# Security Vulnerability Notice

SE-2012-01-ORACLE-12

[Security vulnerabilities in Java SE, Issue 61]

## 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 a security vulnerability in Java SE Platform, Standard Edition. It stems from insecure use of a certain security sensitive Reflection API method. A table below, presents a technical summary of the vulnerability:

| ISSUE # | TECHNICAL DETAILS | |
|---|---|---|
| 61 | origin | `sun.tracing.ProviderSkeleton` class |
| | cause | insecure use of `invoke` method of `java.lang.reflect.Method` class |
| | impact | arbitrary invocation of static methods with user provided arguments |
| | type | complete security bypass vulnerability |

Issue 61 is very similar to Issues 1-7 reported to Oracle in Apr 2012. However, contrary to the past issues, the new weakness is located in a class residing in a restricted package (`sun.tracing.*`). The vulnerable class can be reached via the code path originating in the allowed packages space. It can be also accessed by the means of a public `java.lang.reflect.InvocationHandler` interface.

In our Proof of Concept code, we abuse a vulnerable `invoke` method implementation of a `sun.tracing.ProviderSkeleton` class to successfully:

- issue calls to `forName` method of `java.lang.Class` class in order to obtain references to restricted class (from `sun` package),
- create an instance of `java.lang.invoke.MethodHandles.Lookup` object with a system class object in the `lookupClass` field.

The above is sufficient to achieve a complete compromise of JVM security sandbox. The exploitation scenario relies on a `DefiningClassLoader` class described in our previous vulnerability reports (Issue 32).

It should be also mentioned that `sun.tracing.dtrace.DTraceProvider` is a class implementing similar functionality to the vulnerable `ProviderSkeleton` class. A more thorough investigation is required regarding how it could be triggered though (whether setting "`com.sun.tracing.dtrace`" property to `true` is sufficient, etc.) as Solaris 11.1 returns `NullProviderFactory` by default.

Attached to this report, there is a Proof of Concept code that illustrates the impact of the vulnerability described above. It has been successfully tested in the environment of Java SE 7 Update 21 (JRE version 1.7.0_21-b11) with Internet Explorer 9 and Mozilla Firefox 20.0.1 web browsers.

## 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.