SAFECode Comments on Draft Baseline Criteria for Consumer Software Cybersecurity Labeling

Steve Lipner
Executive Director, SAFECode
16 December 2021

General Comments

SAFECode appreciates the opportunity to comment on the Draft Baseline Criteria for Consumer Software Cybersecurity labeling. We understand that the criteria are intended as an input to the creation of a labeling program rather than a complete definition of such a program, and our comments are consistent with that intent.

In general, we believe that the draft criteria do a good job of identifying the major considerations that should be reflected in a label for consumer software. However, we do have some comments that we believe should be considered as NIST finalizes the criteria required by Executive Order 14028.

The draft criteria require that, to qualify for the label, software be free from known vulnerabilities. While developers strive to ensure that the software they release is free from known vulnerabilities, meeting such an absolute requirement may be impractical and/or inappropriate. If a developer receives a vulnerability report late in the product cycle (say on the day before planned release), the developer will conduct a risk assessment to determine whether the vulnerability report warrants a delay in release. A critical vulnerability will likely be a "ship-stopper"; if a vulnerability is less severe than other vulnerabilities already fixed by the planned release (to pose one hypothetical example), the responsible action is to go ahead with the release and follow up with an update to fix the newly reported problem. It would be better for the criteria to require that the developer remediate known or potential vulnerabilities based on risk and consistent with the developer's secure development process as is required by Task RV.2.2 of the SSDF (NIST SP 800-218).

The draft criteria assume in several places that the developer will release software updates for the product, but they don't appear to make that practice a requirement. The criteria should explicitly require that the developer have and describe a risk-based process for releasing security updates.

We commend the decision to require that developers of consumer software implement a secure development process that is consistent with the NIST SSDF. However, we believe that many developers, especially smaller developers, will be uncertain about what such consistency entails. We believe it would be useful, both for the consumer software labeling criteria and for other applications, if NIST were to publish several full-scale complete and detailed examples of acceptable secure software development process descriptions.

Consumer software, especially for smartphones, will likely inherit many security mechanisms from the underlying operating system. In fact, reliance on operating system security features is almost always a better approach for an application developer than "rolling their own." The draft criteria make this point with regard to use of cryptography (Section 2.3.3.5) but it can equally apply to implementation of other security measures such as protection of secrets and multifactor authentication.

We are aware that the IoT Cybersecurity Labeling Criteria were created with significant awareness of and input from other countries that have created similar labeling criteria and programs. We understand

that other countries are also working on software labeling programs and encourage NIST to collaborate with those countries so that both consumers and developers will benefit from fewer more robust labeling program.

Specific Comments

- Page 4, Section 2.1 – This point may be out of scope, but many consumers do not make a technical distinction between security and privacy protections. The authors of the criteria should consider including attestations about the developer's use of personal data that the application collects.
- Page 6, Section 2.3.1.1 – The attestation as to software provider should identify an email contact alias or telephone associated with the claims. Identification of a specific individual is neither scalable for large development organizations nor practical in a world of frequent turnover of personnel.
- Page 6, Section 2.3.1.2 – On the last line of "Description" change "but is outside" to "but that is outside."
- Page 6, Section 2.3.1.2 – On the last line of "Desired Outcome" change "varying" to "different."
- Page 7, Section 2.3.1.6 – This section requires that the developer "report" vulnerabilities but not "remediate" or "patch" them.
- Page 9, Section 2.3.3.4 – It may be worth referring to "hard-coded" secrets in the "Description" text as well as the title of this section.
- Page 10, Section 2.3.4 – The list of types of PII included in the "Assertions" seems to be pretty limited. Should email addresses, physical addresses, and telephone numbers also be included?
- Page 12, Section 4 – It appears that "the supplier" on line 5 of this section should begin a new sentence.
- Page 13, Section 4 – Under 1 and the first sub-bullet, there is a reference to "operating environment." It is not clear what is meant by operating environment in this context.
- Page 13, Section 4 – Under 2, the first sub-bullet, and in particular the text after "as well as" is unclear.
- Page 14, Section 4 – Under the fourth sub-bullet, it would seem that the developer should maintain the documentation listed whether the using self-declaration or third-party conformity assessment.
- Page 18 – Under the section on "Addressing Potential Weaknesses" it seems that the risks of the "halo effect" are greater if there is no penalty for false or misleading labeling.