**Code Correctness: Class Does Not Implement Equals Development Mitigation SOP**

Code correctness vulnerabilities occur when an Object API is not used properly or as intended. Code correctness vulnerabilities for a class not implementing equals are detected by Fortify because the Equals() method is called on an object that does not implement that method. The Equals() method is typically used to compare properties of objects, however calling *Equals()* on a class (or any super class/interface) does not explicitly implement Equals() results in a call to the Equals() method inherited from System.Object. Object.equals()compares two object instances to see if they are the same instead of comparing object member fields or other properties. This is typically an indication of buggy code.

**Defense Against Code Correctness: Class Does Not Implement Equals**

First, check that Object.equals() is really the method that should be called and if it’s not, implement an equals() method or use a different method for comparing objects.

**Examples**

**General Example**

public class AccountGroup {

private int gid;

public int getGid() { return gid; }

public void setGid( int newGid ) { gid = newGid; }

}

public class CompareGroup {

public boolean compareGroups( AccountGroup group1, AccountGroup group2 ) {

**return group1.equals( group2 );**

}

}

**Explanation**

In the code bolded above, equals() is not implemented in AccountGroup*.*

**Recommendation**

The following code implements an equals() method to the AccountGroup class to effectively compare the AccountGroup objects:

public class AccountGroup {

private int gid;

public int getGid() { return gid; }

public void setGid( int newGid ) { gid = newGid; }

**public boolean equals( Object o ) {**

**if( !( o instanceof AccountGroup ) ) { return false; }**

**AccountGroup other = ( AccountGroup ) o;**

**return ( gid == other.getGid() );**

**}**

}

**Example**

@Override

public Boolean supports(Class<?> clazz) {

return clazz.equals(Address.class) ? true : false;

}

**Explanation**

The code above is recognized by Fortify as a true finding; however, this would be a false finding because class types are being compared. Because the same 2 objects are being compared, this would be implemented correctly.

**Resources**

1. [HP Enterprise Security – Code Correctness: Class Does Not Implement Equals](https://vulncat.fortify.com/en/detail?id=desc.structural.dotnet.code_correctness_class_does_not_implement_equals#C%23%2fVB.NET%2fASP.NET)