**Injection**

1. **Discussion**

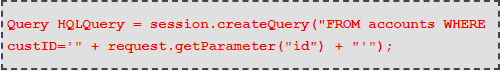
Injection flaws permits the attackers to inject a code or command that is written in other programming languages via shell commands and in sql without authorization. Injection vulnerability occurs every time an application uses an external interpreter. Once the injection was successfully installed in your system, it can control your operating system and crash your database. An example of injection is the SQL injection, the occurrence of injection vulnerability is also because of the untrusted data being filled up by the attackers through a form and pass through the database. In that way, they may be able to modify the id parameters and insert their own commands and after successfully injecting the commands, they can now access or damage your data. Administration operations in a database may be able to perform by the attackers. In addition to that, injection can also be found in other sql databases like NoSQL, LDAP, and XPath.

1. **Prevention**

* There are different ways to prevent this kind of risk. First, avoid using external interpreters. Alternatively, you can use  libraries that functions the same as external interpreters just to lessen the vulnerability of injection.
* Second, keep track of your data by ensuring that the data is separate from the commands and queries. In that way, the attackers will not be able to modify the parameter id. Instead, just use a safe API that uses prepared statements because that is already compiled and it is not possible for the attackers to modify it.
* Third, makes use whitelist input validation for special characters to still allow the users for special character inputs.

1. **Example**

This example shows how does the attacker inputs a malicious data passed on a query using the id parameter. In this example, the attackers changes the query by adding the “SELECT \* FROM items WHERE ‘a’=’a’;” query. This will change the output of the queries.

****

**https://lh6.googleusercontent.com/KicqC1hz0YjCHwJ01zsRvQ4SgOfY_IOtVyLHMv6hFTz71Dyh3HvY4d1Fp4TF2q0naui-dz4S4QfpKfs1N9LcUaHbPzzdETSoH2wTZ1oE7igp3Cmcgm0x7V9E-nIObrJ2h4-H6ovw**