**Fixing Payment Card Industry Java Web Application**

Benjamin M. Brandhorst

University of Maryland Global Campus

SDEV 425 7980 Mitigating Software Vulnerabilities

Doctor Ronald McFarland

May 10th, 2020

**Part 1: Sample Java Application**

I was able to import the java application and configure it to run correctly. Below is a breakdown of the steps taken to accomplish this task:

Figure 1 shows the successful import of the sample project.

A screenshot of a cell phone

Description automatically generated

**Figure 1 – Sample project import**

I setup and configured the database and used the SQL script included with the sample project files to populate the database (see Figure 2).

A screenshot of a cell phone

Description automatically generated

**Figure 2 – Database configuration and population**

When the “Run Project” button is clicked, the web browser is launched and directed to the index page (Figure 3).

A screenshot of a social media post

Description automatically generated

**Figure 3 – Index page**

Selecting “Sign In” brings us to a Login page where credentials can be input (Figure 4)

A screenshot of a cell phone

Description automatically generated

**Figure 4 – Login page**

Upon a successful login we are brought to a landing page that gives a run-down of what functions can be performed with the application (Figure 5)

A screenshot of a cell phone

Description automatically generated

**Figure 5 – Login landing page**

Selecting “Your Account” will bring you to a page which lists all of the logged in user’s account information (Figure 6)

A screenshot of a cell phone

Description automatically generated

**Figure 6 – Account Data**

Clicking the “Sign Out” button gives a message thanking the user for visiting (Figure 7).

A screenshot of a cell phone

Description automatically generated

**Figure 7 – Sign Out**

Finally, clicking the “Home” button brings us back to the initial page (Figure 8).

**A screenshot of a cell phone

Description automatically generated**

**Figure 8 – Home page**

**Part 2: Payment Card Industry (PCI) Compliance**

Requirement 3: Protect stored cardholder data has been violated. Specifically, 3:2 which says:

***Do not store sensitive authentication data after authorization (even if encrypted). If sensitive authentication data is received, render all data unrecoverable upon completion of the authorization process. It is permissible for issuers and companies that support issuing services to store sensitive authentication data if:***

* ***There is a business justification and***
* ***The data is stored securely.***

Sensitive authentication data includes full track data, card validation code or value, and PIN data. The Your Account function returns all of this information (Figure 9). To bring this application into compliance with PCI requirement 3:2, this information needs to be removed from the database.

**A screenshot of a cell phone

Description automatically generated**

**Figure 9 – Your Account data**

To remedy the noncompliance issue, the CustomerAccount table and CustomerAccount insert statements in the SQL script are modified to remove the noncompliant information. Figure 10 shows the modifications to the CustomerAccount table. Figure 11 shows the modifications made to the insert statements. Figure 12 shows the account.jsp changes to remove the noncompliant fields.

A screenshot of a cell phone

Description automatically generated

**Figure 10 – Modified CustomerAccount table**

**A screenshot of a cell phone

Description automatically generated**

**Figure 11 – Modified CustomerAccount insert**

**A close up of a sign

Description automatically generated**

**Figure 12 – Modified account.jsp**

We also must update ShowAccount.java to remove the no longer necessary variables (Figure 13). Then it is on to updating processRequest to remove references to variables which no longer exist (Figure 14). From there we also delete variable references from getData (Figure 15). Afterwards we remove unused fields from the table (Figure 16).

**A picture containing table

Description automatically generated**

**Figure 13 – ShowAccount.java variable updates**

**A close up of a logo

Description automatically generated**

**Figure 14 – Updates to processRequest within ShowAccount.java**

**A screen shot of a computer

Description automatically generated**

**Figure 15 – Updates to getData within ShowAccount.java**

**A screenshot of a cell phone

Description automatically generated**

**Figure 16 – Table updates to removed unused fields**

Finally, we update the database to make the changes to the table and the records inserted (Figure 17).

**A screenshot of a cell phone

Description automatically generated**

**Figure 17 – SQL script table updates**

The results of our changes can be seen when we select the “Your Account” button after logging in (Figure 18).

A screenshot of a cell phone

Description automatically generated

**Figure 18 – Results of the changes applied**