Agenda 07: Tipos De Invasões E Vulnerabilidades

Tecnologias da Informação III

Mônica Zungalo Quintal

"Estudamos sobre diversos tipos de ataques que podem ser realizados, tendo as empresas como alvos. Muitos desses ataques podem ter fins políticos, ideológicos ou simplesmente com o intuito de paralisar um serviço ou toda uma empresa. Os ataques do tipo SQL Injection podem ser executados por qualquer pessoa com um pouco de conhecimentos de páginas HTML, PHP, ASP.net, JavaScript e banco de dados pode executar.

Para sintetizar os conhecimentos sobre esse tipo de ataque, pesquise, exemplifique e explique passo a passo, como podemos obter dados não autorizados de um BD qualquer utilizando SQL Injection.

O exercício deve ser entregue em documento do formato Word no ambiente virtual de aprendizagem. Observação: explique como cada comando em SQL se comporta ao ser executado pelo SGBD."

Conforme estudado nesta agenda, SQL Injection consiste em uma técnica de ataque cibernético que explora vulnerabilidades em sistemas de gerenciamento de banco de dados para executar comandos SQL maliciosos, de forma não autorizada. Essa técnica ocorre quando um aplicativo web ou sistema aceita entradas não confiáveis de usuários e não valida ou filtra essas entradas de maneira adequada antes de incorporá-las em instruções SQL que são enviadas ao SGBD.

O ataque de SQL Injection permite que um invasor insira comandos SQL arbitrários em uma aplicação, com o objetivo de:

1. Acessar, modificar ou excluir dados no banco de dados.
2. Obter informações confidenciais do banco de dados, como senhas, informações de login ou dados pessoais.
3. Realizar ações maliciosas no sistema, como elevar privilégios de usuário.
4. Causar indisponibilidade do sistema, manipulando o banco de dados de forma destrutiva.

Para obter estes dados não autorizados, o invasor deve seguir os passos:

1. Identificar o formulário vulnerável.
2. Determinar a consulta SQL que é executada pelo formulário.
3. Inserir um comando SQL malicioso no campo vulnerável.
4. Enviar o formulário vulnerável.

Existem diversas variedades de vulnerabilidades, ataques e técnicas de SQL Injection, entre elas:

1. **Ao recuperar dados ocultos:**
   - Permite promover mudanças em uma consulta SQL para retornar resultados adicionais.
   - Exemplo:
     - aplicativo de compras que conta com produtos classificados em diferentes categorias.
     - quando o usuário clica em uma delas, neste caso chamada de categoria X, o navegador solicita uma URL: http://website-inseguro.com.br/produtos?categorias=categoriaX, o que faz com que o aplicativo promova uma consulta SQL para recuperar os detalhes sobre os produtos relevantes dentro do banco de dados que correspondem à categoria X.
     - a consulta é exposta da seguinte forma: SELECIONAR * produtos ONDE categoria = categoria X e liberados = 1 (a consulta solicitou um retorno do banco de dados com todos os detalhes da tabela de produtos em que a categoria é X e os produtos lançados correspondem a 1).

- dessa forma, um cibercriminoso pode construir um ataque do seguinte modo: http://website-inseguro.com.br/produtos?categoria=categoriaX'--, que resulta na consulta SQL: SELECIONAR * produtos ONDE categoria = categoria X'-- e lançados = 1 (dois traços "--" é um indicador de comentário em SQL e significa que o restante da consulta é interpretado como um comentário), o que vai remover todo o restante da consulta, ou seja, não incluirá mais o "E lançados = 1". Assim, todos os produtos são exibidos, incluindo aqueles não liberados ou lançados, o que torna a consulta inútil.

2. **Ao subverter a lógica do aplicativo:**
   - Nessa situação, é possível alterar uma consulta para interferir diretamente nos parâmetros da aplicação.
   - Exemplo: aplicativo em que os usuários façam login a partir de um nome de usuário e uma senha. O aplicativo vai verificar as credenciais a partir da seguinte consulta SQL: SELECIONAR * de usuários ONDE nome de usuário = 'cliente01' E senha = 'sigilo01'. Se essa consulta retornar com os detalhes de um usuário, o login ou credenciamento será bem-sucedido. Caso contrário, ele será rejeitado.
   - Dessa forma, o invasor poderá executar o login como qualquer usuário sem uma senha, recorrendo à utilização da sequência de comentário SQL -- para remover a verificação de senha associada à cláusula "ONDE" da consulta. Exemplo: SELECIONAR * DE usuários ONDE nome de usuário = 'administrador'--' E senha = ''.

3. **Recuperando dados de outras tabelas de bancos de dados**
   - Nesse tipo de SQL Injection, é possível recuperar dados de diferentes tabelas de bancos de dados.
   - Nas situações em que os resultados de uma consulta SQL são retornados nas respostas do aplicativo, o invasor pode utilizar uma vulnerabilidade de SQL Injection para recuperar os dados de outras tabelas nos bancos de dados, procedimento feito a partir da utilização da palavra-chave "UNION", que permite a execução de uma consulta adicional e a anexação dos resultados à consulta original.
   - O cibercriminoso pode enviar a seguinte entrada: UNION SELECIONAR nome de usuário, senha DE usuários-- , o que faz com que o aplicativo traga como resultados da consulta todos os nomes de usuários e senhas junto aos nomes e filmagem dos produtos.

4. **Examinando o banco de dados**
   - A partir desse tipo de SQL Injection, é possível extrair informações sobre a versão e a estrutura do banco de dados em questão. São diversas as maneiras pelas quais pode consultar os detalhes da versão do banco de dados em questão. Isso vai depender do tipo de banco de dados.
   - No Oracle, por exemplo, pode executar uma consulta: SELECTIONAR * DE v$version. Também pode definir quais tabelas do banco de dados existem e quais colunas elas contêm.
   - Na maioria dos bancos de dados, é possível executar a seguinte consulta para listar as tabelas: SELECIONAR * DE information_schema.table.

5. **Vulnerabilidades de SQL Injections cegas**
   - Nesse tipo de vulnerabilidade, os resultados de uma consulta que você controla não estão presentes nas respostas do aplicativo. Nessas instâncias, as vulnerabilidades são chamadas de "SQL Injections cegas", o que significa que o aplicativo não vai retornar os resultados da consulta SQL ou demonstrar os resultados de possíveis erros de bancos de dados em suas respostas.
   - Conforme a natureza da vulnerabilidade e do banco de dados envolvidos, uma das técnicas que podem ser utilizadas na exploração das SQL Injections cegas é a alteração da lógica da consulta para acionar uma diferença detectável na resposta do aplicativo, dependendo da veracidade de uma única condição. Esse processo pode envolver, por exemplo, o acionamento condicional de um erro, como uma divisão por zero, ou uma nova condição com alguma lógica booleana.

Para prevenir SQL Injection, os desenvolvedores de software devem tomar as seguintes medidas:

1. Filtrar todos os dados inseridos pelo usuário.

    a. remover caracteres especiais das entradas do usuário.
    b. caracteres especiais podem ser usados para inserir comandos SQL maliciosos.

2. Usar consultas preparadas.
    a. são uma maneira segura de executar consultas SQL.
    b. consultas preparadas são pré-compiladas pelo banco de dados, o que impede que o invasor insira comandos SQL maliciosos.

3. Validar as entradas do usuário.
    a. a validação pode incluir verificação do comprimento, do formato e do conteúdo das entradas.
4. Filtrar todos os dados inseridos pelo usuário.

Ou seja, é fundamental seguir boas práticas de segurança, como a validação de entrada de dados, o uso de consultas parametrizadas e a aplicação de princípios de segurança em todos os níveis de uma aplicação web ou sistema para evitar a inserção de código SQL não autorizado. A prevenção adequada de SQL Injection é crucial para proteger a integridade e a confidencialidade dos dados armazenados em um banco de dados.

Referências:

Blog da EcoIT. SQL Injection. Disponível em: https://blog.ecoit.com.br/sql-injection/. Acesso em: 24 de setembro de 2023.

Blog da EcoIT. Gestão de Riscos Cibernéticos. Disponível em: https://blog.ecoit.com.br/gestao-de-riscos-ciberneticos/. Acesso em: 24 de setembro de 2023.

RMauro.dev. SQL Injection na Prática. Disponível em: https://rmauro.dev/sql-injection-na-pratica/. Acesso em: 24 de setembro de 2023.

GoCache. O que é SQL Injection para Iniciantes. Disponível em: https://www.gocache.com.br/seguranca/o-que-e-sql-injection-para-iniciantes/. Acesso em: 24 de setembro de 2023.

Blog da Trybe. SQL Injection. Disponível em: https://blog.betrybe.com/sql/sql-injection/#2. Acesso em: 24 de setembro de 2023.

Canal TI. SQL INJECTION na PRÁTICA (PHP + MySQL). YouTube, 2018. Disponível em: https://www.youtube.com/watch?v=QmFCoCqNStc. Acesso em: 24 de setembro de 2023.