Security in Code Repositories

Jason Schriner

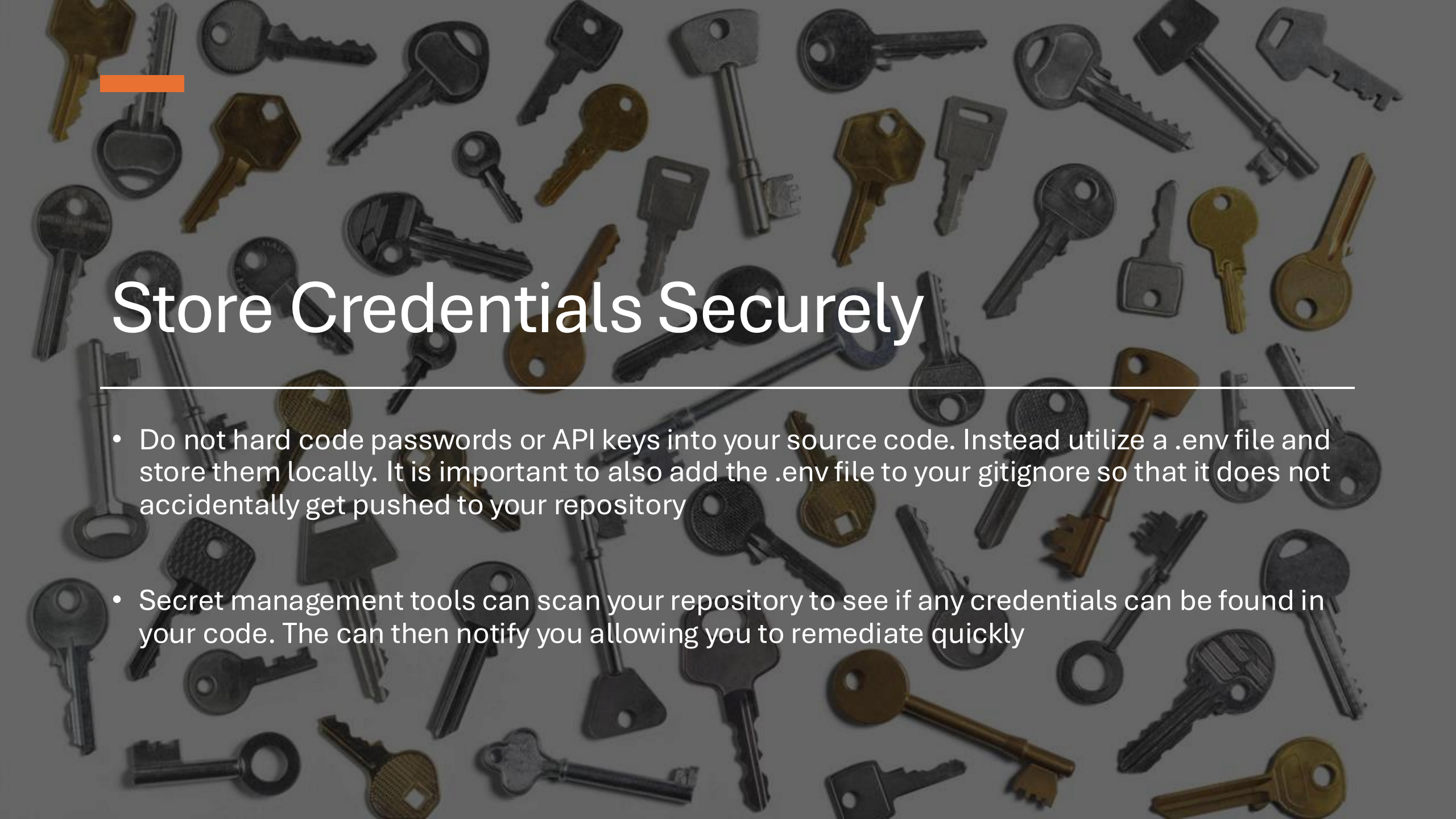
Module 11

7/28/25

Security in Source Code Repositories

- Code repositories are a necessary part of modern software development. As bad actors increasingly target these repositories, the need to secure them has increased.

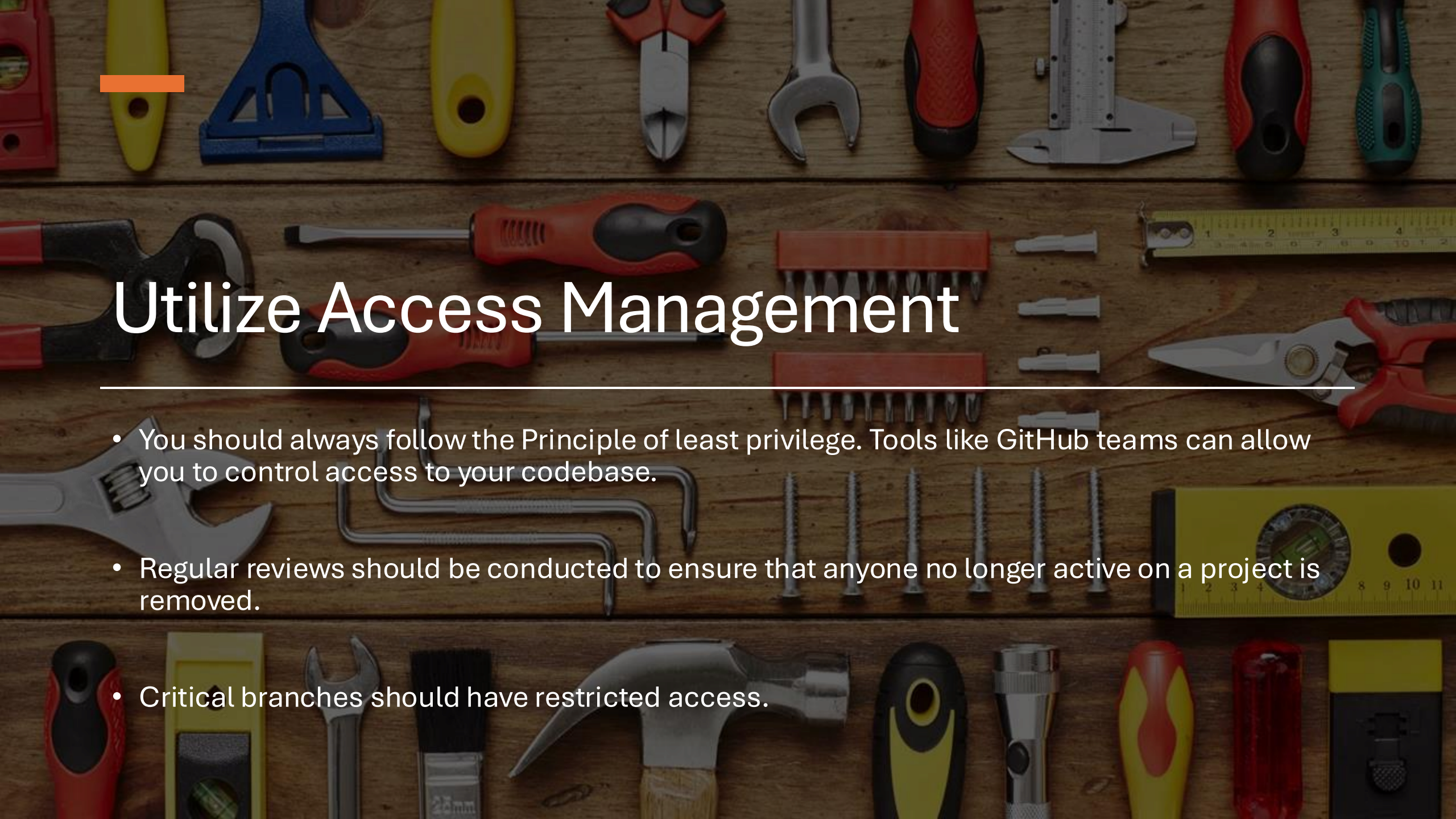
- The following best practices will help you strengthen your source code repository security and reduce the risk of compromise.



Store Credentials Securely

- Do not hard code passwords or API keys into your source code. Instead utilize a .env file and store them locally. It is important to also add the .env file to your gitignore so that it does not accidentally get pushed to your repository

- Secret management tools can scan your repository to see if any credentials can be found in your code. The can then notify you allowing you to remediate quickly



Utilize Access Management

- You should always follow the Principle of least privilege. Tools like GitHub teams can allow you to control access to your codebase.
- Regular reviews should be conducted to ensure that anyone no longer active on a project is removed.
- Critical branches should have restricted access.

Use 2FA

- Require the use of two factor authentication. It provides an additional layer of security and can prevent a bad actor from gaining access even if a password is compromised.
- For higher security, you can require more than two factors also known as multi-factor authentication. This greatly increases protection by making it much harder for attackers to gain access, even if one factor is compromised.

Ensure code branch protection

Protect	Use	Require	Block
Following proper branching strategies can protect against an internal threat from pushing code to a production codebase.	Use signed commits to verify author	Require reviews before merging pull requests	Block force pushed and branch deletions

Create and Maintain a Security .md file

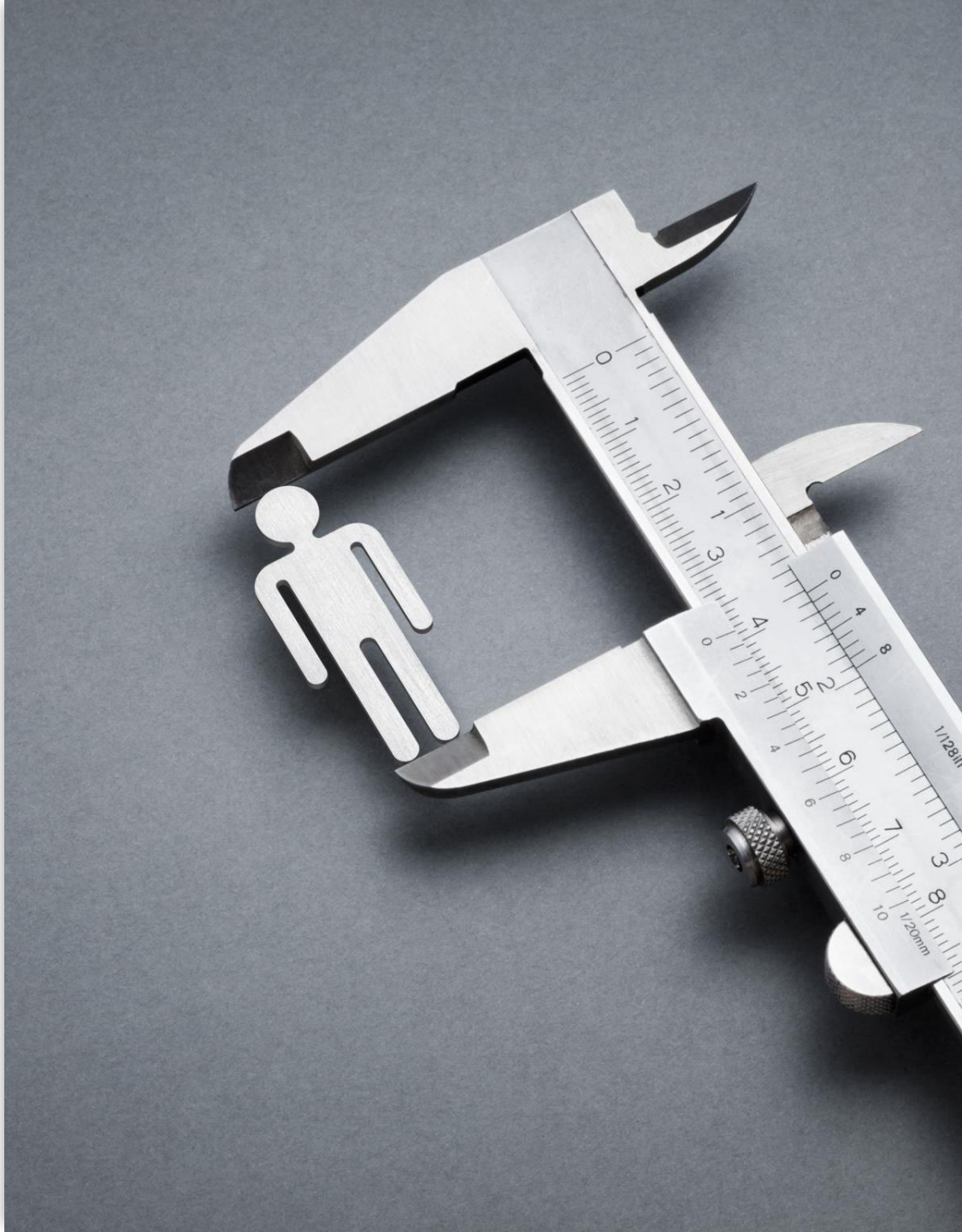
- A security mark down file is used to provide instructions on how to report security vulnerabilities
- Identifies which versions are maintained
- Helps to improve transparency and build trust

Track Activities with Audit logs

- Audit logs can be used to track multiple events such as logins, permission changes and branch rule changes.
- The logs can be used to help enforce policies and support incident response
- Can be used to set up alerts for suspicious activity

Regularly Scan your repositories

- Scans can be integrated into pipeline and offer a proactive approach to help catch issues before they are exploited
- Multiple tools are available to provide for code, secret and dependency scanning.
- These tools can be powerful but require active use to remediate alerts



Disable or Restrict Public Repositories

Prevent	Audit	Use
Disabling public repositories can prevent proprietary code from being accessed by the public	Audit what is publicly accessible. Will help ensure nothing is unintentionally public	Use a .gitignore to prevent credentials from being pushed to the repository

References:

- Rose, J. (2025, June 11). 12 Best Practices for Secure Code Repositories (2025 Guide). Checkmarx. https://checkmarx.com/supply-chain-security/repository-health-monitoring-part-2-essential-practices-for-secure-repositories/
- Kulikova, D. (2023, February 28). GitHub Security Best Practices – 15 Tips To Keep In Mind. Blog | GitProtect.io. https://gitprotect.io/blog/github-security-best-practices-15-tips-to-keep-in-mind/