Author Full Name : Mike Barwise **Received Date :** 12/11/2023 09:43 AM

Comments Received :

The current draft of the attestation form raises several causes for concern:

[1] The definition of "security protections" is somewhat vague. Is it restricted to software immunity from cyber attack, does it extend to protection against adverse consequences of any faults in software, or to some point in between? This should be clarified as it has a major bearing on expectations of development practice. I strongly advocate the widest possible definition as there have been, and are likely to continue to be, numerous incidents with root causes unrelated or only peripherally related to cyber attack but with comparably far reaching consequences (see Annex documents).

[2] What is attested to throughout is merely the presence of processes, rather than their efficacy. For example, "4) The software producer employs automated tools or comparable processes that check for security vulnerabilities" provides no objective assurance that vulnerabilities will actually be detected, particularly in the case of conceptual or logical errors in design. Significantly, current development practices such as 'agile' largely circumvent the formal design stage, greatly increasing the likelihood of conceptual or logical errors that could pass undetected by post-development testing (see Annex 1).

[3] For an attestation of secure software development practice to be meaningful in terms of results it must make adequate reference to levels of designer, developer and tester competence. Regardless of any current official expectations, software development remains the only engineering discipline in which entirely self-taught and self-certified practitioners can engage in critical systems design and implementation. It should be noted that practically all developer training currently concentrates on languages, proprietary development systems and libraries, to the exclusion of the rigorous approach and body of first principles that underpin all other engineering disciplines. Consequently, although both the SSDF and this draft attestation appear to take the adequacy of designer, developer and tester expertise for granted, it is currently the weakest link in the supply chain, for which the use of tools cannot entirely compensate.

[4] As in other domains of compliance (not least operational infosec), the act of implementation tends to imbue management processes locally with a aura of efficacy they commonly do not merit, causing their deficiencies to be overlooked. This self-confidence will be reflected in any self-attestation. External review is typically the only sure way to detect such deficiencies.

[5] Due to the issues identified above, the draft attestation is unlikely to ensure either consistency between implementations or overall improvement of practice where needed. Consequently no common standard of actual performance can be assumed on its basis, and there is significant danger of it supporting meaningless "compliance theatre" whether intentional or not.

[6] Supporting information is provided in the accompanying document:
Integrated InfoSec on draft Secure Software Development Attestation Common Form Explanatory Annex.

END