GitHub

Nine security best practices every software leader should know

1 Use a centralized authentication protocol, such as SSO with LDAP or SAML, for all systems within your organization.

Using SSO across your applications minimizes security risk and keeps your organization in control of user access.

Tips

- Implement SSO across all systems that allow it.
- Have a documented plan to both onboard new hires and offboard exiting employees.
- Keep audit logs for access control.

2 Enable MFA–and require when available–on all systems.

Adding multi-factor authentication provides an extra layer of protection and greatly reduces the chance of compromised user accounts.

Tips

- Audit all systems to document their use of MFA.
- Require MFA on all systems that allow it.
- Document the process of setting up user devices for MFA, such as authenticator apps (Duo, Google, Authy, etc.).

3 Document data use and access policies and ensure they're easy to find.

Keeping current security policies in one document makes them easier for everyone in your organization to find and follow.

Tips

- Outline your security practices in an easy-to-find document for new and existing projects.
- Make it easy for developers to stay secure by regularly checking your organizational repositories' security configurations.

4 Encrypt data both in transit and at rest whenever applicable.

Encrypting data as it moves and while it's being stored cuts down on any opportunity for data leaks or stolen intellectual property.

Tips

- Create a report on the current state of all stored data, including whether or not the data is encrypted.
- Identify what needs to be encrypted, and encrypt everything in order from most to least important.
- Document how to encrypt data and include this reference in the CONTRIBUTING.md file.
- Identify systems that are not using encrypted protocols like HTTPS and SSH and enable these encryption protocols.

5 Make your security team an integral part of development and bring them into key planning and checkpoints.

Keeping your security team and developers in sync leads to stronger application security and shorter development cycles.

Tips

- Notify your security team of all new development so that they can participate in kickoffs and code discussions.
- Have a documented process to report security deficiencies.

6 Scan your team's code for secrets and credentials during code commits.

Scanning credentials automatically prevents developers from accidentally checking in passwords or other credentials.

Tips

- Use credential scanning for all commits to prevent pushing credentials to version control or production environments.
- Regularly perform complete scans of all repository content to track any existing credentials.

7 Make sure everyone at your organization uses a shared credentials vault.

Storing credentials in a centralized vault application means you don't have to keep them in your source code or documentation.

Tips

- Use a shared credential vault tool (LastPass, 1Password, Dashlane, etc.) to generate and store passwords.
- Protect this shared vault with MFA for an extra layer of security.

8 Update user tokens and passwords on a regular and automated cadence.

Rotating your tokens and passwords reduces the possibility of any unauthorized access.

Tips

- Know where passwords are stored for all systems in your organization and document their location.
- Include password rules in your security standards documentation and require everyone to use unique passwords.

9 Track vulnerable dependencies automatically and create an established process for addressing security alerts.

Scanning source code for vulnerabilities reveals flaws and weaknesses that could be exploited or leak information.

Tips

- Set up security alerts with your VCS' API to track and resolve issues and dependencies.
- Automatically scan for vulnerabilities before going live to improve code quality and reveal security risks.

Need help getting started?

Talk to a GitHub security expert:
experts@github.com