Security Vulnerability Notice

SE-2012-01-ORACLE-FIXED[1]

[Security vulnerabilities in Java SE, Issues XX and YY]

[1] The vulnerabilities described in this report were fixed by Oracle on Feb 19, 2013. They were originally found in Feb 2012 and the decision was made to wait with their reporting till the next Java SE CPU cycle release. As a result of a successful discovery by 3rd party researchers, these vulnerabilities were never reported to Oracle by Security Explorations.

DISCLAIMER

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. A table below, presents their technical summary:

ISSUE #	TECHNICAL DETAILS	
XX	Origin	`java.lang.invoke.MethodHandleProxies`
	Cause	Insufficient checks for MethodHandle object implementing given proxy class functionality
	impact	the possibility to implement and call MethodHandle proxy objects for arbitrary interfaces
	Type	partial security bypass vulnerability
YY	Origin	`java.security.AccessController`
	Cause	`doPrivileged` method handle is bound to the privileged class
	impact	the possibility to call `doPrivileged` method from a trusted caller frame
	Type	partial security bypass vulnerability

Issue XX stems from the fact that it is possible to call an arbitrary, user provided MethodHandle object through a target method handle of a fixed type. This can be accomplished with the use of a specially crafted method handle instance which inserts additional arguments, before calling the original method handle object. The type of the new method handle drops the types for the inserted (bound) parameters from the original target type, since the new method handle will no longer require those arguments to be supplied by its callers. In our case, we convert a MethodHandle object of `(SecurityManager)void` type to the `()void` type by creating a new MethodHandle object that binds the `SecurityManager` argument to the NULL value. This is accomplished by the means of `insertArguments` method of `java.lang.reflect.invoke.MethodHandles` class. The idea is to dispatch a call to `setSecurityManager` method of `java.lang.System` class with the use of a MethodHandle of which type corresponds to `run()` method of `java.security.PrivilegedAction` interface.

Issue YY relies on the possibility to call `doPrivileged` method of `java.security.AccessController` class with a privileged class set as a caller. In some of our Proof of Concept codes reported to Oracle in 2012, we relied on a possibility to invoke this method through the wrapper `doPrivilegedWithCombiner` call. At that time, we treated this issue more as a feature than a security bug. However, due to the fact that Oracle has addressed the abovementioned behavior and made it impossible to call a custom `PrivilegedAction` object via the wrapper `doPrivilegedWithCombiner` method call, we now treat it as a bug. A successful call to `doPrivileged` method can be now accomplished with the use of a new Reflection API and a MethodHandle object corresponding to the `doPrivileged` method. Although this MethodHandle object is bound to the non-null Class Loader namespace, the binding is done through a fully privileged trampoline class. This is sufficient for the target call to succeed.

Issues XX and YY, when combined together can be used to successfully achieve a complete JVM sandbox bypass in a target system.

Attached to this report, there is a Proof of Concept codes that illustrate the impact of both vulnerabilities. It has been successfully tested in the environment of Java SE 7 Update 13 (JRE version 1.7.0_13-b20).

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.