**Weak SecurityManager Check: Overridable Method Development Mitigation SOP**

Weak SecurityManager check vulnerabilities from overridden methods occur because non-final methods that perform security checks can be overridden in ways that bypass security checks. When a method is overridden by a child class, that class can bypass security checks in the parent class.

**Defense Against Weak SecurityManager Check: Overridable Method**

Any methods that perform security operations, such as SecurityManager or AccessController, should be declared as final classes or the methods themselves should be declared final.

**Example**

public class BadSecurityCheck {

private int id;

public BadSecurityCheck() {

doSecurityCheck();

id = 1;

}

public void doSecurityCheck() {

SecurityManager sm = System.getSecurityManager();

if(sm != null) {

sm.checkPermission(new SomePermission(

“SomeAction”));

}

}

}

**Explanation**

The code above performs a security check in doSecurityCheck(), but can be overridden by a child class.

**Recommendation**

The code below declared the class GoodSecurityCheck as final so none of its methods can be overridden:

public final class GoodSecurityCheck {

private int id;

public GoodSecurityCheck() {

doSecurityCheck();

id = 1;

}

protected void doSecurityCheck() {

SecurityManager sm = System.getSecurityManager();

if(sm != null) {

sm.checkPermission(new

SomePermission(“SomeAction”));

}

}

}

**References**

1. [HP Enterprise Security – Weak SecurityManager Check: Overridable Method](http://www.hpenterprisesecurity.com/vulncat/en/vulncat/java/weak_securitymanager_check_overridable_method.html)