Jan Rehwaldt
University of Tartu, Secure Programming Techniques, April 2012

Exercise 2: We want to protect a web application against SQL injection attacks. Let's assume that we cannot modify the application itself but the environment around it is under our control (network, server, OS, web server, SQL database server). What different methods can we use to secure the application against SQL injection attacks?

On database level **proper access rights** and user should be created. Applications should generally not have the right granted to drop or alter tables and rights should be limited to only the required operations.

Some programming environments, such as Java, may provide the possibility to **disallow multiple queries within one statement**. In this case it is not possible to append additional statements to end of the initial query. As example consider the query "SELECT * FROM users WHERE name = `" + param + "`" with param set to "`; DELETE FROM users WHERE name LIKE `%", which would result in an ignored select statement and secondly delete all user data.

User inputted data on websites may be filtered. If not by the application, the webserver could provide basic filters, even though an in-build filter in the software is more convenient. Additionally characters can be escaped by the webserver.

Run the database (*and the webserver*) **not as root** or Administrator, as remote users may trigger operations through SQL injection, which can affect the operating system.

Monitor the servers (database etc.) for inconsistencies and block attacker IPs if attacks were recognized. Some blockings may be done automatically according to the number of requests (e.g., tries of form submission) per minute.

Sources:

- Brain
- Slides of Secure Programming Techniques, Meelis Roos