ORACLE CVE ID MAPPING

SE-2012-01

[Security vulnerabilities in Java SE]

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.

On 19-Jun-2013 [1], Oracle provided CVE numbers corresponding to vulnerabilities reported by Security Explorations as part of SE-2012-01 project [2]. They are presented in a table below.

ISSUE #	ORACLE TRACKING ID	CVE NUMBER	ORACLE COMMENT
1-7	S0171818	CVE-2012-5076	
8-9	S0171802	CVE-2012-5075	
10	S0171632	CVE-2012-1725	
11	S0171645	CVE-2012-4681	
12	S0171728	CVE-2012-0547	
13	S0171744	CVE-2012-1726	
14	S0171759	CVE-2012-5072	
15	S0171771	CVE-2012-5073	
16-17	S0171785	CVE-2012-4681	Duplicate of Issue 11
18-19	S0171792	CVE-2012-5074	
20	S0172208	CVE-2012-1682	
21			Duplicate of Issue 13
23-24	S0169569	CVE-2012-5071	
25	S0186854	CVE-2012-5084	
26	S0174636	CVE-2012-1726	Duplicate of Issue 13
27	S0158196	CVE-2012-5067	
28			Duplicate of Issue 16
29	S0151642	CVE-2013-0428	
30			Duplicate of Issue 14
31	S0180576	CVE-2012-5079	
32	S0207543	CVE-2012-5088	
50	S0212060	CVE-2013-1475	
51	S0331258	CVE-2013-1518	
52	S0331262	CVE-2013-0431	
53	S0332748	CVE-2013-1489	
54	S0344573		Not a bug
55	S0344587	CVE-2013-2436	
56	S0346309		Not a bug
57	S0346299	CVE-2013-2421	
58	S0346321	CVE-2013-2421	Duplicate of Issue 57
59	S0346313	CVE-2013-2422	
60	S0346345	CVE-2013-2422	Duplicate of Issue 59
61	S0363000	CVE-2013-2460	

Below, we provide additional comments with respect to the received CVE mapping information:

- Oracle tends to cumulate multiple different security issues under one CVE id, which leads to a misleading and inaccurate vulnerability information. As a result, the number of security fixes announced by the company via Java SE CPUs / Alerts is not necessarily reflecting the number of real security issues addressed. Sample cases include:
  a) CVE-2012-5076, which stands for 7 different issues (code locations) that stem from insecure use of `invoke` method of `java.lang.reflect.Method` class,
  b) CVE-2012-4681, which stands for 4 different issues (code locations) that stem from completely different Reflection API abuses (`forName`, `getMethods`, `getConstructors` and `getFields` calls of `java.lang.Class` class),

c) CVE-2012-1726, which stands for 3 different issues representing scenarios for the abuse of the original implementation of new Reflection API's security model. By design, this model initially relied on security checks conducted solely against the `Lookup` class,

d) CVE-2013-2422, which stands for 2 different issues. One was about the ability to access classes in a restricted package / Class Loader namespace. The other was about the ability to invoke arbitrary methods from a privileged system class, which led to immediate breaking of Oracle's mitigation for `doPrivileged` method call of `java.security.AccessController` class.

- The Risk Matrices footnotes [3][4][5][6][7][8] used by Oracle for most security vulnerabilities reported by Security Explorations suggest that these vulnerabilities applied to client deployment of Java only and that they could be exploited only through untrusted Java Web Start applications and untrusted Java applets. This is not true:
  a) RMI protocol could be successfully used to remotely exploit Java SE vulnerabilities on servers [9] till Apr 2013 when this exploit vector was finally addressed by the company (8 years from a vulnerability report to the fix),
  b) CVE-2013-2460 was proved to affect Server JRE [10], Oracle's runtime environment specifically targeted for deploying Java in server environments. Regardless of this, Oracle claimed CVE-2013-2460 was applicable to client deployments of Java only.
- Oracle's CVSS score of 0.0 for a Click-2-Play bypass vulnerability (CVE-2013-1489) may indicate that these types of issues / the mechanism itself are not that relevant from a security point of view.

REFERENCES

[1] SE-2012-01 Vendors status
`http://www.security-explorations.com/en/SE-2012-01-status.html`

[2] SE-2012-01 Details
`http://www.security-explorations.com/en/SE-2012-01-details.html`

[3] Java SE Critical Patch Update - June 2012
`http://www.oracle.com/technetwork/topics/security/javacpujun2012-1515912.html`

[4] Java SE Critical Patch Update - October 2012
`http://www.oracle.com/technetwork/topics/security/javacpuoct2012-1515924.html`

[5] Java SE Critical Patch Update - February 2013
`http://www.oracle.com/technetwork/topics/security/javacpufeb2013-1841061.html`

[6] Java SE Critical Patch Update - April 2013
`http://www.oracle.com/technetwork/topics/security/javacpuapr2013-1928497.html`

[7] Java SE Critical Patch Update - June 2013
`http://www.oracle.com/technetwork/topics/security/javacpujun2013-1899847.html`

[8] Alert for CVE-2012-4681
http://www.oracle.com/technetwork/topics/security/alert-cve-2012-4681-1835715.html

[9] Proof of Concept code for server side RMI attack
http://www.security-explorations.com/materials/se-2012-01-rmi.zip

[10] Server JRE (Java SE Runtime Environment) 7 Downloads
http://www.oracle.com/technetwork/java/javase/downloads/server-jre7-downloads-1931105.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.