Here is the link for the github repository <https://github.com/Tenukki/F-projekti>

How to install the project:

Install Git to your computer and clone the project to your computer. Use Netbeans to run the program. The idea of the project is fairly simple. With the app you can add yourself to the event. You need to login if you want to submit the form. The app is really simple this is due to that I wanted to show as simply as possible how to fix these kind of problems that app has.

Flaw 1: Security misconfiguration

With this problem usually app has some kind of faulty in it’s security configuration. The problems can vary a lot. In my project I had disabled the csrf token. In spring boot framework the csrf token is enabled as default. This kind of mistake would be rare because you should not disable the csrf token.

How to fix it?

The answer to this is fairly simple. Just deleting the line where you disabled the csrf token should be it. To be more specific go to file named SecurityConficuration.java and delete the code “http.csrf().disable();” on line 24.

Flaw 2: Broken authentication

In broken authentication the authentication is weak or poorly build usually. Some cases hackers can use session hijacking or working as an middle man where they hijack the passwords or some other data. In my project the hashing has a problem it is absent. The user named ted has a password ted1 that is saved as plain text. My project doesn’t use database, but when saving confidential data like passwords, credit card number or social security numbers they should always be hashed. In my case the password is not hashed.

How to fix it?

Fixing this problem in java spring is also simple. Spring uses bcrypt that hashes the password that no man can read it. By adding few methods and overwriting existing code we can add hash to our password. Passwords should all ways be hashed to make them more secure. To be more specific you must add few new methods to your securityconfig file. The method that is added will hash the password

Flaw 3: Broken access controller

User session must be always stored somewhere. The session is saved in the browser usually as a cookie. Session can also be saved in local storage or in session storage. With spring we use cookies to save the session. In my project when you are asked to add your name and address you are also asked to login. When you are logged in you can close the browser and reopen the page without of need to login again. This is the problem. Even if user closes the browser the session is still saved to the browser and other people using the computer can use your session.

How to fix it?

How this kind of problem can be fixed? With spring you can add logout option to your paths meaning that you can make login out mandatory and automated. Better way is to add logout button to my project. The logout works if you go to localhost:8080/logout. When you go to this path and you are logged in you can logout safely and the session will be deleted. So adding a logout button and automated session deletion will remove this problem.

Flaw 4: Insufficient Logging

Insufficient Logging meaning that the users are not monitored anyway. Monitoring logins and user activity is extremely important for security reasons and for performance reasons. Monitoring, for example, where some request are made and how many are they made can help with service denial attacks. In my project the monitoring is absent. There is no kind of monitoring for user activity or for request that are coming to the back-end.

How to fix this?

Fixing this kind of problem is directly an error in code is just poor plain of the architecture. With our project we could implement a way to see what users are logged in and who are sending or getting data from the server. We could use existing libraries for this or make our on implementations of monitoring the user activity.

Flaw 5: Sensitive Data Exposure

Protecting data is extremely important and to be sure that the data is safelystored. In my project one of the api end points is not protected meaning it is public for all. Please login first by using username ted and password ted1. If you go to address <http://localhost:8080/event> you can see your note. This is and security error the end point should be authenticated. We forgot to protect it in our security configuration. In spring all api end points by default are public.

How to fix this?

Fixing this problem is really simple just by adding a small line of code where you tell if the end point is not found from the list you need to use authentication. The code can be found here anyRequest().authenticated().and()

.formLogin().permitAll().and()

.logout().permitAll();