# 一、缺陷列表

## 第一部分，56个缺陷，来源于MuBench

1. 调用Cipher.init前没有调用Cipher.getInstance进行初始化

补丁地址：

<https://github.com/alibaba/druid/commit/e10f2849d046265bf17360ab4aa9eb60fd3ab8de>

关键方法：javax.crypto.Cipher.init

错误调用序列：javax.crypto.Cipher.init

正确调用序列：javax.crypto.Cipher.getInstance javax.crypto.Cipher.init

1. 调用Cipher.doFinal时，传入未制定编码的字符串参数

补丁地址：

<https://sourceforge.net/p/adempiere/svn/1312/tree/trunk/looks/src/org/compiere/util/Secure.java?diff=5139a2ef34309d2ec1827857:1311>

关键方法：javax.crypto.Cipher.doFinal

错误调用序列：java.lang.String.getBytes() javax.crypto.Cipher.doFinal(byte[])

正确调用序列：java.lang.String.getBytes(String) javax.crypto.Cipher.doFinal(byte[])

1. 调用Cipher.doFinal返回的结果在变回String之前没有指定编码

补丁地址：

<https://sourceforge.net/p/adempiere/svn/1312/tree/trunk/looks/src/org/compiere/util/Secure.java?diff=5139a2ef34309d2ec1827857:1311>

关键方法：javax.crypto.Cipher.doFinal

错误调用序列：javax.crypto.Cipher.doFinal java.lang.String.new(byte[])

正确调用序列：javax.crypto.Cipher.doFinal java.lang.String.new(byte[],String)

1. 调用Mac.doFinal时，传入未制定编码的字符串参数

补丁地址：

<https://github.com/android-rcs/rcsjta/commit/04d84799daa51ed7cc0ad270f0eea51ffaf7a53a#diff-bf160ca00204f2ae4c100aabe57a1dfd>

关键方法：javax.crypto.Mac

错误调用序列：java.lang.String.getBytes() javax.crypto.Mac.doFinal(byte[])

正确调用序列：java.lang.String.getBytes(String) javax.crypto.Mac.doFinal(byte[])

1. When a DataOutputStream instance wraps an underlying ByteArrayOutputStream instance, it is recommended to flush or close the DataOutputStream before invoking the underlying instances's toByteArray().

补丁地址：

<https://github.com/apache/gora/commit/4f98503c014c359b6ab018f43920d454b0344160>

关键方法：java.io.ByteArrayOutputStream.toByteArray()

错误调用序列：java.io.DataOutputStream.new(java.io.ByteArrayOutputStream), java.io.ByteArrayOutputStream.toByteArray()

正确调用序列：java.io.DataOutputStream.new(java.io.ByteArrayOutputStream), java.io.DataOutputStream.flush(),java.io.ByteArrayOutputStream.toByteArray()

1. 调用java.lang.Long.parseLong时，未进行异常判断

补丁地址：

<https://github.com/emopers/asterisk-java/commit/41461b41309bf9f027a46f178cb777a1a94b8c3f>

关键方法：java.lang.Long.parseLong

错误调用序列：java.lang.Long.parseLong

正确调用序列：try java.lang.Long.parseLong catch

1. 调用formatter.format对时间进行特定格式输出时，未设置时区

补丁地址：

<https://issues.apache.org/jira/secure/attachment/12327222/ASF.LICENSE.NOT.GRANTED--dateutil.patch>

关键方法：java.text.SimpleDateFormat.format

错误调用序列：

java.text.SimpleDateFormat.new(java.lang.String) java.text.SimpleDateFormat.format(java.util.Date)

正确调用序列：

java.text.SimpleDateFormat.new(java.lang.String) java.text.SimpleDateFormat.setTimeZone

java.text.SimpleDateFormat.format(java.util.Date)

1. 调用java.lang.Byte.parseByte时未进行异常判断，可能抛出NumberFormatException

补丁地址：

<https://github.com/emopers/Confucius/commit/2fefd5eb6d77ab1b83efe3bf50171aabc7128460>

关键方法：java.lang.Byte.parseByte

错误调用序列：java.lang.Byte.parseByte

正确调用序列：try java.lang.Byte.parseByte catch

1. 调用java.lang.Short.parseShort时未进行异常判断，可能抛出异常

补丁地址：

<https://github.com/emopers/Confucius/commit/2fefd5eb6d77ab1b83efe3bf50171aabc7128460>

关键方法：java.lang.Short.parseShort

错误调用序列：java.lang.Short.parseShort

正确调用序列：try java.lang.Short.parseShort catch

1. 调用org.apache.jackrabbit.core.config.ConfigurationParser.getAttribute()之后，需要继续调用ConfigurationParser.replaceVariables()对返回结果进行处理

补丁地址：

<http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-core/src/main/java/org/apache/jackrabbit/core/config/RepositoryConfigurationParser.java?r1=695766&r2=695765&pathrev=695766&diff_format=h>

关键方法：org.apache.jackrabbit.core.config.ConfigurationParser.getAttribute

错误调用序列：org.apache.jackrabbit.core.config.ConfigurationParser.getAttribute

正确调用序列：org.apache.jackrabbit.core.config.ConfigurationParser.getAttribute org.apache.jackrabbit.core.config.ConfigurationParser.replaceVariables

1. 调用java.io.InputStream之后，未关闭。

补丁地址：

<http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-spi-commons/src/main/java/org/apache/jackrabbit/spi/commons/value/QValueFactoryImpl.java?r1=698246&r2=698245&pathrev=698246&diff_format=h>

关键方法：java.io.InputStream

错误调用序列：java.io.InputStream.read

正确调用序列：java.io.InputStream.read java.io.InputStream.close

FileOutputStream未关闭

1. Boolean.parseBoolean() turns a string to a boolean, while Boolean.getBoolean() looks up the boolean value of a system property. The latter method was called here, where it should have been the former.

补丁地址：

<http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-core/src/main/java/org/apache/jackrabbit/core/persistence/xml/XMLPersistenceManager.java?r1=965539&r2=965538&pathrev=965539&diff_format=h>

关键方法：org.apache.jackrabbit.core.util.DOMWalker.getAttribute

错误调用序列：

"org.apache.jackrabbit.core.util.DOMWalker.getAttribute(java.lang.String)","java.lang.Boolean.getBoolean(java.lang.String)","org.apache.jackrabbit.core.state.PropertyState.setMultiValued(boolean)"

正确调用序列：

"org.apache.jackrabbit.core.util.DOMWalker.getAttribute(java.lang.String)","java.lang.Boolean.parseBoolean(java.lang.String)","org.apache.jackrabbit.core.state.PropertyState.setMultiValued(boolean)"

1. javax.jcr.NamespaceRegistry.unregisterNamespace(String prefix) expects the namespace prefix， but org.apache.jackrabbit.jcr2spi.NamespaceStorage.unregisterNamespace(String uri), which expects the namespace uri.

补丁地址：

<http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-jcr2spi/src/main/java/org/apache/jackrabbit/jcr2spi/NamespaceRegistryImpl.java?r1=1133332&r2=1133331&pathrev=1133332&diff_format=h>

关键方法：org.apache.jackrabbit.jcr2spi.NamespaceStorage.unregisterNamespace

错误调用序列：org.apache.jackrabbit.jcr2spi.NamespaceStorage.unregisterNamespace

正确调用序列：

org.apache.jackrabbit.jcr2spi.NamespaceStorage.unregisterNamespace.getURI

org.apache.jackrabbit.jcr2spi.NamespaceStorage.unregisterNamespace.unregisterNamespace

1. org.apache.jackrabbit.core.fs.FileSystemResource.delete调用前，没有调用org.apache.jackrabbit.core.fs.FileSystemResource.exists判断是否存在

补丁地址：

<http://svn.apache.org/viewvc/incubator/jackrabbit/trunk/src/java/org/apache/jackrabbit/core/state/xml/XMLPersistenceManager.java?r1=55854&r2=55853&pathrev=55854&diff_format=h>

关键方法：org.apache.jackrabbit.jcr2spi.NamespaceStorage.unregisterNamespace

错误调用序列：org.apache.jackrabbit.core.fs.FileSystemResource.delete

正确调用序列：org.apache.jackrabbit.core.fs.FileSystemResource.exists

org.apache.jackrabbit.core.fs.FileSystemResource.delet

1. 创建多级目录时，使用File.mkdir会出错，无法创建。应该使用File.mkdirs。谨慎起见，应当都使用File.mkdirs。当路径为单级目录时，使用File.mkdir不会出错；但是当路径为多级目录时，使用File.mkdir一定会错。统一使用File.mkdirs是为了更好的避免缺陷，也可以称之为for refactoring。

补丁地址：

<http://svn.apache.org/viewvc/incubator/jackrabbit/trunk/contrib/bdb-persistence/src/java/org/apache/jackrabbit/core/state/bdb/BerkeleyDBPersistenceManager.java?r1=263892&r2=263891&pathrev=263892&diff_format=h>

关键方法：File.mkdir()

错误调用序列：File.exists() File.mkdir()

正确调用序列：File.exists() File.mkdirs()

1. DataOutputStream未关闭。

补丁地址：

<https://sourceforge.net/p/jmrtd/code/51/tree//passporthostapi/src/sos/mrtd/SecureMessagingWrapper.java?diff=5058d727fd48f84fd52d6740:50>

关键方法：java.io.DataOutputStream.new

错误调用序列：java.io.DataOutputStream.new

正确调用序列：java.io.DataOutputStream.new java.io.DataOutputStream.close

1. 与缺陷6类似， When an ObjectOutputStream instance wraps an underlying ByteArrayOutputStream instance, it is recommended to flush or close the ObjectOutputStream before invoking the underlying instances's toByteArray(). Although in these cases this is not strictly necessary because the writeObject method is invoked right before toByteArray, and writeObject internally calls flush/drain. However, it is a good practice to call flush/close explicitly as mentioned for example [here](http://stackoverflow.com/questions/2984538/how-to-use-bytearrayoutputstream-and-dataoutputstream-simultaneously-java). This pull request flips the order of close and toytBeArray methods. While there are seemingly many changes, they're all just copies of the same change. Please let me know if you want me to extract the common code into one helper method.

补丁地址：

<https://github.com/emopers/joda-time/commit/0e82517a7f4a2e659143ce84ff1f4ea706e9a88c>

关键方法：java.io.ObjectOutputStream.new

错误调用序列：java.io.ObjectOutputStream.new "java.io.ByteArrayOutputStream.toByteArray()"

正确调用序列：java.io.ObjectOutputStream.new java.io.ObjectOutputStream.close() "java.io.ByteArrayOutputStream.toByteArray()"

这是缺陷的一种修复方式，其实还可以用

java.io.ObjectOutputStream.new java.io.ObjectOutputStream.flush() "java.io.ByteArrayOutputStream.toByteArray()" 进行修复

1. 19. Should catch NoSuchElementException around SortedMap.firstKey()

补丁地址：

<http://svn.apache.org/viewvc/lucene/java/trunk/src/java/org/apache/lucene/index/ParallelReader.java?r1=816602&r2=816601&pathrev=816602&diff_format=h>

关键方法：java.util.SortedMap.firstKey

错误调用序列：java.util.SortedMap.firstKey

正确调用序列：try java.util.SortedMap.firstKey catch

1. On some platforms AppInfo.loadIcon() may throw an OutOfMemoryError.

StackOverflow: <http://stackoverflow.com/questions/20415367/android-async-loadicon-outofmemoryerror>

补丁地址：

<https://github.com/lkorth/screen-notifications/commit/afc542066ae1a3c57974dc180374407a7954124d>

关键方法：android.content.pm.ApplicationInfo.loadIcon

错误调用序列：android.content.pm.ApplicationInfo.loadIcon

正确调用序列：try android.content.pm.ApplicationInfo.loadIcon catch

1. Misses to call ByteBuffer.flip() between writing to and reading from the buffer (or vice versa). There''s many ways to read/write buffer, some of which fail if flip() is missing, while others just do nothing.

缺陷地址：

<https://stackoverflow.com/questions/10166011/how-to-use-java-nio-channels-filechannel-to-write-a-byte-to-a-file-basics/10166055>

关键方法：java.nio.channels.FileChannel.write

错误调用序列：java.nio.channels.FileChannel.write(java.nio.ByteBuffer)

正确调用序列：

"java.nio.ByteBuffer.flip()","java.nio.channels.FileChannel.write(java.nio.ByteBuffer)"

1. Examiner.java calls 'scanner.next()' on 'java.util.Scanner scanner' without checking

if there are more elements. Because the scanner is built from the JavaSource parameter

that can be invalid (e.g., no token after opening parenthesis), this can lead to a

runtime exception without a useful error message.

补丁地址：

<https://github.com/emopers/visualee/commit/cb7b08d182b463e78f85f7851cce64d6cf0e36b5>

关键方法：java.util.Scanner.next

错误调用序列：java.util.Scanner.next

正确调用序列：java.util.Scanner.hasNext

java.util.Scanner.next

1. Calls Dialog.dismiss() on dialog that is not showing.

补丁地址：

https://github.com/Tyde/TuCanMobile/commit/02e046e5cc53ad218264dc28a4b96941f2902e38

关键方法：android.app.Dialog.dismiss()

错误调用序列：android.app.Dialog.dismiss()

正确调用序列：android.app.Dialog.isShowing() android.app.Dialog.dismiss()

1. Fails to release Cursor.

补丁地址：

https://github.com/ushahidi/Ushahidi\_Android/commit/16adf0c3cfc9ebef01e7e9042b13d3b5b4da965f

关键方法：android.database.Cursor

错误调用序列：android.database.Cursor

正确调用序列：android.database.Cursor android.database.Cursor.close()

1. StrBuilder.getNullText() may return null.

补丁地址：

http://svn.apache.org/viewvc/commons/proper/lang/trunk/src/java/org/apache/commons/lang/text/StrBuilder.java?r1=627248&r2=627247&pathrev=627248&diff\_format=h

关键方法：StrBuilder.getNullText()r

错误调用序列：StrBuilder.getNullText()

正确调用序列：StrBuilder.getNullText() has\_null\_check

1. org.apache.commons.math3.geometry.euclidean.threed.Line.intersection() may return null.

补丁地址：

<http://svn.apache.org/viewvc/commons/proper/math/trunk/src/main/java/org/apache/commons/math3/geometry/euclidean/threed/SubLine.java?r1=1488866&r2=1488865&pathrev=1488866&diff_format=h>

关键方法：org.apache.commons.math3.geometry.euclidean.threed.Line.intersection()r

错误调用序列：org.apache.commons.math3.geometry.euclidean.threed.Line.intersection()

正确调用序列：org.apache.commons.math3.geometry.euclidean.threed.Line.intersection() has\_null\_check

1. org.apache.commons.math3.geometry.euclidean.twod.Line.intersection() may return null.

补丁地址：

<http://svn.apache.org/viewvc/commons/proper/math/trunk/src/main/java/org/apache/commons/math3/geometry/euclidean/twod/SubLine.java?r1=1488866&r2=1488865&pathrev=1488866&diff_format=h>

关键方法：org.apache.commons.math3.geometry.euclidean.twod.Line.intersection()

错误调用序列：org.apache.commons.math3.geometry.euclidean.twod.Line.intersection()

正确调用序列：org.apache.commons.math3.geometry.euclidean.twod.Line.intersection() has\_null\_check

1. Environment.getExternalStorageState() may throw NullPointerException.

补丁地址：

<https://github.com/nostra13/Android-Universal-Image-Loader/commit/54a9038d7cc13fd46a922ce1c059610d0690115d>

关键方法：android.os.Environment.getExternalStorageState

错误调用序列：android.os.Environment.getExternalStorageState

正确调用序列：try android.os.Environment.getExternalStorageState catch

1. android.content.Intent.getLongExtra传入String对象，导致出错

补丁地址：

<https://github.com/fefe982/gnucash-android/commit/ce50ee116f18a373f30cb135ee48b5ad6c187db0>

关键方法： android.content.Intent.getLongExtra

错误调用序列：android.content.Intent.getLongExtra(java.lang.String)

正确调用序列：android.content.Intent.getStringExtra (java.lang.String)

1. org.apache.commons.httpclient.auth.AuthState.isPreemptive和invalidate调用以后，未调用setAuthRequested(true)。

补丁地址：

<http://svn.apache.org/viewvc/jakarta/commons/proper/httpclient/trunk/src/java/org/apache/commons/httpclient/HttpMethodDirector.java?r1=160490&r2=160489&pathrev=160490&diff_format=h>

关键方法： org.apache.commons.httpclient.auth.AuthState.isPreemptive

错误调用序列：

{"org.apache.commons.httpclient.auth.AuthState.isPreemptive()","org.apache.commons.httpclient.auth.AuthState.invalidate()"};

正确调用序列：

{"org.apache.commons.httpclient.auth.AuthState.isPreemptive()","org.apache.commons.httpclient.auth.AuthState.invalidate()","org.apache.commons.httpclient.auth.AuthState.setAuthRequested(boolean)"}

1. The `ResultSet` fetched in line 644 is never closed.

补丁地址：缺失

关键方法： java.sql.ResultSet

错误调用序列：

{"java.sql.ResultSet"};

正确调用序列：

{"java.sql.ResultSet","java.sql.ResultSet.close()"}

1. The `PreparedStatement` opened in line 702 is never closed.

(Same findings as `mudetec··t-17`)

补丁地址：缺失

关键方法： java.sql.PreparedStatement

错误调用序列：

{"java.sql.PreparedStatement "};

正确调用序列：

{"java.sql.PreparedStatement ","java.sql.PreparedStatement.close()"}

1. UnionTypeBuilder.build() returns a JSType that can never be null.

补丁地址：<https://github.com/google/closure-compiler/commit/67289ae4cbaba3ae70cd2e8fb92f3f2898039dfb>

关键方法： com.google.javascript.rhino.jstype.UnionType.Builder.build()

错误调用序列：

{" com.google.javascript.rhino.jstype.UnionType.Builder.build() ","has\_null\_check"};

正确调用序列：

{" com.google.javascript.rhino.jstype.UnionType.Builder.build() ","com.google.javascript.rhino.jstype.JSType.isNoType"};

该缺陷被MuBench整错了。详细比较<https://github.com/google/closure-compiler/search?q=UnionTypeBuilder&unscoped_q=UnionTypeBuilder>与补丁地址。可发现端倪。

1. org.kohsuke.args4j.spi.Parameters.getParameter() may throw CmdLineException.

补丁地址：https://github.com/google/closure-compiler/commit/43c245f0ff8d409e81e25687e69d34666b7cf26a

关键方法： org.kohsuke.args4j.spi.Parameters.getParameter

错误调用序列：

{"org.kohsuke.args4j.spi.Parameters.getParameter"};

正确调用序列：

{"Try","org.kohsuke.args4j.spi.Parameters.getParameter"};

1. com.google.javascript.rhino.jstype.ObjectType.getImplicitPrototype的返回值可能为空

补丁地址：https://github.com/google/closure-compiler/commit/d1cfe67977d8f3aaa85ec20c262171da394d5977

关键方法： com.google.javascript.rhino.jstype.ObjectType.getImplicitPrototype

错误调用序列：

{"com.google.javascript.rhino.jstype.ObjectType.getImplicitPrototype"}

正确调用序列：

{"com.google.javascript.rhino.jstype.ObjectType.getImplicitPrototyp","has\_null\_check"}

1. Iterator.next() is invoked without prior check that the iterator has more elements.

补丁地址：缺失

关键方法： java.util.Iterator<java.lang.CharSequence>.next()

错误调用序列：

{"java.util.Iterator<java.lang.CharSequence>.next()"}

正确调用序列：

{"java.util.Iterator<java.lang.CharSequence>.hasNext()","java.util.Iterator<java.lang.CharSequence>.next()"}

Size()

1. On line 424, a warning message is issued (`LOG.warn()`) without a prior

checking whether warning messages are enabled (`LOG.isWarnEnable()`).

补丁地址：缺失

关键方法： org.apache.commons.logging.Log.warn

错误调用序列：

{"org.apache.commons.logging.Log.warn"}

正确调用序列：

{"org.apache.commons.logging.Log.warn.isWarnEnable",org.apache.commons.logging.Log.warn"}

1. On line 549, `StringTokenizer.nextToken()` is invoked without prior check of `hasMoreTokens()`. This usage may be safe from context, since the tokenized value is an RBG color value which has a known number of tokens. However, the usage is unsafe against invalid values and we, therefore, consider it a misuse.

补丁地址：缺失

关键方法： java.util.StringTokenizer

错误调用序列：

{"java.util.StringTokenizer.nextToken()"}

正确调用序列：

{"java.util.StringTokenizer.hasMoreTokens","java.util.StringTokenizer.nextToken()"}

hasMoreElements

1. HashMap.get() may return null.

补丁地址：http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-jcr-server/src/main/java/org/apache/jackrabbit/webdav/jcr/JcrDavException.java?r1=677893&r2=677892&pathrev=677893&diff\_format=h

关键方法： java.util.HashMap.get

错误调用序列：

{"java.util.HashMap.get"};

正确调用序列：

{"java.util.HashMap.get","has\_null\_check"};

1. IOManager.getDetector() may return null.

补丁地址：http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-jcr-server/src/main/java/org/apache/jackrabbit/server/io/DefaultHandler.java?r1=880743&r2=880742&pathrev=880743&diff\_format=h

关键方法： org.apache.jackrabbit.server.io.IOManager.getDetector

错误调用序列：

{"org.apache.jackrabbit.server.io.IOManager.getDetector"};

正确调用序列：

{"org.apache.jackrabbit.server.io.IOManager.getDetector","has\_null\_check"};

1. NodeState.getParentId() may return null and ItemStateManager.getItemState() expects non-null parameter.

补丁地址：http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-core/src/main/java/org/apache/jackrabbit/core/query/lucene/IndexingConfigurationImpl.java?r1=926354&r2=926353&pathrev=926354&diff\_format=h

关键方法： org.apache.jackrabbit.core.state.NodeState.getParentId

错误调用序列：

{"org.apache.jackrabbit.core.state.NodeState.getParentId"}; 应当加入ItemStateManager.getItemState否则，误报会很多。

正确调用序列：

{"org.apache.jackrabbit.core.state.NodeState.getParentId","has\_null\_check"};

1. 42. HierarchyEntry.getParent() may return null.

补丁地址：http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-jcr2spi/src/main/java/org/apache/jackrabbit/jcr2spi/state/ItemState.java?r1=1240198&r2=1240197&pathrev=1240198&diff\_format=h

关键方法： org.apache.jackrabbit.jcr2spi.hierarchy.HierarchyEntry.getParent

错误调用序列：

{"org.apache.jackrabbit.jcr2spi.hierarchy.HierarchyEntry.getParent"};

正确调用序列：

{"org.apache.jackrabbit.jcr2spi.hierarchy.HierarchyEntry.getParent","has\_null\_check"};

1. Map.get() may return null and RepositoryConfig.create() expects non-null

parameter.

补丁地址：<http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-jca/src/main/java/org/apache/jackrabbit/jca/JCARepositoryManager.java?r1=1232831&r2=1232830&pathrev=1232831&diff_format=h>

关键方法： java.util.Map<java.lang.String, java.lang.String>.get

错误调用序列：

{" java.util.Map<java.lang.String, java.lang.String>.get"};

正确调用序列：

{" java.util.Map<java.lang.String, java.lang.String>.get","has\_null\_check"};

1. 44. ObjectPersistenceManager.destroy() calls FileSystemResource.delete() (which delegates to FileSystem.deleteFile()) without checking FileSystemResource.fileExists() (which delegates to FileSytem.fileExists()). FileSystem.deleteFile() throws, if the file does not exist. A refactoring was performed in the course of the fix, to remove duplication.

补丁地址：<http://svn.apache.org/viewvc/incubator/jackrabbit/trunk/src/java/org/apache/jackrabbit/core/state/obj/ObjectPersistenceManager.java?r1=55854&r2=55853&pathrev=55854&diff_format=h>

关键方法： org.apache.jackrabbit.core.fs.FileSystemResource.delete

错误调用序列：

org.apache.jackrabbit.core.fs.FileSystemResource.delete

正确调用序列：

org.apache.jackrabbit.core.fs.FileSystemResource.exists org.apache.jackrabbit.core.fs.FileSystemResource.delete

1. 45. PlotRenderingInfo.getOwner() may return null.

补丁地址：<https://sourceforge.net/p/jfreechart/code/103/tree/trunk/source/org/jfree/chart/axis/Axis.java?diff=50b53b485fcbc92b6542a639:102>

关键方法： org.jfree.chart.plot.PlotRenderingInfo.getOwner

错误调用序列：

{"org.jfree.chart.plot.PlotRenderingInfo.getOwner"};

正确调用序列：

{"org.jfree.chart.plot.PlotRenderingInfo.getOwner","has\_null\_check"};

1. 46. StatisticalCategoryDataset.getMeanValue() may return null.

补丁地址：https://sourceforge.net/p/jfreechart/code/164/tree/trunk/source/org/jfree/chart/renderer/category/StatisticalBarRenderer.java?diff=50b53b485fcbc92b6542a639:163

关键方法： org.jfree.data.statistics.StatisticalCategoryDataset.getMeanValue

错误调用序列：

{" org.jfree.data.statistics.StatisticalCategoryDataset.getMeanValue"};

正确调用序列：

{"org.jfree.data.statistics.StatisticalCategoryDataset.getMeanValue","has\_null\_check"};

1. 47. StatisticalCategoryDataset.getStDevValue() may return null.

补丁地址：https://sourceforge.net/p/jfreechart/code/164/tree/trunk/source/org/jfree/chart/renderer/category/StatisticalBarRenderer.java?diff=50b53b485fcbc92b6542a639:163

关键方法： org.jfree.data.statistics.StatisticalCategoryDataset.getStdDevValue

错误调用序列：

{" org.jfree.data.statistics.StatisticalCategoryDataset.getStdDevValue "};

正确调用序列：

{"org.jfree.data.statistics.StatisticalCategoryDataset.getStdDevValue ","has\_null\_check"};

1. 48. XYPlot.getRendererForDataset() may return null.

补丁地址：https://sourceforge.net/p/jfreechart/code/2183/tree/trunk/source/org/jfree/chart/plot/XYPlot.java?diff=50b53b485fcbc92b6542a639:2182

关键方法： org.jfree.chart.plot.XYPlot.getRendererForDataset

错误调用序列：

org.jfree.chart.plot.XYPlot.getRendererForDataset

正确调用序列：

org.jfree.chart.plot.XYPlot.getRendererForDataset has\_null\_check

1. 49. CategoryPlot.getDataset() may return null. The check for null was flipped.

补丁地址：https://sourceforge.net/p/jfreechart/code/2266/tree/trunk/source/org/jfree/chart/renderer/category/AbstractCategoryItemRenderer.java?diff=50b53b485fcbc92b6542a639:2265

关键方法： org.jfree.chart.plot.XYPlot.getRendererForDataset

错误调用序列：

org.jfree.chart.plot.XYPlot.getRendererForDataset

正确调用序列：

org.jfree.chart.plot.XYPlot.getRendererForDataset has\_null\_check

1. Map.get() may return null.

补丁地址：https://sourceforge.net/p/jfreechart/code/811/tree/trunk/source/org/jfree/chart/plot/CategoryPlot.java?diff=50b53b485fcbc92b6542a639:810

关键方法： java.util.Map.get

错误调用序列：

java.util.Map.get

正确调用序列：

java.util.Map.get,"has\_null\_check"

1. RandomAccessFile.close() maybe skipped.

补丁地址：

http://svn.apache.org/viewvc/lucene/java/trunk/src/java/org/apache/lucene/store/FSDirectory.java?r1=642339&r2=642338&pathrev=642339&diff\_format=h

关键方法： java.io.RandomAccessFile

错误调用序列：

java.io.RandomAccessFile

正确调用序列：

java.io.RandomAccessFile java.io.RandomAccessFile.close()

1. Misses to check File.exists()/File.mkdirs() before calling File.createNewFile().

补丁地址：

<http://svn.apache.org/viewvc/lucene/java/trunk/src/java/org/apache/lucene/store/FSDirectory.java?r1=150347&r2=150346&pathrev=150347&diff_format=h>

关键方法： java.io.File.createNewFile

错误调用序列：

java.io.File.createNewFile

正确调用序列：

"java.io.File.exists","java.io.File.mkdirs" java.io.File.createNewFile

1. 53. MultiFields.getDeletedDocs() may return null.

补丁地址：http://svn.apache.org/viewvc/lucene/dev/trunk/lucene/contrib/benchmark/src/java/org/apache/lucene/benchmark/byTask/tasks/DeleteByPercentTask.java?r1=957486&r2=957485&pathrev=957486&diff\_format=h

关键方法： org.apache.lucene.index.MultiFields.getDeletedDocs

错误调用序列：

org.apache.lucene.index.MultiFields.getDeletedDocs

正确调用序列：

org.apache.lucene.index.MultiFields.getDeletedDocs,"has\_null\_check"

1. 54. Registers an IntentFilter to an arbitrary context, rather than the application

context.

补丁地址：

<https://github.com/haku/Onosendai/commit/0e2a7570ab4491d0c4680ef52ee1008bef33fc02>

关键方法： android.content.Context.registerReceiver

错误调用序列：

android.content.Context.registerReceiver

正确调用序列：

android.content.Context.getApplicationContext, android.content.Context.registerReceiver

1. 55. ItemData.getState() may be null.

补丁地址：

<http://svn.apache.org/viewvc/jackrabbit/trunk/jackrabbit-core/src/main/java/org/apache/jackrabbit/core/ItemManager.java?r1=656282&r2=656281&pathrev=656282&diff_format=h>

关键方法： android.content.Context.registerReceiver

错误调用序列：

android.content.Context.registerReceiver

正确调用序列：

android.content.Context.getApplicationContext, android.content.Context.registerReceiver

1. 56. Calls JFrame.setVisible(true) before JFrame.pack(). This may lead to

undesired effects, such as the window being moved to the default plattform

location.

补丁缺失

关键方法：JFrame.setVisible

错误调用序列：

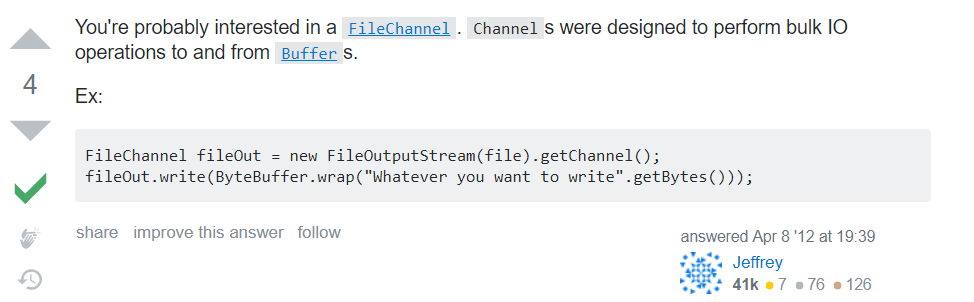
JFrame.setVisible JFrame.pack

正确调用序列：

JFrame.pack JFrame.setVisible

## 第二部分，57-89，来源于论文A Study of API Misuse on Stack Ov{Zhang, 2018 #180}{Zhang, 2018 #180}erflow中的数据 从编号

该文献用于发现Stack Overflow中被采纳为正确答案（“correct usage”）的代码片段是否存在API误用缺陷。以网址[[1]](#footnote-1)中的为例，被采纳为正确答案的代码片段如下。但是，该段代码未关闭FileChannel对象，存在API误用缺陷，因此会对参考这个答案的用户产生误导，导致误用缺陷传播开来。



1. FileChannel对象未关闭。if Alice reuses the example in a long-running program with heavy IO, such lingering file resources may cause file handle leaks. Since most operating systems limit the number of opened fles, unclosed fle streams can eventually run out of file handle resources

Stack Overflow地址：<https://stackoverflow.com/questions/10065723/java-bulk-write/10065852#10065852>

关键方法： java.nio.channels.FileChannel

错误调用序列：

java.nio.channels.FileChannel

正确调用序列：

java.nio.channels.FileChannel, java.nio.channels.FileChannel.close

1. 58. As a novice programmer, Alice may not easily recognize the potential limitation of given Stack Overﬂow examples. In this case, our approach ExampleCheck scans over 380K GitHub repositories and fnds 2230 GitHub snippets that also call FileChannel.write. ExampleCheck then learns two common usage patterns from these relevant GitHub snippets. The mostly frequent usage supported by 1829 code snippets on GitHub indicates that a method call to write() must be contained inside a try and catch block. Another frequent usage supported by 1267 GitHub snippets indicates that write must be followed by close.

Stack Overflow地址：http://stackoverﬂow.com/questions/10506546

关键方法： java.nio.channels.FileChannel.write

错误调用序列：

java.nio.channels.FileChannel.write

正确调用序列：

try java.nio.channels.FileChannel.write catch

1. SQLiteDatabase对象未关闭

Stack Overflow地址：https://stackoverﬂow.com/questions/31531250

关键方法： android.database.sqlite.SQLiteDatabase

错误调用序列：android.database.sqlite.SQLiteDatabase

正确调用序列：

android.database.sqlite.SQLiteDatabase android.database.sqlite.SQLiteDatabase.close

1. String.substring抛出的异常未处理

Stack Overflow地址：https://stackoverflow.com/questions/12742594/extract-src-image-field-from-a-simple-html-string-without-a-parser/12742734#12742734

关键方法： java.lang.String.substring

错误调用序列：java.lang.String.substring

正确调用序列：

try java.lang.String.substring

1. parseInt抛出的异常未处理

Stack Overflow地址：http://stackoverflow.com/questions/15902574

关键方法： java.lang.Integer.parseInt

错误调用序列：java.lang.Integer.parseInt

正确调用序列：

Try java.lang.Integer.parseInt

1. 62. TypedArray.getString可能返回null

Stack Overflow地址：TypedArray.getString

关键方法： android.content.res.TypedArray.getString

错误调用序列：android.content.res.TypedArray.getString

正确调用序列：

android.content.res.TypedArray.getString has\_null\_check

1. 63. Should catch NoSuchElementException around TreeMap.firstKey()

Stack Overflow地址：http://stackoverﬂow.com/questions/21983867

关键方法：java.util.TreeMap.firstKey

错误调用序列：java.util.TreeMap.firstKey

正确调用序列：try java.util.TreeMap.firstKey catch

1. 调用new SimpleDateFormat时未进行异常判断

Stack Overflow地址：http://stackoverflow.com/questions/30105609

关键方法：java.text.SimpleDateFormat.new

错误调用序列：java.text.SimpleDateFormat.new

正确调用序列：try java.text.SimpleDateFormat.new

1. not checking if the input string is null

Stack Overflow地址：http://stackoverflow.com/questions/33775696

关键方法：java.lang.String.split

错误调用序列：java.lang.String.split

正确调用序列：has\_null\_check java.lang.String.split

1. Open a FileInputStream on a non-existing file throws. may throw FileNotFoundException.

Stack Overflow地址：http://stackoverflow.com/questions/7916736

关键方法："java.io.FileInputStream.new";

错误调用序列："java.io.FileInputStream.new";

正确调用序列：Try "java.io.FileInputStream.new";

1. 调用Cipher.init时没有进行抛出异常判断

补丁地址：

<https://github.com/alibaba/druid/commit/e10f2849d046265bf17360ab4aa9eb60fd3ab8de>

关键方法：javax.crypto.Cipher.init

错误调用序列：javax.crypto.Cipher.init

正确调用序列：try Cipher.init catch

1. 调用java.io.BufferedReader.readLine可能抛出异常

Stack Overflow地址：https://stackoverflow.com/questions/2810626/crude-runtime-exec-to-call-java-cp-not-working-in-linux/2810857#2810857

关键方法："java.io.BufferedReader.readLine";

错误调用序列："java.io.BufferedReader.readLine";

正确调用序列："try java.io.BufferedReader.readLine";

1. 68. 调用android.os.Bundle.getString后需要进行空检查

Stack Overflow地址：http://stackoverflow.com/questions/39137999

关键方法："android.os.Bundle.getString";

错误调用序列："android.os.Bundle.getString";

正确调用序列："android.os.Bundle.getString has\_null\_check";

1. 69. 调用android.os.Bundle.getString后未进行异常捕捉

Stack Overflow地址：http://stackoverflow.com/questions/37431352

关键方法："android.os.Bundle.getString";

错误调用序列："android.os.Bundle.getString";

正确调用序列："android.os.Bundle.getString has\_null\_check";

1. 70. 调用Class.forName后未进行异常捕捉

Stack Overflow地址：http://stackoverflow.com/questions/116625

关键方法：" java.lang.Class<T>.ForName";

错误调用序列：" java.lang.Class<T>.ForName";

正确调用序列："TRY java.lang.Class<T>.ForName";

1. 71. 调用Double.parseDouble后未进行异常捕捉

Stack Overflow地址：http://stackoverflow.com/questions/21310279

关键方法："java.lang.Double.parseDouble";

错误调用序列：" java.lang.Double.parseDouble";

正确调用序列："TRY java.lang.Double.parseDouble";

1. 72. 调用HttpClient.execute后未进行异常捕捉

Stack Overflow地址：http://stackoverflow.com/questions/23007311

关键方法："org.apache.http.client.HttpClient.execute";

错误调用序列："org.apache.http.client.HttpClient.execute";

正确调用序列："TRY org.apache.http.client.HttpClient.execute”;

1. 73. 调用org.apache.http.HttpResponse.getEntity后未进行异常捕捉

Stack Overflow地址：https://stackoverflow.com/questions/18316242/cannot-process-url-with-vertical-pipe-bar-in-java-apache-httpclient/18316928#18316928

关键方法：" org.apache.http.HttpResponse.getEntity";

错误调用序列：" org.apache.http.HttpResponse.getEntity";

正确调用序列："TRY org.apache.http.HttpResponse.getEntity”;

1. 74. 调用javax.servlet.http.HttpServletRequest.getParameter后未进行空值检查

Stack Overflow地址：<http://stackoverflow.com/questions/27960657>

关键方法："javax.servlet.http.HttpServletRequest.getParameter";

错误调用序列："javax.servlet.http.HttpServletRequest.getParameter";

正确调用序列：" javax.servlet.http.HttpServletRequest.getParameter has\_null\_check”;

1. 75. 调用android.content.Intent.getStringExtra后未进行空值检查

Stack Overflow地址：<http://stackoverflow.com/questions/27960657>

关键方法：" android.content.Intent.getStringExtra";

错误调用序列：" android.content.Intent.getStringExtra";

正确调用序列：" android.content.Intent.getStringExtra has\_null\_check”;

1. 76. 调用java.awt.Button.new来创建javax.swing.JButton对象

Stack Overflow地址：http://stackoverflow.com/questions/23476519

关键方法：" javax.swing.JButton ";

错误调用序列：" javax.swing.JButton-java.awt.Button.new";

正确调用序列：" javax.swing.JButton- javax.swing.JButton.new”;

1. 77. 调用JSONObject.getString后未进行异常检查

Stack Overflow地址：http://stackoverflow.com/questions/23476519

关键方法：" org.json.JSONObject.getString";

错误调用序列：" org.json.JSONObject.getString";

正确调用序列："try org.json.JSONObject.getString”;

1. 78. 调用Jsoup.connect后未进行异常检查

Stack Overflow地址：http://stackoverflow.com/questions/21656728

关键方法：" org.jsoup.Jsoup.connect;

错误调用序列：" org.jsoup.Jsoup.connect";

正确调用序列："try org.json.JSONObject.getString”;

1. 79. 调用java.security.KeyStore.load后未进行异常检查

Stack Overflow地址：http://stackoverflow.com/questions/37577463

关键方法：" java.security.KeyStore.load;

错误调用序列：" java.security.KeyStore.load";

正确调用序列："try java.security.KeyStore.load”;

1. 80. 调用android.app.ProgressDialog.dismiss之前，未进行空检查

Stack Overflow地址：http://stackoverflow.com/questions/37541746

关键方法：" android.app.ProgressDialog.dismiss"

错误调用序列：" android.app.ProgressDialog.dismiss"

正确调用序列"has\_null\_check android.app.ProgressDialog.dismiss"

1. 81. 调用Random.nextInt之后，未进行异常检查

Stack Overflow地址：http://stackoverflow.com/questions/35488770

关键方法："java.util.Random.nextInt"

错误调用序列："java.util.Random.nextInt"

正确调用序列"TRY java.util.Random.nextInt"

1. 82. 调用java.sql.ResultSet.getString之后，未进行异常检查

Stack Overflow地址：http://stackoverflow.com/questions/29530026

关键方法："java.sql.ResultSet.getString"

错误调用序列："java.sql.ResultSet.getString"

正确调用序列"TRY java.sql.ResultSet.getString"

1. 83. 调用Scanner.nextLine之前，应该调用hasNextLine而非hasNext

Stack Overflow地址：http://stackoverflow.com/questions/36391399

关键方法：" java.util.Scanner.nextLine "

错误调用序列：" java.util.Scanner.hasNext java.util.Scanner.nextLine"

正确调用序列" java.util.Scanner.hasNextLine java.util.Scanner.nextLine "

1. 84. 调用Socket.getOutputStream之后，未进行异常检查

Stack Overflow地址：http://stackoverflow.com/questions/14654133

关键方法：" java.net.Socket.getOutputStream"

错误调用序列：" java.net.Socket.getOutputStream"

正确调用序列"TRY java.net.Socket.getOutputStream"

1. 85. FileOutputStream未关闭

关键方法：java.io.FileOutputStream.write

错误调用序列：java.io.FileOutputStream.write

正确调用序列：java.io.FileOutputStream.write java.io.FileOutputStream.close

|  |
| --- |
| 原来的85号缺陷。其实人家要求是对String.charAt进行前置的检查。我缺理解成了判断是否抛出异常。导致模式录入错误。  调用String.charAt之后，未进行异常检查  Stack Overflow地址：http://stackoverflow.com/questions/6210258  关键方法："java.lang.String.charAt"  错误调用序列："java.lang.String.charAt"  正确调用序列"TRY java.lang.String.charAt" |

1. 86. 调用String.split之前，未进行空值检查

Stack Overflow地址:http://stackoverflow.com/questions/23342401

关键方法："java.lang.String.split"

错误调用序列："java.lang.String.split"

正确调用序列"TRY java.lang.String.split"

1. 87. 调用Thread.sleep之后，未进行异常检查

Stack Overflow地址：<http://stackoverflow.com/questions/14048679>

关键方法："java.lang.Thread.sleep"

错误调用序列："java.lang.Thread.sleep"

正确调用序列"TRY java.lang.Thread.sleep"

1. 88. 调用URL.openConnection之后，未进行异常检查

Stack Overflow地址：http://stackoverflow.com/questions/21288292

关键方法："java.net.URL.openConnection"

错误调用序列："java.net.URL.openConnection"

正确调用序列"TRY java.net.URL.openConnection"

1. 89. 调用URLEncoder.encode之后，未进行异常检查

Stack Overflow地址：http://stackoverflow.com/questions/21353738

关键方法："java.net.URLEncoder.encode"

错误调用序列："java.net.URLEncoder.encode"

正确调用序列"TRY java.net.URLEncoder.encode"

## 第三部分，来源于bugram的数据

1. 90. org.quartz.Job.waitForCompletion可能抛出异常

关键方法："org.quartz.Job.waitForCompletion"

错误调用序列："org.quartz.Job.waitForCompletion"

正确调用序列"TRY org.quartz.Job.waitForCompletion"

1. 91. 调用Log.isDebugEnabled后，调用Log.info，导致权限不一致

关键方法： Log.isDebugEnabled

错误调用序列：

{" org.apache.commons.logging.Log.isDebugEnabled", org.apache.commons.logging.Log.info}

正确调用序列：

{" org.apache.commons.logging.Log.isDebugEnabled ",org.apache.commons.logging.Log.debug "}

## 第四部分，来源于论文A Dataset of Parametric Cryptographic Misuses

该数据集在MUBench的基础上扩展，见地址：<https://github.com/stg-tud/MUBench/pull/427>

1. 92. The first parameter of the constructor of IVParameterSpec is not properly randomized.

关键方法：javax.crypto.spec.IvParameterSpec.new

错误调用序列：

{ javax.crypto.spec.IvParameterSpec.new}

正确调用序列：

{"java.security.SecureRandom" javax.crypto.spec.IvParameterSpec.new"}

1. 93. Salt is not properly randomized while creating PBEParameterSpec object.

关键方法：javax.crypto.spec.PBEParameterSpec.new

错误调用序列：{ javax.crypto.spec.PBEParameterSpec.new}

正确调用序列：

{"java.security.SecureRandom" javax.crypto.spec.PBEParameterSpec.new"}

1. 94. PBEKeySpec(char[]) constructor is forbidden from being used as per CrySL rules, the constructor used should be PBEKeySpec(pw, salt, it, keylength);

关键方法：javax.crypto.spec.PBEKeySpec.new

错误调用序列：{ javax.crypto.spec.PBEKeySpec.new(char[])}

正确调用序列：

javax.crypto.spec.PBEKeySpec.new(char[],byte[],int,int)

1. 95. The first parameter in SecretKeySpec is a not properly generated key.

关键方法：javax.crypto.spec.SecretKeySpec.new

错误调用序列：javax.crypto.spec.SecretKeySpec.new

正确调用序列：

javax.crypto.SecretKeyFactory.getInstance-javax.crypto.spec.PBEKeySpec.new-javax.crypto.SecretKeyFactory.generateSecret-javax.crypto.SecretKey.getEncoded()-javax.crypto.spec.SecretKeySpec.new

1. 96. Third parameter while initializing the Cipher object was not properly preparedIV.

关键方法：javax.crypto.Cipher.init

错误调用序列：javax.crypto.Cipher.init(int,java.security.Key,java.security.spec.AlgorithmParameterSpec)

正确调用序列：javax.crypto.Cipher.init(int,java.lang.String,java.security.SecureRandom)

## 第五部分：来源于Survey调查

1. PrintWriter未关闭

关键方法：java.io.PrintWriter

错误调用序列：java.io.PrintWriter

正确调用序列：java.io.PrintWriter.close()

1. java.io.FileInputStream.new调用时，未对文件是否存在进行判断

关键方法：java.io.FileInputStream.new

错误调用序列：java.io.FileInputStream.new

正确调用序列：java.io.File.exists-java.io.FileInputStream.new

1. Set.iterator().next未判断Set中是否存在元素

关键方法：java.util.Iterator.next() //这里不考虑Iterator <>中的类型

错误调用序列：java.util.Set.iterator() java.util.Iterator.next()

正确调用序列：java.util.Set.iterator()-java.util.Iterator.hasNext()-java.util.Iterator.next()########java.util.Set.iterator()-java.util.Set.isEmpty-java.util.Iterator.next()

Size

1. （该缺陷去除）100. 连续两次调用Cipher.init，多余调用

关键方法：javax.crypto.Cipher.init

错误调用序列：javax.crypto.Cipher.init javax.crypto.Cipher.init

正确调用序列：javax.crypto.Cipher.init

1. 101. IRFactory.initFunction() is called twice along one possible execution

path, which causes an infinite loop.

关键方法：org.mozilla.javascript.IRFactory.initFunction

错误调用序列：org.mozilla.javascript.IRFactory.initFunction org.mozilla.javascript.IRFactory.initFunction

正确调用序列：org.mozilla.javascript.IRFactory.initFunction

1. [↑](#footnote-ref-1)