Change request title:

Major revision of Secure Software Lifecycle Knowledge Area

Change rationale:

CyBOK was originally scoped in 2017 and several Knowledge Areas (KAs) have not seen any significant change requests from the community. This may be a reflection that the KAs continue to represent an up-to-date view of the foundational knowledge in the field. However, it is prudent for the CyBOK editorial team to initiate a proactive review of all five categories to ascertain that this continues to be the case and, if not, identify change requests on which the community may be consulted as part of the normal CyBOK update process.

The CyBOK editorial team, therefore, initiated a pro-active review of the KAs within the Software and Platform Security category. A panel of experts reviewed the KAs within the category and provided input on potential changes. The proposed change request has been distilled from these reviews and panel discussions.

The Secure Software Lifecycle KA continues to represent a strong and cohesive body of knowledge for the topic. However, over the last few years, new developments and considerations have emerged including areas of concern such as software supply chain security, decommissioning as well as new regulations and standards. The proposed change request is to update the KA to incorporate these elements.

Description of change:

Revision to the KA to:

(1) Incorporate the following new topics:

- Risks arising from the software supply chain and how to mitigate these. This should also include relevant strategic initiatives such as Software Bill of Materials and issues such as liability and attestation.
- New regulations and standards that have emerged, for instance, NIST SSDF, FDA's Premarket Cyber Security Guidance. Other regulations should also be discussed that impact secure software lifecycle practices, e.g., NIS2, CCPA, CRA.
- The role and impact of generative AI and automation assistants such as copilot on secure software development practices.
- The challenges and complexity of dependency management arising from a variety of APIs and libraries, and NPM-like patterns.
- The role and impact of vulnerability reward programmes.
- Coverage of human and organisational factors, e.g., challenges that organisations face when adopting secure software lifecycles, how developers delegate responsibility to others, challenges faced by developers when working with security tasks, security culture, the role of security champions and having directly relevant training on secure development practices and coding standards.

(2) Update the following topics:

Date: 6 June 2025

- *Section 2: Secure Software Lifecycle Processes* to include initiatives such as platform hardening, update management and an expanded focus on later stages such as decommissioning. Updates are also needed to discuss how the Microsoft SDL has changed (the new version is centered around zero trust and 10 activities), evolution of the OWASP SAMM (SAMM is now elaborated to 15 security practices with 6 activities each), and BSIMM.
- *Section 3.2: Agile and DevOps* to discuss the automation and integration of security tooling (including Software Composition Analysis) as part of the CI/CD pipeline in more detail.
- *Section 3.3: Cloud Computing* to include coverage of cloud native development and the role of "clean room" computing environments.
- *Section 4: Assessing the Secure Software Lifecycle* to include other national programmes, e.g., the Security Belts maturity model: https://github.com/AppSecure-nrw/security-belts that has seen successful usage in Germany.

Minor points of detail need to be addressed, e.g., pentesting not being only blackbox and addition of references that expand some of the current descriptions, e.g., resources on gamification for learning. Some standards that were in development at the timing of writing, e.g., ISO 21434, have since been finalised and such details will also be checked and updated.

Depends on KAs:

- Distributed Systems Security, Software Security, Cyber-Physical Systems Security

Depends on External Knowledge:

- SWEBOK 3.0: Chapter 3.2. Managing Construction
- SWEBOK 3.0: Chapter 8.2. Software Lifecycles
- SWEBOK 3.0: Chapter 10.2. Software Quality Management Processes

Note: The above dependencies will need to be updated to SWEBOK 4.0

References:

(1) Heather Adkins, Betsy Beyer, Paul Blankinship, Piotr Lewandowski, Ana Oprea, Adam Stubblefield (2020). **Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems**, *O'Reilly Media.*
(2) Adam Shostack (2023). **Threats: What Every Engineer Should Learn from Star Wars**, *Wiley*.
(3) https://shostack.org/games
(4) https://www.lawfaremedia.org/article/incentives-for-improving-software-security-product-liability-and-alternatives
(5) https://safecode.org/blog/secure-by-design-the-u-s-government-and-requirements-for-secure-development/
(6) https://aws.amazon.com/clean-rooms/
(7) S. Taaibi, S. Dziwok, L. Hermerschmidt, T. Koch, S. Merschjohann, and M. Vollmary (2024). **Security Belts: A Maturity Model for DevOps Teams to Increase the Software Security of their Product - An Experience Report**. *Presented at the 30th Americas Conference on Information Systems, Salt Lake City*. https://ris.uni-paderborn.de/record/53811

Date: 6 June 2025