🔒 Security Strategies for Software Engineers

In modern development, security is a central responsibility for software engineers. Threats increase every year: in 2024, critical vulnerabilities grew by **37.1%**, and more than **33%** of issues detected across full technology stacks were classified as high severity. These numbers show that even teams with solid practices can be exposed if security is not integrated from the beginning.

One of today's biggest challenges is the reliance on external libraries. Although only about 1% of releases contain direct vulnerabilities, nearly **46.8%** are affected by **transitive** vulnerabilities coming from indirect dependencies. For this reason, auditing versions, reviewing maintenance history, and running automated CVE scans should be part of the standard development workflow.

Security must start at the design phase through threat modeling, strict input validation, and well-structured access control. Equally important is proper secret management: avoiding hard-coded credentials and using dedicated tools for handling keys and tokens.

In production environments, continuous monitoring and detailed logging help detect abnormal behavior before it escalates into an incident. Complementing this with security-focused code reviews and periodic penetration testing significantly strengthens system protection.

Integrating these strategies not only reduces risk but also increases software reliability and business resilience. Security, when applied consistently, becomes an enabler of modern development.