CIS 655 Assignment 5
Database Stored Procedures, Security and Administration

Instructions:

1. Login to your assigned server and change the administrator password!!!

2. Create the DB and successfully insert data using the SQL_INJECTION.txt file found on your server. Follow the instructions posted on RamCT and ensure that the class server "login" and "products" web pages interact properly with your local CIS_ZZZ database (the connection will use "stephen" SQL account with password "cis655"). Background reading from this document SQL_Injection_Attacks.pdf.

   Rubric (15 points): Browse to http://10.6.55.20/new/login_form.asp (with valid credentials) and http://10.6.55.20/new/products.asp?IP=XX&productID=2 (1 or 3). Check that the pages load and retrieve the proper data from the DB. Try all the combinations to check that the pages work properly (try valid and invalid data).

3. Secure the DB from SQL Injection as discussed, by the use of a defensively coded stored procedure (like the example shown or posted). The stored procedures must be named "sp_GetUser" (with 2 parameters – username, userPass) and "sp_GetProduct" (with 1 parameter - id). Apply increased security user privileges to the "stephen" account.

   Rubric (30 points): Browse to http://10.6.55.20/new/login_form_sp.asp (with valid credentials) and http://10.6.55.20/new/products_sp.asp?IP=XX&productID=2 (1 or 3).

   We will grade this by:
   a. *Attempt to "inject" databases by inserting "**' or 1=1;--**" in the username box, or "**productID=0%20or%201=1**" in the URL. Use other combinations from the SQL injection cheat sheet…*
   b. *running the sqlmap injection software on 10.6.55.25 ("stephen"and "C1s655!")*
   c. *login to your server (with either the "stephen" account or our admin account), and verify the stored procedures*

4. As demonstrated in class, use the proper pieces of the other provided script file to create tables and stored procedures in order to manage client and credit card data. Import the provided client and credit card data into the proper tables, handling any "dirty" data. Ensure that a stored procedure called "sp_RetrieveCC" (taking a clientID as a parameter) returns all the credit cards associated with that client. Secure this SP from injection.

   Rubric (35 points): Browse to "http://10.6.55.20/new/RetrieveCC_Form.asp" (retrieveCC.asp) pages and retrieve CC data. Correct hashes and decrypted CC must be displayed in the correct format, i.e., matching that of server 10.6.55.20 We will also login and verify the table data.