A Java 7 sebezhetség javítása

Tegnap este megjelent a hírhedt Java 7 sebezhetség javítása: http://www.oracle.com/technetwork/topics/security/alert-cve-2012-4681-1835715.html

Kis kutakodással a források között találunk egy érdekességet a ClassFinder.java osztály módosításai között:

ClassFinder.java

```
--- a/src/share/classes/com/sun/beans/finder/ClassFinder.java        Tue May 25 15:58:33 2010 -0700
+++ b/src/share/classes/com/sun/beans/finder/ClassFinder.java        Tue Jun 19 21:04:48 2012 +0400
@@ -1,5 +1,5 @@
 /*
- * Copyright (c) 2006, 2008, Oracle and/or its affiliates. All rights reserved.
+ * Copyright (c) 2006, 2012, Oracle and/or its affiliates. All rights reserved.
  * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.
  *
  * This code is free software; you can redistribute it and/or modify it
@@ -23,6 +23,8 @@
  * questions.
  */
 package com.sun.beans.finder;
+
+import static sun.reflect.misc.ReflectUtil.checkPackageAccess;

 /**
  * This is utility class that provides {@code static} methods
@@ -54,6 +56,7 @@ public final class ClassFinder {
      * @see Thread#getContextClassLoader()
      */
     public static Class<?> findClass(String name) throws ClassNotFoundException {
+        checkPackageAccess(name);
         try {
             ClassLoader loader = Thread.currentThread().getContextClassLoader();
             if (loader == null) {
@@ -94,6 +97,7 @@ public final class ClassFinder {
      * @see Class#forName(String,boolean,ClassLoader)
      */
     public static Class<?> findClass(String name, ClassLoader loader) throws ClassNotFoundException {
+        checkPackageAccess(name);
         if (loader != null) {
             try {
                 return Class.forName(name, false, loader);
```

Egyszeren a problémát okozó metódusok elé bekerült egy *checkPackageAccess* hívás a ReflectUtil osztályból, amely leellenrzi, hogy az adott osztály betölthet-e:

ReflectionUtil.java

```
public static void checkPackageAccess(String name) {
    SecurityManager s = System.getSecurityManager();
    if (s != null) {
        String cname = name.replace('/', '.');
        if (cname.startsWith("[")) {
            int b = cname.lastIndexOf('[') + 2;
            if (b > 1 && b < cname.length()) {
                cname = cname.substring(b);
            }
        }
        int i = cname.lastIndexOf('.');
        if (i != -1) {
            s.checkPackageAccess(cname.substring(0, i));
        }
    }
}
```

Amint láthatjuk, a javítás június 19-én került a forrásokba, az elz Java 7 update 6 kiadás augusztus 14-én kerül ki, felmerül a kérdés, hogy egy ismert hiba javítása miért nem került bele ez utóbbi verzióba, amikor elegend id jutott volna a tesztelésére és javítására...

Oracle release policy

Ahogy a Wikipedia oldalon láthatjuk, az Oracle kiadási ciklusa szerint kéthavonta kerül ki egy frissítés, amelyek felváltva biztonsági és feature kiadások:

Release	Release Date	Highlights
Java SE 7[80]	2011-07-28	Initial release
Java SE 7 Update 1[81]	2011-10-18	20 security fixes, other bug fixes
Java SE 7 Update 2[82]	2011-12-12	No security fixes; reliability and performance improvements; support for Solaris 11 and Firefox 5 and later; JavaFX included with Java SE JDK, improvements for web-deployed applications
Java SE 7 Update 3[83]	2012-02-14	14 security fixes[84]
Java SE 7 Update 4[85]	2012-04-26	No security updates; JDK Support for Mac OS X; New JVM (Java HotSpot Virtual Machine, version 23)
Java SE 7 Update 5[86]	2012-06-12	14 security fixes[87]
Java SE 7 Update 6[88]	2012-08-14	JavaFX and Java Access Bridge included in Java SE JDK and JRE installation, JDK and JRE Support for Mac OS X, JDK for Linux on ARM[89]
Java SE 7 Update 7[90]	2012-08-30	Fixes security issues[60]

Mivel a probléma biztonsági javítás és a júniusi csomagba már nem fért bele, ezért csak az októberi csomagba kerülhetett volna bele, s nem adták ki az augusztusi csomagban, de a súlyossága okán kiadásra került egy rendkívüli csomag. 🙂