A tua empresa está a desenvolver uma solução para promover compra e venda online, de produtos, de todo o tipo de produtos. Terá de haver um particionamento dos utilizadores, como todos os produtos são permitidos, certos utilizadores, por exemplo, menores de 18 anos, não poderão ver nem aceder a produtos derivados do tabaco, alcool ou pornografia.

A solução, donominada de Amazon++, deverá ser acessível por internet em qualquer browser (provavelmente terá de se definir uma lista de browsers inicial, onde a aplicação será efectivamente testada e deverá ser barrada a sua utilização para outros browsers, a não ser que se convença o cliente de que a aplicação pode funcionar noutros browsers sem problemas de segurança).

Do ponto de vista de utilizadores, haverá 3 tipos:

1. Os gestores da plataforma, com acesso total, podendo colocar restrições em certos utilizadores ou ver informação apenas acessível a eles, nomeadamente contactos das pessoas e a imagem de um documento oficial que é obrigatório para o completo registo no sistema. Eles serão também responsáveis pela manutenção e resolução de conflitos caso não haja entendimento entre comprador e vendedor.

2. Utilizador-vendedor, utilizadores que tanto podem comprar como vender, para atingirem este tipo, os utilizadores devem fornecer mais informações à plataforma como o número de identificação nacional, uma morada válida para onde é enviado um pin, e a imagem de um documento oficial;

3. Utilizador comprador, com menos restrições que o anterior, mas que apenas pode efectuar compras no site.

Para além destes tipos, os utilizadores são ainda categorizados conforme a idade legal, havendo certos produtos que não podem ser adquiridos por certos utilizadores.

O sistema deve integrar com diversos meios de pagamento (MBWay, Multibanco, Transferência Bancária, Cartão de Crédito ou Débito, Paypal, etc).

O sistema deve ainda ter um mecanismo de detecção se o produto ou produtos foram colocados na categoria correcta, este mecanismo utiliza um algoritmo inovador de inteligência artificial que a tua empresa desenvolveu, mas que ainda não foi totalmente posto à prova.

A arquitectura da aplicação resta a definir bem como requisitos de backup, segurança, encriptação de dados, certificação do algoritmo de categorização dos produtos, etc.

Software Secure LifeCycle

1. Requisitos de Segurança Funcional do Sistema

1. **Controlo de Acesso Granular:**

    - **Requisito:** Implementar um sistema de controlo de acesso granular, atribuindo permissões específicas com base nos papéis dos utilizadores (gestores, vendedores, compradores). Garantir que apenas utilizadores autorizados possam aceder e modificar informações sensíveis, como dados de transações e detalhes de contacto.

2. **Monitorização de Atividades Suspeitas:**
    - **Requisito:** Estabelecer mecanismos de monitorização em tempo real para detetar atividades suspeitas ou comportamentos anómalos. Configurar alertas para notificar os administradores sobre potenciais violações de segurança, como múltiplas tentativas de autenticação falhadas ou acessos não autorizados.
3. **Encriptação de Dados Sensíveis:**
    - **Requisito:** Aplicar encriptação de ponta a ponta para proteger dados sensíveis, incluindo informações de pagamento, detalhes de identificação do utilizador e documentos oficiais. Garantir que a transmissão e o armazenamento desses dados sejam seguros contra potenciais ameaças de interceptação.
4. **Validação de Dados do Utilizador:**
    - **Requisito:** Implementar um processo rigoroso de validação de dados do utilizador durante o registo, especialmente para utilizadores-vendedores. Exigir informações verificáveis, como número de identificação nacional e morada, para garantir a autenticidade das contas e reduzir a possibilidade de contas fraudulentas.
5. **Proteção contra Ataques de Injeção:**
    - **Requisito:** Utilizar práticas seguras de programação para prevenir ataques de injeção, como SQL injection e cross-site scripting (XSS). Validar e sanitizar todas as entradas de dados do utilizador para evitar a execução de comandos maliciosos ou scripts no sistema.
6. **Atualizações Seguras e Auditoria:**
    - **Requisito:** Garantir que todas as atualizações de software e patches de segurança sejam aplicados de forma segura e oportuna. Implementar um sistema de registo de auditoria para rastrear alterações no sistema, facilitando a identificação de eventos suspeitos e garantindo a conformidade com as políticas de segurança.

2. Requisitos de Design do Sistema

1. **Mecanismo de Autenticação e Autorização:**
    - **Requisito:** Implementar um mecanismo de autenticação robusto que não possa ser contornado ou adulterado. Além disso, autorizar utilizadores após a autenticação com base nos seus papéis e privilégios.
    - **Justificação:** Dada a variedade de níveis de acesso para diferentes tipos de utilizadores (gestores, vendedores, compradores), é crucial garantir que apenas utilizadores autorizados possam realizar ações correspondentes aos seus papéis. Isso ajuda a prevenir acesso não autorizado a dados sensíveis e funcionalidades dentro da plataforma Amazon++.
2. **Validação e Separação de Dados:**
    - **Requisito:** Separar rigorosamente dados e instruções de controlo, garantindo que instruções de controlo recebidas de fontes não confiáveis nunca sejam processadas. Implementar uma abordagem que valide explicitamente todos os dados recebidos para evitar entradas maliciosas.

- **Justificação:** Ao impor uma clara separação entre dados e instruções de controlo, o sistema reduz o risco de comandos não autorizados afetarem o seu comportamento. A validação explícita de dados adiciona uma camada extra de segurança, mitigando os potenciais ataques de injeção ou manipulação de dados.

3. **Criptografia e Manipulação de Dados Sensíveis:**

   - **Requisito:** Utilizar corretamente a criptografia para proteger dados sensíveis, como identificação do utilizador, detalhes de contacto e informações de transações. Identificar claramente dados sensíveis e definir procedimentos seguros de manipulação, garantindo armazenamento e transmissão encriptados.
   - **Justificação:** Com a integração de vários métodos de pagamento e o armazenamento de informações pessoais, a criptografia torna-se crucial. Uma encriptação adequada protege contra acesso não autorizado, proporcionando um ambiente seguro para os dados do utilizador. Este requisito está alinhado com a ênfase do IEEE Center for Secure Design na utilização eficaz da criptografia.

Security Requirements

2. Security Goals

2.1 Threads

Para o sistema Amazon++, é vital identificar potenciais ameaças e estabelecer metas de segurança para mitigar essas ameaças. Aqui estão algumas considerações específicas para o sistema em questão:

1. **Advanced Persistent Threat (APTs):**

   - **Meta de Segurança:** Implementar medidas de deteção avançadas, como sistemas de análise comportamental e monitorização contínua, para identificar padrões de atividade suspeita que possam indicar a presença de APTs. Manter sistemas de segurança atualizados para resistir a técnicas avançadas de evasão.

2. **Backdoors:**

   - **Meta de Segurança:** Realizar auditorias regulares de código para identificar e eliminar potenciais backdoors. Implementar políticas de gestão de acesso rigorosas e monitorização de atividades para detetar comportamentos anómalos que possam indicar a existência de backdoors.

3. **Phishing:**

   - **Meta de Segurança:** Conduzir programas de sensibilização sobre segurança para utilizadores, ensinando-os a reconhecer e evitar ataques de phishing. Implementar filtros anti-phishing robustos no sistema de correio eletrónico e na interface do utilizador para bloquear tentativas de phishing.

4. **Ransomware:**

- **Meta de Segurança:** Realizar backups regulares dos dados do sistema e implementar um sistema de recuperação de desastres eficaz. Utilizar antivírus e anti-malware atualizados para detetar e bloquear ransomware. Educar os utilizadores sobre práticas seguras de navegação e download.

5. **Denial of Service (DoS):**

   - **Meta de Segurança:** Implementar soluções de mitigação de DoS, como firewalls e sistemas de deteção de intrusões. Utilizar serviços de CDN (Content Delivery Network) para distribuir o tráfego e minimizar o impacto de ataques de negação de serviço.

6. **Web Application Security:**

   - **Meta de Segurança:** Realizar testes de segurança regulares, incluindo testes de penetração e varreduras de vulnerabilidades, para identificar e corrigir falhas na segurança da aplicação web. Implementar firewalls de aplicação web (WAF) para proteger contra ataques comuns.

7. **Exploits e Vulnerabilidades:**

   - **Meta de Segurança:** Manter todos os sistemas e software atualizados com os patches de segurança mais recentes. Realizar avaliações regulares de vulnerabilidades e corrigir quaisquer falhas identificadas. Implementar uma política de gestão de patches eficaz.

8. **Spyware e Malware:**

   - **Meta de Segurança:** Utilizar software antivírus e anti-malware atualizado para detetar e remover spyware e malware. Implementar políticas de utilizador que limitem a instalação de software não autorizado. Monitorizar constantemente a atividade do sistema para comportamentos suspeitos.

2.2. Defenses

Considerando as defesas mencionadas, aqui estão algumas estratégias específicas de defesa para o sistema Amazon++:

1. **Controlo de Acesso ao Computador:**

   - Implementar controlos de acesso rigorosos para garantir que apenas utilizadores autorizados tenham acesso ao sistema. Utilizar gestão de identidade e acesso (IAM) para definir e aplicar políticas de acesso com base nos papéis dos utilizadores.

2. **Segurança de Aplicações:**

   - Realizar testes de segurança regulares, incluindo análises estáticas e dinâmicas de código, para identificar e corrigir vulnerabilidades na aplicação. Utilizar firewalls de aplicação web (WAF) para proteger contra ataques comuns.

3. **Software Antivírus:**

   - Implementar um software antivírus eficaz em todas as camadas do sistema para detetar e remover malware. Manter as definições de antivírus atualizadas regularmente para garantir proteção contra as ameaças mais recentes.

4. **Codificação Segura:**
    - Adotar práticas de codificação seguras para desenvolver o sistema. Garantir que a equipa de desenvolvimento está ciente das melhores práticas de segurança e que o código é auditado regularmente para identificar potenciais vulnerabilidades.
5. **Seguro por Padrão e por Design:**
    - Projetar o sistema com segurança desde o início, considerando os princípios de segurança por padrão e por design. Isso inclui a implementação de controlos de segurança em todas as camadas da aplicação e da infraestrutura.
6. **Sistemas Operativos Seguros:**
    - Utilizar sistemas operativos que sejam conhecidos por serem seguros e mantê-los atualizados com os patches de segurança mais recentes. Configurar os sistemas operativos de forma segura, desativando serviços não essenciais e aplicando as configurações recomendadas.
7. **Autenticação e Autenticação Multifatorial (MFA):**
    - Exigir autenticação robusta para todos os utilizadores. Implementar a autenticação multifatorial para adicionar uma camada adicional de segurança, exigindo mais de um método de verificação de identidade.
8. **Autorização e Segurança Centrada nos Dados:**
    - Implementar um sistema de autorização baseado em políticas para controlar o acesso a recursos sensíveis. Adotar uma abordagem de segurança centrada nos dados, garantindo que os dados sejam protegidos independentemente de onde residam.
9. **Encriptação:**
    - Utilizar encriptação para proteger dados em repouso, em trânsito e em processamento. Isso inclui encriptação de comunicações, encriptação de bases de dados e qualquer outra forma de dados armazenados no sistema.
10. **Firewall e Sistema de Detecção de Intrusões (IDS):**
    - Implementar firewalls para monitorizar e controlar o tráfego de rede. Além disso, instalar um Sistema de Detecção de Intrusões para identificar padrões de comportamento suspeitos e alertar sobre potenciais ameaças.
11. **Gateway Móvel Seguro e Autoproteção de Aplicações em Tempo de Execução (RASP):**
    - Se a aplicação for acedida por dispositivos móveis, implementar um Gateway Móvel Seguro para proteger contra ameaças específicas para dispositivos móveis. Considere também a implementação de RASP para proteger a aplicação durante a execução.

    RASP é uma tecnologia de segurança que é implantada junto com a aplicação para monitorizar e proteger a aplicação em tempo de execução. Isso ajuda a proteger a aplicação contra ameaças como injeção de código, ataques de negação de serviço e exploração de vulnerabilidades.

2.3. Confidentality

Para garantir a confidencialidade no sistema Amazon++, é essencial implementar medidas que protejam os dados contra leituras não autorizadas, garantindo que apenas utilizadores autorizados possam aceder a informações sensíveis. Aqui estão algumas estratégias específicas para alcançar a confidencialidade no contexto do sistema Amazon++:

1. **Encriptação de Dados:**
    - **Implementação:** Utilizar algoritmos de encriptação robustos para proteger dados em repouso, em trânsito e em processamento. Certificar-se de que as informações sensíveis, como detalhes de pagamento e dados pessoais, são encriptadas para prevenir acessos não autorizados.
2. **Gestão de Chaves Segura:**
    - **Implementação:** Estabelecer práticas seguras para a gestão de chaves criptográficas. Garantir que as chaves de encriptação são armazenadas de forma segura, com acesso restrito apenas a pessoal autorizado. Implementar rotação periódica de chaves para manter a segurança a longo prazo.
3. **Controlos de Acesso Granular:**
    - **Implementação:** Configurar e aplicar controlos de acesso granulares para garantir que apenas utilizadores autorizados tenham acesso a determinadas informações. Implementar políticas de autorização que restrinjam o acesso com base nos papéis e responsabilidades dos utilizadores.
4. **Monitorização de Acessos:**
    - **Implementação:** Implementar sistemas de monitorização contínua para detetar padrões de acesso anómalos ou tentativas não autorizadas. Alertar imediatamente a equipa de segurança sobre atividades suspeitas para uma resposta rápida.
5. **Proteção contra Ameaças Internas:**
    - **Implementação:** Estabelecer medidas de segurança que protejam contra ameaças internas, como acesso não autorizado por utilizadores privilegiados. Implementar auditorias regulares para identificar atividades suspeitas dentro da organização.
6. **Políticas de Gestão de Informação:**
    - **Implementação:** Desenvolver e implementar políticas claras de gestão de informação que delineiem quem tem acesso a quais tipos de dados e sob quais circunstâncias. Educar os utilizadores sobre a importância de proteger informações confidenciais.
7. **Segurança em Comunicações:**
    - **Implementação:** Utilizar protocolos seguros para comunicações, como HTTPS, para proteger dados em trânsito. Certificar-se de que todas as transações e comunicações entre o utilizador e a plataforma são realizadas de forma segura.
8. **Treinamento de Sensibilização para Segurança:**

- **Implementação:** Realizar programas de treinamento para sensibilização em segurança, focando na importância da confidencialidade dos dados. Ensinar aos utilizadores práticas seguras, como não partilhar credenciais e manter a segurança física dos dispositivos.

2.4. Integrity

1. **Assinaturas Digitais:**
   - **Implementação:** Utilizar assinaturas digitais para verificar a autenticidade e a integridade dos dados. Isso pode ser aplicado especialmente em transações financeiras, confirmações de pedidos e outros registros críticos.
2. **Hashes Criptográficos:**
   - **Implementação:** Utilizar funções de hash criptográfico para calcular e verificar resumos (hashes) dos dados. Armazenar esses hashes de forma segura e compará-los regularmente para detetar alterações não autorizadas.
3. **Controlos de Acesso e Autorização:**
   - **Implementação:** Reforçar controlos de acesso e políticas de autorização para garantir que apenas utilizadores autorizados possam modificar dados sensíveis. Limitar o acesso a funções críticas do sistema e atribuir permissões com base nos princípios do menor privilégio necessário.
4. **Backup e Recuperação:**
   - **Implementação:** Implementar um sistema robusto de backup e recuperação para proteger contra perda de dados e corrupção. Regularmente testar a recuperação de dados para garantir que o processo seja eficaz em preservar a integridade.
5. **Registos de Auditoria:**
   - **Implementação:** Manter registos de auditoria detalhados para todas as atividades críticas no sistema. Isso inclui alterações em dados sensíveis, transações financeiras e alterações nas configurações do sistema. Analisar regularmente os registos para identificar anomalias.
6. **Validação de Dados:**
   - **Implementação:** Implementar processos de validação de dados rigorosos para garantir que apenas dados válidos e autorizados sejam aceites e processados pelo sistema. Isso ajuda a prevenir ataques de injeção e manipulação de dados.
7. **Não-Repúdio:**
   - **Implementação:** Utilizar mecanismos que garantam não-repúdio, especialmente em transações críticas. Isso assegura que as partes envolvidas não podem negar a autenticidade da transação.
8. **Controlo de Mudanças:**
   - **Implementação:** Implementar um processo formal de controlo de mudanças para todas as atualizações no sistema. Certificar-se de que as mudanças são documentadas, autorizadas e testadas antes de serem implementadas para evitar impactos não intencionais.

2.5. Availability

1. **Backup e Recuperação:**
    - **Implementação:** Estabelecer um sistema de backup e recuperação robusto para proteger contra a perda de dados. Garantir backups frequentes e testar regularmente os procedimentos de recuperação para assegurar a rápida restauração dos serviços em caso de interrupções.
2. **Replicação de Dados:**
    - **Implementação:** Utilizar técnicas de replicação de dados para criar cópias redundantes de informações críticas. Isso ajuda a garantir que, em caso de falha de hardware ou outros eventos adversos, os dados ainda estejam disponíveis a partir de fontes alternativas.
3. **Balanceamento de Carga:**
    - **Implementação:** Implementar mecanismos de balanceamento de carga para distribuir equitativamente o tráfego entre diferentes servidores e recursos. Isso não apenas melhora o desempenho, mas também ajuda a prevenir a sobrecarga de sistemas específicos.
4. **Redundância de Servidores e Infraestrutura:**
    - **Implementação:** Configurar redundância em níveis críticos da infraestrutura, como servidores, fontes de energia e conexões de rede. Isso assegura que, em caso de falha em um componente, o sistema possa continuar a operar de forma ininterrupta.
5. **Monitorização Proativa:**
    - **Implementação:** Implementar sistemas de monitorização proativa para identificar potenciais problemas antes que causem interrupções significativas. Isso inclui monitorizar o desempenho do sistema, a utilização de recursos e alertas para eventos críticos.
6. **Planos de Continuidade de Negócios:**
    - **Implementação:** Desenvolver planos de continuidade de negócios que incluam estratégias específicas para manter a disponibilidade dos serviços em cenários de emergência. Isso pode incluir a ativação de sistemas de contingência ou a transferência de operações para ambientes alternativos.
7. **Resposta a Incidentes Rápida:**
    - **Implementação:** Estabelecer procedimentos eficazes de resposta a incidentes para lidar rapidamente com eventos que possam afetar a disponibilidade. Isso inclui a identificação rápida da causa raiz e a implementação de medidas corretivas.
8. **Atualizações Programadas:**
    - **Implementação:** Planejar atualizações e manutenções programadas de forma cuidadosa, minimizando o impacto na disponibilidade do sistema. Comunicar proativamente aos utilizadores sobre qualquer tempo de inatividade planejado.
9. **Contingência de Redes:**

- **Implementação:** Garantir contingência de redes para evitar a interrupção dos serviços devido a falhas na conectividade. Isso pode incluir a utilização de múltiplos fornecedores de serviços de internet ou a implementação de redes resilientes.