This example requires the creation of a user table (use the SQL Injection Example Setup script).

The example has two string variables for username and password. The first example of password is how a programmer would expect users to enter their password. NOTE: typically a user enters a password and the password is hashed to compare to a HASHED password in a database. This example shows a BAD example of storing passwords in plain text in a database (i.e., NEVER do this!).

Having said that we can pretend that the password is the hashed version and the password column in the database contains hashed passwords – the injection attack works even when passwords are hashed.

Note that the first password returns nothing. There are no username/password combinations that match what was provided.

However, the second (malicious) password entry returns the entire user table.

Things to learn from this

* NEVER store passwords in plain text
* Protect your database schema design – once hackers know the schema then they know what tables to target
* Protect against SQL injection
  + Avoid dynamic SQL unless absolutely necessary
  + IF dynamic SQL is required then use parameterize queries (do not just build them using concatenation).