Security is an important issue in the development of software applications/systems. Considering the most common issues a piece of code might face; we can pre-emptively prepare security tests to perform on our code. Whilst this is a substantial decrease in time taken, it can be made quicker and more efficient with the use of Security plugins like SpotBugs.

**Untrusted Query String/SQL Injection:**

One of the most prevalent issues a system that interacts with a database can face is a SQL Injection. What this basically means is that a user can input an SQL query within one of the accepted input fields and use this method to trick the database into dumping its data into a readable format for the unauthorized user to abuse.

Common fixes for this include reversing ‘If’ statements, rewriting SQL queries with the intent to remove potential injections through ‘Stored Procedures’, etc. An example is displayed below;

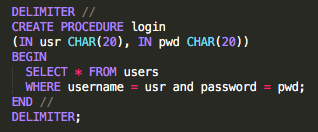


Figure 1: Stored Procedures

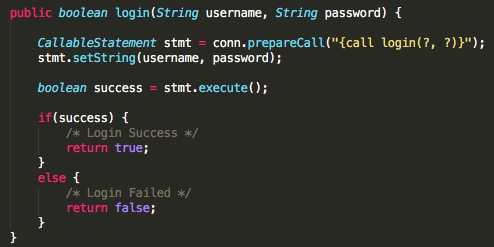


Figure 2: revised login with Stored Procedures

**Avoid Java Serialization:**

Typically, when dealing with serialization our data is either taken from disk or saved to disk. This can become a serious security issue as it’s impossible to know what’s in a byte object until it’s deserialized. Say that an unauthorized user was to serialize something to infiltrate our system. It would be impossible to know, or defend against, this attack unless we work around this.

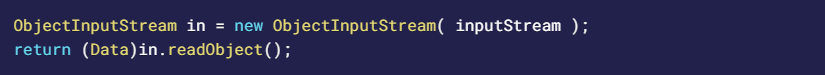


Figure 3: An example of bad security practices

The best practice found through research is to indirectly remove deserialization altogether. Whilst Oracle has plans for this, it is not yet implemented. As a result, if you have to implement serializable objects in your application, then it is best to implement the ‘readObject()’ class as shown below.

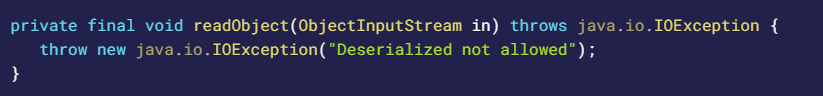


Figure 4: The current best practice for avoiding deserialization security issues

**References:**

<https://logz.io/blog/defend-against-sql-injections/>

<https://snyk.io/blog/10-java-security-best-practices/>