Security Vulnerability Notice

SE-2012-01-ORACLE-2

[Security vulnerabilities in Java SE, Issues 20-21]

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 more security issues in the latest version of Java Platform, Standard Edition. They are similar to the issues presented in the previous report (problems with Reflection API). A table below, presents their technical summary:

ISSUE #	TECHNICAL DETAILS	
20	origin	`com.sun.beans.decoder.MethodElementHandler` 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
21	origin	`java.lang.invoke.MethodHandles` class
	cause	public `Lookup` based on a system class available to any caller
	impact	the ability to obtain `java.lang.invoke.MethodHandles.Lookup` object with a system `lookupClass`, this allows to obtain method handles from restricted classes and to issue calls on them
	type	partial security bypass vulnerability

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

- Issue 20 can be exploited with the use of a specially crafted XML data fed at the `java.beans.XMLDecoder` object's input. In our case, we use the following `object` tag in order to invoke `forName` method of `java.lang.Class` class:

```
<object class=\"java.lang.Class\"  method=\"forName\">
<string>sun.awt.SunToolkit</string></object>
```

- Issue 21 obtains a reference to the public `Lookup` object via a standard API available to any caller (`MethodHandles.publicLookup()` method). The problem with a public `Lookup` object stems from the fact that it is based on a system class, thus a `Lookup` object can access all other system classes (classes from the same classloader namespace) regardless of the package access restrictions.

Issues 20 and 21, when combined together allow for a complete compromise of JVM security sandbox. The exploitation scenario proceeds in a similar way as for the Issues 12 and 13.

Attached to this report, there is a Proof of Concept code that illustrates both reported vulnerabilities. It has been successfully tested in a Windows OS environment and with the following versions of Java SE:

- JRE/JDK 7u2 (version 1.7.0_02-b13)
- JRE/JDK 7u3 (version 1.7.0_03-b05)

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.