Security Vulnerability Notice

SE-2012-01-ORACLE-8

[Security vulnerabilities in Java SE, Issues 51 and 52]

DISLAIMER

INFORMATION PROVIDED IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW NEITHER SECURITY EXPLORATIONS, ITS LICENSORS OR AFFILIATES, NOR THE COPYRIGHT HOLDERS MAKE ANY REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR THAT THE INFORMATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS, OR OTHER RIGHTS. THERE IS NO WARRANTY BY SECURITY EXPLORATIONS OR BY ANY OTHER PARTY THAT THE INFORMATION CONTAINED IN THE THIS DOCUMENT WILL MEET YOUR REQUIREMENTS OR THAT IT WILL BE ERROR-FREE.

YOU ASSUME ALL RESPONSIBILITY AND RISK FOR THE SELECTION AND USE OF THE INFORMATION TO ACHIEVE YOUR INTENDED RESULTS AND FOR THE INSTALLATION, USE, AND RESULTS OBTAINED FROM IT.

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL SECURITY EXPLORATIONS, ITS EMPLOYEES OR LICENSORS OR AFFILIATES BE LIABLE FOR ANY LOST PROFITS, REVENUE, SALES, DATA, OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, PROPERTY DAMAGE, PERSONAL INJURY, INTERRUPTION OF BUSINESS, LOSS OF BUSINESS INFORMATION, OR FOR ANY SPECIAL, DIRECT, INDIRECT, INCIDENTAL, ECONOMIC, COVER, PUNITIVE, SPECIAL, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND WHETHER ARISING UNDER CONTRACT, TORT, NEGLIGENCE, OR OTHER THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE INFORMATION CONTAINED IN THIS DOCUMENT, EVEN IF SECURITY EXPLORATIONS OR ITS LICENSORS OR AFFILIATES ARE ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS.

Security Explorations discovered two security vulnerabilities in Java SE Platform, Standard Edition. They are similar to the weaknesses discussed in our previous reports (problems with Class Loader's access and Reflection API). A table below, presents their technical summary:

ISSUE #	TECHNICAL DETAILS	
51	origin	`com.sun.org.apache.xalan.internal.xsltc.trax.TemplatesImpl`
	cause	the possibility to define user provided classes in a privileged (no package access in `loadClass` method) class loader (`TransletClassLoader`)
	impact	arbitrary access to restricted classes
	type	partial security bypass vulnerability
52	origin	`com.sun.jmx.mbeanserver.Introspector`
	cause	insecure use of `invoke` method of `java.lang.reflect.Method` class
	impact	arbitrary invocation of no-argument methods on user provided objects / classes
	type	partial security bypass vulnerability

Issue 51 relies on a definition of a custom instance of `TemplatesImpl` subclass that is further instantiated with the use of serialization. Due to the complexity of the exploit implementation (`BlackBox` class deserializes `TemplatesImpl` instance from the `ObjectInputStream` defined in an array of bytes), we provide the source code for the translet class that is defined by the instantiated object (`Helper.java` file from `translet` directory).

The whole exploitation process takes place inside the constructor of the `Helper` class, which is a user provided subclass of `AbstractTranslet` class. The bytecodes for the body of this class are initialized at the time of deserialization of `TemplatesImpl` instance (`_bytecodes` field). As an exploitation vector we again rely on `DefiningClassLoader` class from `sun.org.mozilla.javascript.internal` package.

Issue 52 relies on the possibility to call no-argument methods on arbitrary objects or classes. For the purpose of our Proof of Concept code we invoke `getDeclaredMethods` of `java.lang.Class` class to get access to methods of restricted classes. This is accomplished with the use of the following code sequence:

```
Introspector.elementFromComplex((Object)clazz,"declaredMethods")
```

Issues 51 and 52, when combined together can be used to successfully achieve a complete JVM sandbox bypass in a target system. It might be possible that Issue 52 could be used alone to achieve a complete sandbox bypass. That however requires more thorough investigation.

Attached to this report, there is a Proof of Concept codes that illustrate the abovementioned impact of both vulnerabilities. It has been successfully tested in the environment of Java SE 7 Update 11 (JRE version 1.7.0_11-b21).

About Security Explorations

Security Explorations (http://www.security-explorations.com) is a security start-up company from Poland, providing various services in the area of security and vulnerability research. The company came to life in a result of a true passion of its founder for breaking security of things and analyzing software for security defects. Adam Gowdiak is the company's founder and its CEO. Adam is an experienced Java Virtual Machine hacker, with over 50 security issues uncovered in the Java technology over the recent years. He is also the hacking contest co-winner and the man who has put Microsoft Windows to its knees (vide MS03-026). He was also the first one to present successful and widespread attack against mobile Java platform in 2004.