Lab 4

Database Design for Secure Software Applications

Overview:
In this lab, you will demonstrate how to identify database code that is susceptible to SQL Injection and make recommendations to fix the code. You will also review an application for privacy violations and make recommendations for improvement.

Existing Application Description:
An existing PHP-based application connects to a database providing functionality for retrieving table information for a specific employee. The application also has the ability to update specific user fields. In both cases, a Web-based form is used allowing a user to search for an employee and then either display their employee information or update it.

The web-form display functionality asks the user to enter a specific employee ID and then retrieves the following fields:

- Employee_id,
- firstname,
- lastname,
- salary,
- birthdate,
- SSN,
- phonenumber,
- address,
- email,
- nickname,
- Password

The update form allows the user to enter a specific employee ID and then update most of the fields in the list above.

As you look at this list, you should definitely be concerned. Part of this exercise will be to identify those concerning fields and make recommendations for changes. However, you were also given access to the PHP code and flagged the following code as being suspect.

```
$conn = getDB();
$sql = "SELECT id, firstname, lastname, salary, birth, ssn,
phonenumber, address, email, nickname, Password
FROM data
WHERE id= '$input_id' and password='$input_pwd'";
$result = $conn->query($sql))
```

Figure 1. PHP Retrieve Data code

```
$conn = getDB();
$sql = "UPDATE data
SET nickname='$nickname',
```

```
email='$email',
address='$address',
phonenumber='$phonenumber',
Password='$pwd',
Firstname = '$firstname',
Lastname = '$lirstname'
WHERE id= '$id' ";
$conn->query($sql))
```

Figure 2. PHP Update Data code

Lab Requirements

1. As you review the overall functionality of the application. What concerns do you have with the data and display of the data. For example, are there any privacy concerns? What are best practices for protecting private data? What are best practices for changing a password? Note: you can ignore the SQL Injection issues in this discussion as we will be addressing that in a separate requirement. **(30 points)**
2. Now that your concerns about the application have been documented, what specific recommendations do you have to address your concerns? Be specific with your recommendations. You should consider items such as using roles to restrict access, limiting access to the form, encrypting fields, applying security controls and others. Discuss if the recommended changes will impact the functionality of the application and if so why that is Okay. **(30 points)**
3. Review the code and determine is SQL Injection is a problem. Describe specifically how you would test to see if SQL injection was a problem. For example, show exactly what you would input into the form fields to determine if SQL injection was a problem. Discuss the code issues pointing to where the SQL injection could happen. **(30 points)**
4. Now, that you have identified the code issues, describe what you need to do to fix it. You don't have to rewrite the code, but you do need to provide specific mitigation strategies, explain why the strategy works**. (30 points)**
5. The file should be well-organized, well-written using full sentences and paragraphs and include all required components. **(30 points)**

Deliverables:

One word (or PDF) document should be provided with your name, date, course information (e.g. SDEV 350 6380) and professor name fulfilling all of the requirements listed above.