Software Assurance Tips

A product of the Software Assurance Tips Team[1]

Kevin Keen

Monday 17th October, 2022

1 Java, Inner Classes, and Checkmarx Unused Variable Findings

Updated Wednesday 11th February, 2026

Java has some odd, non-intuitive rules when it comes to the visibility of inner classes. One of which is that an inner class not marked `private` will be accessible to other classes in the same package. This can lead to some odd code that Checkmarx doesn't handle well, especially when the inner class is a POD (plain old data) class.

A good argument can be made that Java applications really shouldn't have many, if any, POD classes. After all, data exists to be operated on, and if there is behavior defined somewhere to operate on this data, why not have it live in the same place? Nevertheless, POD classes do tend to creep their way into some Java applications, particularly if the programmer hails from a C background and is accustomed to working with structs. While avoiding POD classes might prevent the issue we are about to highlight, we will set the POD v. no-POD debate aside for the time being.

When an inner class is not declared `private`, it is entirely possible for members of that class to be unused locally, in any of its sibling classes, or in the enclosing class. Instead, because of the package level visibility, the use may reside in a completely different class. Take, for example, a class `Outer`, which contains a private (no public, protected, etc) class `Inner`. A 3rd unrelated class could then contain code like that shown in Listing 1.

```
(In class Unrelated)
Outer myOuter = new Outer();
Inner myInner = myOuter.new Inner();
if(myInner.thoughtItWasUnused == 5) {...

}
```

Listing 1: Tertiary Class with Unused Inner Variables

Unfortunately, in these cases, Checkmarx is likely to flag the members of `Inner` as unused variables unless they are used by either the `Inner` class itself or perhaps by `Outer`. In this example, the member `thoughtItWasUnused` may have been flagged in `Outer.java` as an unused variable.

References

[1] Jon Hood, ed. SwATips. `https://www.SwATips.com/`.