**Code Correctness: Comparison of Boxed Primitive Types**

Comparing boxed primitives using equality operators instead of their equals() method can result in unexpected behavior.

**Explanation**

When dealing with boxed primitives, when comparing equality, the boxed primitive's equals() method should be called instead of the operators == and !=. The Java Specification states about boxing conversions:

"If the value p being boxed is an integer literal of type int between -128 and 127 inclusive, or the boolean literal true or false, or a character literal between '\u0000' and '\u007f' inclusive, then let a and b be the results of any two boxing conversions of p. It is always the case that a == b."

This means that if a boxed primitive is used (other than Boolean or Byte), only a range of values will be cached, or memoized (*in computing, memoization or memoisation is an optimization technique used primarily to speed up computer programs by storing the results of expensive function calls and returning the cached result when the same inputs occur again*). For a subset of values, using == or != will return the correct value, for all other values outside of this subset, this will return the result of comparing the object addresses.

**Example**

The following example uses equality operators on boxed primitives.

Integer mask0 = 100;

Integer mask1 = 100;

...

if (file0.readWriteAllPerms){

mask0 = 777;

}

if (file1.readWriteAllPerms){

mask1 = 777;

}

...

if (mask0 == mask1){

//assume file0 and file1 have same permissions

...

}

...

**Recommendation**

For most cases, developers do not want to compare the boxed primitives' addresses, so the boxed primitives' equals() method should be used instead of equality operators. Alternatively the equation can be rewritten to use other comparison operators, such as <, >, <=, or >=, since these cause automatic unboxing of the primitive values.

**References**

* http://www.hpenterprisesecurity.com/vulncat/en/vulncat/java/code\_correctness\_comparison\_of\_boxed\_primitive\_types.html