Kalled Abdala – 234846

Questão 2: Teoria de Criptografia

Diferenças entre Criptografia Simétrica e Assimétrica

Criptografia Simétrica:

- **Definição**: A criptografia simétrica usa uma única chave para tanto criptografar quanto descriptografar os dados. Isso significa que tanto o remetente quanto o destinatário precisam ter a mesma chave secreta.
- **Exemplo prático**: O AES (Advanced Encryption Standard) é um algoritmo simétrico comum. Um exemplo de aplicação seria quando você criptografa arquivos em seu computador com uma senha e, em seguida, usa a mesma senha para descriptografá-los.
- **Vantagens**:
    - Mais rápido que a criptografia assimétrica, pois utiliza operações matemáticas mais simples.
    - Menor uso de recursos computacionais.
- **Desvantagens**:
    - A chave precisa ser compartilhada de forma segura. Caso a chave seja interceptada, o atacante poderá descriptografar os dados.
    - Não é escalável, já que se a quantidade de remetentes e destinatários for grande, o número de chaves necessárias cresce exponencialmente.

Criptografia Assimétrica:

- **Definição**: Na criptografia assimétrica, são usadas duas chaves distintas: uma pública e uma privada. A chave pública é usada para criptografar as informações e a chave privada é usada para descriptografá-las.
- **Exemplo prático**: O RSA é um exemplo clássico de criptografia assimétrica. Quando você envia um e-mail criptografado usando a chave pública de um destinatário, somente ele poderá descriptografá-lo com sua chave privada.
- **Vantagens**:
    - Mais segura para comunicação entre partes que nunca se encontraram, pois não há necessidade de compartilhar uma chave secreta.
    - Escalável, pois uma chave pública pode ser compartilhada com múltiplas pessoas.

- **Desvantagens**:
    - Lenta em comparação com a criptografia simétrica, devido à complexidade dos algoritmos.
    - Exige mais poder computacional.

Quando Usar Cada Tipo de Criptografia

- **Criptografia Simétrica** é adequada para cenários onde a chave secreta pode ser compartilhada com segurança, como em ambientes fechados onde há controle total sobre a comunicação, como comunicação interna em uma empresa.
- **Criptografia Assimétrica** é mais indicada para a troca segura de informações entre partes que não se conhecem ou que não têm um canal seguro para compartilhar uma chave, como em sistemas de e-mail seguro ou comunicação em redes públicas.

Questão 3: Mitigação de Ataques

Ataques Comuns em Aplicações Web

1. **Cross-Site Request Forgery (CSRF)**:
    - **Descrição**: O CSRF é um ataque onde um usuário é induzido a executar ações indesejadas em um site no qual está autenticado. O atacante envia uma solicitação maliciosa em nome da vítima sem seu consentimento.
    - **Mitigação**: A melhor prática para mitigar CSRF é usar tokens CSRF, que são gerados de forma única para cada requisição. O token é incluído no formulário da aplicação, e a requisição só será processada se o token enviado na solicitação coincidir com o token esperado pelo servidor, como implementado no Flask-WTF.
2. **Cross-Site Scripting (XSS)**:
    - **Descrição**: O XSS permite que atacantes injetem scripts maliciosos nas páginas web que são visualizadas por outros usuários. Isso pode ser usado para roubar informações sensíveis como cookies e credenciais de login.
    - **Mitigação**: Para mitigar XSS, deve-se sempre validar e higienizar qualquer dado de entrada que possa ser executado como código, como em campos de texto. Além disso, é recomendada a implementação de políticas de segurança como o Content Security Policy (CSP).
3. **SQL Injection**:
    - **Descrição**: O SQL Injection ocorre quando um atacante insere código SQL malicioso em um campo de entrada, o que permite manipular a base de dados da aplicação, podendo vazar dados sensíveis ou modificar informações.

- **Mitigação**: A maneira mais eficaz de prevenir SQL Injection é usar **prepared statements** (declarações preparadas), que separam os dados da consulta SQL. Isso garante que os dados fornecidos pelo usuário sejam tratados como dados e não como comandos SQL.