Security Vulnerability Notice

SE-2012-01-ORACLE-9

[Security vulnerabilities in Java SE, Issue 53]

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. The vulnerability can be used to bypass default security settings introduced in Java SE 7 Update 10. More specifically, we have found a way to run unsigned Java code without displaying ANY warning to the user.

According to [1], starting from JDK 7u10 release, a user may control, via the Java Control Panel, the level of security that will be used when running unsigned Java apps in a browser. Apart from being able to completely disable Java content in the browser, the following four security levels are also available for configuration:

- *Low*
  Most unsigned Java apps in the browser will run without prompting unless they request access to a specific old version or to protected resources on the system.
- *Medium*
  Unsigned Java apps in the browser will run without prompting only if the Java version is considered secure. User will be prompted if an unsigned app requests to run on an old version of Java.
- *High*
  User will be prompted before any unsigned Java app runs in the browser. If the JRE is below the security baseline, user will be given an option to update.
- *Very High*
  Unsigned (sandboxed) apps will not run.

Unfortunately, the implementation of the above security levels does not take into account that Java Applets can be instantiated with the use of serialization. This can be accomplished with the use of the following `<applet>` HTML tag:

```
<applet object="BlackBox.ser">
```

The data for the serialized Applet (`BlackBox.ser` file) can be easily created with the use of the following code:

```
BlackBox b=new BlackBox(); // target Applet instance

ByteArrayOutputStream baos=new ByteArrayOutputStream();
ObjectOutputStream oos=new ObjectOutputStream(baos);

oos.writeObject(b);

FileOutputStream fos=new FileOutputStream("BlackBox.ser");
fos.write(baos.toByteArray());
fos.close();
```

We have verified that the above technique can be successfully used to run unsigned (and malicious!) Java applet under Java SE 7 Update 11 and regardless of the security level configured in Java Control Panel. We confirmed that arbitrary Java code could be run without any prompt even if "High" or "Very High" security level was configured in a target system.

Attached to this report, there is a Proof of Concept code that illustrates the abovementioned vulnerability. It has been successfully tested in the environment of Java SE 7 Update 11 (JRE version 1.7.0_11-b21).

REFERENCES

[1] Setting the Security Level of the Java Client http://docs.oracle.com/javase/7/docs/technotes/guides/jweb/client-security.html

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.