**Code Correctness: toString on Array Development Mitigation SOP**

Code correctness vulnerabilities occur when an Object API is not used properly or as intended. Code correctness vulnerabilities can occur on a toString() on an Array because a call to toString()indicates a developer is intending to return the contents of the array as a String. However, a direct call to toString() on an array will return a string value containing the array’s type.

**Defense Against Code Correctness: toString on Array**

The Array.toString() method was introduced in Java 5 that allows the return of a string representation of the array contents in comma delimited format.

**Example**

String[] strList = new String[5];

…

System.out.println(strList.toString());

**Explanation**

The code above will output [Ljava.lang.String;@1232121*.* The toString() method on an array will not achieve the desired results of printing out the contents of the array. It will instead print a string value containing the array’s type and hash code in memory.

**Recommendation**

The code below shows how to return a string representation of the array contents in comma-delimited format:

String[] strList = new String[5];

…

System.out.println(Arrays.toString(strList));

**Resources**

1. [HP Enterprise Security – Code Correctness: toString on Array](https://vulncat.fortify.com/en/detail?id=desc.structural.dotnet.code_correctness_tostring_on_array#C%23%2fVB.NET%2fASP.NET)