SECURITY CONTROLS IN SHARED SOURCE CODE REPOSITORIES

Matthew Gamboa

Module 11.2

12/15/24

WHAT ARE REPOSITORIES?

Repositories = Places where code is stored and shared.

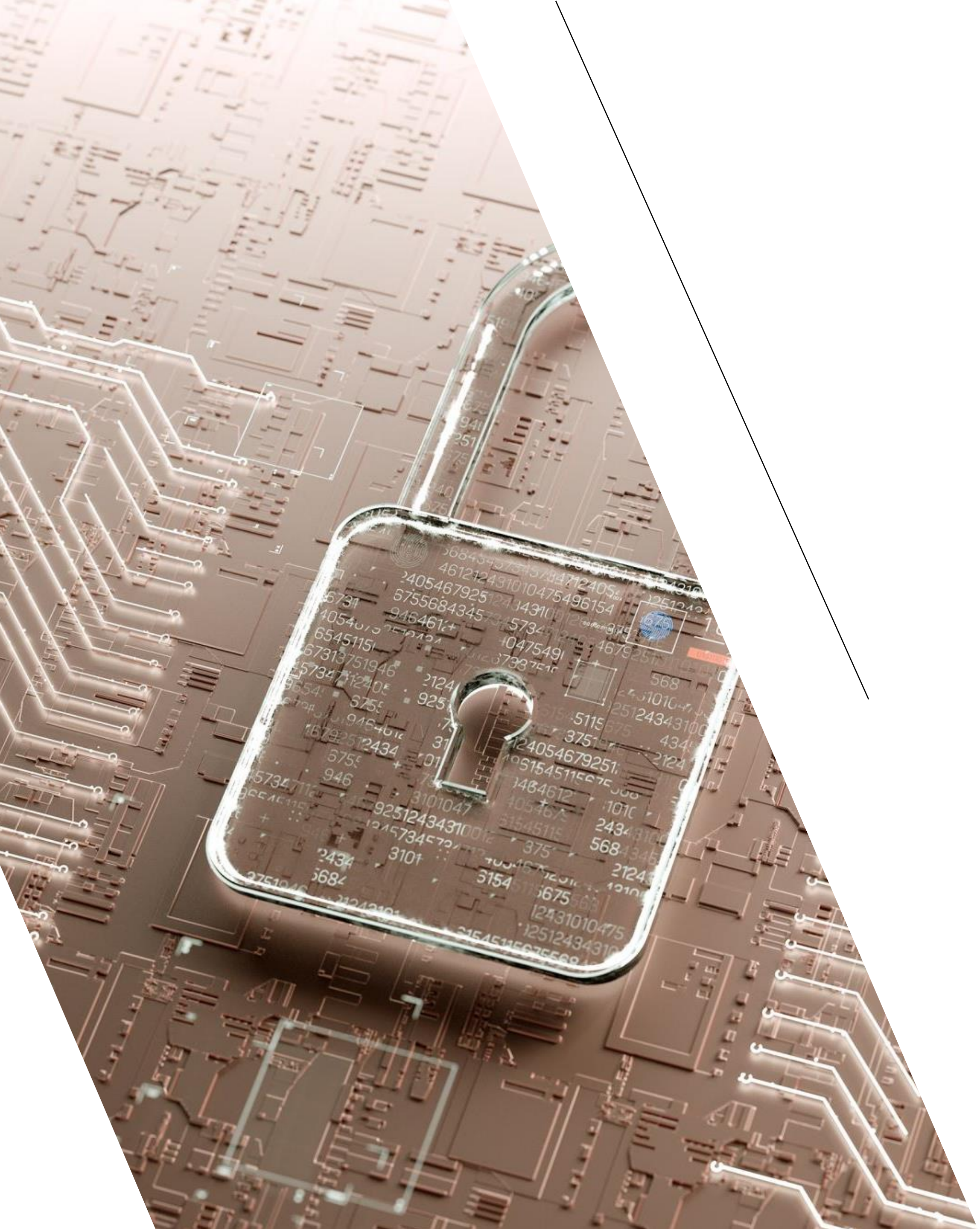
Examples: GitHub, GitLab, Bitbucket.

Developers use these to work on code together.

Why care about security? (Stops hacks or leaks)

THE RISKS

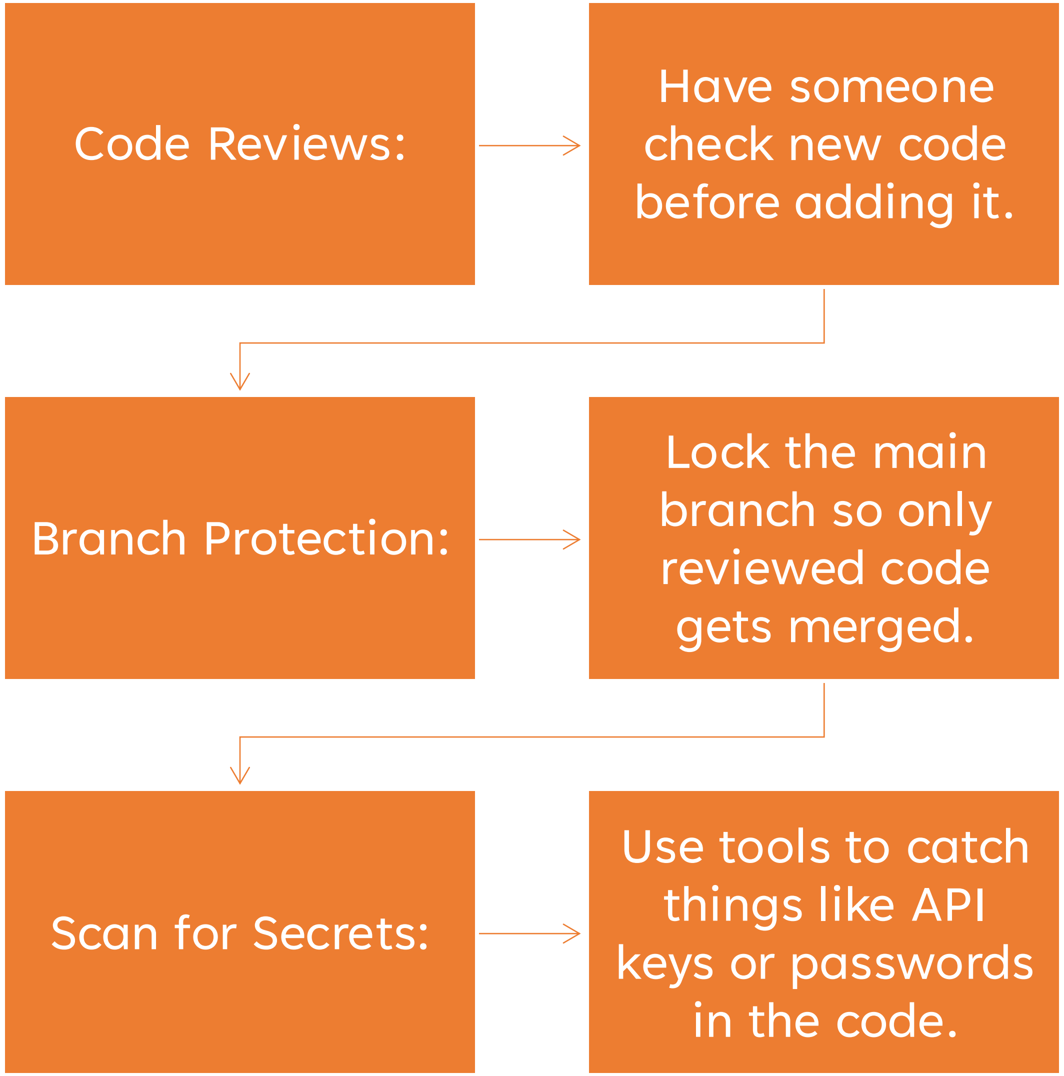
- People getting into your repo without permission.
- Someone sneaking bad code into your project.
- Private info like passwords getting leaked.
- Attacks on tools or libraries your code depends on.



LOCK IT DOWN (ACCESS CONTROL)

- Only let people in who need access.
- Use roles—like admin, contributor, or viewer.
- Require strong passwords and multi-factor authentication (MFA).

KEEP THE CODE CLEAN



Code Reviews:

Have someone check new code before adding it.

Branch Protection:

Lock the main branch so only reviewed code gets merged.

Scan for Secrets:

Use tools to catch things like API keys or passwords in the code.

WATCH FOR SUSPICIOUS ACTIVITY

Automated Scans:

Use tools that look for bugs or malware in your code.

Update Dependencies:

Keep libraries and tools you use up-to-date.

Logs:

Keep track of who did what in the repo.

BACKUP YOUR WORK

Why?

- If something breaks or gets hacked, you don’t lose everything.

How?

- Have regular backups of your repo.
- Store backups somewhere safe.

EXAMPLES OF TOOLS

Tools That Help:

- **Dependabot (GitHub):** Keeps dependencies updated.
- **SonarQube:** Finds bugs and security issues.
- **GitGuardian:** Detects secrets in your code.

CONCLUSION

- Repositories are good but need protection.
- Use access controls, scan for bad activity, and keep things up-to-date.
- Stay on top of backups and reviews.

SOURCES

- GitGuardian Blog: "GitHub Security 101"

https://blog.gitguardian.com/github-security-101/

- Wiz.io Academy: "Source Code Security"

https://www.wiz.io/academy/source-code-security

- Snyk.io: "Securing Source Code Repositories"

https://snyk.io/articles/securing-source-code-repositories/