Curso de Pós-Graduação em Cibersegurança - Turma 2024.2

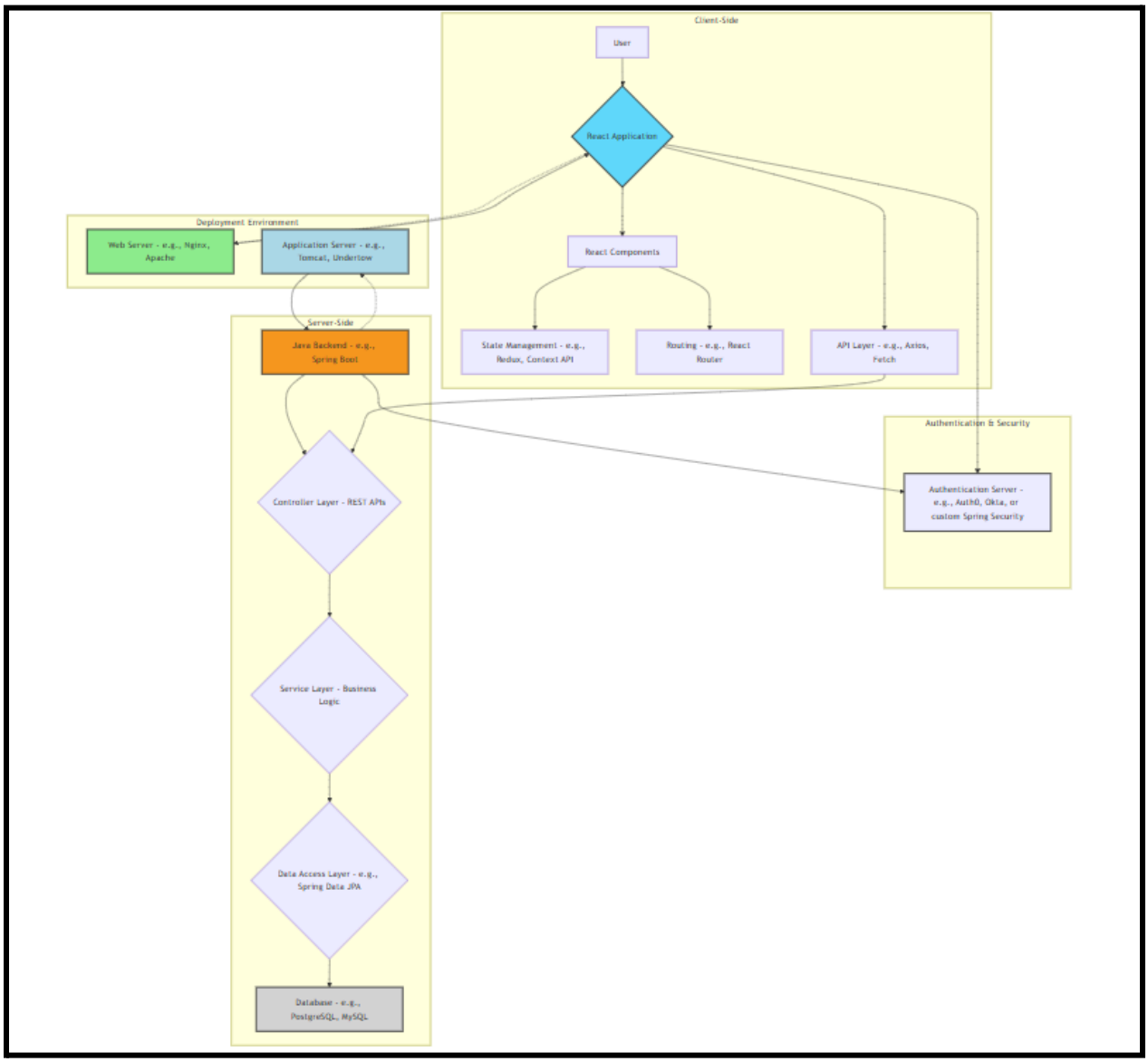
Modelagem de Ameaças

CESAR SCHOOL

Aluna : Adyellen Alves

Professor: Thiago Prota

Recife, 28 de Junho de 2025


Client-Side
User
React Application
React Components
State Management - e.g., Redux, Context API
Routing - e.g., React Router
API Layer - e.g., Axios, Fetch
Deployment Environment
Web Server - e.g., Nginx, Apache
Application Server - e.g., Tomcat, Undertow
Server-Side
Java Backend - e.g., Spring Boot
Controller Layer - REST APIs
Service Layer - Business Logic
Data Access Layer - e.g., Spring Data JPA
Database - e.g., PostgreSQL, MySQL
Authentication & Security
Authentication Server - e.g., Auth0, Okta, or custom Spring Security

Com base no diagrama de arquitetura da aplicação fornecido, segue a análise de ameaças utilizando o padrão STRIDE:

Agentes de Ameaça

- TA-1: Atacante Externo Não Autenticado
- TA-2: Atacante Externo Autenticado
- TA-3: Atacante Interno Malicioso (usuário legítimo comprometido ou funcionário mal-intencionado)
- TA-4: Atacante com Acesso Físico ou Lógico (comprometendo a infraestrutura)

Ameaças (STRIDE)

S - Spoofing (Falsificação de Identidade)

- T-1: Falsificação de identidade do usuário: Um atacante pode tentar se passar por um usuário legítimo para obter acesso indevido à aplicação React ou aos recursos do backend.
- T-2: Falsificação de identidade do servidor: Um atacante pode falsificar o servidor de autenticação (Auth0, Okta, ou Spring Security) ou o servidor web/aplicação para interceptar credenciais ou redirecionar usuários para sites maliciosos.
- T-3: Falsificação de requisições de API: Um atacante pode criar requisições falsas para as APIs REST do backend, fazendo-se passar por uma aplicação cliente legítima.

T - Tampering (Violação de Dados)

- T-4: Violação de dados em trânsito: Dados trafegando entre a aplicação React, o backend e o servidor de autenticação podem ser interceptados e modificados por um atacante.
- T-5: Violação de dados no lado do cliente: Um atacante pode manipular o estado da aplicação React (via Redux, Context API) para contornar validações ou exibir informações incorretas.
- T-6: Violação de dados no banco de dados: Um atacante com acesso ao backend ou

diretamente ao banco de dados pode modificar, excluir ou inserir dados indevidamente.
- T-7: Violação de arquivos de configuração ou binários: Um atacante com acesso ao ambiente de deployment pode alterar arquivos de configuração do servidor web/aplicação ou os binários da aplicação, introduzindo comportamento malicioso.

R - Repudiation (Repúdio)

- T-8: Repúdio de ações do usuário: Um usuário malicioso pode negar ter realizado certas ações dentro da aplicação (e.g., transações, modificações de dados) se não houver um log de auditoria adequado.
- T-9: Repúdio de ações do administrador: Um administrador comprometido ou mal-intencionado pode negar alterações feitas na configuração ou nos dados do sistema.

I - Information Disclosure (Divulgação de Informação)

- T-10: Divulgação de dados sensíveis: Informações confidenciais (e.g., credenciais de usuário, dados pessoais, informações financeiras) podem ser expostas se não forem devidamente protegidas em trânsito ou em repouso (banco de dados, logs).
- T-11: Divulgação de informações de erro: Mensagens de erro detalhadas no frontend ou backend podem vazar informações sobre a arquitetura interna, versões de software ou dados sensíveis.
- T-12: Divulgação de segredos (chaves de API, credenciais de banco de dados): Segredos armazenados no ambiente de deployment ou no código-fonte podem ser acessados por atacantes.

D - Denial of Service (Negação de Serviço)

- T-13: Negação de serviço da aplicação React: Um atacante pode sobrecarregar a aplicação React com requisições maliciosas ou explorar vulnerabilidades de performance que causem travamentos ou lentidão.
- T-14: Negação de serviço das APIs REST: Um atacante pode inundar o Controller Layer com requisições, esgotando recursos do Java Backend e impedindo que usuários legítimos acessem os serviços.
- T-15: Negação de serviço do banco de dados: Consultas complexas ou excessivas, ou ataques diretos ao banco de dados, podem exaurir seus recursos e tornar a aplicação indisponível.
- T-16: Negação de serviço nos servidores de deployment: Ataques de inundação de rede ou exploração de vulnerabilidades nos servidores web/aplicação (Nginx, Apache, Tomcat) podem derrubar o serviço.
- T-17: Negação de serviço no servidor de autenticação: Um ataque ao servidor de autenticação pode impedir que os usuários se autentiquem, tornando a aplicação inacessível.

E - Elevation of Privilege (Elevação de Privilégio)

- T-18: Elevação de privilégio através de vulnerabilidades no backend: Um atacante pode explorar falhas no Controller Layer, Service Layer ou Data Access Layer (e.g., injeção de SQL, deserialização insegura) para obter permissões de acesso mais elevadas.
- T-19: Elevação de privilégio no lado do cliente: Manipulação de tokens de sessão, cookies ou armazenamento local para obter acesso a funcionalidades restritas ou dados de outros usuários.
- T-20: Elevação de privilégio em componentes de autenticação: Vulnerabilidades no servidor de autenticação ou no fluxo de autenticação podem permitir que um atacante obtenha privilégios de administrador ou de outro usuário.

Contramedidas

Contramedidas para Spoofing (Falsificação de Identidade)

- C-1: Autenticação Forte: Implementar mecanismos robustos de autenticação (e.g., MFA, senhas fortes, autenticação baseada em certificados) para usuários e serviços.
- C-2: Uso de OAuth2/OIDC: Utilizar um provedor de identidade confiável (Auth0, Okta) para gerenciar a autenticação e autorização, garantindo a validação de tokens JWT.
- C-3: Validação de Certificados SSL/TLS: Garantir que todos os componentes da aplicação validem os certificados SSL/TLS dos serviços com os quais se comunicam para prevenir ataques Man-in-the-Middle.
- C-4: HSTS (HTTP Strict Transport Security): Forçar o uso de HTTPS em todas as comunicações para proteger contra ataques de downgrade e cookies inseguros.

Contramedidas para Tampering (Violação de Dados)

- C-5: Criptografia em Trânsito (TLS/SSL): Todas as comunicações entre o cliente, backend, servidores de deployment e servidor de autenticação devem ser criptografadas usando TLS/SSL.
- C-6: Validação de Entrada: Implementar validação rigorosa de todos os dados de entrada no lado do cliente (para usabilidade) e, crucialmente, no lado do servidor (Controller Layer) para prevenir injeção de código, scripts maliciosos, etc.
- C-7: Integridade de Dados no Backend: Usar checksums, hashing ou assinaturas digitais para verificar a integridade de dados críticos armazenados ou transmitidos.
- C-8: Controle de Acesso Baseado em Função (RBAC): Implementar RBAC no Service Layer para garantir que os usuários só possam acessar e modificar dados para os quais têm permissão explícita.
- C-9: Proteção contra XSS e CSRF: Implementar cabeçalhos de segurança (CSP,

X-Content-Type-Options), sanitização de entrada e tokens CSRF para proteger a aplicação React e as APIs.
- C-10: Imutabilidade de Dados: Para dados sensíveis ou históricos, considerar a imutabilidade para prevenir alterações não autorizadas.

Contramedidas para Repudiation (Repúdio)

- C-11: Logs de Auditoria Abrangentes: Registrar todas as ações críticas dos usuários (autenticação, acesso a dados, modificações) e administradores, incluindo timestamp e identificação do usuário.
- C-12: Proteção de Logs: Proteger a integridade e a confidencialidade dos logs para evitar manipulação ou exclusão por atacantes.
- C-13: Não Repúdio Digital: Para transações muito críticas, considerar o uso de assinaturas digitais para garantir a autoria.

Contramedidas para Information Disclosure (Divulgação de Informação)

- C-14: Criptografia em Repouso: Criptografar dados sensíveis no banco de dados (K) e quaisquer outros locais de armazenamento.
- C-15: Filtragem de Mensagens de Erro: Evitar mensagens de erro detalhadas que possam vazar informações sensíveis. Implementar mensagens de erro genéricas para o usuário final e logs detalhados apenas para a equipe de desenvolvimento/operação.
- C-16: Gerenciamento Seguro de Segredos: Utilizar um gerenciador de segredos (e.g., HashiCorp Vault, AWS Secrets Manager) para armazenar chaves de API, credenciais de banco de dados e outros segredos, em vez de codificá-los ou armazená-los em arquivos de configuração.
- C-17: Controle de Acesso Mínimo: Conceder o menor privilégio necessário aos usuários, processos e serviços para desempenhar suas funções.
- C-18: Remover Informações Desnecessárias: Garantir que o código-fonte, comentários e metadados não contenham informações sensíveis que possam ser expostas.

Contramedidas para Denial of Service (Negação de Serviço)

- C-19: Limitação de Taxa (Rate Limiting): Implementar limitação de taxa nas APIs REST (H) e no servidor de autenticação (O) para prevenir ataques de força bruta e inundações.
- C-20: Validação de Parâmetros de Entrada: Validar o tamanho e a complexidade dos parâmetros de entrada para evitar ataques de estouro de buffer ou processamento excessivo.
- C-21: Balanceamento de Carga: Utilizar balanceadores de carga para distribuir o tráfego entre múltiplas instâncias dos servidores web/aplicação e backend.
- C-22: Web Application Firewall (WAF): Implementar um WAF para filtrar tráfego malicioso e proteger contra ataques comuns de negação de serviço e outras vulnerabilidades.
- C-23: Monitoramento e Alerta: Monitorar continuamente o desempenho e a

disponibilidade da aplicação e dos servidores, com alertas para anomalias.
- C-24: Auto-Escalabilidade: Configurar a infraestrutura para escalar automaticamente em resposta ao aumento da carga.
- C-25: Otimização de Queries de Banco de Dados: Otimizar as consultas SQL e o esquema do banco de dados para garantir alta performance e evitar bloqueios.

Contramedidas para Elevation of Privilege (Elevação de Privilégio)

- C-26: Princípio do Menor Privilégio: Garantir que cada componente (aplicação React, backend, banco de dados) e cada usuário opere com o conjunto mínimo de permissões necessárias.
- C-27: Gerenciamento de Sessão Seguro: Usar tokens de sessão seguros (e.g., JWT com expiração curta, assinados), armazená-los de forma segura (HTTP-only cookies) e invalidá-los corretamente após o logout ou inatividade.
- C-28: Sanitização de Entrada e Saída: Além da validação de entrada, garantir a sanitização de todas as saídas para prevenir ataques como XSS.
- C-29: Atualizações de Segurança Regulares: Manter todos os componentes do software (sistema operacional, servidores web/aplicação, bibliotecas, frameworks Java, banco de dados) atualizados com os últimos patches de segurança.
- C-30: Testes de Segurança: Realizar testes de penetração, varreduras de vulnerabilidade e revisões de código regulares para identificar e corrigir falhas de segurança.
- C-31: Validação de Tokens JWT: No backend, validar a assinatura, a expiração e as reivindicações (claims) de todos os tokens JWT recebidos do servidor de autenticação.

Markdown

```
### Agentes de Ameaça

* TA-1: Atacante Externo Não Autenticado
* TA-2: Atacante Externo Autenticado
* TA-3: Atacante Interno Malicioso (usuário legítimo comprometido ou funcionário mal-intencionado)
* TA-4: Atacante com Acesso Físico ou Lógico (comprometendo a infraestrutura)

### Ameaças (STRIDE)

**S - Spoofing (Falsificação de Identidade)**
```

* T-1: Falsificação de identidade do usuário: Um atacante pode tentar se passar por um usuário legítimo para obter acesso indevido à aplicação React ou aos recursos do backend.
* T-2: Falsificação de identidade do servidor: Um atacante pode falsificar o servidor de autenticação (Auth0, Okta, ou Spring Security) ou o servidor web/aplicação para interceptar credenciais ou redirecionar usuários para sites maliciosos.
* T-3: Falsificação de requisições de API: Um atacante pode criar requisições falsas para as APIs REST do backend, fazendo-se passar por uma aplicação cliente legítima.

****T - Tampering (Violação de Dados)****

* T-4: Violação de dados em trânsito: Dados trafegando entre a aplicação React, o backend e o servidor de autenticação podem ser interceptados e modificados por um atacante.
* T-5: Violação de dados no lado do cliente: Um atacante pode manipular o estado da aplicação React (via Redux, Context API) para contornar validações ou exibir informações incorretas.
* T-6: Violação de dados no banco de dados: Um atacante com acesso ao backend ou diretamente ao banco de dados pode modificar, excluir ou inserir dados indevidamente.
* T-7: Violação de arquivos de configuração ou binários: Um atacante com acesso ao ambiente de deployment pode alterar arquivos de configuração do servidor web/aplicação ou os binários da aplicação, introduzindo comportamento malicioso.

****R - Repudiation (Repúdio)****

* T-8: Repúdio de ações do usuário: Um usuário malicioso pode negar ter realizado certas ações dentro da aplicação (e.g., transações, modificações de dados) se não houver um log de auditoria adequado.
* T-9: Repúdio de ações do administrador: Um administrador comprometido ou mal-intencionado pode negar alterações feitas na configuração ou nos dados do sistema.

****I - Information Disclosure (Divulgação de Informação)****

* T-10: Divulgação de dados sensíveis: Informações confidenciais (e.g., credenciais de usuário, dados pessoais, informações financeiras) podem ser expostas se não forem devidamente protegidas em trânsito ou em repouso (banco de dados, logs).
* T-11: Divulgação de informações de erro: Mensagens de erro detalhadas no frontend ou backend podem vazar informações sobre a arquitetura interna, versões de software ou dados sensíveis.
* T-12: Divulgação de segredos (chaves de API, credenciais de banco de dados): Segredos armazenados no ambiente de deployment ou no código-fonte podem ser acessados por atacantes.

D - Denial of Service (Negação de Serviço)

* T-13: Negação de serviço da aplicação React: Um atacante pode sobrecarregar a aplicação React com requisições maliciosas ou explorar vulnerabilidades de performance que causem travamentos ou lentidão.
* T-14: Negação de serviço das APIs REST: Um atacante pode inundar o Controller Layer com requisições, esgotando recursos do Java Backend e impedindo que usuários legítimos acessem os serviços.
* T-15: Negação de serviço do banco de dados: Consultas complexas ou excessivas, ou ataques diretos ao banco de dados, podem exaurir seus recursos e tornar a aplicação indisponível.
* T-16: Negação de serviço nos servidores de deployment: Ataques de inundação de rede ou exploração de vulnerabilidades nos servidores web/aplicação (Nginx, Apache, Tomcat) podem derrubar o serviço.
* T-17: Negação de serviço no servidor de autenticação: Um ataque ao servidor de autenticação pode impedir que os usuários se autentiquem, tornando a aplicação inacessível.

E - Elevation of Privilege (Elevação de Privilégio)

* T-18: Elevação de privilégio através de vulnerabilidades no backend: Um atacante pode explorar falhas no Controller Layer, Service Layer ou Data Access Layer (e.g., injeção de SQL, deserialização insegura) para obter permissões de acesso mais elevadas.
* T-19: Elevação de privilégio no lado do cliente: Manipulação de tokens de sessão, cookies ou armazenamento local para obter acesso a funcionalidades restritas ou dados de outros usuários.
* T-20: Elevação de privilégio em componentes de autenticação: Vulnerabilidades no servidor de autenticação ou no fluxo de autenticação podem permitir que um atacante obtenha privilégios de administrador ou de outro usuário.

Contramedidas

Contramedidas para Spoofing (Falsificação de Identidade)

* C-1: Autenticação Forte: Implementar mecanismos robustos de autenticação (e.g., MFA, senhas fortes, autenticação baseada em certificados) para usuários e serviços.
* C-2: Uso de OAuth2/OIDC: Utilizar um provedor de identidade confiável (Auth0, Okta) para gerenciar a autenticação e autorização, garantindo a validação de tokens JWT.
* C-3: Validação de Certificados SSL/TLS: Garantir que todos os componentes da aplicação validem os certificados SSL/TLS dos serviços com os quais se comunicam para prevenir ataques Man-in-the-Middle.
* C-4: HSTS (HTTP Strict Transport Security): Forçar o uso de HTTPS em todas as

comunicações para proteger contra ataques de downgrade e cookies inseguros.

****Contramedidas para Tampering (Violação de Dados)****

* C-5: Criptografia em Trânsito (TLS/SSL): Todas as comunicações entre o cliente, backend, servidores de deployment e servidor de autenticação devem ser criptografadas usando TLS/SSL.
* C-6: Validação de Entrada: Implementar validação rigorosa de todos os dados de entrada no lado do cliente (para usabilidade) e, crucialmente, no lado do servidor (Controller Layer) para prevenir injeção de código, scripts maliciosos, etc.
* C-7: Integridade de Dados no Backend: Usar checksums, hashing ou assinaturas digitais para verificar a integridade de dados críticos armazenados ou transmitidos.
* C-8: Controle de Acesso Baseado em Função (RBAC): Implementar RBAC no Service Layer para garantir que os usuários só possam acessar e modificar dados para os quais têm permissão explícita.
* C-9: Proteção contra XSS e CSRF: Implementar cabeçalhos de segurança (CSP, X-Content-Type-Options), sanitização de entrada e tokens CSRF para proteger a aplicação React e as APIs.
* C-10: Imutabilidade de Dados: Para dados sensíveis ou históricos, considerar a imutabilidade para prevenir alterações não autorizadas.

****Contramedidas para Repudiation (Repúdio)****

* C-11: Logs de Auditoria Abrangentes: Registrar todas as ações críticas dos usuários (autenticação, acesso a dados, modificações) e administradores, incluindo timestamp e identificação do usuário.
* C-12: Proteção de Logs: Proteger a integridade e a confidencialidade dos logs para evitar manipulação ou exclusão por atacantes.
* C-13: Não Repúdio Digital: Para transações muito críticas, considerar o uso de assinaturas digitais para garantir a autoria.

****Contramedidas para Information Disclosure (Divulgação de Informação)****

* C-14: Criptografia em Repouso: Criptografar dados sensíveis no banco de dados (K) e quaisquer outros locais de armazenamento.
* C-15: Filtragem de Mensagens de Erro: Evitar mensagens de erro detalhadas que possam vazar informações sensíveis. Implementar mensagens de erro genéricas para o usuário final e logs detalhados apenas para a equipe de desenvolvimento/operação.
* C-16: Gerenciamento Seguro de Segredos: Utilizar um gerenciador de segredos (e.g., HashiCorp Vault, AWS Secrets Manager) para armazenar chaves de API, credenciais de banco de dados e outros segredos, em vez de codificá-los ou armazená-los em arquivos de

configuração.
* C-17: Controle de Acesso Mínimo: Conceder o menor privilégio necessário aos usuários, processos e serviços para desempenhar suas funções.
* C-18: Remover Informações Desnecessárias: Garantir que o código-fonte, comentários e metadados não contenham informações sensíveis que possam ser expostas.

****Contramedidas para Denial of Service (Negação de Serviço)****

* C-19: Limitação de Taxa (Rate Limiting): Implementar limitação de taxa nas APIs REST (H) e no servidor de autenticação (O) para prevenir ataques de força bruta e inundações.
* C-20: Validação de Parâmetros de Entrada: Validar o tamanho e a complexidade dos parâmetros de entrada para evitar ataques de estouro de buffer ou processamento excessivo.
* C-21: Balanceamento de Carga: Utilizar balanceadores de carga para distribuir o tráfego entre múltiplas instâncias dos servidores web/aplicação e backend.
* C-22: Web Application Firewall (WAF): Implementar um WAF para filtrar tráfego malicioso e proteger contra ataques comuns de negação de serviço e outras vulnerabilidades.
* C-23: Monitoramento e Alerta: Monitorar continuamente o desempenho e a disponibilidade da aplicação e dos servidores, com alertas para anomalias.
* C-24: Auto-Escalabilidade: Configurar a infraestrutura para escalar automaticamente em resposta ao aumento da carga.
* C-25: Otimização de Queries de Banco de Dados: Otimizar as consultas SQL e o esquema do banco de dados para garantir alta performance e evitar bloqueios.

****Contramedidas para Elevation of Privilege (Elevação de Privilégio)****

* C-26: Princípio do Menor Privilégio: Garantir que cada componente (aplicação React, backend, banco de dados) e cada usuário opere com o conjunto mínimo de permissões necessárias.
* C-27: Gerenciamento de Sessão Seguro: Usar tokens de sessão seguros (e.g., JWT com expiração curta, assinados), armazená-los de forma segura (HTTP-only cookies) e invalidá-los corretamente após o logout ou inatividade.
* C-28: Sanitização de Entrada e Saída: Além da validação de entrada, garantir a sanitização de todas as saídas para prevenir ataques como XSS.
* C-29: Atualizações de Segurança Regulares: Manter todos os componentes do software (sistema operacional, servidores web/aplicação, bibliotecas, frameworks Java, banco de dados) atualizados com os últimos patches de segurança.
* C-30: Testes de Segurança: Realizar testes de penetração, varreduras de vulnerabilidade e revisões de código regulares para identificar e corrigir falhas de segurança.
* C-31: Validação de Tokens JWT: No backend, validar a assinatura, a expiração e as reivindicações (claims) de todos os tokens JWT recebidos do servidor de autenticação.