International Journal of Advancements in Technology

Perspective

Security Quality Assurance of the Vulnerabilities in the Software Systems

Davletova Karazhanov*

Department Manufacturing Engineering, Norwegian University of Science and Technology, Trondheim, Norway

INTORDUCTION

A security pattern is one of the reusable building blocks of a secure software architecture that provides a solution to a specific, recurring security problem in a specific context. Incomplete or non-standard implementation of security patterns can create vulnerabilities and invite attackers. Recognition of security patterns therefore improves the quality of security features. This document proposes a Security Pattern Detection (SPD) framework and its internal pattern matching techniques. This framework provides a platform for data extraction, pattern matching, and semantic analysis techniques. It implements Ordered Matrix Matching (OMM) and Non-Uniform Distributed Matrix Matching (NDMM) techniques. OMM technology recognizes a security pattern matrix within the Target System Matrix (TSM). The NDMM methodology determines whether the relationships between all classes of security patterns are similar to the relationships between some classes of TSM. Use semantic analysis to reduce false positive rates. We evaluate and compare the performance of the proposed SPD framework using both matching techniques based on four independent case studies. The results show that the NDMM methodology provides a place for security patterns and is highly flexible, scalable, and highly accurate with acceptable memory and time consumption for large projects.

DESCRIPTION

Maintaining security in software systems is a dynamic struggle. Securing legitimate access from unauthorized use in software systems is a daunting task. Security patterns have always existed in all security applications for decades. Since 1998, researchers have begun to discover them and unravel their uses in software system security. These patterns are repeated to provide specific security features such as: "Login window" security features. "Login window" is an example of the Single Access Point (SAP) security pattern, but "Login window" was used before the discovery of the SAP security pattern. Researchers classify security patterns in a standard catalog of security patterns. These catalogs provide application details for safety patterns using standard safety pattern templates. Discovering and documenting security patterns is essential for designing, implementing, and improving the security features of software systems. This is due to the following characteristics: A software security pattern is one of the building blocks of software security that prevents attackers from misusing software systems. Security patterns are solutions to security design problems. They are reusable security design components of software systems that provide solutions to specific recurring security problems in specific contexts. It is easy to build and delivers sophisticated software security systems quickly and efficiently. Security patterns are transplanted with security design knowledge from experienced security professionals. Thus, novice security developers can use security patterns to easily achieve high quality security functionality when developing software systems.

Secure software development uses security patterns as one of the security micro-architecture components. Ensuring correct, standardized implementation of security patterns in software systems is an essential task of the testing and quality assurance process. Standard application of the safety pattern means that the safety pattern is implemented in the software system according to the detailed documentation of the safety pattern in the available catalog. However, to ascertain the quality of security patterns, a software system must detect the patterns. Software has many quality attributes, including: Performance, modifiability, and security. Security quality attributes are inherent in protecting software systems that can run securely on any device, anywhere, anytime. However, these quality features influence each other. For example, implementing advanced security features degrades software system performance. Frequent mutability also has a negative impact on security.

In forward software engineering, the status of security patterns can be incomplete, abused, or incomplete after security requirements have been implemented. Testing and other quality assurance processes are used to identify vulnerabilities and provide another opportunity to correctly implement security patterns. In case of deviations, the source model is updated by backtracking from implementation to design and requirements phases. However, once a security pattern has been successfully implemented, an evaluation process is applied to use the security patterns in combination to enhance the security level of the software system and the security objectives of the software system

Correspondence to: Davletova Karazhanov, Department Manufacturing Engineering, Norwegian University of Science and Technology, Trondheim, Norway; E-mail: Davlekarazhanov@ife.no

Received: 06-Dec-2022, Manuscript No. IJOAT-22-20645; **Editor assigned:** 08-Dec-2022, PreQC No. IJOAT-22-20645 (PQ); **Reviewed:** 22-Dec-2022, QC No. IJOAT-22-20645; **Revised:** 20-Feb-2023, Manuscript No. IJOAT-22-20645 (R); **Published:** 27-Feb-2023, DOI: 10.35248/0976-4860.23.14.231.

Citation: Karazhanov D (2023) Security Quality Assurance of the Vulnerabilities in the Software Systems. Int J Adv Technol. 14:231.

(such as confidentiality). For example, multiple security patterns can be used to harden the system access control architecture of a software system. The combined use of single access point, checkpoint, and security session security patterns ensures the effectiveness of users against legitimate use of software system resources.

CONCLUSION

Quality assurance processes cannot achieve high quality implementation of security patterns without pattern recognition, which provides information about the structure and location of implemented patterns. Identifying security patterns is the first step in measuring security at the design and architectural level of a software system. Security measurements provide information about the actual security expectations of software systems and provide information about quality improvements. The discovery process gives Security Quality Assurance (SQA) team confidence. The SQA team checks the status of security patterns as countermeasures against security attacks in target software systems. Lack of security patterns creates security vulnerabilities in software systems. These loopholes are commonly called security holes, flaws, or vulnerabilities. This document uses the term vulnerability. Security patterns are used to mitigate security vulnerabilities. Vulnerabilities allow attackers to exploit software vulnerabilities to gain unauthorized access to targeted software systems. Recognizing security patterns is therefore securing software systems and supporting security quality attributes. This helps in the elaboration and construction phases of the secure software development life cycle. Detecting instances of security patterns in target systems helps security developers understand missing security features.