INTEGRATING SECURE CODING PRACTICES AND RISK ASSESSMENT IN MODERN SOFTWARE DEVELOPMENT

Cand. Sc., assoc. prof. V. Savchenko, Senior Lecturer. O. Mnushka, NTU "KhPI", Kharkiv

Secure coding involves the practices, methods, and principles software developers use to create programs that meet the latest security standards. Secure programming aims to avoid vulnerabilities and attacks that could compromise the confidentiality, integrity, and availability of the program and its data [1-3].

ISO/IEC 15408 The Common Criteria for Information Technology Security Evaluation (Common Criteria or CC) is used to evaluate the security properties of information technology (IT) products and systems. Assessment of Assurance Levels (EAL) is a framework implemented in Common Criteria used in information technology security assessment. First, this is related to assessing security functions and the guarantee of IT products and systems. EAL provides a structured approach to evaluating the security of a product or system based on a set of predefined criteria. The EAL structure consists of seven levels, ranging from EAL1 (lowest) to EAL7 (highest). Each level represents a higher degree of confidence and rigor in security assessment. The evaluation process includes examining various aspects, such as the design, implementation, and configuration of the product or system and its documentation and security testing. Despite its general acceptance, software certification, according to Common Criteria, does not mean the product is free of security vulnerabilities. It only meets the security requirements specified in the protection profile or security goal. Based on the general practices of secure programming and security assessment, we will form the main tasks the developer faces during the development of secure software: analysis of security requirements, design based on the principles of secure architecture and design (least privilege, defense in depth, fail-safe defaults, etc.); security testing; code verification and auditing; security training and awareness; compliance with safety standards and regulations; incident response and monitoring.

The discussion covers alternative and additional approaches to software development that impact quality and security, including a risk assessment-based approach to security [4]. It is demonstrated that, under modern conditions, comprehensive solutions are required to ensure software security.

References: 1. SEI CERT C Coding Standard: Rules for Developing Safe, Reliable, and Secure Systems Software Engineering Institute / Carnegie Mellon University, 2016. **2.** *Kohnfelder L.* Designing secure software: A guide for developers / San Francisco: No Starch, 2022. **3.** *Mnushka, O., Savchenko V.* Security Model of IOT-based Systems // 15th IEEE TCSET Int. Conf. – Lviv-Slavske, Ukraine, 2020. – pp. 398-401, doi: 10.1109/TCSET49122.2020.235462. **4.** St*amatis D.H.* Risk management using failure mode and effect analysis (FMEA) / Milwaukee, WI: ASQ Quality Press, 2019.