08

Conclusão

Transcrição

[00:00] Parabéns, chegamos agora ao final da segunda parte do curso de segurança utilizando o Spring vmc. Vamos recapitular o que é que nós vimos aqui nessa segunda parte? Então nós tínhamos inicialmente aqui nessa nossa aplicação, essa nossa aba de depoimentos, onde um usuário deveria colocar a experiência dele com a Alura shows, deveria ser uma mensagem com algum dizer do tipo.

[00:22] Mas nós vimos que um desses usuários, o que é que ele fez? Ele, na verdade, ao invés de deixar uma mensagem de depoimento, ele colocou o código java script, que foi persistido no nosso banco, e essa mensagem depois foi levada para todos os nossos clientes, todos os usuários que foram acessar essa aba de depoimentos.

[00:43] Então nós vimos que uma das formas de nós protegermos isso era o quê? É nós fazermos uma validação aqui do que que o usuário está passando nesse campo título e do depoimento. Então nós vimos aqui, nós criamos essa nossa classe, que era depoimento validator, para verificar o caso quando o usuário abrir ou ele fechar tags, tanto no caso do campo do título, como no campo da mensagem.

[01:05] Caso isso aconteça, nós mostrávamos aquela mensagem para o usuário dizendo que a mensagem que ele passou não é uma mensagem aceita aqui pela nossa aplicação.

[01:15] Feito isso, o que nós vimos? Nós vimos que na hora de um usuário fazer o registro aqui na Alura shows, o que acontece? Ele poderia vir aqui, clicar com o botão direito do mouse, inspecionar, e ele poderia fazer uma edição desse formulário de registro para ele poder manipular aquele atributo que estava na classe usuário, aquele atributo roles.

[01:40] E esse nosso usuário poderia alterar esse atributo para que ele fosse o administrador do sistema. E nós vimos no caso da Joviane que ela conseguiu alterar esse atributo para que ela fosse administrador da Alura shows e ela teve acesso ao painel administrativo.

[01:57] Então nós vimos que uma das formas que nós podemos nos proteger contra esse tipo de ataque, que é utilizando justamente uma classe intermediária que é responsável só por fazer o transporte dessas informações desse objeto para montar a nossa classe, para montar o nosso objeto usuário de fato.

[02:16] Nós montamos essa nossa face usuário DTO para pegar só os parâmetros que de fato nós esperamos receber do formulário. Então nós configuramos aqui com os atributos somente que nós esperamos estar recebendo do formulário, e nós montamos o nosso objeto do tipo usuário.

[02:29] Com isso, mesmo que na hora de fazer o registro nós manipulássemos o html para tentar mudar a roles, agora esse nosso usuário DTO não tem nenhum atributo roles e com isso a nossa aplicação estaria protegida.

[02:44] Esse método funcionaria para qualquer framework, mas como nós estamos trabalhando com spring, nós utilizamos recursos que o spring nos oferece e nós chamamos esse método, setAllowedFields, indicando quais são os métodos da nossa classe que podem ser permitidos de serem manipulados.

[03:03] E nós falamos que da nossa classe usuário, só podem ser editados, manipulados, esses quatro atributos, que é o nome, o e-mail, a senha e o nome da imagem, que o usuário vai passar na hora de fazer o registro. Aquela outra atributo, roles, vamos só lembrar dele aqui, esse atributo roles ele não teria permissão de fazer a fazer a manipulação.

[03:27] E, com isso, nós poderíamos voltar aqui na nossa classe usuário controller, uma vez que estamos trabalhando com esse método do setAllowedFields, e poderíamos fazer a associação diretamente aqui com a nossa face usuário que não teria nenhum problema.

[03:40] Por fim, nós vimos que o usuário pode, na hora de fazer o registro, colocar um arquivo que na verdade não é uma imagem, é um arquivo que possui um código JAVA e como nós vimos, nós fomos capazes até de remover um diretório aqui na nossa máquina, onde o Tomcat está sendo executado aqui no Windows.

[03:59] Nós vimos que é importante nós fazermos essa ligação de que tipo de arquivo é esse que o usuário está passando para a nossa aplicação. Nós poderíamos pensar inicialmente em fazer aquela verificação da extensão seria já uma resolução. Mas como nós vimos, o usuário pode facilmente manipular isso.

[04:17] Então, nós vimos que o ideal era o que? É nós fazermos uma leitura do conteúdo desse arquivo para que nós consigamos determinar se é um arquivo de imagem ou não. E nós conseguimos ter uma melhor certeza do que é esse arquivo que esse nosso usuário está passando para nós.

[04:33] Nós criamos essa classe, imagem validator, para validar essa entrada do arquivo e nós só aceitaríamos se fosse de fato uma imagem compatível com esses nossos requisitos. Eu agradeço muito vocês por terem chegado até o fim desse treinamento e eu espero encontrá-los em uma próxima oportunidade.