Secure Software Engineering

Cheat Sheet for Junior Developers

Principle 1: Least Privilege

Key Ideas

- Always give the minimum required access.
- Never assume “read-only” is harmless.
- Define clear roles and scopes.

Principle 2: Data is Always Sensitive

Key Ideas

- Non-critical data can still be aggregated or misused.
- Context may change: what’s harmless now may be sensitive tomorrow.

Principle 3: Auditability

Key Ideas

- Keep logs of who did what, when, and why.
- Use audit trails to ensure accountability.

Principle 4: Don’t Ship Raw Data

Key Ideas

- Use structured APIs or dashboards.
- Separate internal models from external exposure.

Principle 5: Defense in Depth

Key Ideas

- Combine access control, validation, monitoring, and limits.
- Design for failure and containment.

As a Junior Developer, Always:

- Ask: “Is this access really necessary?”
- Challenge: “It’s internal, so it’s fine.”
- Learn about secure defaults and data classification.
- Speak up if something feels wrong.

You don’t need to be a security expert to think securely. You just need to care — and ask the right questions.