Security Vulnerability Notice

SE-2012-01-ORACLE-3

[Security vulnerabilities in Java SE, Issues 23-26]

DISLAIMER

Security Explorations discovered additional security issues in Java Platform, Standard Edition. This in particular includes one issue in the latest version of Java SE 7 as well as two attack vectors for Java SE 6 environment and previously reported issues. All issues are similar to those presented in our previous reports (problems with Reflection API). A table below, presents their technical summary:

ISSUE #	TECHNICAL DETAILS	
23	origin	`javax.management.modelmbean.DescriptorSupport` class
	cause	insecure use of `getConstructor` and `newInstance` methods of `java.lang.Class` class
	impact	creation of restricted public classes (scope limited to the classes with the instance initialization method denoting one `java.lang.String` argument)
	Type	exploitation vector (requires a security bypass precondition)
24	origin	`javax.media.jai.OperationRegistry` class
	cause	insecure use of `invoke` method of `java.lang.reflect.Method` class
	impact	arbitrary invocation of methods with user provided arguments
	Type	partial security bypass vulnerability
25	origin	`javax.swing.text.DefaultFormatter` class
	cause	insecure use of `getConstructor` and `newInstance` methods of `java.lang.Class` class
	impact	creation of restricted public classes (scope limited to the classes with the instance initialization method denoting one `java.lang.String` argument)
	type	exploitation vector (requires a security bypass precondition)
26	origin	`java.lang.invoke.MethodHandles.Lookup` class
	cause	access to package scoped classes via a specially chosen system class as `lookupClass` value
	impact	obtaining access to inner classes to which a caller of the `Lookup` object has no access
	type	complete security bypass vulnerability

Below, we provide additional comments with respect to the issues presented in the table above:

- Issue 23 is presented as an exploit vector for Issue 8 (privileged `OrderClassLoaders` as Thread's context classloader) and Java SE 6 environment. This exploit vector allows for the creation of certain objects of classes from `sun.security.action` package. This in particular includes such classes as `GetPropertyAction` or `OpenFileInputStreamAction`. Once created, the objects of these classes can be provided as an input to the `doPrivilegedWithCombiner` method call of `java.security.AccessController` class. In a result, arbitrary read access to system properties or user files could be obtained.
- Issue 24 was verified in the environment of a fully patched MacOS X Snow Leopard system only. It's Java VM environment contains additional classes beyond those distributed as part of a standard Java SE software available from Oracle. This in particular includes JAI classes. One of them (`javax.media.jai.OperationRegistry` class) contains a security vulnerability in the way Reflection API is used. As a result, it is possible to call methods of arbitrary classes and obtain references to class objects from restricted packages. This can be

achieved by the means of a proper `forName` method invocation of `java.lang.Class` class.

- Issue 25 is presented as an exploit vector for Issue 24 (and similar) and Java SE 6 environment. Similarly to Issue 23, this exploit vector also allows to obtain arbitrary read access to system properties or user files. Proof of Concept code for Issues 24 and 25 was successfully tested in a MacOS environment only.
- Issue 26 can be used to achieve a complete JVM security bypass. It allows for the creation of instances of non-public classes such as those with access rights limited to a given package only. In our Proof of Concept code, we make use of the `Lookup` object based on a `javax.swing.JOptionPane` class to obtain a method handle to the constructor of `javax.swing.JOptionPane$ModalPrivilegedAction` inner class. This constructor is later used to successfully create instances of the aforementioned class and to obtain references to restricted `java.lang.reflect.Method` objects. The exploitation scenario proceeds as following:
  - A partially initialized instance of a Thread object is created that overloads `getContextClassLoader` method in such a way, so that it always returns `null`.
  - A privileged (`override` field set to true) method reference to the private `start0` method of `java.lang.Thread` class is obtained. This method is later called in order to successfully start the partially initialized Thread object.
  - The `null` value set as Thread's context class loader is used to obtain references to restricted classes and `sun.awt.SunToolkit` class in particular. The loading occurs in the context of a started thread (the value of its `contextclassloader` is `null`).
  - A privileged method reference to the private `privateGetPublicMethods` method of `java.lang.Class` class is obtained. This method is later called in order to obtain a list of public methods declared by a given restricted class.
  - The exploitation scenario proceeds further in a similar way to the one presented in our first report (SE-2012-01-ORACLE).

Attached to this report, there are several Proof of Concept code that illustrates all reported vulnerabilities. They have been successfully tested in a Windows (Issues 23 and 26), and Mac OS (Issues 24 and 25) environments and with the latest versions of Java SE 6 (Issues 23-25) and 7 (Issue 27).

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.