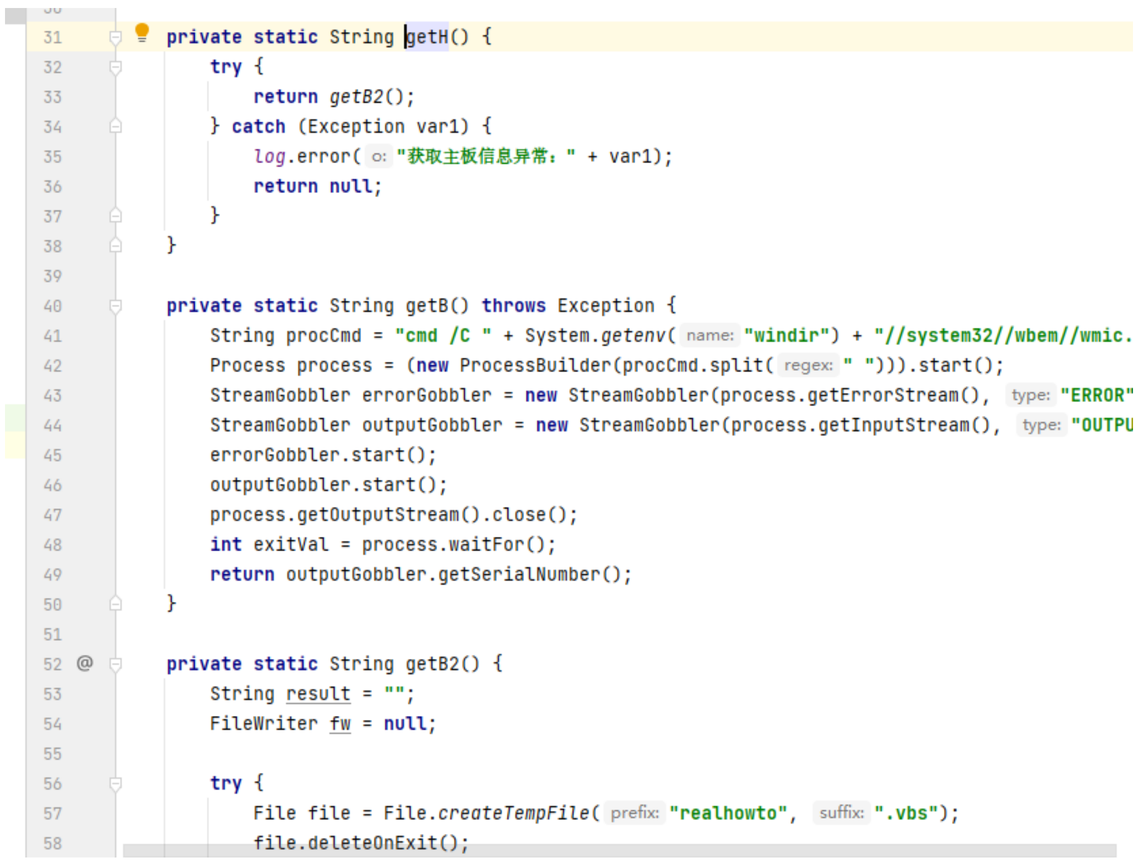


```
PS C:\Users\Tom\Desktop\patch> java -jar .\DEEPatch.jar
请输入AuthorizeDate参数: .i<d?f<f?l<fa)%.<g1z
输入的数据为: .i<d?f<f?l<fa)%.<g1z
================================================
将数据替换至DEE_Configurator\webapps\ROOT\WEB-INF\classes\config\serial.properties文件
AuthorizeDate: .i<d?f<f?l<fa)%.<g1z
Deadline: ?g?h?hb_DW2V/`CS
keycode: C2E3370BFD8C03DF4D155118F5159131
SerialNumber: PLR8ZC-855705-6158865174889836
PS C:\Users\Tom\Desktop\patch>
```

serial.properties - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
#KEYCODE
#Tue Dec 21 11:19:27 CST 2021
autodeploy=true
SerialNumber=PLR8ZC-855705-6158865174889836
KeyCode=C2E3370BFD8C03DF4D155118F5159131
Deadline=?g?h?hb_DW2V/`CS
AuthorizeDate=.i<d?f<f?l<fa)%.<g1z
resource.export.filter=com.seeyon,com,seeyon.son
```

- 注意点：在生成`keycode`的`AuthorizeUtil.getK(str2, Deadline);`方法中有一个`getH()`方法获取主板信息，因此需要将破解包放到程序安装的服务器进行运行。



```java
    private static String getH() {
        try {
            return getB2();
        } catch (Exception var1) {
            log.error( o: "获取主板信息异常：" + var1);
            return null;
        }
    }

    private static String getB() throws Exception {
        String procCmd = "cmd /C " + System.getenv( name: "windir") + "//system32//wbem//wmic.
        Process process = (new ProcessBuilder(procCmd.split( regex: " "))).start();
        StreamGobbler errorGobbler = new StreamGobbler(process.getErrorStream(),  type: "ERROR"
        StreamGobbler outputGobbler = new StreamGobbler(process.getInputStream(),  type: "OUTPU
        errorGobbler.start();
        outputGobbler.start();
        process.getOutputStream().close();
        int exitVal = process.waitFor();
        return outputGobbler.getSerialNumber();
    }

    private static String getB2() {
        String result = "";
        FileWriter fw = null;

        try {
            File file = File.createTempFile( prefix: "realhowto",  suffix: ".vbs");
            file.deleteOnExit();
```

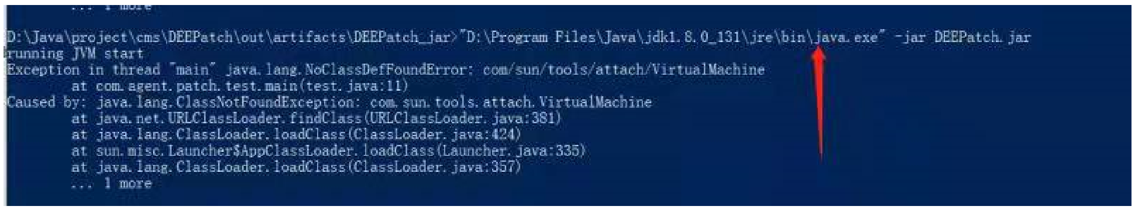
`java agent`破解

说明

> 这个点比较有意思，通过`java agent`动态修改内存中的过滤器源代码，然后实现破解。主要步骤，首先通过`VirtualMachine`获取系统内的全部虚拟机，然后找到`org.apache.catalina.startup.Bootstrap`虚拟机，之后将`java agent`注入到该虚拟机，然后获取虚拟机加载的全部`Class`类，找到用于授权验证的`com.seeyon.dee.configurator.interceptor.SerialInterceptor`类，之后通过`javaassit`修改类字节码，然后重新加载类。这个和之前冰蝎源码中注入内存马的方式相同。

注意

1. 关于`VirtualMachine`的使用，这个类存在于`tools.jar`包中，`tools.jar`是`jdk`中自带的，我们需要将`tools.jar`复制到`jre`的路径下面，另外还有就是需要在`jre\bin`目录下找到一个叫`attach.dll`的文件，复制到`jdk\bin`目录，不然会一直提示类`com/sun/tools/attach/VirtualMachine`缺失。



```
D:\Java\project\cms\DEEPatch\out\artifacts\DEEPatch_jar>"D:\Program Files\Java\jdk1.8.0_131\jre\bin\java.exe" -jar DEEPatch.jar
running JVM start
Exception in thread "main" java.lang.NoClassDefFoundError: com/sun/tools/attach/VirtualMachine
        at com.agent.patch.test.main(test.java:11)
Caused by: java.lang.ClassNotFoundException: com.sun.tools.attach.VirtualMachine
        at java.net.URLClassLoader.findClass(URLClassLoader.java:381)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:335)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
        ... 1 more
```

2. `agent`包的编译。

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <parent>
        <artifactId>javaagent</artifactId>
        <groupId>org.example</groupId>
        <version>1.0-SNAPSHOT</version>
    </parent>
    <modelVersion>4.0.0</modelVersion>

    <artifactId>agentMain</artifactId>
    <build>
        <plugins>
            <plugin>
                <artifactId>maven-assembly-plugin</artifactId>
                <configuration>
                    <archive>
                        <!--自动添加META-INF/MANIFEST.MF -->
                        <manifest>
                            <addClasspath>true</addClasspath>
                        </manifest>
                        <manifestEntries>
                            <Agent-Class>AgentMain</Agent-Class>
                            <Can-Redefine-Classes>true</Can-Redefine-Classes>
                            <Can-Retransform-Classes>true</Can-Retransform-
Classes>
                        </manifestEntries>
                    </archive>
                    <descriptorRefs>
                        <descriptorRef>jar-with-dependencies</descriptorRef>
                    </descriptorRefs>
                </configuration>
            </plugin>
        </plugins>
    </build>

    <properties>
        <maven.compiler.source>8</maven.compiler.source>
        <maven.compiler.target>8</maven.compiler.target>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.javassist</groupId>
            <artifactId>javassist</artifactId>
            <version>3.27.0-GA</version>
        </dependency>
    </dependencies>
</project>
```

> mvn clean package -DskipTests

- `agent注入代码`

```
import javassist.ClassClassPath;
import javassist.ClassPool;
import javassist.CtClass;
import javassist.CtMethod;

import java.io.*;
import java.lang.instrument.ClassDefinition;
import java.lang.instrument.ClassFileTransformer;
import java.lang.instrument.IllegalClassFormatException;
import java.lang.instrument.Instrumentation;
```

```java
import java.nio.charset.StandardCharsets;
import java.security.ProtectionDomain;

public class AgentMain {
    public static void agentmain(String agentArgs, Instrumentation
instrumentation) {
        Class[] cLasses = instrumentation.getAllLoadedClasses(); //当前jvm加载的所
有类
        try {
            for (Class classs:cLasses){
                /*find zhe target class*/
                String
targetClassName="com.seeyon.dee.configurator.interceptor.SerialInterceptor";
                if(classs.getName().contains(targetClassName)){
                    String code="return invocation.invoke();";
                    ClassPool cPool = ClassPool.getDefault();
                    ClassClassPath classClassPath = new ClassClassPath(classs);
                    cPool.insertClassPath(classClassPath);
                    CtClass ctClass = cPool.get(targetClassName);
                    CtMethod doIntercept =
ctClass.getDeclaredMethod("doIntercept", new CtClass[]
{cPool.get("com.opensymphony.xwork2.ActionInvocation")});
                    doIntercept.insertBefore(code);
                    ctClass.detach();
                    byte[] bytes = ctClass.toBytecode();
                    File file = new File("C:\\httpServelt.class");
                    FileOutputStream fileOutputStream = new
FileOutputStream(file);
                    BufferedOutputStream bufferedOutputStream = new
BufferedOutputStream(fileOutputStream);
                    bufferedOutputStream.write(bytes);
                    bufferedOutputStream.close();
                    fileOutputStream.close();
                    instrumentation.redefineClasses(new ClassDefinition[]{new
ClassDefinition(classs, bytes)});
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        instrumentation.addTransformer(new DefineTransformer(), true);
    }

    static class DefineTransformer implements ClassFileTransformer {

        @Override
        public byte[] transform(ClassLoader loader, String className, Class<?>
classBeingRedefined, ProtectionDomain protectionDomain, byte[] classfileBuffer)
throws IllegalClassFormatException {
            System.out.println("premain load Class:" + className);
            return classfileBuffer;
        }
    }
}
```

> 实现逻辑非常粗暴，找到`org.apache.catalina.startup.Bootstrap`这个`JVM`加载的全部类中的`com.seeyon.dee.configurator.interceptor.SerialInterceptor`类，然后通过`javaassit`动态修改类字节码然后重新载入类。

- 主程序编写

```java
package com.agent.patch;

import com.sun.tools.attach.*;

import java.io.IOException;
import java.util.List;

public class test {
    public static void main(String[] args) {
        System.out.println("running JVM start ");
        List<VirtualMachineDescriptor> list = VirtualMachine.list();
        for (VirtualMachineDescriptor vmd : list) {
            //如果虚拟机的名称为 xxx 则 该虚拟机为目标虚拟机，获取该虚拟机的 pid
            //然后加载 agent.jar 发送给该虚拟机
            if
(vmd.displayName().contains("org.apache.catalina.startup.Bootstrap")) {
                System.out.println(vmd.displayName());
                System.out.println(vmd.id());
                VirtualMachine virtualMachine = null;
                try {
                    System.setProperty("jdk.attach.allowAttachSelf", "true");
                    virtualMachine = VirtualMachine.attach(vmd.id());
                    String path = System.getProperty("user.dir");
                    virtualMachine.loadAgent(path+"\\agentMain-1.0-
SNAPSHOT.jar");
                    virtualMachine.detach();
                } catch (AgentLoadException e) {
                    e.printStackTrace();
                } catch (AgentInitializationException e) {
                    e.printStackTrace();
                } catch (IOException e) {
                    e.printStackTrace();
                } catch (AttachNotSupportedException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```