**Object Model Violation: Erroneous clone() Method Development Mitigation SOP**

The erroneous clone() method vulnerability comes from a clone() method that is not calling super.clone() to obtain the new object, otherwise the subclass’s clone() method will return an object of the wrong type.

**Defense Against Object Model Violation: Erroneous clone() Method**

The new object should be obtained by calling super.clone() is ensure an object of the correct type is ensured.

**Example**

public class Kibitzer implemented Cloneable {

public Object clone() throws CloneNotSupportedException {

Object returnMe = new Kibitzer();

…

}

}

public class FancyKibitzer extends Kibitzer implement Cloneable {

public Object clone() throws CloneNotSupportedException {

Object returnMe = super.clone();

…

}

}

**Explanation**

The above code shows how not calling super.clone() can return an object of the wrong type. Because Kibitzer calls new Kibitizer() rather than super.clone(), it will cause FancyKibitizer to return an object of type Kibitizer rather than FancyKibitizer.

**Recommendation**

To implement the correct fix, Kibitzer should call super.clone()in the clone() method instead of new Kibitzer().

**References**

1. [HP Enterprise Security – Object Model Violation: Erroneous clone() Method](http://www.hpenterprisesecurity.com/vulncat/en/vulncat/java/object_model_violation_erroneous_clone_method.html)