What is a secret in software, and how should you manage secrets?

3 November 2026 · 4 min read · blogs.iictp.org/secrets-management-for-beginners

A secret is any value that grants access, such as an API key, a database password, a token or a private key. Manage secrets by keeping them out of source code, storing them in a secrets manager or protected environment, limiting who and what can read them, and rotating them regularly.

TL;DR The short version

- A secret is any value that grants access, such as an API key, password, token or private key.
- Keep secrets out of code and version history; store them in a secrets manager or the environment, limit who can read them and rotate them.
- If one leaks, revoke or rotate it immediately and check logs; prevent leaks with secret scanning and per-service credentials.

Read online, listen to it, or change the text size: https://blogs.iictp.org/secrets-management-for-beginners

Why is putting a key in the code a problem?

Code is copied, shared and kept in version history forever. A key committed even once can be found by anyone who sees the repository, including automated scanners that look for exactly this. Deleting it in a later commit does not remove it from history.

Where should secrets live instead?

The twelve-factor method says configuration that varies between environments belongs in the environment, not in code. For anything sensitive, use a dedicated secrets manager or your platform's secret store, which encrypts the value, controls who can read it and records access.

What should you do if a secret leaks?

1. Revoke or rotate the secret immediately, before anything else.
2. Check logs for any use of it you did not expect.
3. Remove it from the code and, where practical, from history.
4. Find out how it leaked and close that path.

How do you stop leaks before they happen?

Add secret scanning to the repository and to the pre-commit and pipeline checks, keep local configuration files in the ignore list, and give each service its own credentials so one leak has a small reach. This applies least privilege.

Why rotate secrets?

A secret that never changes stays valuable to an attacker forever. Regular rotation, and short-lived credentials where the platform supports them, limit how long a stolen value remains useful.

> Related credential: Associate Technical Practitioner. Explore the exam facts and syllabus.

Sources and further reading

1. Wiggins, A. The Twelve-Factor App, III. Config. 12factor.net/config.
2. OWASP. Secrets Management Cheat Sheet. cheatsheetseries.owasp.org.

Published by IICTP, an independent technical credential body, on 3 November 2026. The current version of this post is at https://blogs.iictp.org/secrets-management-for-beginners. More guides: blogs.iictp.org.