1.SQL Injection

O criminoso virtual explora uma vulnerabilidade, inserindo uma instrução SQL mal-intencionada em um campo de entrada. Mais uma vez, o sistema não filtra a entrada do usuário corretamente para os caracteres em uma instrução de SQL em sites ou qualquer banco de dados SQL. Os criminosos podem falsificar uma identidade, modificar os dados existentes, destruir os dados ou se tornar os administradores do servidor do banco de dados. **O objetivo é explorar a falta de validação ou sanitização de dados**, permitindo ao atacante acessar, modificar ou excluir dados do banco de dados, além de executar operações não autorizadas.

Os ataques de SQL Injection acontecem quando o aplicativo web incorpora diretamente dados não confiáveis em consultas SQL sem uma validação adequada. O invasor explora essa falta de segurança injetando instruções SQL maliciosas nos campos de entrada. O processo geralmente envolve os seguintes passos:

- **Identificação de vulnerabilidades:** O atacante procura campos de entrada, como caixas de pesquisa ou formulários de login, onde dados inseridos pelo usuário são diretamente incorporados em consultas SQL.
- **Inserção de instruções maliciosas:** O invasor insere instruções SQL maliciosas nos campos de entrada. Por exemplo, em vez de inserir um nome de usuário válido, o atacante pode inserir ' OR 1=1 – em um campo de login.
- **Execução de instrução maliciosa:** O aplicativo web, sem uma validação adequada, incorpora a instrução SQL maliciosa na consulta e a executa no banco de dados. No exemplo acima, a consulta se tornaria SELECT * FROM usuarios WHERE nome_usuario = '' OR 1=1 --'.
- **Exploração:** Como o 1=1 sempre será verdadeiro, a consulta retornará todos os registros da tabela de usuários, permitindo ao invasor acessar informações de outros usuários ou até mesmo contornar a autenticação.

2.Exemplos de ataques de SQL injection

Um atacante insere ' OR 'a'='a em um campo de login, enganando o aplicativo a pensar que um usuário válido está tentando fazer login, concedendo acesso não autorizado. Um invasor envia uma consulta SQL maliciosa que exclui todas as entradas de um banco de dados, resultando na perda de dados.

3.Prevenção e mitigação de SQL injection

A prevenção e a mitigação de SQL Injection são fundamentais para proteger aplicativos web contra esse tipo de ataque. Utilizar consultas parametrizadas, validar entradas de dados e adotar boas práticas de segurança são passos cruciais para garantir a integridade e a confidencialidade dos dados em um aplicativo.

3.1 Validação de entradas

Valide e filtre rigorosamente todas as entradas de dados fornecidas pelos usuários. Certifique-se de que os dados sejam do tipo esperado e não contenham caracteres especiais não autorizados.

3.2 Consultas parametrizadas

Consultas parametrizadas são uma das formas mais eficazes de prevenir SQL Injection. Use consultas parametrizadas ou consultas preparadas, dependendo da linguagem de programação e do sistema de gerenciamento de banco de dados (DBMS) que você está utilizando. Elas separam os valores dos parâmetros SQL, garantindo que os dados do usuário não sejam tratados como parte do comando SQL e assim, impede que os invasores injetem código malicioso nas consultas. A técnica correta é utilizar marcadores de posição (placeholders) na consulta SQL e inserir diretamente os valores de entrada no lugar dos placeholders. Essa abordagem separa os dados dos comandos SQL, prevenindo o SQL Injection.

3.3 Evite construção de consultas manuais

Evite construir consultas SQL manualmente concatenando strings. Isso torna seu aplicativo vulnerável a injeções de SQL.

3.4 Princípio do menor privilégio

Configure as permissões do banco de dados para que o aplicativo tenha apenas as permissões necessárias para realizar suas operações. Evite permissões de gravação quando somente a leitura é necessário.

3.5 Monitoramento e registros de auditoria

Implemente monitoramento e registros de auditoria para detectar atividades incomuns ou tentativas de injeção de SQL em tempo real.