**Poor Error Handling: Throw Inside Finally Development Mitigation SOP**

Poor error handling through throws inside a finally block makes the application vulnerable by disrupting the normal program. The logical progression of a try-catch-finally block is broken when using a throw statement inside a finally block. Finally blocks in Java are always executed after their corresponding try-catch blocks because the finally blocks are typically used to free allocated resources, such as file handles or database cursors. This can bypass the necessary cleanup code because normal program execution will be disrupted.

**Defense Against Poor Error Handling: Throw Inside Finally**

Do not throw exceptions from within finally blocks. If an exception must be re-thrown, it should be done inside the catch block so the normal execution of the finally block is not interrupted.

**Example**

public void processTransaction( Connection conn ) throws FileNotFoundException {

FileInputStream fis = null;

Statement stmt = null;

try {

stmt = conn.createStatement();

fis = new FileInputStream( “baseFile.text” );

…

} catch ( FileNotFoundException fe ) {

log( “File not found.” );

} catch ( SQLException se ) {

// handle error

} finally {

if( fis == null ) {

throw new FileNotFoundException();

} if( stmt != null ) {

try {

**stmt.close();**

} catch ( SQLException e ) {

log(e);

}

}

}

}

Explanation:

1. The code above would bypass *stmt.close()* (bolded above) when the *FileNotFoundException* is thrown
2. The code below re-throws the *FileNotFoundException* in the catch block (bolded below) to avoid any piece of code being bypassed:

public void processTransaction( Connection conn ) throws FileNotFoundException {

FileInputStream fix = null;

Statement stmt = null;

try {

stmt = conn.createStatement();

fis = new FileInputStream( “badFile.txt” );

…

} **catch( FileNotFoundException fe ) {**

**log( “File Not Found.” );**

**throw fe;**

**}** catch ( SQLException se ) {

// handle error

} finally {

if( fis == null ) {

try {

fis.close();

} catch ( IO Exception ie ) {

log( ie );

}

}

if( stmt != null ) {

try {

stmt.close();

} catch( SQLException e ) {

log( e );

}

}

}

}

**References**

1. [HP Enterprise Security – Poor Error Handling: Throw Inside Finally](http://www.hpenterprisesecurity.com/vulncat/en/vulncat/java/poor_error_handling_throw_inside_finally.html)