Homework 2

Implementing Low-Impact Security Controls
Adding Select Security Controls to a Login Application

Overview

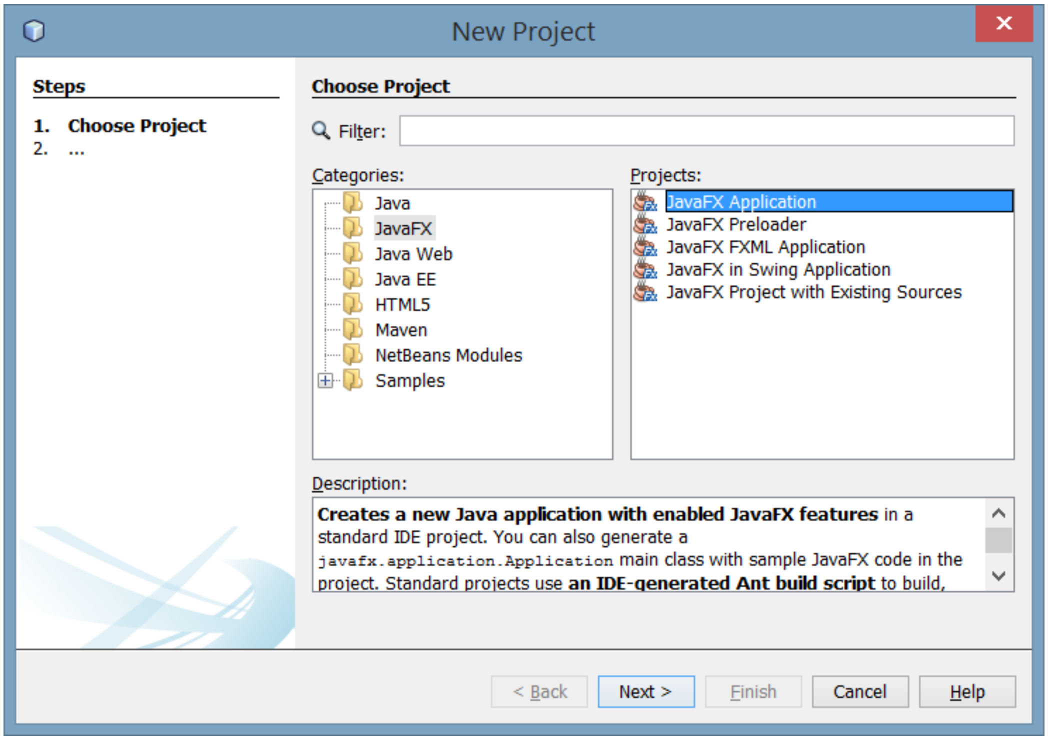
In this homework you will enhance an existing JavaFX Login application to add select low-impacts security controls based on recommendations from the NIST security controls database. We won't be able to implement all of the baseline controls for this exercise. However; the assignment details below provide direction on which ones should be implemented. You will also asked to choose one additional low-impact security control, of your choice and implement it.

Assignment

Part I: Review and Understand the Sample JavaFX application

More than likely you have not coded a JavaFX application before. However; if you successfully completed the CMIS242 prerequisite, you have seen how to create Simple GUI's using the Swing class. JavaFX is quite similar to the Swing classes. In fact, most of the classes and methods provide the same functionality. The sample code provided will refresh and enhance your GUI building skills.

1. Using Netbeans, start a new Java project. Select the JavaFX application option.



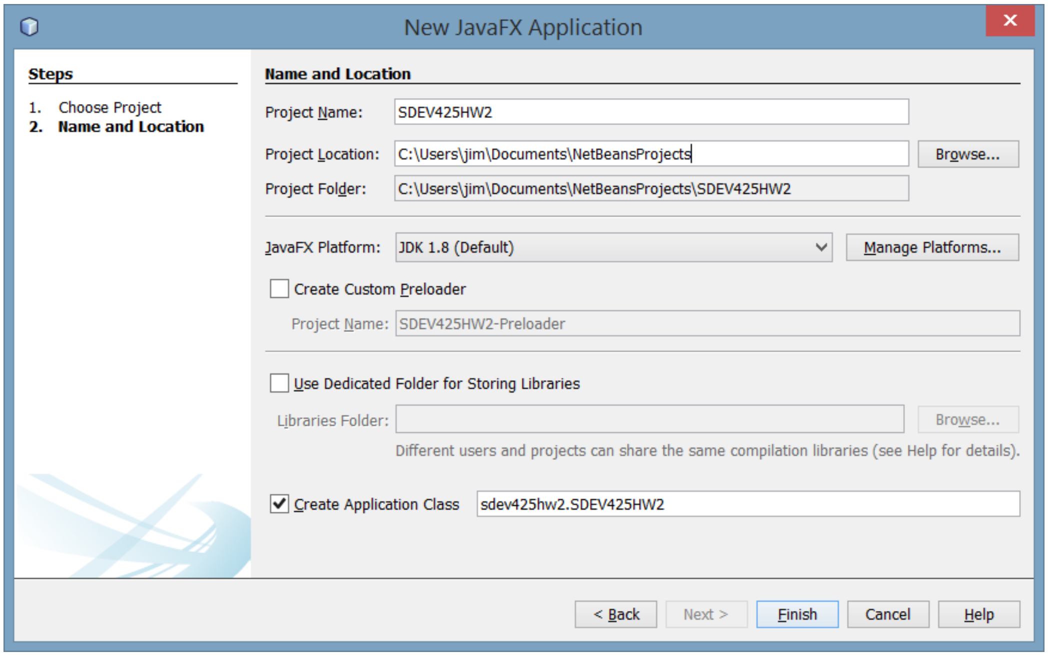
Categories: JavaFX

Projects: JavaFX Application

Description:

Creates a new Java application with enabled JavaFX features in a standard IDE project. You can also generate a `javafx.application.Application` main class with sample JavaFX code in the project. Standard projects use **an IDE-generated Ant build script** to build,

2. Name the project SDEV425HW2 (or a name of your choice).



Name and Location

Project Name: SDEV425HW2

Project Location: C:\Users\jim\Documents\NetBeansProjects [Browse...]

Project Folder: C:\Users\jim\Documents\NetBeansProjects\SDEV425HW2

JavaFX Platform: JDK 1.8 (Default) [Manage Platforms...]

□ Create Custom Preloader

Project Name: SDEV425HW2-Preloader

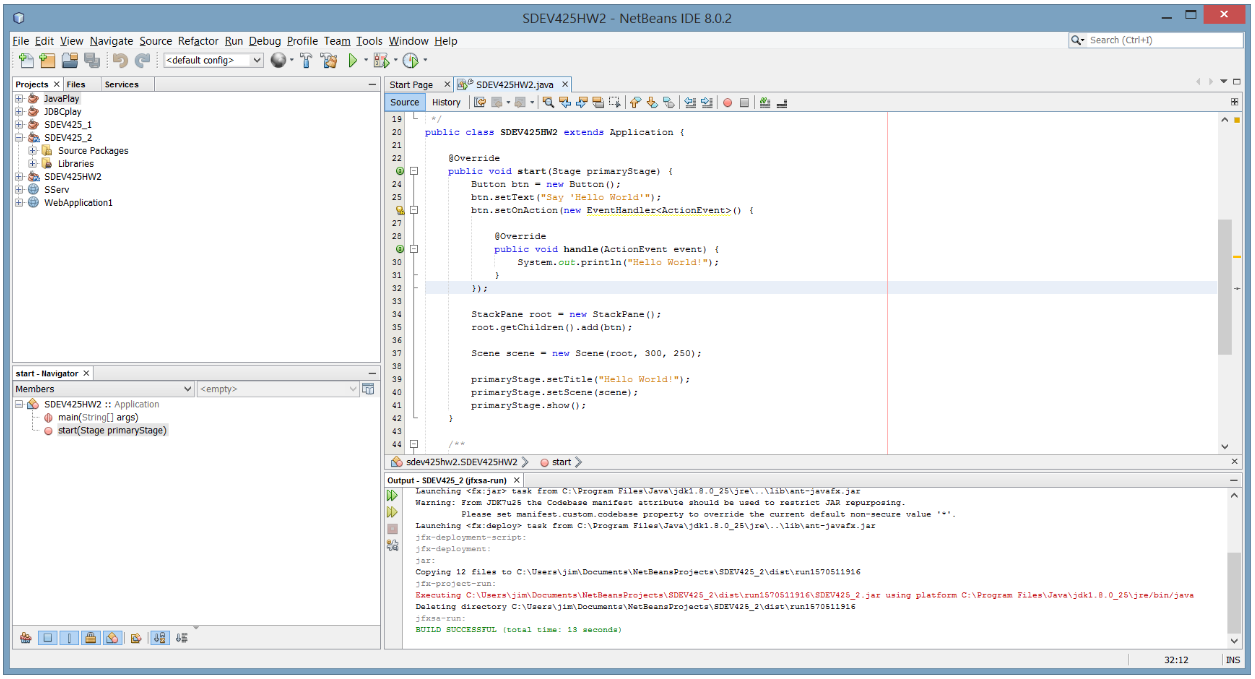
□ Use Dedicated Folder for Storing Libraries

Libraries Folder: [Browse...]

Different users and projects can share the same compilation libraries (see Help for details).

☑ Create Application Class sdev425hw2.SDEV425HW2

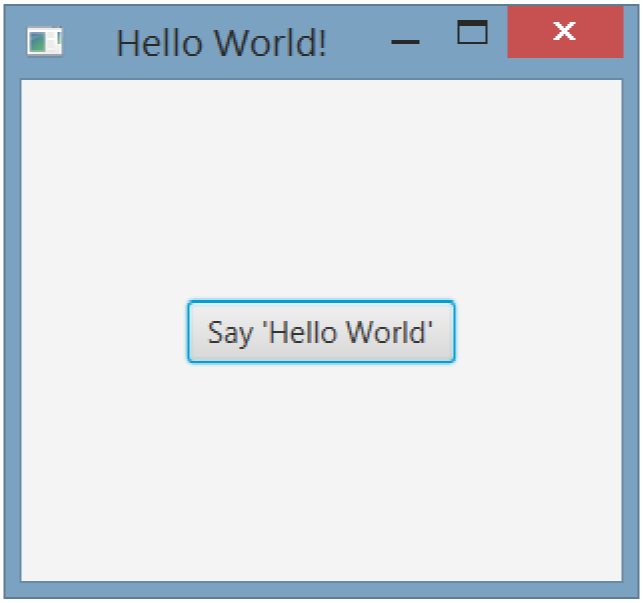
3. Click Finish and the template application code will be provided.



```java
 */
public class SDEV425HW2 extends Application {

    @Override
    public void start(Stage primaryStage) {
        Button btn = new Button();
        btn.setText("Say 'Hello World'");
        btn.setOnAction(new EventHandler<ActionEvent>() {

            @Override
            public void handle(ActionEvent event) {
                System.out.println("Hello World!");
            }
        });

        StackPane root = new StackPane();
        root.getChildren().add(btn);

        Scene scene = new Scene(root, 300, 250);

        primaryStage.setTitle("Hello World!");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    /**
```

```
Launching <fx:jar> task from C:\Program Files\Java\jdk1.8.0_25\jre\..\lib\ant-javafx.jar
Warning: From JDK7u25 the Codebase manifest attribute should be used to restrict JAR repurposing.
         Please set manifest.custom.codebase property to override the current default non-secure value '*'.
Launching <fx:deploy> task from C:\Program Files\Java\jdk1.8.0_25\jre\..\lib\ant-javafx.jar
jfx-deployment-script:
jfx-deployment:
jar:
Copying 12 files to C:\Users\jim\Documents\NetBeansProjects\SDEV425_2\dist\run1570511916
jfx-project-run:
Executing C:\Users\jim\Documents\NetBeansProjects\SDEV425_2\dist\run1570511916\SDEV425_2.jar using platform C:\Program Files\Java\jdk1.8.0_25\jre/bin/java
Deleting directory C:\Users\jim\Documents\NetBeansProjects\SDEV425_2\dist\run1570511916
jfxsa-run:
BUILD SUCCESSFUL (total time: 13 seconds)
```

4. Click the Green arrow to run the application.



5. Clicking the button will display "Hello World" in the Netbeans output.

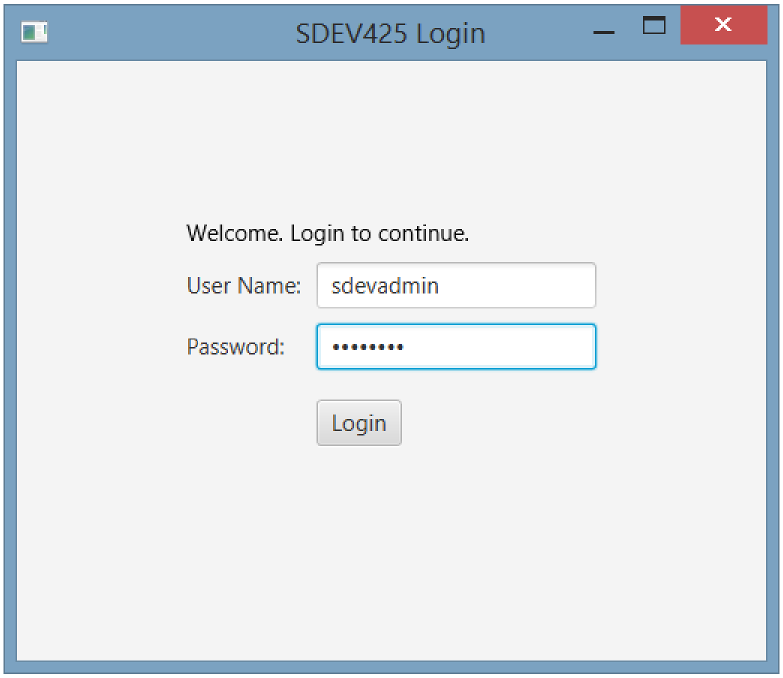
6. Study the template code paying careful attention to the methods, and the use of the event handlers.

7. Using the Login Application, **provided as an attachment in this assignment**, copy and paste it into your project. Note: you may need to change the class and package name of the Login application.

8. Run and **experiment** with the code so you understand what each line of code is doing.

9. Demonstrate you successfully completed this part of the assignment by providing screen captures generated from your development environment showing both the FX Hello, World and SDevAdmin login apps running with no additional modifications.

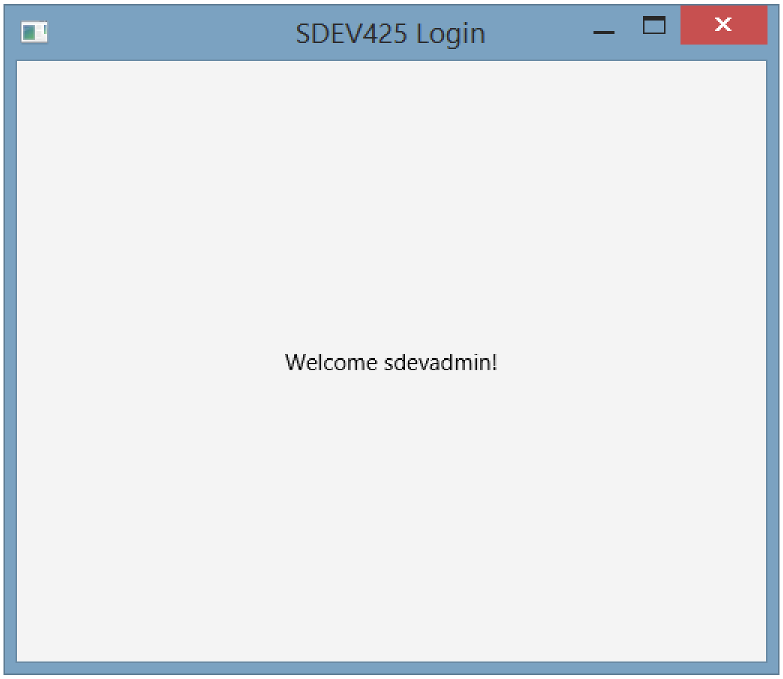
10. Describe an overview of the functionality of the SDEVAdmin login application pointing out the functionality of critical code components.



Welcome. Login to continue.

User Name: sdevadmin

Password: ••••••••



Welcome sdevadmin!

Part II: Apply select NIST low-impact security controls to the JavaFX Login application.

The following security controls should be applied to the application (check the NIST Security Controls Database for details, description and guidance for each control:

- AC-7 - UNSUCCESSFUL LOGON ATTEMPTS
- AC-8 - SYSTEM USE NOTIFICATION
- AU-3 - CONTENT OF AUDIT RECORDS
- AU-8 - TIME STAMPS
- IA-2(1) IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS) | NETWORK ACCESS TO PRIVILEGED ACCOUNTS (Note this is an enhancement of an existing low-impact security control)
- Select one additional low-impact security control and implement it. This can be an enhancement or a required low-impact security control. Selecting a control that provides documentation as opposed to code changes is also acceptable and encouraged.

Hints:

a. Start with the baseline Login Application and add methods (or additional classes) as needed to comply with each of the security controls.
b. You will need to make some decisions for your implementation for the security audit/log files format. Each student may have a slightly different implementation.
c. For the multi-factor authentication, keep it simple. One approach is to send an email to the user with a security code. Then, have them check their email and enter the code. If the code matches, they are properly authenticated.
d. There are examples for using JavaMail and writing to files in the materials for this week. Be sure to use those as needed.
e. Pay attention to the details of the NIST database description and make sure all of the selected security controls for this project are fully implemented.
f. Start on this early. This will take you longer than you think.

Deliverables

Provide your security fixed Java source code along with a PDF document describing how you addressed each security control. For example, you should list the security control and the descriptions and show and describe the code that addresses the security control. You should also provide screen shots and descriptions of the successful executing the code and the resultant output as applied to each security control. Be sure to submit all of your Java source code if you used multiple classes.

Your code should be well-documented with comments, include header comments, use proper variable and naming conventions and properly formatted. (Use the Google Java style guidelines provided in the content-> Course Resources). 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 JavaFX application	**15 points** Provides screen captures generated from your development environment showing both the FX Hello, World and SDevAdmin login apps running with no additional modifications. (8 points) Describes an overview of the functionality of the SDEVAdmin login application pointing out the functionality of critical code components. (7 points)	**0 points** Does not provide screen captures generated from your development environment showing both the FX Hello, World and SDevAdmin login apps running with no additional modifications. Does not describe an overview of the functionality of the SDEVAdmin login application pointing out the functionality of critical code components.
NIST low-impact security controls	**60 points** Applies AC-7 - UNSUCCESSFUL LOGON ATTEMPTS to the application. (10 points) Applies AC-8 - SYSTEM USE NOTIFICATION to the application. (10 points) Applies AU-3 - CONTENT OF AUDIT RECORDS to the application. (10 points) Applies AU-8 - TIME STAMPS to the application. (10 points) Applies IA-2(1) IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS) | NETWORK ACCESS TO PRIVILEGED ACCOUNTS to the application. (10 points) Selects one additional low-impact security control and implements it. (10 points)	**0 points** Does not apply AC-7 - UNSUCCESSFUL LOGON ATTEMPTS to the application. Does not apply AC-8 - SYSTEM USE NOTIFICATION to the application. Does not apply AU-3 - CONTENT OF AUDIT RECORDS to the application. Does not apply AU-8 - TIME STAMPS to the application. Does not apply IA-2(1) IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS) | NETWORK ACCESS TO PRIVILEGED ACCOUNTS to the application. Does not select one additional low-impact security control and implements it.
Documentation and Submission	**25 points** Provides all Java source code including “fixed” code. (5 points) Provides screen shots and descriptions of the successful	**0 points** Does not provide all Java source code including “fixed” code. Does not provide screen shots and descriptions of the successful executing

	executing the code and the resultant output as applied to each security control. (5 points) Code is well-documented with comments, including header comments, use of proper variable and naming conventions and properly formatted. (5 points) Document is neat, well-organized and is well-written with minimal spelling and grammar errors. (5points) All references used should be included in your document. (5 points)	the code and the resultant output as applied to each security control. Code is not well-documented with comments, including header comments, use of proper variable and naming conventions and properly formatted. Document is not neat, well-organized and is not well-written with minimal spelling and grammar errors. All references used were not included in your document.