

SQL Injection 1 : Bypass password saat login

Kenapa berhasil?

1. Data inputan tidak disanitasi.
2. Tidak ada validasi format email
3. ' mengeluarkan inputan dari parameter dan menambahkan kondisi baru menggunakan OR
4. 1=1 = true
5. '-- menyebabkan kondisi where di belakangnya menjadi comment, memungkinkan pemeriksaan password diabaikan.

Risiko dampak

1. Akses user lain secara illegal
2. Pencurian data

Rekomendasi perbaikan

1. Sanitasi data inputan.
2. Gunakan validasi format email.
3. Gunakan mekanisme login yang lebih aman.

The screenshot shows the OWASP Juice Shop login interface. In the 'Email*' field, the value "' or 1=1--" is entered, which is a common SQL injection payload. Below the login button, a success message reads: "You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)". The main page below shows a grid of products:

All Products	
Apple Juice (1000ml) 1.99€ Add to Basket	Apple Pomace 0.89€ Add to Basket
Banana Juice (1000ml) 1.99€ Add to Basket	Carrot Juice (1000ml) 2.99€ Add to Basket
Only I left Best Juice Shop Salesman Artwork	Eggfruit Juice (500ml) 8.99€ Add to Basket

SQL Injection 2 : Menambahkan karakter yang menyebabkan query error

Kenapa berhasil?

1. Data inputan tidak disanitasi.
2. Tidak ada validasi format email.
3. ‘ menyebabkan error pada query yang tidak dihandle dengan benar.

Risiko dampak

1. Bila error handling kurang baik, bisa jadi pesan error mengekspos kerentanan lebih jauh

Rekomendasi perbaikan

1. Sanitasi data inputan.
2. Gunakan validasi format email.
3. Perbaiki error handling agar tidak mengekspos informasi terlalu banyak.

The screenshot shows a browser developer tools interface with the Network tab selected. At the top, a green success message says "You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)". Below it, a "Login" form is displayed with an email field containing "2@google.com". The Network tab shows a single request labeled "login" with a status of "Success". The response body is a JSON object:

```
1 {  
2   "error": {  
3     "message": "SQLITE_ERROR: near '@google': syntax error",  
4     "stack": "Error\n  at database.<anonymous> (/juice-shop/node_modules/sequelize/lib/dialects/sqlite/query.js:185:27)\n    at /juic  
5     \"name\": \"SequelizeDatabaseError\",  
6     \"parent\": {  
7       \"errno\": 1,  
8       \"code\": \"SQLITE_ERROR\",  
9       \"sql\": \"SELECT * FROM Users WHERE email = '2@google.com' AND password = '0cc175b9c0f1b6a831c399e269772661' AND deletedAt IS NULL\"  
10    },  
11  }  
12 }  
13 
```

The search bar on the right shows the query "getnib" with no matches found.