Security Vulnerability Notice

SE-2012-01-IBM-4

[Security vulnerabilities in Java SE, Issue 67#2]

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 that Issue 67 [1] reported to IBM in May 2013 was improperly fixed. According to the company, the vulnerability was addressed in a release of IBM Java from Jul 2013. Below, technical details of the flawed fix implementation are provided.

Issue 67 had its origin in an insecure use of `invoke` method of `java.lang.reflect.Method` class, which was called inside `AccessController`'s `doPrivileged` block. It could be successfully exploited to call `setSecurityManager` method of `java.lang.System` class. As a result, a complete Java security sandbox escape could be gained.

IBM patch for Issue 67 implemented the change to the code of `com.ibm.CORBA.iiop.ClientDelegate` class as illustrated on Fig. 1. Prior to the patch, the `ClientDelegate` class itself implemented `java.lang.reflect.InvocationHandler` interface functionality and the vulnerable `invoke` call was directly exposed to all other classes. As a result of the patch, the `InvocationHandler` functionality along with the insecure call were simply moved into the private (inner) class.

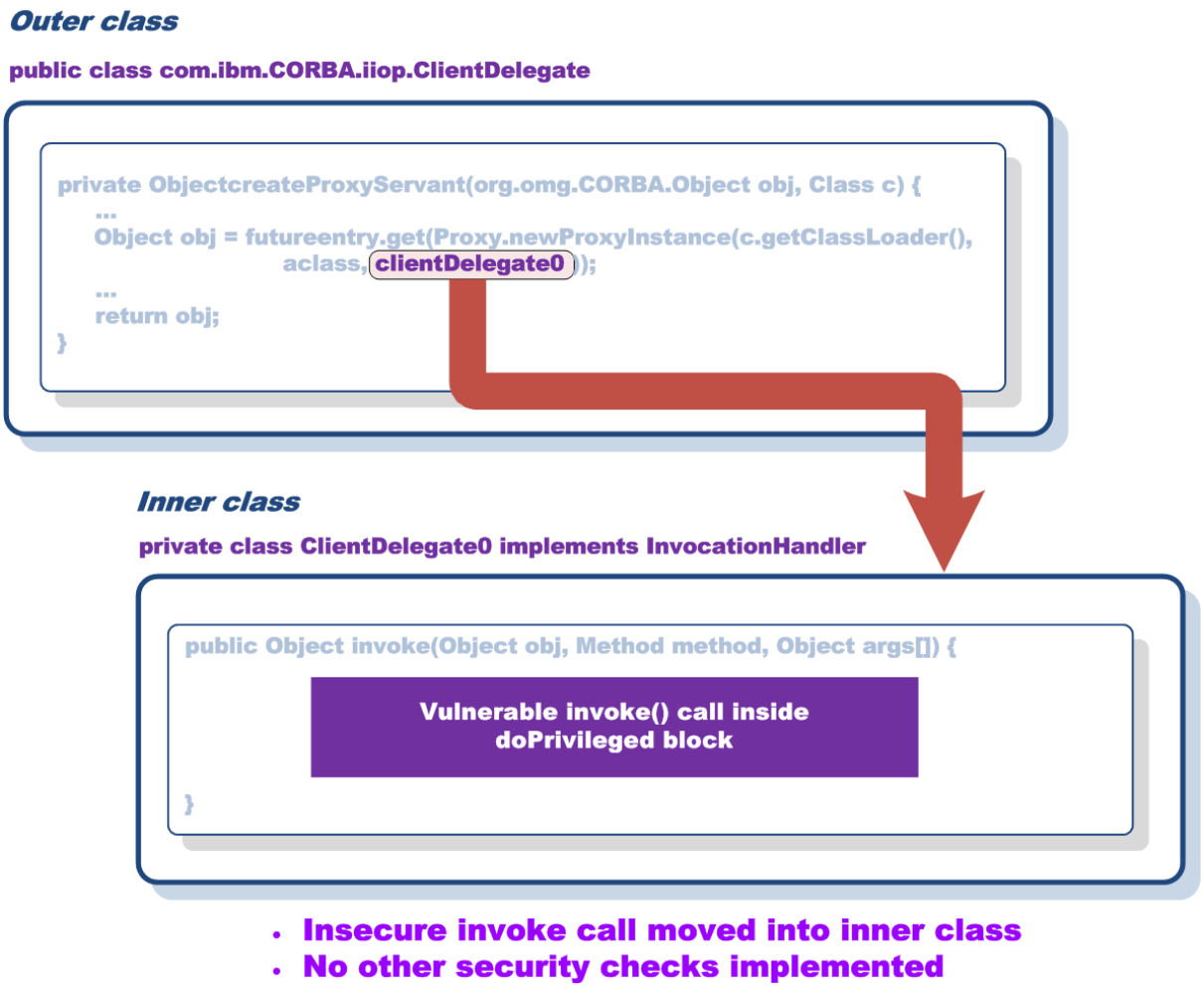

Fig. 1 Illustration of a fix for Issue 67.

This functionality could be however still accessed through the dedicated Proxy object instance implementing `java.lang.reflect.InvocationHandler` interface. The actual root cause of the issue hasn't been addressed at all. The `invoke` method of

`java.lang.reflect.Method` class was still insecurely used inside `AccessController`'s `doPrivileged` block. There were no security checks introduced anywhere in the code. The patch relied solely on the idea that hiding the vulnerable method deep in the code and behind a Proxy class would be sufficient to address the issue.

Breaking IBM patch for Issue 67 requires only several minor changes to our original Proof of Concept code published in Jul 2013. It is simply a matter of gaining access to the Proxy object instance implementing the vulnerable method invocation and its invocation handler in particular. The required changes to our original POC code are illustrated on Fig. 2.

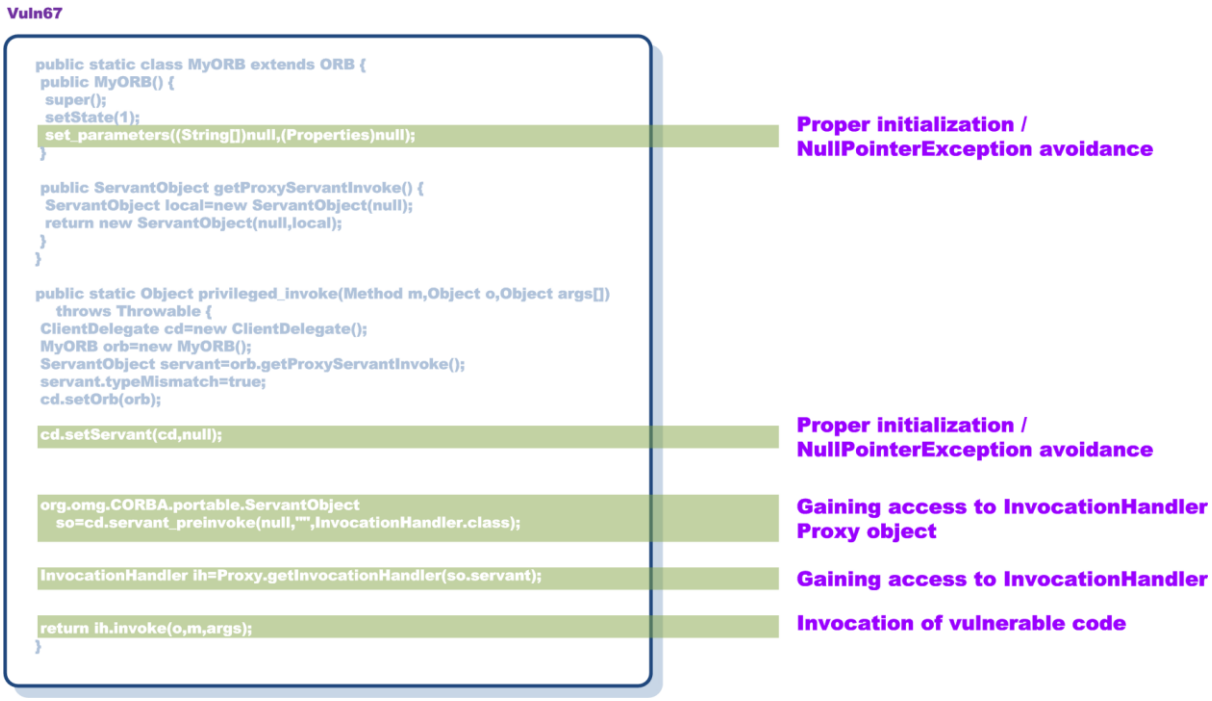

Fig. 2 The changes required to the original POC allowing for Issue 67 fix bypass.

We implemented a Proof of Concept code that illustrates the impact of the broken fix described above. It has been successfully tested in a 32-bit Linux OS environment and with the following versions of IBM SDK:

- IBM SDK, Java Technology Edition, Version 7.1 for Linux (32-bit x86) released on 2016-01-26 (build pxi3270_27sr3fp30-20160112_01(SR3 FP30))
- IBM SDK, Java Technology Edition, Version 8.0 for Linux (32-bit x86) released on 2016-01-26 (build pxi3280sr2fp10-20160108_01(SR2 FP10))

We verified that, a complete Java security sandbox escape could be achieved with it.

REFERENCES

[1] SE-2012-01-IBM-2, Issues 62-68
http://www.security-explorations.com/materials/SE-2012-01-IBM-2.pdf

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.