SQL INJECTION (Keamanan jaringan komputer)

Muhammad Ikhsan (09011381621102)

Injeksi SQL terjadi ketika data yang masuk dari pengguna yang tidak terpercaya dimasukkan ke dalam aplikasi web dan data tersebut kemudian digunakan untuk secara dinamis membuat kueri SQL yang akan dijalankan oleh server database. Aplikasi Web umumnya menggunakan bahasa skrip dari sisi-server, seperti ASP (Active Server Pages), JSP (Java Server Pages), PHP (Hypertext Preprocessor), dan CGI (Common Gateway interface), untuk membuat kueri string yang diteruskan ke database sebagai pernyataan SQL tunggal. Aplikasi PHP dan ASP cenderung terhubung ke server basis data menggunakan Antarmuka Pemrograman Aplikasi (API) yang lebih tua yang secara alami lebih mudah dieksploitasi sementara aplikasi J2EE (Java 2 Enterprise Edition) dan ASP.NET tidak mudah dieksploitasi dengan mudah. Jika aplikasi web rentan terhadap injeksi SQL, maka penyerang memiliki kemampuan untuk mempengaruhi SQL yang digunakan untuk berkomunikasi dengan database. Efeknya sangat besar. Database sering mengandung informasi sensitif; Oleh karena itu, penyerang dapat membahayakan kerahasiaan dengan melihat tabel. Penyerang juga dapat membahayakan integritas dengan mengubah atau menghapus catatan basis data menggunakan injeksi SQL [1].

Dengan kata lain, serangan injeksi SQL yang berhasil dapat mengakibatkan akses tidak sah ke data sensitif, seperti kata sandi, detail kartu kredit, atau informasi pengguna pribadi. Banyak pelanggaran data profil tinggi dalam beberapa tahun terakhir telah menjadi hasil dari serangan injeksi SQL, yang menyebabkan kerusakan reputasi dan denda peraturan. Dalam beberapa kasus, seorang penyerang dapat memperoleh backdoor terus - menerus ke dalam sistem organisasi, yang mengarah ke kompromi jangka panjang yang bisa tanpa diketahui untuk waktu yang lama [2].

Sebagian besar kerentanan injeksi SQL muncul dalam sekumpulan beberapa data, WHERE dari kueri SELECT. Pada prinsipnya dapat terjadi di setiap lokasi dalam kueri, dan dalam berbagai jenis kueri. Paling umum di mana injeksi SQL muncul adalah:

- Dalam pernyataan UPDATE, dalam nilai yang diperbarui atau sekumpulan data WHERE.
- Dalam pernyataan INSERT, dalam nilai yang dimasukkan.
- Dalam pernyataan SELECT, dalam nama tabel atau kolom.
- Dalam pernyataan SELECT, dalam klausa ORDER BY.

Attack Scenario

Kode berikut merupakan dari SQL dengan menggabungkan string yang dimasukkan oleh pengguna dengan string kode :

```
String query = "SELECT * FROM items WHERE owner = '" + userName + "'  AND itemName = '" + ItemName.Text + "'";
```

Maksud dari permintaan ini adalah untuk mencari semua item yang cocok dengan nama item yang dimasukkan oleh pengguna. Dalam contoh di atas, userName adalah pengguna yang diautentikasi saat ini dan ItemName.Text adalah input yang disediakan oleh pengguna. Misalkan pengguna normal dengan nama pengguna smith memasukkan manfaat dalam formulir web. Nilai

itu diekstraksi dari formulir dan ditambahkan ke kueri sebagai bagian dari kondisi SELECT. Permintaan yang dieksekusi kemudian akan terlihat mirip dengan yang berikut:

SELECT * FROM items WHERE owner = 'smith' AND itemName = 'benefit' Namun, karena kueri dibuat secara dinamis dengan menggabungkan string kueri basis konstan dan string yang disediakan pengguna, kueri hanya berlaku dengan benar jika itemName tidak mengandung karakter tanda kutip tunggal ('). Jika penyerang dengan nama pengguna smith memasuki string j:

anything' OR 'a'='a

Kueri yang dihasilkan akan:

SELECT * FROM items WHERE owner = 'smith' AND itemName = 'anything' OR 'a' = 'a'

Penambahan kondisi OR 'a' = 'a' menyebabkan WHERE klausa untuk selalu mengevaluasi ke true. Kueri kemudian menjadi setara secara logis dengan kueri yang kurang selektif:

SELECT * FROM items

Permintaan yang disederhanakan memungkinkan penyerang untuk melihat semua entri yang disimpan dalam tabel item, menghilangkan kendala bahwa permintaan hanya mengembalikan item yang dimiliki oleh pengguna yang diautentikasi. Dalam hal ini penyerang memiliki kemampuan membaca informasi yang seharusnya tidak dapat diaksesnya.

Sekarang asumsikan bahwa penyerang memasuki yang berikut ini: apa saja '; jatuhkan item tabel— Dalam kasus ini, kueri berikut dibuat oleh skrip:

SELECT * FROM items WHERE owner = 'smith' AND itemName = 'anything'; drop table items

Tanda titik koma (;) menunjukkan akhir dari satu permintaan dan awal dari yang lain. Banyak server database memungkinkan beberapa pernyataan SQL yang dipisahkan oleh titik koma untuk dieksekusi bersama. Ini memungkinkan penyerang untuk mengeksekusi perintah sewenang-wenang terhadap database yang memungkinkan beberapa pernyataan dieksekusi dengan satu panggilan. Tanda hubung ganda (-) menunjukkan bahwa sisa baris saat ini adalah komentar dan harus diabaikan. Jika kode yang dimodifikasi secara sintaksis benar, itu akan dieksekusi oleh server. Ketika server database memproses dua pertanyaan ini, pertama-tama akan memilih semua catatan dalam item yang cocok dengan nilai apa pun yang dimiliki oleh pengguna smith. Kemudian server database akan menjatuhkan, atau menghapus, seluruh tabel item.

BLIND SQL INJECTION

Bentuk lain dari injeksi SQL disebut injeksi SQL *blind*. Biasanya, jika penyerang menyuntikkan kode SQL yang menyebabkan aplikasi web membuat kueri SQL yang tidak valid, maka penyerang harus menerima pesan kesalahan sintaksis dari server database. Namun, kode kesalahan spesifik dari database tidak boleh dibagikan dengan pengguna akhir suatu aplikasi. Ini dapat mengungkapkan informasi tentang desain basis data yang dapat membantu penyerang. Dalam upaya untuk mencegah eksploitasi injeksi SQL, beberapa pengembang mengembalikan halaman generik daripada pesan kesalahan atau informasi lain dari database. Hal ini membuat eksploitasi kerentanan injeksi SQL potensial lebih sulit, tetapi bukan tidak mungkin. Seorang

penyerang akan tahu apakah suatu kueri valid berdasarkan halaman yang dikembalikan. Jika aplikasi rentan dan kueri valid, halaman tertentu akan dikembalikan. Namun, jika kueri tidak valid, halaman yang berbeda mungkin dikembalikan. Oleh karena itu, penyerang masih bisa mendapatkan informasi dari database dengan mengajukan serangkaian pertanyaan benar dan salah melalui pernyataan SQL yang disuntikkan. Halaman yang sama harus dikembalikan terlepas dari apakah kueri SQL yang tidak valid dieksekusi [1].

Cara mendeteksi kerentanan injeksi SQL

Mayoritas kerentanan injeksi SQL dapat ditemukan dengan cepat dan andal menggunakan pemindai kerentanan web Burp Suite. Injeksi SQL dapat dideteksi secara manual dengan menggunakan serangkaian tes sistematis terhadap setiap titik masuk dalam aplikasi. Ini biasanya melibatkan:

1. Mengirimkan karakter kutipan tunggal 'dan mencari kesalahan atau anomali lainnya.
2. Mengirimkan beberapa sintaks khusus SQL yang mengevaluasi ke nilai dasar (asli) dari titik masuk, dan ke nilai yang berbeda, dan mencari perbedaan sistematis dalam respons aplikasi yang dihasilkan.
3. Menyerahkan kondisi Boolean seperti OR 1 = 1 dan OR 1 = 2, dan mencari perbedaan dalam respons aplikasi.
4. Mengirimkan muatan yang dirancang untuk memicu penundaan waktu ketika dieksekusi dalam kueri SQL, dan mencari perbedaan waktu yang diperlukan untuk merespons.
5. Mengirimkan muatan OAST yang dirancang untuk memicu interaksi jaringan out-of-band ketika dieksekusi dalam kueri SQL, dan memantau interaksi yang dihasilkan [2].

[1] S. Reetz and S. O. C. Analyst, “SQL-Injection-White-Paper,” no. January, pp. 1–5, 2013.

[2] Portswigger, “SQL INJECTION,” 2018. [Online]. Available: https://portswigger.net/web-security/sql-injection#retrieving-hidden-data.