# CWE Detail – CWE-1235

## Description

The code uses boxed primitives, which may introduce inefficiencies into performance-critical operations.

## Extended Description

Languages such as Java and C# support automatic conversion through their respective compilers from primitive types into objects of the corresponding wrapper classes, and vice versa. For example, a compiler might convert an int to Integer (called autoboxing) or an Integer to int (called unboxing). This eliminates forcing the programmer to perform these conversions manually, which makes the code cleaner. However, this feature comes at a cost of performance and can lead to resource exhaustion and impact availability when used with generic collections. Therefore, they should not be used for scientific computing or other performance critical operations. They are only suited to support "impedance mismatch" between reference types and primitives.

## Threat-Mapped Scoring

Score: 1.9

Priority: P3 - Important (Medium)

## Modes of Introduction

**•** Implementation: The programmer may use boxed primitives when not strictly necessary.

## Common Consequences

**•** Impact: DoS: Resource Consumption (CPU), DoS: Resource Consumption (Memory), DoS: Resource Consumption (Other), Reduce Performance — Notes: Incorrect autoboxing/unboxing would result in reduced performance, which sometimes can lead to resource consumption issues.

## Potential Mitigations

**•** Implementation: Use of boxed primitives should be limited to certain situations such as when calling methods with typed parameters. Examine the use of boxed primitives prior to use. Use SparseArrays or ArrayMap instead of HashMap to avoid performance overhead. (Effectiveness: N/A)

## Applicable Platforms

**•** Java (Class: None, Prevalence: Undetermined)

**•** C# (Class: None, Prevalence: Undetermined)

## Demonstrative Examples

**•** In the above loop, we see that the count variable is declared as a boxed primitive. This causes autoboxing on the line that increments. This causes execution to be magnitudes less performant (time and possibly space) than if the "long" primitive was used to declare the count variable, which can impact availability of a resource.

**•** N/A