OpenRefactory

INTELLIGENT CODE REPAIR (iCR)

iCR for Java Supported Fixers
Release 2.0

This information sheet enumerates the currently supported set of Fixers for the **iCR for Java** Release 2.0 Analysis Engine. They are organized using Bug Categories with specific Fixers identified within that category.

OpenRefactory is constantly updating this list as new algorithms are developed for additional Fixers. Please contact OpenRefactory at info@openrefactory.com to stay current on available Fixers.

API Usage Issues (7):

Add Controller Class Restrictions –
A Spring `@Controller` class that uses `@SessionAttributes` have to call `setComplete()` on the `SessionStatus` object from an `@RequestMapping` method. This is specific to the Spring framework.

Add Component Package Location –
A class with annotation `@ComponentScan` should include all component (Service, Repository, Controller, RestController) packages. Otherwise, the classes will not be available in the Spring application context. This is specific to the Spring framework.

Add Default Package Restrictions –
The default package should not contain a class with `@ComponentScan`, `@SpringBootApplication`, or `@ServletComponentScan` annotation. This is specific to the Spring framework.

Block Serialization with Append –
An object output stream that is opened in append mode should not be serialized because the data will be stored in the wrong format and a deserialization attempt will result in an exception being thrown.

Replace Confusing Scope Combination –
Classes with annotation `@Controller`, `@Service`, or `@Repository` are singleton classes and if they have `@Scope` annotation, that should be explicitly specified. This is specific to the Spring framework.

Replace EnableAutoConfiguration with Import –
A class with annotation `@EnableAutoConfiguration`, should be replaced by `@Import`, as `@EnableAutoConfiguration` may include unnecessary beans which slows down an application. This is specific to the Spring framework.

Remove Method Call –
Remove unnecessary method calls that have been deprecated.

Arithmetic Issues (1):

Fix Zero Division –
Removes zero division opportunities in code. Fixes CWE 369, CERT Secure coding standard NUM02-J.

Bad Control Flow (4):

Add Missing Breaks –
Unexpected control flow because of missing `break` statements in `switch`. Fixes CWE 484.

Fix Equality Check –
Confusing object equality (equals method) with reference equality (== operator) and vice versa lead to inappropriate control flow, thus leading to hard-to-debug root causes. Perform appropriate equality checks according to the context. Fixes CWE 595, CWE 597, CERT Secure coding standard EXP03-J, CERT Secure coding standard EXP50-J.

Move Default Statement –
Switch statements should handle default case after everything else is handled.

Remove Unused Semicolon –
Remove unexpected control flow scenarios because of bad use of delimiters.

Broken Authentication (2):

Fix Hard-coded Key –
Cryptographic keys or other credentials should not be kept hard-coded in the source code. An attacker can extract the strings or byte arrays from an application source code or binary. Fixes CWE 798, OWASP A2-Broken Authentication, CERT Secure coding standard MSC03-J.

Fix Hard-coded Password –
User passwords should not be kept hard-coded in the source code. An attacker can extract the strings or byte arrays from an application source code or binary. Fixes CWE 259, OWASP A2-Broken Authentication, CERT Secure coding standard MSC03-J.

Concurrency Issues (4):

Avoid Value-Based Class Locks –
Synchronization should avoid value-based classes as locks. Fixes CERT Secure coding standard LCK01-J.

Avoid String and Boxed Primitive Locks –
Synchronization should avoid `Strings` and boxed primitives that can be reused. Fixes CERT Secure coding standard LCK01-J.

Remove Servlet Mutable Fields –
Make the instance fields of the servlet classes to be `static` or `final`, or remove them. The servlet container creates one instance of each servlet for each HTTP request and the threads will share the instance fields, leading to concurrency issues. Fixes CERT Secure coding standard MSC11-J.

Synchronize with Proper Class –
Synchronization should avoid using `getClass()` methods. Fixes CERT Secure coding standard LCK02-J.

Improper Access Control (2):

Get Proper Permission –
Get Proper Permission from the super `ClassLoader` if any class extends the `URLClassLoader`. Fixes CERT Secure coding standard SEC07-J.

Prevent Persistent Entity Short-circuiting –
Persistent objects annotated with `@Entity` or `@Document` should not be used as arguments in methods annotated with `@RequestMapping` and similar other annotations. This is specific to the Spring framework. Fixes CWE 915, OWASP A5-Broken Access Control Issue.

Improper Method Call (4):

Check Return Result –
Method `return` values that return error codes should be checked against error codes before being used. Fixes CWE 252.

Fix Finalize Method Implementation –
`finalize` method should be avoided or if used, called properly with reference to `Object.finalize()`. Fixes CWE 568, CERT Secure coding standard MET12-J.

Call Super Method –
Overriding methods should reference the method in the parent class.

Prevent Incompatible Transactional Calls –
Methods should not call same-class methods with incompatible "`@Transactional`" values.

Injection (8):

Prevent SQL Injection –
Constructing SQL queries with untrusted user provided data, e.g., URL parameters, enables attackers to inject code in place of data that changes the meaning of the SQL query. Identify potential SQL injection opportunities. Fixes CWE 20, CWE 85, CWE 943, OWASP A1-Injection Issue, CERT Secure coding standard IDS00-J.

Prevent Cross-site Scripting –
When endpoints reflect back tainted, user-provided data such as `POST` content, URL parameters, etc., it may allow attackers to inject code that will eventually be executed on the browser of a user. Identify potential SQL injection opportunities. Fixes CWE 79, CWE 80, CWE 81, CWE 82, CWE 83, CWE 84, CWE 85, CWE 86, CWE 87, OWASP A7-XSS.

Prevent Path Manipulation –
Constructing file system paths from untrusted user-provided data such as `POST` content, URL parameters, etc., enables attackers to inject specific path browsing symbols, such as "..", to manipulate the file path and to access files that they are not allowed to access otherwise. Identify potential path manipulation opportunities. Fixes CWE 22, CWE 23, CWE 36, CWE 99, CWE 641, OWASP A1-Injection, OWASP A5-Improper Access Control.

Prevent OS Command Injection –
Applications that execute operating system calls should not use untrusted user-provided data to create the command or command parameters. Identify potential OS command injection opportunities. Fixes CWE 77, CWE 78, CWE 88, OWASP A1-Injection.

Prevent XPath Injection –

Constructing `XPath` expressions using untrusted user-provided data such as `POST` content, URL parameters, etc., enables attackers to inject specially crafted values that change the way the expression is supposed to interpreted under normal circumstances. Identify potential `XPath` injection opportunities. Fixes CWE 643, OWASP A1-Injection, CERT secure coding standard IDS53-J.

Prevent LDAP Injection –

Constructing `LDAP` names or search filters using untrusted user-provided data enables attackers to inject values that change the way the name or the filter is supposed to interpreted under normal circumstances. Identify potential `LDAP` injection opportunities. Fixes CWE 90, OWASP A1-Injection, CERT secure coding standard IDS54-J.

Prevent Regular Expression Denial of Service –

Using external strings as regular expressions leads to potential denial of service attack since evaluating the regular expressions is CPU intensive. Identify potential regular expression injection opportunities. Fixes CWE 400, OWASP A1-Injection.

Prevent SQL Injection in Prepared Statement –

`Prepared Statement` is used to prevent SQL injection attacks. But, constructing SQL queries with string concatenation using untrusted user provided data, e.g., URL parameters, undermines the benefits of a `Prepared Statement`. Identify potential SQL injection opportunities. Fixes CWE 20, CWE 85, CWE 943, OWASP A1-Injection Issue, CERT Secure coding standard IDS00-J.

Null Pointer Issues (2):

Bad Return Value –

Methods with boxed type `return` values should not return `null`. Fixes CWE 476, CERT Secure coding standard EXP01-J.

Fix Null Dereference –

A pointer which has not been initialized is used as if it pointed to a valid memory area in the heap. A null pointer issue happens because the developer mistakenly did not allocate an object or has mistakenly assumed that that the object is allocated when in fact it is null. Fixes CWE 476, CERT Secure coding standard EXP01-J.

Object Visibility (2):

Add Qualifier for Static –

Access inherited static fields using the parent class as the qualifier.

Limit Field Access –

Accessibility of fields in Java classes should be limited. Fixes CWE 582, CWE 607, CERT Secure Coding Standard OBJ01-J and OBJ13-J.

Security Misconfiguration Issues (11):

Declare EJB Connectors Properly –

Following EJB 3.0 conventions, application security interceptors must be listed in the `ejb-jar.xml` file, or they will not be treated as default interceptors. Fixes OWASP A6-Security Misconfiguration Issue.

Remove Duplicate Validation Forms –

The names of Struts validation forms should be unique. When there are duplicate validation form names, the Struts Validator arbitrarily chooses one of the forms to use for input validation and discards the other. This is specific to the Struts framework. Fixes CWE 102, OWASP A6-Security Misconfiguration Issue.

Use Declared Filter –

Every filter declared in web.xml file should be used in an element. Otherwise such filters are not invoked. Fixes OWASP A6-Security Misconfiguration Issue.

Limit Scope of Maven Dependencies –

System dependencies in Maven are sought at a specific path that matches a configuration and cannot be ported. If an artifact is deployed in an environment that is different from the original configuration, the build would fail.

Track Messages During Restart –

In Spring, `DefaultMessageListenerContainer` is implemented as a Java Message Service (JMS) polling component. While the Spring container is going through a restart/shut down, the message listener may discard messages and will therefore lose them. The message listener container should be declared such that the messages are not discarded. This is specific to the Spring framework.

Allow Automatic Connection Recovery –

Spring framework uses a factory object (`SingleConnectionFactory`) that returns the same connection for all connection requests. This should be declared with a setting to allow automatic recovery when the connection goes bad. This is specific to the Spring framework.

Stop Debugging Web Remoting –

Direct Web Remoting (DWR) is a Java and JavaScript library that enables RPC calls in an Ajaxian application. If the debug mode of DWR is turned on, it will allow users to access information exposed under the debugging servlet. This is specific to the DWR library.

Remove Duplicate Servlet Definition –

When a deployment descriptor contains the same name for multiple servlets, only the first one is deployed, and the others are ignored.

Avoid GET Mix –

Spring framework annotations should use `HTTP GET` method and not mix it with other methods. Fixes CWE 352, OWASP A6-Security Misconfiguration Issue.

Use Proper Request Mapping –

Ensures that Proper Request Mappings are used. Fixes CWE 352, OWASP A6-Security Misconfiguration Issue.

Annotate Public Method –

Spring framework annotations should be accompanying a public method.

Sensitive Data Exposure (4):

Replace Random Generator –

Weak random number generator should be replaced with a strong random number generator. Fixes CWE 330, CWE 332, CWE 336, CWE 337, OWASP A6-Security Misconfiguration Issue, Secure coding standard MSC63-J.

Remove Weak Seed –

A strong random number generator does not need a seed value to be set. Setting the seed with a constant or a predictable value will weaken the random generator itself. If a seed value is set explicitly, it should be removed. Fixes CWE 337, OWASP A6-Security Misconfiguration Issue, CERT secure coding standard MSC63-J.

Use Proper Dependency Injection –

Non-static members and constructors in classes with annotation `@Controller`, `@Service`, or `@Repository` should have proper annotations (any one of `@Autowired`, `@Value`, `@Resource`, or `@Inject`). This is specific to the Spring framework. Fixes OWASP A3-Sensitive Data Exposure Issue.

Prevent XML eXternal Entity –

XML Document Type Definition (DTD) should be disabled to prevent information disclosure via XXE attacks. Fixes CWE 611, CWE 827, OWASP A4-XML eXternal Entities.

Weak Cryptography Issues (6):

Use Strong Password Encoder –

Authentication manager should use a strong password encoder. This is specific to the Spring framework. Fixes CWE 327, CWE 328, OWASP A2-Broken Authentication Issue, OWASP A6-Security Misconfiguration Issue.

Use Strong Hash Function –

Hashing should be done using strong hashing algorithm such as SHA-256 or SHA-3. Fixes CWE 327, CWE 328, OWASP A6-Security Misconfiguration Issue.

Use Secure Socket Protocol –

SSL context objects should use a secure socket protocol such as TLS or DTLS. Fixes CWE 326, CWE 327, OWASP A3-Sensitive Data Exposure, OWASP A6-Security Misconfiguration Issue.

Restrict Access to Broadcast Receiver –

While registering broadcast receivers in Android, broadcast permission should be specified so that a receiver only receives broadcasts sent by components having proper permission. This is specific to Android applications. Fixes CWE 925, OWASP A1-Injection.

Restrict Access to Broadcast Sender –

While sending broadcast messages in Android, broadcast permission should be specified so that only receivers with proper permission can receive it. This is specific to Android applications. Fixes CWE 927, OWASP A3-Sensitive Data Exposure.

Prevent Sharing of User Preferences –

Android `getPreferences` and `getSharedPreferences` should use private mode when invoked, so that the preferences are not exposed globally. Fixes OWASP A3-Sensitive Data Exposure.