**Portability Flaw: Locale Dependent Comparison Development Mitigation SOP**

When comparing data that may be locale-dependent, an appropriate locale should be specified.

**Defense Against Locale Dependent Comparison**

To prevent this from occurring, always make sure to either specify the default locale, or specify the locale with APIs that accept them such as toUpperCase().

**Example**

The following example tries to perform validation to determine if user input includes a <script> tag.

...

public String tagProcessor(String tag){

if (tag.toUpperCase().equals("SCRIPT")){

return null;

}

//does not contain SCRIPT tag, keep processing input

...

}

...

The problem with the above code is that java.lang.String.toUpperCase() when used without a locale uses the rules of the default locale. Using the Turkish locale "title".toUpperCase() returns "T\u0130TLE", where "\u0130" is the "LATIN CAPITAL LETTER I WITH DOT ABOVE" character. This can lead to unexpected results, such as in Example 1 where this will prevent the word "script" from being caught by this validation, potentially leading to a Cross-Site Scripting vulnerability.

**References**

1. [HP Enterprise Security - Portability Flaw: Locale Dependent](http://www.hpenterprisesecurity.com/vulncat/en/vulncat/java/portability_flaw_locale_dependent_comparison.html)