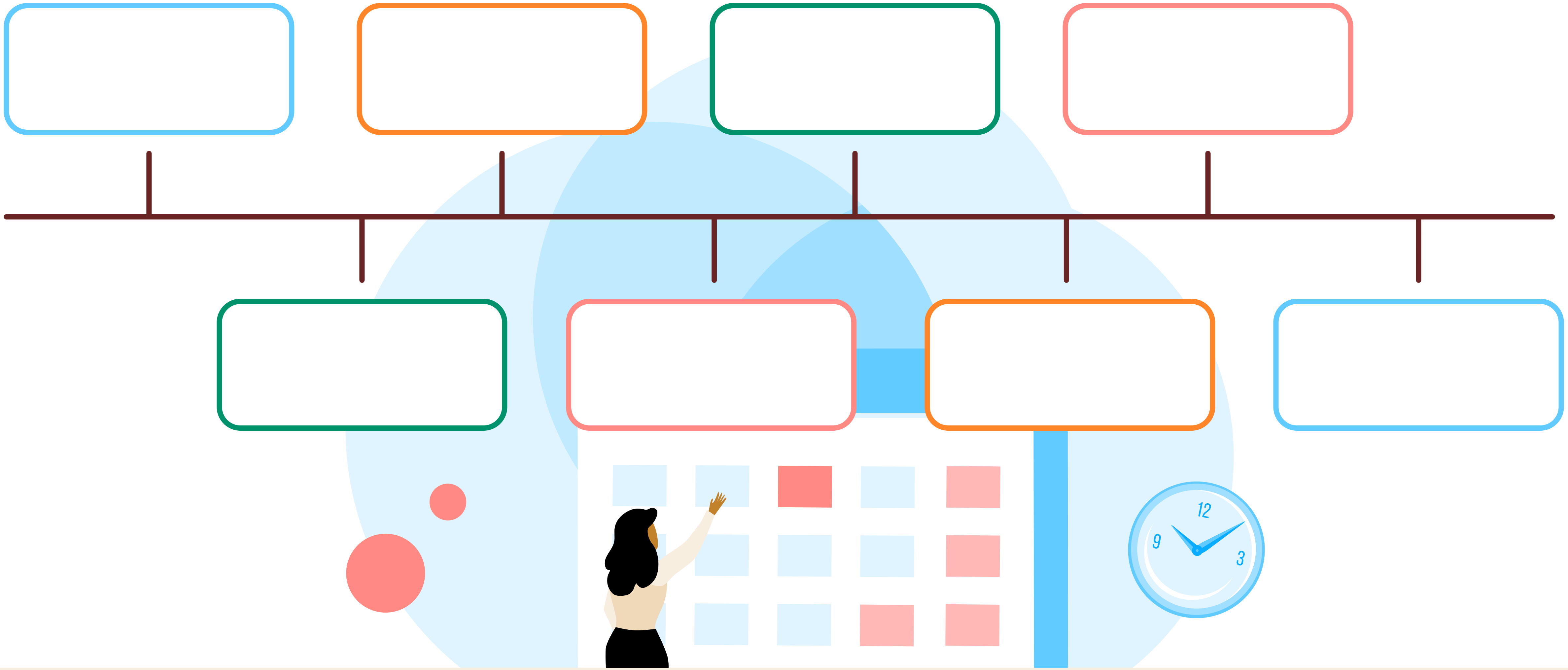


Introduction to Security in Shared Source Code Repositories

Thunder Harding

Introduction to Security Controls in Shared Source Code Repositories

Shared source code repositories are essential for collaboration, but they also introduce security risks. Unauthorized access, exposed secrets, and unverified dependencies can lead to vulnerabilities. Implementing strong security controls ensures the integrity, confidentiality, and availability of source code. This presentation outlines best practices to secure shared repositories, covering authentication, access control, code review policies, and automated security scanning. Organizations that follow these best practices reduce the risk of supply chain attacks, insider threats, and accidental data leaks while maintaining a secure development environment.

The first line of defense for securing a shared repository is enforcing strong authentication and access control mechanisms. Multi-factor authentication (MFA) should be required for all users, preventing unauthorized access even if credentials are compromised. Role-based access control (RBAC) ensures that only authorized personnel can push, merge, or modify critical code. Additionally, implementing least privilege principles limits access to only what is necessary for each user, reducing the risk of accidental or malicious modifications.

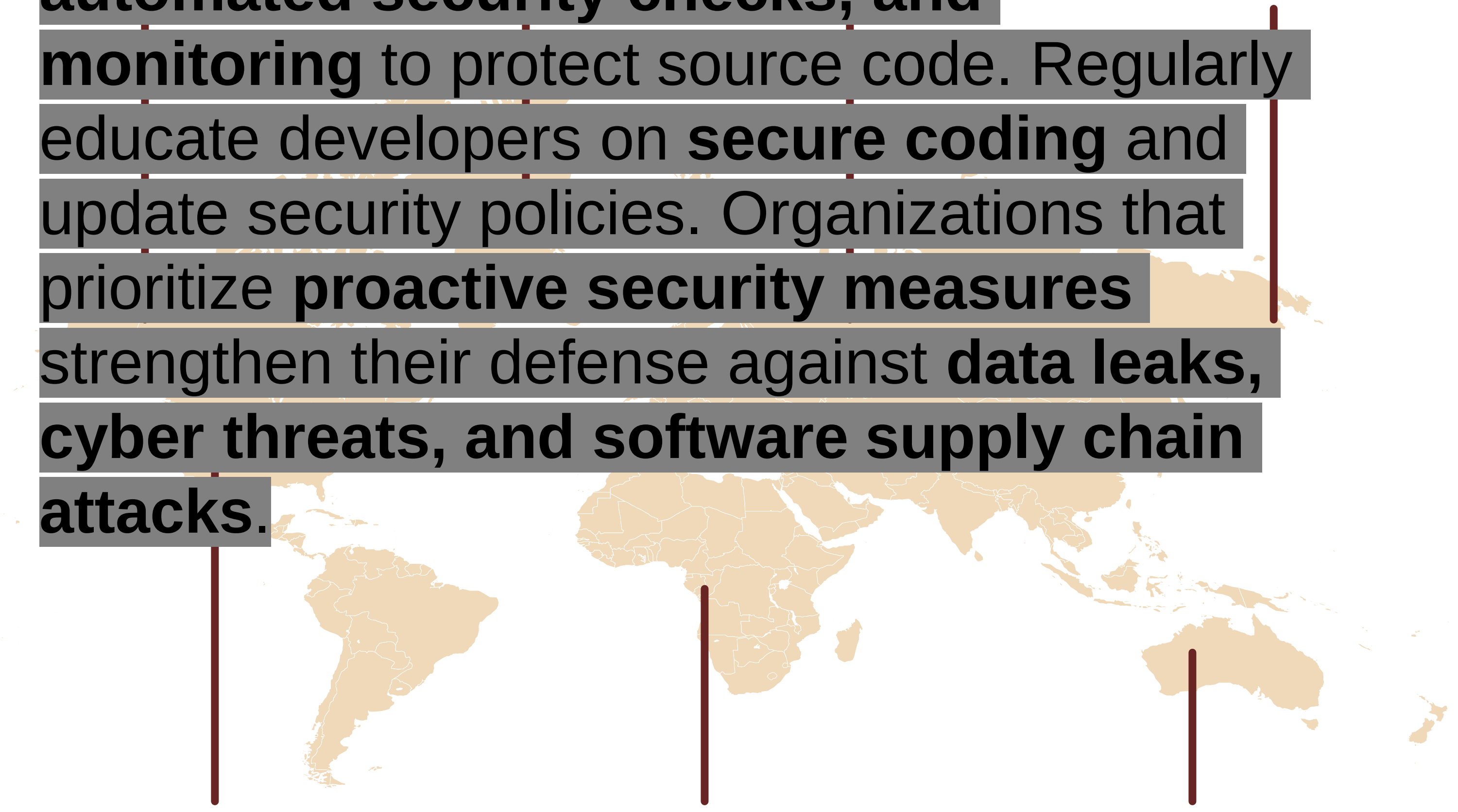
Authentication and Access Control

Secure Coding and Secrets Management

Prohibit storing **API keys, credentials, and sensitive data** in source code. Use **secrets management tools** like HashiCorp Vault or AWS Secrets Manager. Implement **peer-reviewed pull requests** and **automated security linting** to enforce secure coding practices and prevent vulnerabilities from entering the codebase.

Automated Security Scanning and Monitoring

Use tools like **Snyk, SonarQube, and Dependabot** to detect **code vulnerabilities** and outdated dependencies. Enable **audit logging** to track repository activity and flag suspicious modifications. Continuous **security monitoring** ensures rapid detection and response to **unauthorized access or malicious code injections**.



Implement **strong authentication, controlled access, secure coding practices, automated security checks, and monitoring** to protect source code. Regularly educate developers on **secure coding** and update security policies. Organizations that prioritize **proactive security measures** strengthen their defense against **data leaks, cyber threats, and software supply chain attacks**.

Conclusion and Best Practices

Sources:

Protect your code repository. NCSC. (n.d.). https://www.ncsc.gov.uk/collection/developers-collection/principles/protect-your-code-repository?utm_source=

Tal, L. (2023, November 16). *Securing source code in repositories is essential: How to get started*. Snyk. https://snyk.io/articles/securing-source-code-repositories/?utm_source=