Chapter 5
Towards a Security Competence of Software Developers: A Literature Review

Nana Assyne
https://orcid.org/0000-0003-0469-6642
University of Jyväskylä, Finland

ABSTRACT

Software growth has been explosive as people depend heavily on software on daily basis. Software development is a human-intensive effort, and developers' competence in software security is essential for secure software development. In addition, ubiquitous computing provides an added complexity to software security. Studies have treated security competences of software developers as a subsidiary of security engineers' competence instead of software engineers' competence, limiting the full knowledge of the security competences of software developers. This presents a crucial challenge for developers, educators, and users to maintain developers' competences in security. As a first step in pushing for the developers' security competence studies, this chapter utilises a literature review to identify the security competences of software developers. Thirteen security competences of software developers were identified and mapped to the common body of knowledge for information security professional framework. Lastly, the implications for, with, and without the competences are analysed and presented.

INTRODUCTION

The current explosive growth being observed in the software industry requires high-level corresponding software security. This is because "software vulnerabilities or flaws are often key entrance door for attackers" (Sametinger, 2013). They include buffer overflows, SQL injection, cross-site scripting, stack overflow, inconsistent error handling, and so on (McGraw, 2004). Previously, software security used to be an afterthought, but recently it is being addressed actively from the planning stage of software development. Additionally, in today's software development process, software testing includes security testing instead of only functional testing (Mano, Duhadway, & Striegel, 2006), thus making the security

DOI: 10.4018/978-1-7998-3149-5.ch005

competences of the developers more eminent in software development. Coupled with the fact that research work on software developers' competence is not lacking (Lenberg, Feldt, & Wallgren, 2015), the security competences of software developers should be well recorded in literature. But on the contrary, that is not the case. However, when they are recorded, they are recorded as a subsidiary of security engineers' competence instead of software engineers' competence, thus making it counterproductive to develop and maintain the security competences of software developers to the benefit of the possessors (developers), those who train the possessors of the competences (educators), and users of the competences (industry).

McGraw (2004) defines software security as "the idea of engineering software so that it continues to function correctly under malicious attack". And, Hazeyama & Shimizu (2012), goes further with the definition by stating that "software security deals with security during the whole software development process". On the other hand, software engineering competence is defined by the Institute of Electrical and Electronics Engineers (IEEE) as knowledge, skills, and attitudes of software developers to fulfil a given task in a software development project (IEEE, 2014). Thus, the author of this chapter defines security competence of software developers as those specific security competences required by a developer to deal with security during the whole software development process. An example is an SQL injection skills and security pattern skills.

As mentioned above, one cannot afford to leave software security as an afterthought; developers must strive to improve software security issues from the planning stage to the maintenance stage. The works of Cheng et al. (2008), Hilburn and Mead (2013), and Riehle and Nürnberg (2015) are studies that investigated methods to handle software security using the lifecycle of software development. It is also well established that vulnerabilities and flaws are the doors attackers exploit. Works such as Kaur and Kaur (2016), McGraw (2004), Park et al. (2010), and Wegerer and Tjoa (2016) confirm this assertion in literature. In addition, assailants of software systems are persons or entities, who are active and keep on improving their skills in attacking software systems to satisfy their desire (Cheng et al., 2008). However, the security competences of the developers of the software are not well established in literature.

Whilst introducing security engineering environment studies for software developers, Cheng et al. (2008) point out that there is urgent need to create an environment that integrates various tools and provides comprehensive facilities to the designers, developers, users, and maintainers of a software system (Cheng et al., 2008). The development and maintenance of such an environment requires knowledge of security competences of the developers to prepare and develop them to withstand the intrinsic difficulty of assailants of a software system (Cheng et al., 2008). This implies that security know-how of the developer is very crucial. Hazeyama and Shimizu (2012) and Hilburn and Mead (2013) reiterate the need for awareness to be channelled towards developers' skills regarding security. However, previous studies provide less concise and coordinated information on security competences of developers.

Summarily, these competences are scattered in several different studies. Thus, the following questions arise: *what are the security competences of software developers? How can they be improved?* As part of broader research on software developers' competences, we set our research question as *what are the security competences of a software developer that are available in literature?* The remainder of this work includes: Section 2 presents previous studies and background. Section 3 looks at the methodology used in this study. Section 4 looks at the results. Section 5 and 6 presents the discussion and conclusion.

PREVIOUS STUDIES AND BACKGROUND

In this section of the study, three literature review studies on software developers' competences are identified. These literature reviews are Cruz et al. (2015); Moustroufas et al. (2015) and Vishnubhotla et al. (2018). Two of the studies utilized systematic literature review methods and the last study employed a traditional literature review method. Cruz et al. (2015) and Vishnubhotla et al. (2018) that used systematic literature review, focused on specific areas of software developers' competence. Cruz et al. (2015) investigated the personality of software engineers and their roles in software development. Vishnubhotla et al. (2018) also presented the capability and competence measurement of software engineers, including team working in agile software development. Moustroufas et al. (2015) utilized a traditional literature review to evaluate the adequacy of software engineer competences and created a software competence profiling model for recruiting software engineers. Moustroufas et al. (2015) investigated and reviewed software developers' competence in general contrary to the first two that focused on specific areas. The software security competence of developers did not appear in any of the three studies, thus the need for this paper.

It is also worth mentioning that there are several efforts being made to improve security matters in the development of software. They include the development processes and the methods to reduce vulnerabilities and flaws in software. Hazeyama & Shimizu (2012) proposed a software security learning process using the traditional software development cycle. Cheng et al. (2008) reiterated for security engineering environment for software development since security requires continuous support. Thus, they make use of the lifecycle of software engineering for their solution which is based on International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) standards. The work of Verdon (2006) and McGraw (2004) examined the security policies and best practices that are essential for software developers.

The Open Web Application Security Project (OWASP) that is OWASP top 10 -2017 that focused on software developers and designers stated that "insecure software is undermining our financial, healthcare, defense, energy, and other critical infrastructure." The increasing complexity and the connectedness of software, is making it more difficult in attaining an increase in application security. Additionally, we face the rapid process of developing software which increases our common security risks. This makes it impossible to accept simple security problems as listed in the OWASP top 10 – 2017. The top five on the list are (i) Injection, (ii) Broken Authentication, (iii) Sensitive Data Exposure, (iv) XML External Entities (XXE), and (v) Broken Access Control. The rest of the OWASP top 10 – 2017 are (vi) Security Misconfiguration, (vii) Cross-Site Scripting (XSS), (viii) -Insecure Deserialization, (ix) Using Components with Known Vulnerabilities, and (x) Insufficient Logging & Monitoring (OWASP, 2017). Such security problems require corresponding skills to handle them. Given this, software developers' need to develop their security competences. For them to be able to develop and maintain such competences, it requires that such competences are identified and placed in the appropriate domain. Thus, the need for this study.

A survey to identify the guidance available on the web to help software developers' to fix security matters was conducted by Acar et al. (2017). They concluded that not all the information on the web is readily made for fixing security issues (Acar et al., 2017). Therefore, it may require security competences of the developers' to adjust the available code to meet the security demand. Hilburn & Mead (2013), developed a software security assurance model by providing capabilities. The capability of the assurance model was addressed by utilizing the knowledge areas. The main knowledge areas of assurance model that were identified were: assurance across lifecycles, risk management, assurance assessment, assur-

ance management, system security assurance, system functionality assurance and system operational assurance (Hilburn & Mead, 2013). Even though, this work focused on assurance in software security, it also provided some capabilities or knowledge areas that are useful for this paper. Work such as Meng et al. (2018); Miller and Heymann (2018) and Qian et al. (2018) provide some information on the security competences of software developers. Therefore, we employ these studies stated above and other existing studies to set the agenda for identifying the security competences of software developers and highlight the importance of software developers' security competences for further studies. Thus, this study seeks to employ traditional literature reviews to identify the security competences of software developers as the first step in broader research.

In presenting Common Body of Knowledge (CBK) for Information security professionals, Theoharidou & Gritzalis (2007) made a case for technical and behavioural skills for information security professionals. The framework was achieved using 135 academic intuitions from Africa, Asia, Australia, Europe, and South and North America to provide a skill set for information security professionals. The framework can be utilized in identifying and assessing the skills of information security professionals. The framework has three major areas: information communications technology skills area, security skills area and behavioural skills area. This study aimed at identifying the security competences of software developers from literature using traditional literature review and maps the result to the Common Body of Knowledge for information security professional skills framework (CBK). As a result, the CBK framework will be employed as a theoretical lens for this study.

METHODOLOGY

Primarily a literature review will be mainly employed in this study. Fink defines a research literature review as "a systematic, explicit and reproducible method for identifying, evaluating and synthesizing the existing body of completed and recorded work produced by researchers, scholars, and practitioners" (Fink, 2010, p. 3). In this section, an attempt is also made to distinguish between a traditional literature review and a systematic literature review. Systematic literature review is defined by Kitchenham and Charters as "a form of secondary study that uses a well-defined methodology to identify, analyse and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable" (Kitchenham & Charters, 2007, p. vi, pp. 8). A traditional literature review is used to demonstrate a gap or a problem in an area one seeks to research without an explicit method for reviewing the literature (Moustroufas et al., 2015). Since this is the first step towards broader research, a traditional literature review will be utilized.

Given this, the IEEE database was used as the database to find studies that investigated software security. The identified competences were grouped into two areas: programming related competences and non-programming related competences. The detail of the classification is explained in the result section. The identified competences were then mapped to technical and behavioural skills of information security professionals' skill set framework. With regard to data collection, data was collected from the IEEE database. The search strings that were utilized for the search were: software engineers/developers' skills, competence, and security knowledge. This was done without any strict protocol. Only peer-review papers were employed for the study. The names of the competences were extracted, descriptions of the competences were recorded into an excel sheet for the next stage of the research. On data analysis, competences with the same meaning were group together. Different implications of the competences

were analysed and recorded against the individual competences identified. Using conventional content analysis guideline of Hsieh & Shannon (2005), competences were classified into two areas. They are programming related competences and non-programming related competences. Lastly, the identified competences were mapped to the information security professional skills set framework.

RESULTS

The identified competences were categorized into two. They are programming related competences and non-programming related competences. Programming related competences are those that involve coding. Non-programming related competences are those that do not directly deal with coding. The competences were mapped to the common body of knowledge information security professional skills framework. Table 1 depicts the competence area, the competence name, the citation of the papers that the competences were extracted from and the CBK of information security professional's framework.

Table 1 shows the competences identified, their classifications, the literature from which the competence is extracted from and their relationship to CBK of information security professionals' framework. In all 13 competences were identified, nine competences were programming related and 4 competences were non-programming related. Seven of the competence maps to both information communication technology and security criterial and 6 maps to information communication technology. The next section provides the definition/descriptions of the competences and implications.

PROGRAMMING RELATED COMPETENCES

Secure Programming/Coding Skills

Description

The art of adopting a secure practice in the development of software. This includes the skill of being able to guide against vulnerabilities and flaws in software development. The majority of vulnerabilities and flaws in software appear when developers ignore secure practices in programming. More details of secure programming/coding competences can be found in the works of Mano et al. (2006); Miller & Heymann (2018) and Zainuddin & Normaziah (2011).

Implication

Without the adoption of secure coding, developers may create software with flaws and vulnerabilities. As pointed out by Sametinger (2013), vulnerabilities and flaws are the key entrants for attackers. Improving secure coding or programming will reduce security flaws. Secure coding must be part of a software development curriculum. There is a need to include fundamental security principles programming courses. Organizations must continue to introduce fresh courses on secure coding. In today's software development, secure coding must be started from the planning stage of the development to the end of the software development lifecycle. This implies that developers' competence in secure coding is essential.

Table 1. Security competences of software developers

Competence area	Competence name	Reference	CBK of information security professionals framework (Theoharidou & Gritzalis, 2007)
Programming related skills	Secure programming or coding skills	(Acar et al., 2017; Mano et al., 2006; Miller & Heymann, 2018; Qian, Lo, et al., 2018; Zainuddin & Normaziah, 2011)	Information communications technology/ security
	Secure mobile software development skills	(Meng et al., 2018; Qian, Parizi, & Lo, 2018)	Information communications technology/ security
	Secure socket layer/transport layer security (SSL/TLS) skills	(Verdon, 2006)	Information communications technology/ security
	Web Application security development skills	(Qian, Lo, et al., 2018)	Information communications technology/ security
	Integrated development environment (IDE) security skill	(Meng et al., 2018)	Information communications technology
	Code Analysis tools skills	(Meng et al., 2018)	Information communications technology
	Modelling SQL injection skills	(Kaur & Kaur, 2016; Wegerer & Tjoa, 2016)	Information communications technology/ security
	Handling buffer overflow skills	(Park et al., 2010)	Information communications technology/ security
	Security patterns skills	(Hazeyama & Shimizu, 2012)	Information communications technology/ security
Non-Programming related skills	Software security policy skills	(Verdon, 2006)	Information communications technology
	Software security best practice and standard skills	(McGraw, 2004)(Hazeyama & Shimizu, 2012)(Cheng et al., 2008)	Information communications technology
	System Security assurance skills	(Hilburn & Mead, 2013)(Miller & Heymann, 2018)	Information communications technology
	Vulnerability assessment tool skills	(Miller & Heymann, 2018)	Information communications technology

As suggested by Mano et al. (2006), secured programming must be taught in the early part of a software program. It must also be recognized as important skill for software developers.

Secure Mobile Software Development Skills

Description

Mobile devices may have software applications that we utilize frequently or perhaps even daily. The process of developing apps for these devices differ from the main devices. Furthermore, the database and the storage for these devices also differ. Thus, requiring different programming and security competences for the development of mobile apps. More about secure mobile software development skills can be found in the works of Meng et al. (2018); Qian, Lo, et al. (2018); Qian, Parizi, et al. (2018).

Implication

Most of the developers of these apps lack the necessary skill for developing mobile apps, thereby creating vulnerabilities for attackers to exploit those devices. The common nature (maybe you could be more specific here?) of the devices makes them more vulnerable. Thus, delays in providing bug fixings for new versions of applications can provide a door for attackers. Un-updated operating systems (OS) on mobile devices can allow attackers to exploit the vulnerabilities on the OS to attack the software application. Developers must pay attention to secure mobile development skills since techniques used for developing mobiles are different from that of normal devices. Fundamentally the increased usage of mobile technology is putting pressure on mobile developers. Both the trainers and users of the security competence of developers must adopt modern techniques to upgrade the developers to withstand the modern attackers.

Secure Socket Layer Skills

Description

Communication – data transmission between devices - is important in the applications function. This requires developers' skills in standard cryptographic protocol and technology for communicating on the internet. More importantly the use of transport layer security (TLS). Developers need to have skills in socket programming to enable them to develop this type of communication. More details of secure socket layer skills can be found in the work of Verdon (2006)

Implications

Most attackers take advantage of eavesdropping on transmission and launch their attack. This happens when strong encryptions are not used. Developers are to have skills in SSL or TLS encryptions technology. This is because most devices use the internet as a means to transmit data. Without such skills will mean that most attackers can eavesdrop on the communication and launch attacks. Developers should understand and have skills in symmetric encryption.

Web Application Security Skills

Description

Skills to protect devices or applications against web attacks such as cross-site scripting, SQL injection, denial-of-service, etc. Most attackers use vulnerabilities of web applications to attack. It is important to know that web application security directly relates to websites, web applications and web services such as APIs. Again, one needs to distinguish between network security and web application security. Therefore, the competences may defer. More details of secure socket layer skills can be found in the works Anand & Ryoo (2017); Uskov (2013) and Uskov & Avenue (2013).

Implication

In today's world, most of our business is done using the internet. Thus, not having the skills of developing software that can reduce web vulnerability will mean that most businesses could face catastrophes in their dealings. There is the need to have developers who understand using up-to-date skills in proper authentication methods, encryptions and development of patching for discovered vulnerabilities.

Integrated Development Environment (IDE) Security Skills

Description

Most developers of software make use of IDE for the development of software. They are software applications that provide the environments for software development. Thus, they are attitude, skills, and knowledge for using IDE securities in developing software. More details of IDE security skills can be found in the work of Meng et al. (2018).

Implication

Such environments sometimes if not well protected, can leave vulnerabilities in the software being developed and can be exploited by attackers. Having the skills related to the security of the use of the said IDE provides the developer with an environment free of vulnerabilities and flaws. Security updates are important and other security in the transmission of data. Developers must understand such security environments and use them appropriately to avoid leaving vulnerabilities that can be taken advantage of attackers.

Code Analysis Tools Skills

Description

Code analysis tools are used during coding to aid in analysing the code of the developer. Such tools help in identifying bugs and guide the developer to fix them before deploying the applications. They are attitude, skills, and knowledge for performing code analytics in software development. More details of code analysis tools skills can be found in the work of Meng et al. (2018)

Implication

If developers do not have the skill of using code analysis tools it may mean that time to identify bugs during coding may be long. It can result in leaving bugs to be exploited by attackers. It is also important to note that most of these bugs are difficult to be identified by the human eye. Examples of such tools are PMD java and SonarQube.

Modelling SQL Injection Skills

Description

It is a code injection technique that attackers take advantage of data-driven applications using SQL statements. It mostly happens when user inputs are not well-typed. They are attitude, skills, and knowledge for developing software free of SQL injection. More detail of SQL injection skills can be found in the works of Kaur & Kaur (2016) and Wegerer & Tjoa (2016).

Implication

It allows attackers to use malicious SQL statements to attack. This can be used on websites and databases. This is done by using spoof identity to temper with existing data. Such attacks are known as vector. Without skills in SQL injection handling in web applications and applications using databases, it will give attackers the chance to attack just systems since such vulnerability is commonly committed by developers.

Handling Buffer Overflow Skills

Description

It happens when a program writing to the buffer, which is a memory area set aside to hold data overflow. Mostly, when malformed inputs are used. they are attitude, skills, and knowledge needed to avoid buffer overflows. More details of handling buffer overflow skills can be found in the work of Park et al. (2010).

Implication

This happens when programmers or developers assume that all inputs may be smaller, but this may not always be the case. In case there is an overflow, the system may write beyond the allocated size causing erratic in execution leading to access error or crashing of the system. There is the need to write code that has built-in protections in the programming codes. The possession of such skills may reduce buffer overflows in memory, since not all input size can be predicted well by the developer.

Security Patterns Skills

Description

Security patterns are applied during software development by developers to achieve security goals. Such security patterns are pre-defined to guide developers. Having such skills will enable developers to know what security pattern can be used to achieve a particular security goal. That is the protected system patterns for confidentiality and integrity of information and error detection/correction pattern for deducing errors for corrections. More detail of security patterns skills can be found in the work of Hazeyama & Shimizu (2012).

Implication

Without such patterns, developers are to start from scratch to develop such protections. Understanding or having such skills, they can also develop security patterns to meet a specific goal that is not available.

NON-PROGRAMMING RELATED SKILLS

Software Security Policy Skills

Description

A software security policy defines the specific rules of security that software to be developed must have. That means that developers must frequently reference to make sure that the software obeys such policy. Understanding software security policy as a skill will enable the developer to develop software that will meet the security policy of the organization, the state and the world in general. Thus, they are attitude, skills, and knowledge needed to develop software to meet software security policies of the organization, the state, and the international community. More details of software security policy skills can be found in the work of Verdon (2006).

Implication

If developers do not have the skill to understand security policies and cannot develop software to meet what the organization, the state, and the international community have set as their policy for software security, consumers may not trust those software products. Furthermore, software security policies are standards, established to help reduce security threats. This means that, without them, developers may develop software according to their skills. This can lead to a lower security standard for the software they develop.

Security Best Practice and Standard Skills

Description

Best practice and standard are what has been used, tested and agreed as the best way of handling security in software development. Security best practices and standards can guide developers in secure software development. Thus, they are the attitude, skills, and knowledge needed to develop software security best practices and standards. More details of software security policy skills can be found in the works Cheng et al. (2008); Hazeyama & Shimizu (2012) and McGraw (2004).

Implication

If developers do not have such skills, it will mean they may not follow the best way of developing secure software. Mostly, security best practices and standards serve as a guide, but also provide a means to develop to meet certain accepted way that leads to trust.

This will mean that software developed by such developers with security best practices and standards skills will develop secured software, thereby, reducing the vulnerabilities that an attacker can exploit.

System Security Assurance Tools Skills

Description

These are tools that help developers of software from protecting the data and resources controlled by the software. They are the first line in for defending the attackers and also assessing the software security. Thus, they are the attitudes, skills, and knowledge needed to use system security assurance tools when developing software. More details of system security assurance tools skills can be found in the works of Hilburn & Mead (2013) and Miller & Heymann (2018).

Implication

Mostly, the human resources of the developer alone may not be enough for handling the development of software. Therefore, tools are needed to support the development of secured software. System security assurance tools support developers in such a situation. Not having the skill of using such tools will require more human hand in the development process. Alternatively, they will develop software that does not provide the required assurance for the people.

Vulnerability Assessment Tool Skills

Description

Tools are needed to identify the threats and risks that may be in software during development. In using such tools developers will need some special skills. Thus, they are attitude, skills, and knowledge needed by developers to use vulnerability assessment tools during software development. More detail of vulnerability assessment tool skills can be found in the work of (Miller & Heymann, 2018)

Implication

Without such tools, the human factor is to be used for such identification of vulnerability and threats thus, making such skills important for developers. It is important to note that most of such vulnerabilities are difficult to be identified by the human eye, thus if developers have no skills in using these tools, it may mean suck vulnerabilities and threats may be left in the software for attackers to exploit.

DISCUSSIONS

As stated in the related works, there were three review papers on software developers' competences. Two made use of a systematic review and one used a traditional review. None of these reviews mentioned the security competences of software developers. Nevertheless, there are some similarities. The work of Moustroufas et al. (2015) also used a traditional review, which was the same method used by this paper.

The difference between this paper and Moustroufas et al. (2015) is that they looked at software developers competence in general, whereas this paper looked at is security competence of the developers which is a specific area in software developers' competence. On the other hand, the other two reviews also looked at specific areas of developers' competence similar to this paper but used a systematic literature review as a method. This paper agrees with these authors that competences of software developers are essential for software development and effort must be made to maintain them especially in academia.

In proposing a security engineering environment for software developers, Cheng et al. (2008) claimed that the tools and the developers must integrate for a secure engineering environment. We support their assertion, but their work falls short of the implication of not having such an environment. To add to their work, this paper has provided the security competences of the developers which are essential for the security engineering environment they proposed. Furthermore, this paper has responded to the call by Hazeyama and Shimizu (2012) and Hilburn and Mead (2013), that there is the need to pay attention to security competences of the developers'. This paper has provides some of the competences, therefore agreeing with Hazeyama and Shimizu (2012) and Hilburn and Mead (2013) that the security competences of the developers are an essential parts of software developers' competences. For that reason, we support their call for more research on security competences of software developers'.

Researchers such as Cheng et al. (2008); Hilburn & Mead (2013) and Riehle & Nürnberg (2015) have called for security competence development through the lifecycle of developers. We concede, we could not do that, but we have identified some security competences of the developer that can be used as a starting stage for security competences of the developers' studies. Acar et al. (2017)stated that not all web security resources can be used fully to solve security problems by developers. Therefore, with the identification of the security competences of software developers, industry players can add to such work (web resources) by using the competences they have. Thus, this chapter supports the work of Hilburn & Mead (2013) that, knowing those security competences of software developers will help the users, possessors, and educators. Meng et al. (2018); Miller and Heymann (2018) and Qian et al. (2018) provided individual security competences of software developers, though this paper could not provide a full list, the paper has provided the basis for more work to be done. Theoharidou & Gritzalis (2007) work identified the technical and behavioural competences of information security professionals. This assertion has been established in the literature. We did not identify any behavioural security competences of software developers. Nevertheless, we hold the belief that there are behavioural security competences of developers and that empirical work must be conducted to identify them.

CONCLUSION

This chapter proposes a security competence for software developers. It uses a literature review to identify and classify security competence of software developers. Thirteen security competences of software developers were identified. They were classified as programming related competence and non-programming related competence. The author agrees that the methodology used has some limitations. Nevertheless, the competence identified and the linkage provided between the security competence of software developers and the information security professional framework will serve as a base for the development of the security competence of software developers. Furthermore, this chapter also makes a

call for empirical research to identify the security competence of software developers. By that, the author calls for a systematic literature review on the security competence of software developers. Again, there is the need also to identify those security competences using the lifecycle of the software development process.

REFERENCES

Acar, Y., Stransky, C., Wermke, D., Weir, C., Mazurek, M. L., & Fahl, S. (2017). *Developers Need Support, Too: A Survey of Security Advice for Software Developers. In 2017 IEEE Cybersecurity Development IEEE Secure Development Conference Developers* (pp. 22–26)., doi:10.1109/SecDev.2017.17

Anand, P., & Ryoo, J. (2017). Security Patterns As Architectural Solution - Mitigating Cross-Site Scripting Attacks in Web Applications. In *2017 International Conference on Software Security and Assurance (ICSSA)* (pp. 25–31). IEEE. 10.1109/ICSSA.2017.30

Cheng, J., Goto, Y., Morimoto, S., & Horie, D. (2008). A Security Engineering Environment Based on ISO / IEC Standards: Providing Standard, Formal, and Consistent Supports for Design, Development, Operation, and Maintenance of Secure Information Systems. In *2008 International Conference on Information Security and Assurance* (pp. 350–354). 10.1109/ISA.2008.106

Cruz, S., Fabio, Q. B., & Fernando, L. (2015). Forty years of research on personality in software engineering: A mapping study. *Computers in Human Behavior*, *46*, 94–113. doi:10.1016/j.chb.2014.12.008

Fink, A. (2010). Conducting Research Literature Reviews: From the Internet to Paper (3rd ed.). SAGE.

Hazeyama, A., & Shimizu, H. (2012). Development of a Software Security Learning Environment. In *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (pp. 518–523). IEEE. 10.1109/SNPD.2012.65

Hilburn, T. B., & Mead, N. R. (2013). Building Security In. *IEEE Security and Privacy*, *11*(October), 89–92. doi:10.1109/MSP.2013.109

Hsieh, H.-F., & Shannon, S. E. (2005). Three Approaches to Qualitative Content Analysis. *Qualitative Health Research*, *15*(9), 1277–1288. doi:10.1177/1049732305276687 PMID:16204405

IEEE. (2014). *Software Engineering Competency Model (SWECOM)*. IEEE. Retrieved from http://www.dahlan.web.id/files/ebooks/SWECOM.pdf

Kaur, N., & Kaur, P. (2016). Modeling a SQL Injection Attack. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 77–82). Bharati Vidyapeeth.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3. Engineering (Vol. 45). doi:10.1145/1134285.1134500

Lenberg, P., Feldt, R., & Wallgren, L. G. (2015). Behavioral software engineering: A definition and systematic literature review. *Journal of Systems and Software*, *107*, 15–37. doi:10.1016/j.jss.2015.04.084

Mano, C. D., Duhadway, L., & Striegel, A. (2006). A Case for Instilling Security as a Core Programming Skill. In *Proceedings. Frontiers in Education. 36th Annual Conference* (pp. 13–18). IEEE. 10.1109/FIE.2006.322347

McGraw, G. (2004). *Software Security*. IEEE Security & Privacy. doi:10.1109/MSECP.2004.1281254

Meng, X., Qian, K., Lo, D., & Wu, F. (2018). Secure Mobile Software Development with Vulnerability Detectors in Static Code Analysis. *2018 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–4. 10.1109/ISNCC.2018.8531071

Miller, B. P., & Heymann, E. (2018). *Tutorial: Secure Coding Practices, Automated Assessment Tools and the SWAMP. In 2018 IEEE Cybersecurity Development (SecDev)* (pp. 124–125). IEEE; doi:10.1109/SecDev.2018.00025

Moustroufas, E., Stamelos, I., & Angelis, L. (2015). Competency profiling for software engineers: Literature review and a new model. In *Proceedings of the 19th Panhellenic Conference on Informatics* (pp. 235–240). Athens, Greece: ACM. 10.1145/2801948.2801960

OWASP. (2017). *OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks*. OWASP.

Park, C. S., Lee, J. H., Seo, S. C., & Kim, B. K. (2010). Assuring software security against buffer overflow attacks in embedded software development life cycle. In *2010 The 12th International Conference on Advanced Communication Technology (ICACT)* (Vol. 1, pp. 787–790). IEEE.

Qian, K., Lo, D., Parizi, R., & Wu, F. (2018). Authentic Learning Secure Software Development (SSD) in Computing Education. *2018 IEEE Frontiers in Education Conference (FIE)*, 1–9.

Qian, K., Parizi, R. M., & Lo, D. (2018). OWASP Risk Analysis Driven Security Requirements Specification for Secure Android Mobile Software Development. In *2018 IEEE Conference on Dependable and Secure Computing (DSC)* (pp. 1–2). IEEE. 10.1109/DESEC.2018.8625114

Riehle, D., & Nürnberg, F.-A.-U. E. (2015). How Open Source Is Changing the Software Developer's Career. *Computer Practice*, *48*(5), 51–57. doi:10.1109/MC.2015.132

Sametinger, J. (2013). Software Security. In *2013 20th IEEE International Conference and Workshops on Engineering of Computer Based Systems (ECBS)* (p. 216). IEEE. 10.1109/ECBS.2013.24

Theoharidou, M., & Gritzalis, D. (2007). Common Body of Knowledge for Information Security. *IEEE Security & Privacy*, 64–67.

Uskov, A. V. (2013). Software and Web Application Security: State-of-the-Art courseware and Learning Paradigm. In *IEEE Global Engineering Education Conference (EDUCON)* (Vol. 0, pp. 608–611). 10.1109/EduCon.2013.6530168

Uskov, A. V., & Avenue, W. B. (2013). Hands-On Teaching of Software and Web Applications Security. 2013 3rd Interdisciplinary Engineering Design Education Conference, 71–78. 10.1109/IEDEC.2013.6526763

Verdon, D. (2006). *Security Policies and the Software Developer*. IEEE Security & Privacy. doi:10.1109/MSP.2006.103

Vishnubhotla, S. D., Mendes, E., & Lundberg, L. (2018). An Insight into the Capabilities of Professionals and Teams in Agile Software Development A Systematic Literature Review. In *ICSCA 2018* (pp. 10–19). Kuantan, Malaysia: ACM. doi:10.1145/3185089.3185096

Wegerer, M., & Tjoa, S. (2016). Defeating the Database Adversary Using Deception - A MySQL Database Honeypot. In *2016 International Conference on Software Security and Assurance (ICSSA)* (pp. 6–10). IEEE. 10.1109/ICSSA.2016.8

Zainuddin, H. N., & Normaziah, A. A. (2011). Secure Coding in Software Development. In *2011 Malaysian Conference in Software Engineering* (pp. 458–464). IEEE. 10.1109/MySEC.2011.6140716

KEY TERMS AND DEFINITIONS

Competence: A set of knowledge, skills, and attitudes for performing a task.

Non-Programming-Related Competences: Software security skills that do not directly deal with coding. For example, software security policy skills and system security assurance tools skills.

Programming Related Competences: Software security skills needed for coding. For example, secure programming/coding skills and secure mobile software development skills.

Security Competence of Developers: A set of specific security competencies required by a developer to deal with security during the whole software development process; For example, SQL injection skills, and security pattern skills.

Software Developer: Individuals who employ software development skills to design, construct, test, and maintain computer software.

Software Engineering Competence: A set of knowledge, skills, and attitudes of software developers to fulfill a given task in a software development project.

Software Security: An art of providing protection to software against hackers and attackers during the life cycle of the software.

Traditional Literature Review: A method used to demonstrate a gap or a problem in an area one seeks to research without an explicit method for reviewing the literature.