

Task A - SQL Injection

The screenshot shows a Postman collection named "New Collection / New Request Copy". A GET request is made to `http://localhost:3000/rest/products/search?q=<script>alert(1)</script>`. The "Params" tab shows a query parameter `q` with the value `'<script>alert(1)</script>`. The response status is 500 Internal Server Error, with a timestamp of 59 ms and a size of 726 B. The response body is a JSON object containing an error message:

```
1 {  
2   "error": {  
3     "message": "SQLITE_ERROR: near \"/": syntax error",  
4     "stack": "Error: SQLITE_ERROR: near \"/": syntax error",  
5     "errno": 1,  
6     "code": "SQLITE_ERROR",  
7     "sql": "SELECT * FROM Products WHERE ((name LIKE '%<script>alert(1)</script>%' OR description LIKE '%<script>alert(1)</script>%') AND deletedAt IS NULL"  
8   }  
9 }
```

Kenapa ?

- query seharusnya terlihat seperti name LIKE '%somequery%'.
- Karena server menyusun query dengan menggabung (concatenate) string user langsung, input `<script>...` dimasukkan tanpa pembungkus yang benar.
- SQL parser melihat '%'`<script>` — yaitu: string literal '%' diikuti token `<script>` yang bukan token SQL valid → SQLite mengeluh near "/" : syntax error (karena simbol '/' dalam `<script>` dianggap bagian token yang tidak valid).
- Jadi payload memecah literal string SQL, dan menyebabkan syntax error.

Perbaikan

- Jangan menyusun SQL dengan string concatenation.
- Escape wildcard saat memakai LIKE dan pakai parameter binding untuk pola (%...%).
- Validasi input (panjang, karakter yang diizinkan)
- Jangan pamerkan stack trace / SQL ke client, log internal saja.

```

POST http://localhost:3000/rest/user/login
{
  "email": "or 1=1", "password": "1"
}

```

Body { } JSON

```

1 {
  "authentication": {
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.
eyJzdGF0dXMiOiJzdWNjZXNlIiwidXNlcm5hbWUiOii1LcJ1bwFpbC161mPkbWlUQGp1awNLXNlNm9wiwicGFzc3vcmQ10i1wMTkyMDIzYtdYnQ3MzI1MDUxNmYwNj1kZjE
4YJuWMCisInJvbGUiOihGpbilsImRlhV4ZVRva2Uljo11iwFzdxvZ21uSXAiOii1LcJwm9maw15W1hZ2Ui0i1hc3NLdmVvcH1j121tYwdlcyc91cgvxYmRzl2R1zFn1bHRBZG1pb15wbm
c1LCJ0b3RwU2VjcnwIjoiLiwiwNBY3RpdmUionRydwUsImNyZwf0ZWRBdC161j1wMjUtMTEtMDYgNDM6MTY6MDE0UeOTwICswMDoMcIsInVwZGf0ZwR8dcI61j1wMjUMTETMDYgMDM6MTY6D
CsMwDowMCIsImRLbgVO2WFBdC16bnVsbl0sImldc16tC2MjQwMDA0X0.
ITqvYmTfITMD90Eh1Yj0Gpqm5likNSkWhQnpxlmlUvI_uoW0_inX12Woe8nogBawaFgy5YmUiUoEJytpLbsWsKj1Jlg-t1iW32KRSJXYuzHae44yxMInkt1iE6G4zcalquZkG-_pzGse17C8ka3343cC
Txhde_XB8Ym6xis1",
    "bid": 1,
    "umail": "admin@juice-sh.op"
  }
}

```

200 OK 65 ms 1.16 KB Save Response

Kenapa ?

- Server menyediakan input user langsung ke string query tanpa parameterization.
- Kutip tunggal ' dalam input memecah literal SQL sehingga bisa menulis kondisi sendiri.
- -- adalah komentar SQL yang mengabaikan sisa kondisi (misal pengecekan password).

Perbaikan

- Prepared statements / parameterized queries.
- Validasi input (contoh: valid email format, panjang maksimal).

Task B – BAC (Broken Access Control)

GET http://localhost:3000/rest/basket/2

This request does not have a body

Body { } JSON

```

1 {
  "status": "success",
  "data": {
    "id": 2,
    "coupon": null,
    "userId": 2,
    "createdAt": "2025-11-06T02:05:58.260Z",
    "updatedAt": "2025-11-06T02:05:58.260Z",
    "Products": [
      {
        "id": 4,
        "name": "Raspberry Juice (1000ml)",
        "description": "Made from blended Raspberry Pi, water and sugar.",
        "price": 4.99,
        "deluxePrice": 4.99,
      }
    ]
  }
}

```

200 OK 79 ms 943 B Save Response

Kenapa ? Server tidak memverifikasi kepemilikan objek sebelum mengembalikan data.

TOKEN autentikasi hanya memastikan kamu adalah *user yang login*, tetapi tidak memastikan bahwa kamu *berhak atas resource dengan ID tersebut*.

Perbaikan

- Tambahkan validasi kepemilikan (object ownership check)
- Idealnya, endpoint /rest/basket cukup mengembalikan basket milik user yang sedang login tanpa memerlukan parameter id

The screenshot shows a Postman collection named "New Collection / New Request Copy". A GET request is made to `http://localhost:3000/rest/track-order/5267-1217ce5d30c53298`. The "Authorization" tab is selected, showing "Inherit auth from parent". The response is a 200 OK with a total price of 26.97.

```
1 {
2   "status": "success",
3   "data": [
4     {
5       "orderId": "5267-1217ce5d30c53298",
6       "email": "admin@j***-sh.*p",
7       "totalPrice": 26.97,
8       "bonus": 3,
9       "products": [

```

Kenapa ?

- Bisa mendapatkan data order berdasarkan order id milik orang lain berdasarkan order id

Perbaikan

- Tambahkan validasi kepemilikan (object ownership check)

Task C – XSS

The screenshot shows a browser window for the OWASP Juice Shop. A modal dialog box appears with the text "localhost:3000 says" and "xss". Below the dialog, the page content shows a search results section with the message "No results found" and "Try adjusting your search to find what you're looking for." At the bottom, the developer tools Network tab is open, showing a list of network requests and their details.

Name	Status	Type	Initiator	Size	Time
socket.io/?EIO=4&transport=polling&i=PfNLFd&sid=7MsUCTb6ZE0beuzYAAAU	200	xhr	polyfills.js:1	0.2 kB	84 ms
socket.io/?EIO=4&transport=polling&t=PfNLFdd&sid=7MsUCTb6ZE0beuzYAAAU	200	xhr	polyfills.js:1	0.3 kB	87 ms
search?q=	200	xhr	polyfills.js:1	0.9 kB	106 ms
search?q=	200	xhr	polyfills.js:1	4.2 kB	80 ms

Kenapa ?

- Tidak ada proses sanitasi input.
- Browser langsung menganggap teks <script>...</script> sebagai HTML sah yang bisa dieksekusi.

Perbaikan

- Validasi dan encode input
- Validasi nilai hash, query string, atau field input sebelum digunakan.
- Encode karakter berbahaya seperti <, >, &, ', dan " jika akan ditampilkan.