**Code Correctness: String Comparison of Float Development Mitigation**

**SOP**

Code correctness vulnerabilities occur when an Object API is not used properly or as intended. Code correctness vulnerabilities can occur when comparing a floating-point value with a String object.

**Defense Against Code Correctness: String Comparison of Float**

To compare a floating-point value to a String object, it must be changed to a String object first. When converting to a String, it is important to be aware of the type and value of the floating-point variable. If converted to a String object, it could be “NaN”, “Infinity”, “-Infinity”, have a certain amount of trailing decimal places containing zeroes, or may contain an exponent field. The representation may also greatly change if it is converted to a hexadecimal String.

**Examples**

**General Example**

…

int initialNum = 1;

…

String resultString = Double.valueOf(initalNum/10000.0).toString();

if(s.equals(“0.0001”)){

//do something

…

}

…

**Explanation**

The example above shows the comparison of a floating-point variable with a String.

**Resources**

1. [HP Enterprise Security – Code Correctness: String Comparison of Float](https://vulncat.fortify.com/en/detail?id=desc.dataflow.java.code_correctness_string_comparison_of_float#Java%2fJSP)