API, Data, Permissions, and Privacy

Lesson 6 · Prove boundaries hold

Today’s outcome

- Read safe API evidence
- Test roles and tenant isolation
- Recognize privacy/security risk
- Write responsible security findings

Authorization is a chain

Identity → role/permission → server decision → organization filter → database policy

Any one layer can be wrong. Multiple layers protect the customer.

Hidden UI ≠ security

A disabled or invisible button helps usability.

The server and data layer must still deny a forbidden request.

Safe evidence collection

- synthetic accounts only
- redact headers, tokens, cookies, PII
- record status and safe response summary
- never test unknown systems or production boundaries

Demo: UI denial vs. API denial

Compare:

- expected wrong-role UI state
- forbidden server/API result
- correct organization-scoped data result

All three matter.

The tenant question

“Can a user from organization A see or change organization B’s data—even by guessing an ID?”

That is a high-impact QA question.

Lab: Tenant-Boundary Escape Room

Test four controls: list/detail isolation, PATCH permission, positive owner control and UI/API send denial.

PATCH does not persist. Unsupported writes are not authorization tests.

Deliver: four-lock evidence table and regression recommendation.

Report responsibly

State:

- exact scope
- synthetic account and environment
- observed/expected result
- evidence
- customer impact
- remediation direction

Avoid sensational claims or live exploitation.

Exit ticket

Name two independent layers that should enforce tenant isolation.