



Plano de Segurança

	Created by	joao victor leonardo lopes Matheus Machado Felipe Wesley
	Last edit	@May 13, 2021 9:46 PM
	Last edited by	Felipe Wesley
	Status	Completed

Apresentação

Este documento tem o intuito de apresentar um plano de segurança referente à aplicação University Market, destacando alguns pontos relevantes referentes a vulnerabilidades do app, informações e dados sensíveis e demais aspectos relacionados a segurança e integridade de informações.

Aspectos analisados

- Principais pontos de vulnerabilidade do sistema

- Considerando também: arquitetura, vulnerabilidade de tecnologia, processo que dependia de meios externos, etc.
- Informações relevantes
 - Dados de usuários, informações de negócio, etc.
- Meios de acesso
 - Web, aplicativo, API, etc.

Principais pontos de vulnerabilidade do sistema



Armazenamento de dados de sessão do usuário no navegador

Conforme o planejamento da aplicação, e devido também aos recursos oferecidos pelas tecnologias escolhidas, pode-se citar como possível ponto de vulnerabilidade o controle de session de usuário com JavaScript no front-end.

Uma validação mais "forte" é, também, realizada no back-end, mas o ideal seria armazenar as sessões em algum banco (possivelmente NoSql), para evitar o acesso facilitado ao dados através do browser.



Configurações no controle de permissões no servidor de hospedagem da API

Talvez algumas configurações de controle de permissões RWX (*Read*, *Write* e *Execute*) não tenham sido feitas corretamente, tornando este ponto significativamente vulnerável, considerando a possibilidade de acesso inesperado não autorizado ao servidor onde a aplicação esteja hospedada.



Configuração de CORS pode apresentar vulnerabilidade de acordo com a origem da requisição recebida

Enquanto não há uma versão de produção da aplicação, a configuração de controle de CORS da API é habilitada para requisições de todas as origens, permitindo acesso à todos os verbos HTTP contidos nos routes.



Nem todos os dados sensíveis foram pensados para serem submetidos a hashes

Informações sensíveis, sobretudo de usuários, não foram completamente pensadas para serem submetidas à aplicação de hashes (sejam eles uni ou bi-direcionais). Informações como senha, por exemplo, serão submetidas à aplicação de hash normalmente, porém, outras informações que possam vir a ser consideradas sensíveis não foram planejadas (questões de tipagem, estrutura de armazenamento, apresentação em layouts, etc) para serem criptografadas.



Ainda não há estratégia de criptografia de registros marcados como deletados

Usuários, publicações, movimentações, etc.

Sabe-se que, com a corrente legislação referente à Lei Geral de Proteção de Dados, os registros armazenados e tratados pela instituição devem atender às políticas de segurança definidos no contrato de uso da aplicação, e que, tal contrato, assegura ao usuário o tratamento adequado de seus dados de acordo com o que é estipulado pela LGPD.



Entidades do banco de dados podem ser expostas ao front-end

Com a atual arquitetura de back-end planejada, ainda não foi definida uma estratégia efetivamente segura para envio de dados do banco de dados ao front-end da aplicação. Ainda é planejado a criação de modelos, com dados específicos estritamente necessários, para a estabelecer comunicação back-front, evitando, assim, a exposição de entidades completas à origem da requisição.

Informações relevantes sensíveis

1. Nome do usuário
2. Endereço do Usuário
3. Data nascimento
4. Senha do Usuário
5. Movimentação da conta

Pesquisa de notícias relacionadas a segurança da informação

- **eBay**

Durante 2014, um grupo de hackers tiveram o acesso aos dados de mais de 140 milhões de usuários do eBay durante 229 dias, tempo mais do que suficiente para comprometer o banco de dados dos usuários. Eles fizeram isso através das informações de alguns funcionários que obtiveram.

Os criminosos só conseguiram acesso a dados superficiais, como: nome, senha, e-mail, endereços, telefone e data de nascimento. Os clientes podem deixar de utilizar o serviço pela falta de confiança na empresa, o que irá causar uma grande perda de dinheiro. E esse risco se agrava ainda mais quando falamos do maior site de leilões e vendas que é o eBay

<https://backupdados.com.br/blog/11-empresas-que-sofreram-ataques-ciberneticos/#eBay>

<https://computerworld.com.br/seguranca/os-15-maiores-vazamentos-violacoes-de-dados-do-seculo-21/>

- **NEWEGG**

A loja online foi invadida pela gangue virtual Magecart, que injetou um código de skimming de cartão de crédito no site da Newegg. Sempre que um cliente comprava algo online, essas informações de pagamento iam direto para o branco de dados controlado pela Magecart, para passarem despercebidos o grupo utilizou um domínio semelhante e usou um certificado HTTPS para se misturar.

<https://blog.avast.com/pt-br/os-ultimos-10-maiores-vazamentos-de-dados>

- **Netshoes**

O Comércio eletrônico de artigos esportivos, também foi vítima de um ataque e dados como números de CPF, e-mail e data de nascimento de seus clientes foram roubados. Depois de uma reunião com o Ministério Público do Distrito Federal, a Netshoes se comprometeu a ligar para os clientes afetados para explicar o que aconteceu e orientá-los.

<https://assisemendes.com.br/vazamento-de-dados-nas-empresas/>

⚠ Com essas pesquisas chegamos a conclusão que com o aprimoramento da internet, todos os dias teremos novos desafios para deixar cada vez mais seguros os dados dos nossos usuários.

Trabalharemos para implementar um bom padrão de criptografia em dados chaves, como, por exemplo, a senha de acesso do usuário, e manteremos uma rotina de atualizações para manter o sistema íntegro e seguro para todos os usuários, subindo versões atualizadas para `main / master` constantemente, com novas atualizações de segurança a cada merge realizado.

Referência da organização do projeto

Recentemente, uma nova organização do projeto foi criada, afim de manter um controle maior sobre as funcionalidades implementadas, separando devidamente as responsabilidades entre os membros da equipe.

Esta separação também possibilitou uma organização melhor das camadas de front-end e back-end da aplicação, mantendo ambas centralizadas num projeto, mas em repositórios separados, facilitando a geração de uma possível build do front ou deploy da API em algum servidor.

O projeto pode ser encontrado e acompanhado na seguinte organização no Github:

<https://github.com/university-market>

