The Secure Software Development Framework (SSDF)

Following the Cybersecurity Executive Order, NIST released version 1.1 of 'The Secure Software Development Framework (SSDF): Recommendations for Mitigating the Risk of Software Vulnerabilities'. This is a quick primer to the framework and you can read the entire report here: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-218.pdf.

Few software development life cycle (SDLC) models explicitly address software security in detail. The secure software development framework (SSDF) addresses this gap by describing a set of high-level practices. The practices are divided into 4 groups. Each group outlines practices which in turn provide tasks that may be needed to perform the practice. Each task has examples and references.

Prepare the Organization - Examples

- Make sure the security requirements are defined and understood by your entire organization early. Update security requirements annually - at least. (PO 1.1)

Protect the Software - Examples

- Use code signing to help protect the integrity of executables (PS 1.1)
- Use an established certificate authority for verifying release integrity (PS 2.1)
- Share provenance data e.g. in a software bill of materials [SBOM] (PS 3.2)
- Reuse existing, well secured software (e.g. open source frameworks) instead of duplicating functionality. (PW 4.1)
- Implement "clean builds" and perform all builds in a dedicated, highly controlled build environment (PW 6.2)

Produce Well Secured Software - Examples

- Treat build systems like production systems by securing and hardening development endpoints. (PO 5.2)
- Establish a vulnerability disclosure program (RV 1)
- Monitor vulnerability databases (RV 2)
- Have a security response playbook (RV 3)
- Deliver remediations with automation (RV 4)
- Record root causes in a wiki (RV 5)
- Analyze root causes overtime (RV 6)

LEARN MORE: Read the Chainguard blog series on the SSDF at https://blog.chainguard.dev.