Homework 4

Fixing Payment Card Industry Java Web Application

Overview

In this homework, you will modify an existing Java Web application that violates several Payment Card Industry guidelines and recommendations. Your task is to locate the issues, based on the readings for this course, identify what is wrong and then fix the code. You will discuss each issue in terms of why the issue may cause a security vulnerability, and how you specifically fixed the issue.

Assignment

Review, Run and Understand the Sample Java Web application.

The current code, uses Java JSP and Servlets to allow a user to login to their account and view credit card data stored in the database. The functionality is relatively simple but several PCI compliance rules have been violated that will prevent the application from being approved by a PCI software auditor.

You should first load up the application, populate the database and make sure the application is working in your environment as expected. The application uses the Java Derby relational database. The script used to populate the application is attached in your project folder as well as the Java web project itself. You should be able to open the existing project using Netbeans. However; you may need to load the Derby drivers to the libraries for the project.

You can create a new database connection by clicking the services folder and then right mouse-click on Databases select new connection. You can then create a connection for the SDEV425 database. Below is a screen capture that represents my configuration.



There are some tutorials on the Internet on using the Derby database within Netbeans. (e.g. <u>https://netbeans.org/kb/docs/ide/java-db.html</u>) The key is making sure you have the drivers in the library of your project and you run the scrips to populate the tables. To execute a command in the database from within Netbeans you right click the connection and select execute command. A window will then pop-up for that connection and you can execute any command, including all of those database scripts provided to you.



Once you have the database loaded, you can try the application. (This assumes you have properly installed the Java EE when you installed Netbeans).

To launch the application, just highlight the java project you loaded (SDEV425_HW4) and click the green arrow. Once launched, your glassfish server will start and your default browser will be invoked. The application will automatically launch and the home page will be displayed.

SDEV 425 Final Project × +	×				
🗧 🕑 🕼 localhost:8080/SDEV425_HW4/index.jsp 🔍 C 🔍 Java Derby set-up in t 🔿 💟 🏠 🧔 😫	=				
SDFV425 Final Project					
SDL v +2.5 F mar F Toject					
Home Sign In Your Account Sign Out					
J UMUC					
University of Maryland University College					
The online bachelor's degree in software development and security from University of Maryland University College is designed to teach you in-demand programming languages and the best practices in software development in today's workplace.					
Your degree can prepare you to pursue technical and leadership roles in diverse and challenging application development and security settings, including high-demand positions such as software development and security analyst, software development and security manager, application and software architect, information security officer, intrusion analyst, penetration tester, programmer, software engineer, security and code auditor, or system architect.					
Your software development degree courses will focus on developing your skills using multiple programming languages and relational databases while maintaining component security using industry and government best practices. You'll learn to design, develop, and test secure software applications, conduct software penetration testing, and provide recommendations for reducing computer security risks.					
©2015 UMUC					

You can (and definitely should) review the database script to see the current users and associated passwords. As a test, you can click the Sign in link and enter this account information

Email: james.robertson@umgc.edu Password: mypassword

SDEV425 Login	< \ +			_ 🗆 🗙	
Contemporary Conte	IW4/login.jsp	I C Q Java Derby	set-up in t → 💟 1	ñ 4 9 ≡	
	SDEV425 Final Project				
<u>Home</u>	<u>Sign In</u>	Your Account	Sign Out		
		Login			
	Email: james.robe	ertson@umuc.edu			
	Password:	Sign In			

After successful sign-in you will see the welcome message displayed:

Welcome >	<hr/> +		-	. 🗆	×
Calhost:8080/SDEV425_H	W4/Authenticate	🔻 🖉 🔍 Java Derby set-up in t 🔶 💟 🏫		ø	≡
	SDF	EV425 Final Project			
<u>Home</u>	<u>Sign In</u>	Your Account Sign Out			
	Hell	o james.robertson@umuc.edu!			
	Selec	t from any of menu items above.			
	Menu Item	Description			
	Home	Return to the initial landing page.			
	Sign-in	Sign in to the database.			
	Your Account	Update your name and connection information.			
	Sign-out	Sign out of the system. This invalidates your session.			

Clicking on the Your Account menu will display the credit card information.

Your Account Information	× \ +			-		×
Contemporary Conte	25_HW4/ShowAccount) î		Ø	≡
	SDE	V425 Final Project				
Home	<u>Sign In</u>	Your Account Sign O	<u>ut</u>			
		Account Data				
	Email:	james.robertson@umuc.edu				
	User ID:	1				
	Card Holder Name:	James Robertson				
	Card Type:	MasterCard				
	Service Code:	27aD				
	Card Number:	111111111111				
	CAV CCV2:	321				
	Expire Date:	2016-02-23				
	Full Track Data:	3323344ASDfc23442				
	PIN:	3Ds2q				

Clicking on Sign out will invalidate the current session.

Logout	×	\ +				×
Certification (Contraction)	t:8080/SDEV425_HW4	4/logout.jsp		et-up in t → 🔽 🏫	A 9	≡
		SDEV4	25 Final Project	t		
	<u>Home</u>	<u>Sign In</u>	Your Account	<u>Sign Out</u>		
		Thank you f	or visiting our Web S	ite.		

Be sure to demonstrate the application runs properly in your development environment.

Carefully, review the code and perform analysis as needed. You should experiment with running the application as well as reviewing the code to identify possible areas of security concerns. You don't have to be an expert in JSP/Servlets, html or css to be able to find some of the issues that you have read about in the PCI documentation. However; it is recommended you experiment with the code so you have a baseline familiarity with the model and know how the JSP and servlets communicate with each other.

Focus on the PCI compliance issues found in sections 6 through 9 as you look for issues. There are multiple issues and you should work to fix and document as many as possible.

Hints:

- a. Make sure your Java EE Netbeans is working properly.
- b. Work to get the Derby database populated and working with the SDEV425_HW4 Web application.
- c. Study and experiment with the code. For example, even if you haven't used CSS style sheets before you should be curious about how color schemes are used and how easily they can be changed in the css file. You should also note how the database connections are made and how JSP and servlets communicate.
- d. Again, start on this early. This will take you longer than you think.

Deliverables

Provide all of your modified Java code, your modified database script and a PDF document describing how you addressed each issue. You should clearly describe the code and what PCI compliance issue were violated and how you fixed it. You should provide screen captures as needed to support your findings and improvements.

Be sure your PDF document is neat, well-organized and is well-written with minimal spelling and grammar errors. All references used should be included in your document.

Grading rubric:

Attribute	Meets	Does not meet
Sample Java	10 points	0 points
Application	Demonstrates the Java application	Does not demonstrate the Java
	is running properly in your	application is running properly in your
	development environment. (10	development environment.
	points)	
PCI compliance	70 points	0 points
	Identifies PCI compliance issues	Does not identify PCI compliance issues
	found within the application. (35 points)	found within the application.

	Fixes and documents the PCI	Does not fix or document the PCI
	compliance found during the	compliance found during the analysis.
	analysis. (35 points)	
Documentation and	20 points	0 points
Submission	Provides all modified Java code,	Did not provide all modified Java code,
	modified database script and a	modified database script or a PDF or
	PDF or word document describing	word document describing how each
	how each issue was addressed. (5 points)	issue was addressed.
		Did not provide screen shots and
	Provides screen shots and	descriptions of the successful executing
	descriptions of the successful	the code and the resultant output as
	executing the code and the	applied to each security control.
	resultant output as applied to	
	each security control. (5 points)	Document is not neat, well-organized
		and is not well-written with minimal
	Document is neat, well-organized and is well-written with minimal	spelling and grammar errors.
	spelling and grammar errors.	All references used were not included in
	(5points)	your document.
	All references used should be	
	included in your document. (5	
	points)	