Práticas de segurança incorporadas no projeto.

- Variáveis de Ambiente: Todos os dados sensíveis, como chaves de APIs e configurações de conexão com bancos de dados, serão armazenados em arquivos de variáveis de ambiente (.env). Esses arquivos não serão incluídos no controle de versão (e.g., Git) .
- Kubernetes Secrets: Dados sensíveis serão gerenciados usando Kubernetes Secrets.
- TLS/SSL: Todas as comunicações entre serviços, bem como com usuários finais, serão criptografadas usando TLS/SSL para garantir a confidencialidade e integridade dos dados transmitidos.
- MTLS (Mutual TLS): Implementação de MTLS para autenticação mútua entre os microsserviços, garantindo que apenas serviços autenticados possam se comunicar entre si.
- Autenticação e Autorização: Uso de OAuth2 para autenticação e autorização, garantindo que apenas usuários e serviços autenticados possam acessar recursos.
- RBAC (Role-Based Access Control): Configuração de políticas de RBAC no Kubernetes para limitar as permissões de usuários e serviços apenas ao necessário.
- Network Policies: Uso de políticas de rede do Kubernetes para controlar o tráfego de rede entre os pods, limitando a comunicação apenas aos serviços necessários.
- Pod Security Policies: Implementação de políticas de segurança de pod para garantir que os pods sejam executados com permissões mínimas e em conformidade com as práticas de segurança.
- Imagens de Container Seguras: Uso de imagens de container provenientes de fontes confiáveis e verificáveis, ou seja, imagens oficiais.
- Imagens Imutáveis: Uso de tags imutáveis para imagens de container, garantindo que uma vez que uma imagem é implantada, ela não seja alterada.
- Logs Centralizados: Implementação de soluções de logging centralizadas, como ELK Stack (Elasticsearch, Logstash, Kibana), para coletar e analisar logs de todos os serviços.
- Monitoramento Contínuo: Usar ferramentas como Prometheus e Grafana para monitoramento contínuo da infraestrutura e serviços, detectando anomalias e respondendo rapidamente a incidentes de segurança.
- Análise de Vulnerabilidades: Uso de ferramentas automatizadas de análise de vulnerabilidades para identificar e corrigir vulnerabilidades em código e infraestrutura.
- Atualizações e Patches: Aplicação regular de patches e atualizações para todos os componentes do sistema, incluindo imagens de container, serviços e infraestrutura de Kubernetes.
- Planos de Resposta a Incidentes: Desenvolvimento e manutenção de um plano de resposta a incidentes bem definido para lidar com potenciais violações de segurança.