Security Vulnerability Notice

SE-2012-01-ORACLE-5

[Security vulnerabilities in Java SE, Issue 32]

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 additional security issue in Java SE Platform, Standard Edition. It is similar to the weaknesses discussed in our previous reports (problems with Reflection API). A table below, presents its technical summary:

ISSUE #	TECHNICAL DETAILS	
32	origin	`java.lang.invoke.MethodHandle`
	cause	the possibility to call `invokeExact` from a system wrapper method
	impact	bypass of security checks based on the immediate caller
	type	partial security bypass vulnerability

The new weakness has its origin in `java.lang.invoke.MethodHandle` class. It allows to invoke arbitrary (both static and virtual) methods with an immediate caller originating in a null class loader namespace. The problem stems from the fact that `invokeWithArguments` method of `MethodHandle` class can be used as a wrapper method for the actual `invokeExact` method invocation. Such a wrapper call leads to the additional stack frame of a system class being asserted into the call stack, which allows to bypass security checks based on the immediate caller of a given security sensitive method. Examples, of such methods include Reflection API based ones, but also `getUnsafe` of `sun.misc.Unsafe` class.

Issue 32 could be potentially used alone to achieve a complete JVM sandbox bypass. Such a scenario might be possible if Issue 32 is used for the instrumentation of `invoke` method of `java.lang.reflect.Method` class. This however requires more thorough investigation.

Issue 32 was tested in the environment of a recently released Java SE 7 Update 7. We verified that when combined with one of some of still unpatched security vulnerabilities (Issue 1-7), it can be successfully used to achieve a complete JVM sandbox bypass in a target system.

Attached to this report, there is a Proof of Concept codes that illustrates this. It has been successfully tested in a Windows environment and with the latest versions of Java SE 7 (JRE version 1.7.0_07-b10).

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.