SQL INJECTION

NAMA : RAFI IKHWANSYAH

NIM : 4342401026

KELAS : TRPL 1A PAGI

PENDAHULUAN

SQL Injection adalah salah satu jenis serangan keamanan yang sering terjadi pada aplikasi berbasis web. Serangan ini memungkinkan penyerang untuk mengeksekusi perintah SQL yang tidak sah ke dalam basis data melalui aplikasi yang rentan. Hal ini dapat menyebabkan kerusakan yang serius, seperti pencurian data, penghapusan data, bahkan pengambilalihan server basis data secara keseluruhan. SQL Injection adalah salah satu metode yang sangat populer digunakan oleh peretas karena seringkali aplikasi web yang tidak dilindungi dengan baik bisa dengan mudah dieksploitasi.

Makalah ini akan mengulas SQL Injection secara mendalam, mencakup pengertian, jenis-jenisnya, cara kerjanya, serta langkah-langkah pencegahan yang dapat dilakukan untuk menghindari serangan tersebut.

PENGERTIAN SQL INJECTION

SQL Injection adalah serangan yang dilakukan dengan memanfaatkan kerentanannya aplikasi berbasis web yang mengakses database menggunakan perintah SQL. Penyerang memasukkan kode SQL berbahaya ke dalam input aplikasi, seperti form pencarian, form login, atau URL, yang kemudian dieksekusi oleh server database. Dengan demikian, perintah SQL yang tidak sah ini dapat mengakses atau merusak data yang ada dalam database.

Serangan SQL Injection umumnya terjadi karena kurangnya validasi atau penyaringan input yang dimasukkan oleh pengguna. Tanpa validasi yang baik, aplikasi web dapat mengeksekusi query SQL yang telah dimodifikasi oleh penyerang untuk mengakses data yang seharusnya tidak bisa diakses.

JENIS – JENIS SQL INJECTION

SQL Injection dapat dibagi menjadi beberapa jenis tergantung pada teknik yang digunakan oleh penyerang:

A. In-Band SQL Injection

In-band SQL Injection adalah jenis yang paling umum dan mudah dilakukan. Serangan ini memungkinkan penyerang untuk mendapatkan hasil langsung dari query SQL yang telah dieksekusi. Ada dua jenis in-band SQL Injection:

1). Error-based SQL Injection:

Penyerang memanfaatkan pesan kesalahan yang diberikan oleh database untuk mendapatkan informasi tentang struktur database. Pesan kesalahan yang dihasilkan dari query yang gagal dapat memberikan petunjuk yang berguna, seperti nama tabel atau kolom.

Contoh: Jika penyerang memasukkan input yang salah, database mungkin memberikan pesan kesalahan yang mengungkapkan informasi penting.

2). Union-based SQL Injection:

Teknik ini memungkinkan penyerang untuk menggabungkan hasil query yang sah dengan query lain menggunakan operator UNION. Penyerang dapat menyisipkan query kedua untuk mendapatkan data yang seharusnya tidak terlihat.

Contoh: Penyerang mungkin mencoba untuk menggabungkan data dari tabel lain menggunakan UNION SELECT.

B. Blind SQL Injection

Blind SQL Injection terjadi ketika aplikasi tidak memberikan pesan kesalahan yang informatif kepada penyerang. Dalam hal ini, penyerang harus membuat asumsi dan menggunakan teknik percakapan bertahap untuk memperoleh informasi, biasanya dengan cara "True/False" atau "Yes/No".

1). Boolean-based Blind SQL Injection:

Penyerang mengubah input sehingga query SQL yang dihasilkan memberikan hasil yang berbeda berdasarkan kondisi tertentu. Penyerang kemudian mengamati apakah aplikasi mengubah perilakunya berdasarkan kondisi ini.

Contoh: Mengubah input sehingga query SQL yang dihasilkan bernilai TRUE atau FALSE.

2). Time-based Blind SQL Injection:

Penyerang memasukkan query yang menyebabkan server database untuk menunda eksekusi query (misalnya dengan menggunakan fungsi SLEEP atau WAITFOR DELAY). Berdasarkan waktu tanggapan dari server, penyerang dapat menentukan apakah query tersebut berhasil atau tidak.

C. Out-of-Band SQL Injection

Out-of-Band SQL Injection terjadi ketika penyerang mengandalkan saluran komunikasi lain (misalnya DNS atau HTTP request) untuk menerima data dari hasil eksekusi query. Jenis ini tidak terlalu umum dan bergantung pada kemampuan database untuk melakukan aksi "out-of-band", seperti membuat permintaan jaringan ke server eksternal.

CARA KERJA SQL INJECTION

SQL Injection bekerja dengan memanfaatkan aplikasi yang tidak memvalidasi input pengguna secara tepat. Berikut adalah langkah-langkah umum bagaimana serangan SQL Injection dapat terjadi:

1). Identifikasi Titik Masuk: Penyerang mencari titik masuk di aplikasi web, seperti form login, URL, atau parameter query string yang tidak aman. Misalnya, formulir pencarian atau URL yang meminta input berupa ID produk.

2). Masukkan Kode Berbahaya: Penyerang memasukkan perintah SQL yang dimodifikasi ke dalam input tersebut. Misalnya, pada formulir login, penyerang bisa memasukkan sesuatu seperti ' OR '1'='1 dalam kolom nama pengguna atau kata sandi.

3). Manipulasi Query SQL: Input yang dimasukkan akan digabungkan dengan query SQL yang sudah ada. Query yang dihasilkan menjadi sesuatu seperti:

SELECT * FROM users WHERE username = '' OR '1'='1' AND password = '';

Query ini selalu bernilai TRUE, sehingga memungkinkan penyerang untuk login tanpa kredensial yang sah.

4). Eksekusi Query oleh Database: Query yang sudah dimodifikasi ini kemudian dikirim ke server basis data untuk dieksekusi. Server database mengeksekusi query yang telah diubah oleh penyerang, sehingga memberikan hasil yang diinginkan oleh penyerang.

5). Mendapatkan Data atau Akses Tidak Sah: Jika berhasil, penyerang dapat mengakses data sensitif, seperti kredensial pengguna, atau bahkan mengubah data di dalam database. Dalam beberapa kasus, penyerang bisa mendapatkan akses penuh ke server basis data.

CARA MENCEGAH SQL INJECTION

1). Penggunaan Prepared Statements (Parameterized Queries): Prepared statements adalah cara terbaik untuk mencegah SQL Injection. Dengan menggunakan prepared statements, parameter input pengguna diperlakukan sebagai data, bukan bagian dari query SQL. Hal ini memastikan bahwa input pengguna tidak dapat mengubah logika query.

Contoh dalam PHP menggunakan MySQLi:

$stmt = $conn->prepare("SELECT * FROM users WHERE username = ? AND password = ?");

$stmt->bind_param("ss", $username, $password);

$stmt->execute();

2). Penggunaan ORM (Object-Relational Mapping): Framework dan pustaka ORM biasanya telah mengimplementasikan proteksi terhadap SQL Injection dengan cara yang aman dan efisien.

3). Validasi dan Sanitasi Input: Semua input dari pengguna harus divalidasi dan disaring dengan benar. Misalnya, pastikan bahwa input yang diharapkan adalah angka atau string yang tidak mengandung karakter berbahaya seperti ', ", ;, dan lainnya.

4). Penggunaan Hak Akses Minimun: Berikan hak akses yang terbatas pada aplikasi untuk berinteraksi dengan database. Pastikan bahwa akun yang digunakan aplikasi web hanya memiliki akses yang dibutuhkan, bukan akses penuh ke seluruh database.

5). Penggunaan Web Application Firewall (WAF): WAF dapat membantu memantau dan melindungi aplikasi web dari berbagai jenis serangan, termasuk SQL Injection.

6). Pemantauan dan Pencatatan Aktivitas: Monitor aktivitas aplikasi secara terus-menerus dan simpan log untuk menganalisis potensi percakapan yang mencurigakan. Ini membantu dalam mendeteksi serangan lebih awal.