- modyfikacja dzialania konstruktora, metody, ingerencja w mytecode z zewnarz, dodanie swojej funkcjonalnosci na przyklad

- w 3 krokach, klasa co odpowiednia static metode wywoluje, rejestracja klasy w manifescie i odpalenie ze zbudowanym jar agenta aplikacji klienta

---

import java.lang.instrument.Instrumentation;

public class Agent {

public static void premain(String args, Instrumentation instrumentation){

ClassLogger transformer = new ClassLogger();

instrumentation.addTransformer(transformer);

}

}

--

- dodatkowo potrzebny transformer

--

public class ClassLogger implements ClassFileTransformer {

@Override

public byte[] transform(ClassLoader loader,

String className,

Class<?> classBeingRedefined,

ProtectionDomain protectionDomain,

byte[] classfileBuffer) throws IllegalClassFormatException {

try {

Path path = Paths.get("classes/" + className + ".class");

Files.write(path, classfileBuffer);

} catch (Throwable ignored) { // ignored, don’t do this at home kids

} finally { return classfileBuffer; }

}

}

--

Mozna konkretna klase z puli, np. Za pomoca javaassist

|  |
| --- |
| final ClassPool classPool = ClassPool.getDefault(); |

|  |  |
| --- | --- |
| 13 | final CtClass clazz = |

|  |  |
| --- | --- |
| 14 | classPool.get("sun.net.www.protocol.http.HttpURLConnection"); |

|  |  |
| --- | --- |
| 15 |  |

|  |  |
| --- | --- |
| 16 | for (final CtConstructor constructor: clazz.getConstructors()) { |

|  |  |
| --- | --- |
| 17 | constructor.insertAfter("System.out.println(this.getURL());"); |

|  |  |
| --- | --- |
| 18 | } |

|  |  |
| --- | --- |
| 19 |  |

|  |  |
| --- | --- |
| 20 | byte[] byteCode = clazz.toBytecode(); |

|  |  |
| --- | --- |
| 21 | clazz.detach(); |

|  |  |
| --- | --- |
| 22 |  |

|  |  |
| --- | --- |
| 23 | return byteCode; |

--

Trza zbudowac jar, tu gradle przyklad :

<https://zeroturnaround.com/rebellabs/how-to-inspect-classes-in-your-jvm/>

- teraz odpalamy apke z agentem

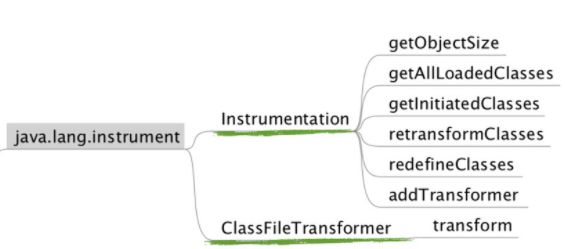
java -javaagent:/path/to/agent.jar -jar myapp.jar

--

Tu przyklad wywolania metody, przed po – obliczanie czasu dzialania konkretnej metody w javassist i ASM

<http://www.tomsquest.com/blog/2014/01/intro-java-agent-and-bytecode-manipulation/>

--



--

Mozna refleksji uzyc

https://javabeat.net/introduction-to-java-agents/

<https://www.javacodegeeks.com/2015/09/java-agents.html>

<https://chimpler.wordpress.com/2013/11/05/implementing-a-java-agent-to-instrument-code/>

----

Javaassist

<http://jboss-javassist.github.io/javassist/tutorial/tutorial.html>

<https://jboss-javassist.github.io/javassist/tutorial/tutorial2.html>

mvn clean compile assembly:single

java -javaagent:C:\OTHER\_REPO\javaagent\target\javaagent-1.0-SNAPSHOT-jar-with-dependencies.jar="appId=com.citi.vaultencryptor,serviceURL=http://localhost:8080/,testConfigFile=C:\agentConfigs\appId.1.json,testId=1,connectionsConfigFile=C:\agentConfigs\abc.json,metricsConfigFile=C:\agentConfigs\metrics.json" -jar target\javaagentsample-1.0-SNAPSHOT-jar-with-dependencies.jar

java -javaagent:C:\OTHER\_REPO\javaagent\target\javaagent-1.0-SNAPSHOT-jar-with-dependencies.jar="appId=com.citi.vaultencryptor,serviceURL=http://localhost:8080/,testConfigFile=C:\agentConfigs\appId.1.json,testId=1,connectionsConfigFile=C:\agentConfigs\abc.json,metricsConfigFile=C:\agentConfigs\metrics.json" -cp "target\javaagentsample-1.0-SNAPSHOT-jar-with-dependencies.jar;spring-web-4.3.13.RELEASE.jar" com.citi.SampleClass

Boot libs

<dependency>

<groupId>javax.validation</groupId>

<artifactId>validation-api</artifactId>

<version>1.0.0.GA</version>

</dependency>

<dependency>

<groupId>org.hibernate</groupId>

<artifactId>hibernate-validator</artifactId>

<version>4.3.1.Final</version>

</dependency>

<!--<dependency>-->

<!--<groupId>javax.servlet</groupId>-->

<!--<artifactId>javax.servlet-api</artifactId>-->

<!--<version>3.0.1</version>-->

<!--</dependency>-->

<!--<dependency>-->

<!--<groupId>org.apache.geronimo.specs</groupId>-->

<!--<artifactId>geronimo-servlet\_3.0\_spec</artifactId>-->

<!--<version>1.0</version>-->

<!--</dependency>-->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>javax.servlet-api</artifactId>

<version>4.0.0</version>

</dependency>

<dependency>

<groupId>org.apache.tomcat</groupId>

<artifactId>tomcat-servlet-api</artifactId>

<version>8.5.23</version>

</dependency>