A Knowledge-based Expert System for Supporting Security in Software Engineering Projects

Ahmad Azzazi[1]
Dept. of Software Engineering
Applied Science Private University, Amman, Jordan

Mohammad Shkoukani[2]
Dept. of Computer Science
Applied Science Private University, Amman, Jordan

***Abstract*—Building secure software systems requires the intersection between two engineering disciplines, software engineering and security engineering. There is a lack of a defined security mechanism for each of the software development phases, which affects the quality of the software system intensively. In this paper, the authors are proposing a framework to consider the security aspects in all the phases of the software development process from the requirements until the deployment of the software product, with three additional phases that are important to automatically produce a secure system. The framework is developed after analyzing the existing models for secure system development. The key elements of the framework are the addition of the phases like physical, training, and auditing, where they improve the level of security in software engineering projects. The authors found so a solution for the replacement of the knowledge of the security engineer through the construction of an intelligent knowledge-based system, which provides the software developer with the security rules needed in each phase of the software development lifecycle and it improves the awareness of the software developer about the security-related issues in each phase of the software development lifecycle. The framework and the expert system are tested on a variety of software projects, where a significant improvement of security in each phase of the software development process is achieved.**

Keywords—Knowledge-based systems; security engineering; software development process; expert systems

I. Introduction

Software-intensive systems are a major factor in many business areas. There is an increasing need for such software systems that could help us in the daily life. These software systems must fulfil certain requirements. The usage of the internet as a platform for electronic commerce and online banking pushes the need for securely reliable software systems. The software systems must be secured against potential threats and attacks. Attackers are looking for security holes in these software systems to increase their chance of getting sensible information that could be used in an illegal way. As a result of all these facts, the software developers must think about building secure software systems before beginning with the real software systems, and a process of ensuring a secured software system must be followed [1].

There are many disciplines involved in such a process for developing a secure software system, where software engineering is one of the most important disciplines. To develop software in general the authors need a software engineer. The task of such an engineer is to produce quality software within the given constraints such as time, cost, etc. The software engineer must follow a generalized development process suitable for the desired software [2].

There are many stages in a software development process, the requirement definition, system design, systems implementation, and other stages. In all these different important stages security must be considered and developed in each stage and for each stage. Therefore, there is a need to integrate the security related tasks into each stage of the software development process. The role of a security engineer appears as a part of the software development team. The need for a security engineer or software development enforces the whole development of a software system to interact with different influencers on the final software product [2, 17].

One of the most important dangerous practices during software development is the lack of detailed security requirements. The system requirements must include those ones related to security requirements. The security requirements must be specified and integrated into the whole software development.

The security of existing software majority was built as ad hoc solution that means after the development of the software, security was added to the system, even for security critical systems. The Development of secured software gives us the interaction between two disciplines, software engineering and security engineering. Building secure software is a process in which software security is considered in all phases of a software engineering life cycle [3, 16].

The focus will be on the security activities at each phase of software development, and the authors will present a Knowledge-Based Expert System for Supporting Security in Software Engineering Projects.

The paper is organized as follows: section one is the introduction, section two is the background section which provides a description about software, software engineering, software process, software engineering for Security, security engineering and knowledge-based system for secure software development. Section three is the security software engineering framework using a knowledge-based system provides the proposed framework solution for the research problem stated in this paper. The fourth section includes the Web-Based Security Expert System for Software Engineering Projects. Section five discusses the contribution. Finally, section six is the conclusion.

II. Background

In this section, the authors are giving an overview of the most important terms used in this research. First, the authors are introducing the term software, software engineering essentials, and software engineering for security-critical systems. Secondly, the authors are describing the information security essentials, the common protection mechanisms, and security engineering. Finally, a brief description of the knowledge-based system to support security in the software process is introduced.

A. *Software*

Software could be defined as the computer programs, data and documentation which support processes to do an automatic problem-solving task [2].

Due to “pressman,” the software is a vehicle for delivering a product, which supports or directly ides system functionality, controls other programs like an operating system, effects communications like, networking software or helps building other software like, software tools [4].

To deliver software as a product, many properties should be considered, among the most important characteristics is the security of the software as a software system, which could be defined as the system attribute that reflects the ability of the system to protect itself from external attacks that may be accidental or deliberate the resources, the prevention of unauthorized disclosure of information, the extent that the software itself must be hidden or obscured, the trustworthiness of data or resources [3].

The main principal security issues are availability, confidentiality, and integrity. There are other extended properties like non-reputability, accountability, and authenticity.

B. *Software Engineering*

Software engineering deals with a detailed production of quality software. There must be a way of organizing the various stages of software development, a process. There is a need for different tools, technologies, and techniques due to the given software problem with the consideration of resources and constraints to be applied. It is the targeted provision and systematic use of principles, methods; process models, concepts, and tools for the development of software systems with a quality focus [4].

C. *Software Engineering for Security*

Secure Software development usually requires the engagement and usage of a defined software process which includes the intensive usage of tools, methods, techniques, and technologies. Security errors appear from the lack of a detailed definition of security in the requirements stage, design stage, or coding stage. Therefore, an urgent need for a security engineer appears before going into a stage of the software development process. One must carefully consider the security aspects of the software product from the beginning with requirements and moving on through later lifecycle activities, ending with the deployment and the administration of the software product [5, 15].

D. *Security Engineering*

The duties of a security engineer are to ensure the security of software systems during and after the software development process. Security engineering requires cross-disciplinary expertise, ranging from cryptography and computer security through hardware tamper-resistance and formal methods to knowledge of applied psychology, organizational and audit methods [3, 18].

E. *Security in the Software Engineering Life Cycle*

Through all the phases of the software engineering phases, a deep consideration of the security aspects must be done for each phase. Beginning with the requirement phase, one should describe the security requirements of the software. In the analysis, a deep analysis of the security requirements should be made. Also, in the design phase, the security design consideration of each designed software component should be considered. In the coding phase, an immense value is given to secure coding and with the deployment phase, all aspects of the security issues must be considered [6].

III. Proposed Framework

Knowledge is a justified true belief. Knowledge is a higher level than data or information in a way that it is higher than both, the information is higher than data in its level of abstraction. It is the richest, deepest, and most valuable of the three [7].

There are two kinds of knowledge. The First one is the explicit knowledge, which can be expressed in words and numbers and shared in the form of data, scientific formulae, product specifications, manuals, universal principles, and so forth. This type of knowledge is transmitted in a formal and systematic way among all individuals. Knowledge is another type of knowledge; it is a personal level of knowledge, with difficulties in formulation of it, sharing and communicating it with others. The term knowledge-based stands for the internal structure to process symbols, which represents the information of the real world to achieve intelligent behavior.

The main components of a knowledge-based system are the knowledge base component, the inference component, the user interface, the knowledge acquisition component, and the explanation component [7]. There are diverse types of known knowledge-based systems; the most important types of them like the rule-based systems and expert system, which is a class of software systems, which serves based on expert knowledge to the solution or evaluation of certain problem definitions. Knowledge is represented in diverse ways in expert systems like, the production rules, the semantic networks, and the logic statements representation [7, 14].

Secure Software Systems are associated with a solid software process implementation. Therefore, one must have a security engineer in the software team, who will then guide the software team with security related knowledge through all the phases of the software development life cycle. The lack of security engineers could represent an obstacle in the software development process [8, 13].

Security Engineers could not be available for each software engineering project. In some cases, security engineers are too rare to be found for all the needs of security critical software projects. In other cases, it is too expensive to get extra security engineers into the development team. Also, in some geographical areas it is difficult to find these security engineers. If the security experience of the software development team is not sufficient, this leads to no secure software and the failure of other software. At the end, the software product may complete fail due the fact of the lack of the security engineers [9, 12].

Therefore, the authors suggest in this work the use of a knowledge-based system to assist the software engineers with the needed security engineering activities. This knowledge-based system is then integrated into the software development framework.

The researchers are proposing a new software engineering framework using some of the knowledge gathered.

Some reasons to choose such a framework are that:

- It covers all aspects of software development life cycle.
- It includes a knowledge base for each phase with the appropriate security related knowledge for each phase.
- It checks the security related activities in each phase.
- The knowledge is adaptive and increased with time.
- Additional phases are easily added to include the security activities in additional phases to ensure more security.
- It is easily implemented and accessed.

The new framework should consider the security aspects in all the phases of the software development process from the specification of the requirements until the construction and after that the deployment of the software product to the security training of the end users.

The framework focuses on the security activities on each phase of the software development process using the general process activities of the software engineering process with knowledge-based system, which helps the developer to get the expert knowledge of a security expert in each phase of the development lifecycle [10, 11].

There are many advantages when using a knowledge-based system (expert system) in the proposed framework like:

- A security knowledge-based system is an intelligent information system, in which security knowledge with methods of knowledge representation and knowledge modeling.
- A knowledge-based system is easy to understand.
- It is more easily to update it according to the increasing level of security knowledge.
- The security expertise feature, where a security expert can make expert level decisions about security.
- The symbolic reasoning feature, where the knowledge is represented symbolically and is given back through a reasoning mechanism.
- The deep security knowledge feature, where a security knowledge base for the different software development lifecycle contains complex security knowledge.
- The security self-knowledge feature, where the system can examine their own reasoning and is able to explain why a specific conclusion is reached.
- They have advantages over conventional systems like; they do not require all initial facts, changes in rules are easily implemented, execution may be done by heuristics or logic, and the effective manipulation of large knowledge.
- They have many benefits when they are used like; increased outputs, increased productivity, decreased decision-making time, increased process, and product quality, reduced downtime, capture of scarce expertise and flexibility.

The are some limitations when using expert system like; that the knowledge is not always readily available, it is sometime difficult to extract expertise from humans, where we have different approaches for the knowledge extraction, and one faces lack of end user trust when using the expert system. Therefore, the researchers have divided the knowledge-based system into many knowledge-based systems for each phase of the different phases of the software development lifecycle to make it easier to get new rules for the expert system extracted from the different sentences of the security expertise.

The proposed framework consists of several steps leading to the development of secured software as shown in Fig. 1.

The Framework begins with the requirement phase, which is specially designed for secured systems and in which the Framework collects the security engineering activities needed for obtaining the security requirements of the software products.

After obtaining the security requirements, the system performs a security test of obtained requirements. The analysis phase of the previous phases follows with the conjunction of the security activities designed for this phase with feedback to the requirements phase to make changes on the requirements of any new requirements that arise in the analysis phase. After it the system begins with the design phase for building the secured product, which has its own security engineering activities designed for this phase. The authors have then a special test for the security validation of the design phase. After that the authors begin with the coding phase with the consideration of the security engineering activities for this phase. The next step in the proposed framework is the security testing of the code that the authors have constructed in the previous step with feedback to the coding phase, to repair any security holes in the code of the product. In the next step, the authors have the deployment of the security-critical system with the security engineering activities that must be done for this phase. The last step in the framework is the security test for the software after the deployment; it includes the feedback to the deployment phase for making any changes regarding the secure deployment of the software product.

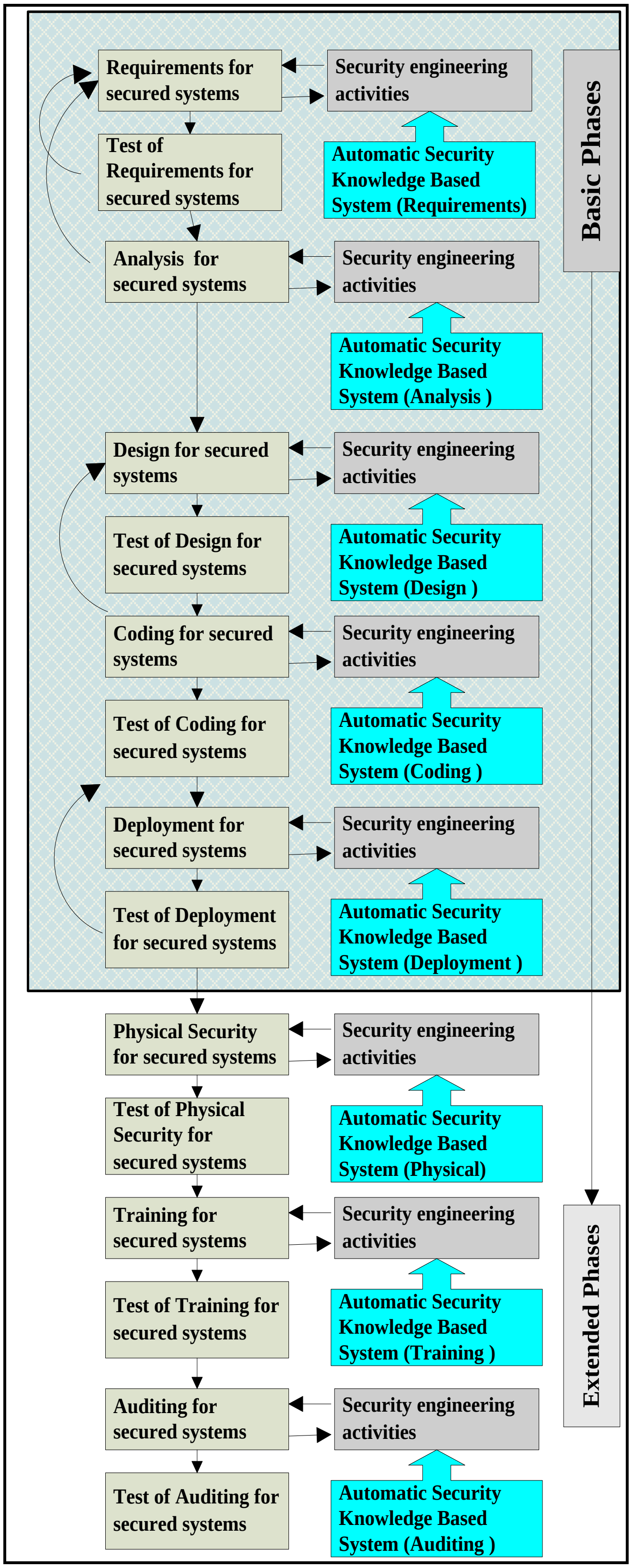
Basic Phases
Requirements for secured systems
Security engineering activities
Test of Requirements for secured systems
Automatic Security Knowledge Based System (Requirements)
Analysis for secured systems
Security engineering activities
Automatic Security Knowledge Based System (Analysis)
Design for secured systems
Security engineering activities
Test of Design for secured systems
Automatic Security Knowledge Based System (Design)
Coding for secured systems
Security engineering activities
Test of Coding for secured systems
Automatic Security Knowledge Based System (Coding)
Deployment for secured systems
Security engineering activities
Test of Deployment for secured systems
Automatic Security Knowledge Based System (Deployment)
Extended Phases
Physical Security for secured systems
Security engineering activities
Test of Physical Security for secured systems
Automatic Security Knowledge Based System (Physical)
Training for secured systems
Security engineering activities
Test of Training for secured systems
Automatic Security Knowledge Based System (Training)
Auditing for secured systems
Security engineering activities
Test of Auditing for secured systems
Automatic Security Knowledge Based System (Auditing)

Fig. 1. Security Software Engineering Framework using Knowledge-based System.

The proposed framework is a framework that concentrates on highlighting the security activities to be done in each phase of the general software development lifecycle. It includes an intensive testing of security after each phase. All that leads to the development of a highly secure software product.

The Framework is simple; it could be followed easily from a software engineering perspective. The use of a knowledge-based system assists the software engineers with the needed security engineering activities in different phases of the software development lifecycle.

As shown in Fig. 1 the authors gain the security engineering activities in the requirement phase through an automatic security knowledge-based system especially designed for the requirements phase. For the design phase of the software process lifecycle, the authors have another automatic knowledge-based system for the security activities in this phase.

For the coding phase the authors have also another automatic knowledge-based system as a source for the security engineering activities. For the deployment phase the authors have an automatic knowledge bases system for the security engineering activities as a source. The complicated work the security engineer is done through the usage of different automatic intelligent knowledge-based systems.

Three additional phases are added to the general phases of the software engineering lifecycle, the physical security phase, where it is concerned about the physical security of every essential asset of the whole system, the security training phase, where it is concerned about receiving appropriate information about security training in a software project and the security auditing phase, where it is concerned about involving auditing and monitoring activities of the security requirements of the whole system.

Some limitations of the proposed framework are:

- The availability of knowledge in each phase.
- The changeability of the knowledge over time.
- The amount of knowledge which can increase dramatically.

IV. Web-Based Security Expert System for Software Engineering Projects

In this research, the authors tried to find a suitable solution for security in software projects, where most of the existing software, even software for security-critical systems, has been built and is being built in an ad hoc, unsystematic fashion. In this work, the researchers are representing the interaction between the software engineering discipline and the security engineering discipline. The researchers have made a deep study of the concepts from scientific literature, gathered the knowledge needed to propose a new software engineering framework for security critical systems.

In the proposed framework the researchers are considering the security aspects in all the phases of the software development process from the specification of the requirements until the construction and after that the deployment of the software product. This consideration of all the security activities is not an ad hoc solution to the software product. In the proposed framework security issues are implemented more efficiently. In the proposed framework the researchers considered all the phases of the general software engineering activities with the three additional for security important phases, the physical phase, the training phase, and the auditing phase. In all these phases, the researchers build up an appropriate knowledge base system (expert system) that should help the developers in applying the needed security engineering activities simple. The separation of the knowledge base system into knowledge base systems for each phase of the framework is made, so that the addition of new rules is specially done for each phase separately, which is simpler to do and requires minimal knowledge of security engineering knowledge.

To prove the need of the proposed framework the researchers conducted a deep survey, which provided us with more information about the security implementation issues in the software project.

This survey helped us in developing rules for each phase of the software development life cycle for the security knowledge-based system. This survey is a sample for descriptive analysis and hypotheses testing, to set up conclusions and recommendations about the usability of the proposed framework.

To prove the proposed framework, the researchers constructed an expert system case tool, the Web-Based Security Expert System for Software Engineering Projects (WB-SES-SEP). The construction of the distinct phases of the overall proposed framework is done in a web environment for many important named reasons. In the (WB-SES-SEP) one could select each phase of the 8 phases of the proposed framework, where one could begin with each phase, or the user could easily add new security rules to the knowledge-based system of each phase.

V. Contribution

The main contribution of this research is the new framework for secure software development using a knowledge-based system, where the modeling of security activity rules on each phase of the software development process using the general process activities of the software engineering process in done. In this framework the researchers added three additional phases, which are essential to get better secured software through the software development lifecycle. These three additional phases are not common in the software development lifecycles. The Framework is simple; it could be followed easily from a software engineering perspective. The use of a knowledge-based system assists the software engineers with the needed security engineering activities in different phases of the software development lifecycle, which solves the problem of the availability of security experts at software engineering projects.

The researchers found that the proposed framework consists of several steps leading to the development of better secured software, through the implementing of a case study, where the proposed framework is simply used through the case tool, which we built for this work.

Another contribution of the work is the results of the deep survey, which provided us with important information about the security implementation issues in software projects like:

- Most of the projects have security objectives, but the integration of security to the product is on average done.
- Only on average of the projects was the notice taken that security must be manageable.
- Very few of the project members have enough exposure to principles and techniques of secure application development.
- Security is considered and implemented as an ad hoc solution.
- Only very few of the projects have a person responsible for reviewing security.
- The security rules are very weak implemented in all the phases of the general software engineering phases.
- The availability of any information security policy is very weak.
- The physical security of the software project is rarely implemented.
- The security related training of people is not done in a good way.

VI. Conclusion

The researchers proposed a new framework for secure software development using a knowledge-based system, where the modeling of security activity rules on each phase of the software development process using the general process activities of the software engineering process in done. In this framework the researchers added three additional phases, which are essential to get better secured software through the software development lifecycle.

The researchers conducted an analytical survey and the test of the framework with the own built case tool led to the following features of the framework:

- One constructs better secured software when following the proposed framework.
- The security knowledge-based system for each phase of the framework consists of all the needed security rules for each phase.
- The framework is easily understandable and easily used. It is simple implemented and followed.
- Through the analysis the researchers gained very important security rules for each phase of the software development lifecycle and the additional cycles of the framework.
- New rules could be added very simply to the existing rules.

- The inference engine of the knowledge-based system, in which the researchers built, contains very accurate decision tables with a rule importance and a rule implementation level.
- The inference engine of the knowledge base system could decide about any implementation level of the rules in each phase separately, or about all the rules in the phase or about the security implementation level of the phase itself.
- With the case tool, the (WB-SES-SEP), the developer can gain experience about the rules to be implemented in a specific phase and one can repeat the processing of the phase rules until he reaches a satisfied level of security for the phase very easily.
- With the case tool, the (WB-SES-SEP), the developer can get a graphical analysis of the current implementation of all security rules in a specific phase very easily.
- The framework is easy assessing the software developer with the needed security engineering activities through a web-based interface accessible globally and all the time.
- The added phases to general software devolvement life cycle phases are improving the level of security in a software engineering project.
- The case tool, the (WB-SES-SEP), could be used as training tool for software developer in a security critical software project so that, they could identify certain security related problems that face the project members during each phase of the software development lifecycle.
- The case tool, the (WB-SES-SEP), could make the project members more concerned about security by displaying the needed security activities needed in each software engineering phase.

Acknowledgment

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the full financial support granted to this research.

References

[1] M. Mirakhorli,M. Galster & L. Williams, Understanding Software Security from Design to Deployment, ACM SIGSOFT Software Engineering Notes, Volume 45, Issue 2,2020, pp 25–26.

[2] Sommerville, I., Software Engineering, 10th Edition, Pearson India, 2018.

[3] Jens Bürgera et al, A framework for semi-automated co-evolution of security knowledge and system models, Jthenal of Systems and Software, Volume 139, May 2018, Pages 142-160.

[4] Pressman, R.S. Software Engineering: A Practitioner's Perspective, kindle edition, McGraw-Hill, New York, 2019.

[5] Lada Gonchar &Lada Gonchar, Implementation of Secure Software Development Lifecycle in a Large Software Development Organization, Proceedings of the 21st International Workshop on Computer Science and Information Technologies, 2019.

[6] R.Matulevičius, Fundamentals of secure system modelling,1st ed, Springer, 2017.

[7] Anthony J Rhem, Knowledge Management in Practice, Auerbach Publications; 1st edition,2016.

[8] Ross J. Anderson, Security Engineering, second edition, Wiley, 2008.

[9] A. Johanson and W. Hasselbring, Software Engineering for Computational Science: Past, Present, Future, Computing in Science & Engineering, vol. 20, no. 2, pp. 90-109, 2018.

[10] Ahmad AlAzzazi, Asim El Sheikh, Security Software Engineering: Do it the right way, Proceeding of the 6th WSEAS International Conference on SIGNAL PROCESSING, ROBOTICS and AUTOMATION, Corfu Island, Greece, 2007.

[11] Ahmad AlAzzazi, Asim El Sheikh, Security Software Engineering with a Knowledge Based Engineering, WSEAS TRANSACTIONS ON COMPUTER RESEARCH, 2007, pp.276-282.

[12] Riad, ABM Kamrul, et al. "Plugin-based Tool for Teaching Secure Mobile Application Development." INFORMATION SYSTEMS EDUCATION JOURNAL 19.2 (2021): 2.

[13] Yurin, Aleksandr Yu, and Nikita O. Dorodnykh. "Personal knowledge base designer: Software for expert systems prototyping." SoftwareX 11 (2020): 100411.

[14] Wang, Yingxu, and Omar A. Zatarain. "Design and implementation of a knowledge base for machine knowledge learning." 2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC). IEEE, 2018.

[15] Burnashev, R. A., Ismail Amer, and A. I. Enikeev. "Expert system building tools based on dynamically updated knowledge." Journal of Physics: Conference Series. Vol. 1352. No. 1. IOP Publishing, 2019.

[16] Burnashev, Rustam A., et al. "Research on the Development of Expert Systems Using Artificial Intelligence." International Conference on Information Systems Architecture and Technology. Springer, Cham, 2019.

[17] Alguliyev, Rasim, Yadigar Imamverdiyev, and Lyudmila Sukhostat. "Cyber-physical systems and their security issues." Computers in Industry 100 (2018): 212-223.

[18] Sultan, Sari, Imtiaz Ahmad, and Tassos Dimitriou. "Container security: Issues, challenges, and the road ahead." IEEE Access 7 (2019): 52976-52996.