SWE 681 – Secure Software Engineering

Dr. David A. Wheeler

Minor Assignment | Due 26th Feb 2014

Shaeq Khan

Source	National Institute of Standards and Technology [NIST] link
Test Case ID	146967
Contributor	SAMATE Team Staff
Language	Java
Date	05/21/13

Where, Why and What kind of vulnerability is this?

It is a failure to sanitize data within a SQL Query (SQL Injection) in the Source Code.

```
String data = ""; /* Initialize data */
File file = new File("C:\\data.txt");
FileInputStream streamFileInput = null;
InputStreamReader readerInputStream = null;
BufferedReader readerBuffered = null;

try
{
    /* read string from file into data */
    streamFileInput = new FileInputStream(file);
    readerInputStream = new InputStreamReader(streamFileInput, "UTF-8");
    readerBuffered = new BufferedReader(readerInputStream);

    /* POTENTIAL FLAW: Read data from a file */
    /* This will be reading the first "line" of the file, which
     * could be very long if there are little or no newlines in the file
     */
    data = readerBuffered.readLine();
}
```

```
.
.
.
Connection dbConnection = null;
Statement sqlStatement = null;

try
{
     /* IO is the helper class to connect to the database */
     dbConnection = IO.getDBConnection();
     sqlStatement = dbConnection.createStatement();

     /* POTENTIAL FLAW: data concatenated into SQL statement used in
      * executeUpdate(), which could result in SQL Injection */
     int rowCount = sqlStatement.executeUpdate("insert into users (status)
     values ('updated') where name='"+data+"'");

     IO.writeLine("Updated " + rowCount + " rows successfully.");
}
```

How can this be fixed?

A bad source for the variable data is to read from a file C:\\data.txt.

A good source for the variable data is as a hard coded string.

The problem is that content of variable “data” concatenated into SQL statement used in executeUpdate() could result in SQL Injection.

The solution is to use prepared statement and executeUpdate() properly.

```
String insertStatement =
     “insert into users (status) values ('updated') where name=? ";

PreparedStatement preparedStatement =
       dbConnection.prepareStatement(insertStatement);

preparedStatement.setString(1,data);
int rowCount = preparedStatement.executeUpdate();
```