SONiC software development lifecycle

This document describles SONiC development lifecycle, it particularly focuses on addressing security risks that could be introduced during the development process.

Overall development of SONiC includes the feature design, coding, test, review, merge and release etc. See also: Developing Guide

For security aspects, SONiC is following: Microsoft Security Development Lifecycle

Open source contributions

SONiC is open source. All source code is available to the public and anyone with CLA signed can submit changes to a review process which results in accepting or abandoning the changes.

SONiC's contibuting guidelines:

Becoming a contributor

Contributing Workflow

Code review process

Code reviews are conducted as described in the contributing guidelines, any one can review the code but the code reviews are only accepted (merged) when a maintainer is satisfied with the quality of the change.

The github review tool (Pull request) helps providing high quality reviews:

- Contributors can work on private forks/branches and send pull request against SONiC official branches
- All changes and comments can be seen by all reviewers
- Re-worked material can be seen in context with previous commits etc.

Security aspects of code review: Security code review.

Reviewers would pay attention to stack buffer overflow, heap overflow and integer arithmetic errors etc security related aspects.

SONiC repository maintainers and merge process

The repository maintainers are responsible for maintaining the quality and integrity and therefore the security of the whole project.

See also: Governance

There were tens of repositories that is used to build the SONiC image. Each SONiC repository accepts changes only from the maintainers.

Pull-request sent by contributor should resolve conflicts before approval, merge process is done by maintainers. The merge is done by fast-foward without conflict. Git commit message include the PR# and issue ID if applicable.

Use of open source projects

SONiC build process pulls in other open source projects, including debian pacakges, docker, redis, and many others. The exact list is provided by a seperate dependecy tracker document: SONiC Dependency Tracker

The quality of these projects matters because they can introduce security vulnerabilities into the SONiC project. The open source packages are from well-known sources, specific releases are used, and the cryptographic hashes are obtained from trusted sources and validated during the build process. All such hashes are stored in the SONiC build repository sonic-buildimage as submodule commit-id or tag/commit-id in Makefile. This mechanism can control the exact set of code SONiC retrieves.

The SONiC development team reviews the packages SONiC uses for known security vulnerabilities, and updates the dependecies accordingly. See: SONiC Dependency Tracker

Development tools

SONiC relies on web-based tools such as Github and Jenkins for development and testing. SONiC maintainers have administration access to these sites, and rely on the tools to enforce their security requirements.

GitHub offers 2-factor authentication (2FA) for user authentication. GitHub also offers a way for project owners to require 2FA as a prerequisite to edit the project.

Work items

Project management uses work items to track features and defects to help ensure SONiC code is the way it was intended. Specifically, work items help identify which security items are reported and fixed.

Work items are recorded as Git issues in each repo such as sonic-buildimage issues and in other sonic repositories. These generally include defects, features, and wishes. Anybody can read, create, and comment on issues. The SONiC maintainers can assign owners, tag, and close issues.

The responsible disclosure model is to not create public work items for high-impact vulnerabilities (easier to reproduce and have larger impacts) until a mitigation is available. See: SONiC reporting security issues

Test

Testing is a core part of security assurance.

SONiC runs continuous integration tests for core componment like sonic-swss-common and sonic-swss etc repos by Jenkins servers during the pull-request review process.

For security aspects, static code analysis is conducted during this phase.

SONiC community has nightly testbed that runs on varies hardware with SONiC image from master and release branches. Testbed details are here: SONiC testbed

The SONiC community also has active downstream development teams that use SONiC and submit issues and fixes back into the SONiC project, including security items.

When releasing SONiC, penetration test is done and reported during the release cycle. It will be reworked if any security issues are poped up.

Release

The SONiC project has regular releases every 3 months since a few years ago.

The SONiC release checklist is performed for each release going forward.