Public Comments for ICR 202311-1670-001

Author Full Name : Moti Gindi **Received Date :** 12/17/2023 10:24 AM

Comments Received :

Section III, clause 4

We suggest to rephrase sub-clause 4a), as following (suggested changes marked in ***):

a) The software producer operates these processes on an ongoing basis and, at a minimum, *** upon any material change, *** and prior to product, version, or update releases;

Rationale:
In modern software development processes, the software is changed continuously. CICD practices create a situation in which for most software companies the notion of a “release” is no longer effective. Changes in software, across first and 3rd parties, are happening continuously. Every code change has the potential to introduce risk, whether in the form of a vulnerability, the expansion of your application attack surface, the introduction of a business logic flaw, etc. And as development and deployment velocity accelerates, the pace of change and, thus, the potential for risk also accelerates.

A material code change can be defined as any change that can potentially introduce risk into an application. For example, a material code change could be: the addition of or changes to entry points (e.g., APIs), the introduction of new technologies and frameworks (e.g., DB or authentication framework), exposure of sensitive data (e.g., PII, PCI, PHI), new or unexpected changes within open source dependencies, configuration changes to pipelines or infrastructure and more.

In order to understand, manage and disclose the true state of security risks in a software, it is crucial to be able to identify material changes and deeply inspect their design and contents for security vulnerabilities. Although they are not explicit vulnerabilities, they may expand or alter the application attack surface or introduce business or application logic flaws, thus potentially exposing your organization to risk.

In our view, monitoring and reviewing material changes to software is the only effective way to assure PW.1 in SSDF Practices and Tasks - and specifically PW.1.1 (Use forms of risk modeling to help assess the security risk for the software) and PW.1.2 (Track and maintain the software’s security requirements, risks, and design decisions) - in modern software development, and should therefore be one of the items attested directly as part of this initiative.