Towards a Context-Based Approach for Software Security Learning

Shao-Fang Wen and Basel Katt

Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Gjøvik, Norway

{shao-fang.wen, basel.katt}ntnu.no

Towards a Context-Based Approach for Software Security Learning

Learning software security is one of the most challenging tasks in the information technology sector due to the vast amount of security knowledge and the difficulties in understanding its practical applications. Conventional teaching approaches give little attention to how to improve the effectiveness of learning in the domain of software security. Context-based learning has been proven to be a sound pedagogical methodology; however, it is still unclear how to synthesize the prescription in the domain of software security. In this paper, a context-based approach to software security learning is proposed for structuring and presenting security knowledge. To evaluate the proposed approach, a quasi-experiment was designed and executed in the setting of a university learning environment. The experiment results indicate that the proposed context-based learning approach not only yields significant knowledge gains compared to the conventional approach. but also gains better learning satisfaction of students

Keywords: Software security, context, context-based learning, knowledge management

Introduction

Information technology is one of the world's fastest growing industries. In fact, the rate at which software and software products are evolving is many times greater than the rate at which software security is evolving. According to CVE[1] vulnerability data (CVE), the number of software vulnerabilities disclosed in 2017 grew by 128% compared to the number in 2016, reaching an all-time high of 14,714. In an age of cybercrime, some of the most widespread software-based crimes include stealing information via hacking, carrying out virus attacks to take down computer systems and implanting spyware with the intent to watch a person or his or her computer activities. Due to the increasing importance and complexity of computer systems, insufficient knowledge and skills

[1] Common Vulnerabilities and Exposures (CVE) is a dictionary-type list of standardized names for vulnerabilities and other information related to security exposures. https://cve.mitre.org/

related to software security will result in more serious breaches in the future.

Software security knowledge is multifaceted and can be applied in diverse ways (McGraw, 2006). Learning software security is a complex and difficult task because learners must not only deal with a vast amount of knowledge about a variety of concepts and methods but also have to demonstrate the applicability of the knowledge through experience in order to understand their practical use. Conventional security learning materials are usually subject-oriented, which is useful for rote memorization of a specific subject or for information recall later. However, such an approach makes it difficult for learners to understand the rationale of the topics and correlate those topics with real software cases. Learners often feel that security knowledge is so extensive and software security is so difficult to achieve that they simply cast it aside (Apvrille & Pourzandi, 2005).

In traditional software security teaching, little attention is given to what the security knowledge really means to learners, and there is not much content addressing the connections between real-world situations and security concepts. According to Jonassen and Land (2012), "...learners must be introduced to the context of the problem and its relevance, and this must be done in a way which challenges and engages them. Context, and the particulars of that context, can provide a powerful motivation for learning" (Cooper & Cunningham, 2010, Perin, 2011). If learners do not learn the knowledge well in the first place, they cannot possibly transfer it to new situations (Council, 2000). We argue that, in order to regulate learning about software security effectively, security knowledge should be contextualized and embedded in a meaningful scenario that makes sense to the students to enhance their understanding and make the concepts more relatable.

The concept of the learning context has been widely addressed in education and psychology literature over the years, and the effectiveness of context-based learning has been demonstrated in the setting of interactive school classrooms. However, it is still unclear how this concept can be synthesized and applied in the domain of software security. To mitigate this research gap, we proposed a context-based approach to structure security knowledge and facilitate software security learning in a way that can motivate learners. We conducted experiments to evaluate the effectiveness of this approach in the setting of a university learning environment. This paper presents the rationale of the proposed approach and the findings of our experimental studies.

Conventional Security Learning Materials

In conventional security learning materials, the knowledge content is commonly organized topically, focusing on security aspects. One approach may first introduce attack patterns or security vulnerabilities (the black-hat side), such as Cross-Site Scripting (XSS) and SQL injection (SQLi), while another might start with secure design practices or coding standards (the white-hat side), such as input validation and output encoding. The security-centric materials are often written in the form of a reference manual or a guide to a particular security certification. Learners usually finish reading such materials with little understanding of the context in which the security knowledge should be applied. This relates to what is known as the knowing-doing gap; that is, knowing better but not doing better.

On the other hand, security learning materials usually emphasize concepts first rather than facts or context to transmit knowledge. Consequently, learners may struggle to finish reading them due to a learning style mismatch. Several studies (McCaulley, 1976, McCaulley et al., 1983, Felder & Silverman, 1988) have shown that the majority of engineering students are sensor-type learners, who like facts, data, and observable

phenomena as opposed to theoretical abstractions. Since many security tasks require awareness of one's surroundings, attentiveness to detail, experimental thoroughness, and practicality, the learning material presented must provide meaning and motivation for learners, allowing them to learn security principles and processes through a real-world situation that is of particular interest to them.

General Concepts of Context-Based Knowledge for Learning

According to Oxford Dictionaries[2], context is defined as "The circumstances that form the setting for an event, statement, or idea, and in terms of which it can be fully understood." Meanwhile, Dey (2001) defined context as "a set of information used to characterize the situation of an entity." Nonaka and Konno (1998) noted that knowledge reflects a particular stance, perspective, or intention in accordance with the characteristics of a specific context, which is different from information. Knowledge comes from a variety of contexts, and it cannot be accurately understood without context (Klemke, 2000, Brézillon, 2002). Without proper contextual information, knowledge can be isolated from other relevant knowledge, resulting in limited or distorted understanding (Brézillon & Pomerol, 1999, Goldkuhl & Braf, 2001). Since context can provide guidance regarding when, where, and why a piece of knowledge is used, it is crucial to consider the context to enhance the applicability of the knowledge.

Context can increase the information content of natural language utterances and facilitate learning (Brézillon, 1996, Brézillon, 2003). Psychology and education researchers have demonstrated that when knowledge is learned in a context similar to that in which the skills will actually be needed, the application of the learning to the new context may be more likely (Dey, 2001, Dolmans et al., 2005, Perin, 2011). Predmore

[2] https://en.oxforddictionaries.com/definition/context

(2005) showed that learning about knowledge content through real-world experience is important because "once [students] can see the real-world relevance of what they're learning, they become interested and motivated." The book *How People Learn* (Council, 2000) also pointed out that motivation is critical for learning, enabling knowledge transfer to occur. If students do not learn the material well in the first place, they cannot possibly transfer it to new situations. As stated the book "Learners of all ages are more motivated when they can see the usefulness of what they are learning and when they can use that information to do something that has an impact on others" (Council, 2000) (page 49).

Bennett et al. (2007) offered a definition of a context-based approach to science education: "Context-based approaches are approaches adopted in science teaching where contexts and applications of science are used as the starting point for the development of scientific ideas." The authors reported that context-based science courses motivate students and help them become more positive about science by representing real-world situations of the learning subject. When students are more interested and motivated by the experiences they are having in their lessons, their increased engagement may result in improved learning (Bennett et al., 2007). In computer science education, there is also a broad agreement that teaching units should start from a "real-world" context or phenomenon, aiming to create connections to prior knowledge, increase the relevance of the material to students, or show applications of the intended knowledge, thereby increasing motivation (Guzdial, 2006, Cooper & Cunningham, 2010, Guzdial, 2010, Diethelm et al., 2012). These contrast with more traditional approaches that cover abstract ideas first, before looking at practical applications.

Likewise, in software engineering, studying in one context and then abstracting the knowledge gained for use in a new context is a common way of learning programming that has been observed extensively in both new and experienced programmers (Apvrille

& Pourzandi, 2005, Ko & Myers, 2008). In order to capture and use security knowledge appropriately, it is necessary to first specify which context information is to be handled. Then, it must be represented in a format that is understandable and acceptable to the individuals. Thus, a context for a software security topic includes the circumstances in which its technical content exists. Therefore, when talking about software security in a given context, the knowledge would not only include the basic principles and processes of software security but also consider how security knowledge is used in one or more particular domains or application areas.

The Proposed Context-Based Approach

To facilitate contextual learning about software security, we proposed a context-based approach to structuring and presenting software security knowledge using three strategies: (1) Using a meaningful application scenario; (2) Simulating learners' mental models for security learning, and (3) Moving from concrete to abstract security knowledge. Figure 1 shows the conceptual view of the proposed context-based learning approach with three strategies.

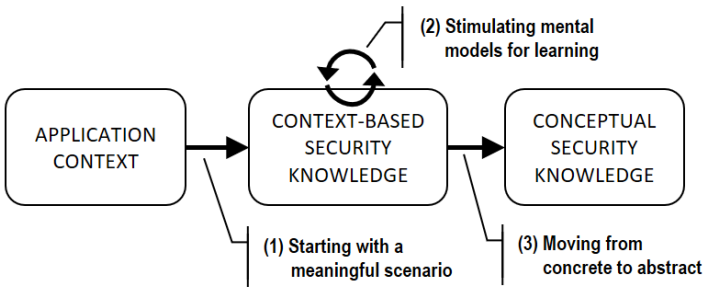


Figure 1. A conceptual representation of the proposed learning approach for software security

Starting with a Meaningful Scenario

Contextualized learning often takes the form of real-world examples or problems that are meaningful to the learners personally (Rivet & Krajcik, 2008). To begin the process of

learning, a meaningful situation for learners must first be established. In our approach, we set the application context as the starting point for learning security concepts on a need-to-know basis. Figure 2 presents the main components of the application context, which include application paradigms, application functionalities, and application scenarios. The application paradigm is a combination of security-independent data that characterize software applications; for example, the domain area that the application belongs to or the technologies that the application uses. The software functionality represents any aspect of software applications that can perform for users or other systems in a particular paradigm, such as dynamically generating HTML in web applications and *cleartext* transmission of sensitive information in network applications. Under a given application paradigm and functionality, a series of scenarios are identified, each of which deals with one specific scenario in the context.

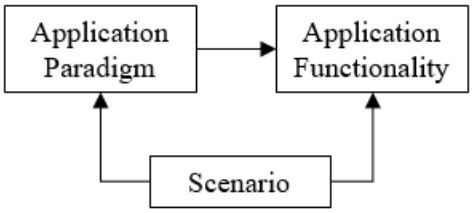


Figure 2. Components of the application context

A scenario is made up of practical demonstrations of the pre-described application functionality and the code fragments behind it that bridge the corresponding security knowledge. In this manner, a scenario constitutes a form of an anchoring event (Cognition Technology Group at Vanderbilt, 1992), which provides an experiential practice in software development from which learners can relate to new information about security. Research has shown that using anchoring events in learning promotes memory recall and the subsequent transfer of information to a new setting (Cognition Technology Group at Vanderbilt, 1992), which helps to render abstract ideas more concretely and thus provides a cognitive mooring around which newly learned ideas can be linked with learners' prior

understandings (Sherwood et al., 1987, Cognition Technology Group at Vanderbilt, 1992). When learners see applications and software function with the code they are already familiar with, (i.e., the anchor event), the consequence of exploiting vulnerabilities hits close to them and becomes more real, further motivating them to learn.

Stimulating Mental Models for Learning

In order to help learners create a strong and lasting bond that makes navigating the security knowledge efficient, we developed a knowledge structure to guide them in approaching personal mental models in the software security domain. Mental models combine a schema or a knowledge structure with a process for manipulating the information in the memory (Merrill, 2000), while knowledge structure interrelates a collection of facts or concepts about a particular topic. Craik (1967) suggested that the human mind builds and constructs "small-scale models" to anticipate events. Such mental models allow learners to gain insight regarding their world by building a work scheme (Gentner & Stevens, 2014), which makes it easier for them to access the information needed to understand the knowledge domain, make predictions, and decide upon action to take (Rouse & Morris, 1986). This can result in successful learning by engaging students, fostering their concentration, and assisting them in organizing systemic information (Seel et al., 2000).

To design a security knowledge structure (schema) that is easier to store in the learners' memory, we simplified the schema and reduce the content load of the knowledge structure. We identified the critical security concepts that are most widely used throughout the security domain and concentrated learning approaches on them. Ultimately, three security concepts were incorporated into the knowledge structure: security attack, security weakness, and security practice. Table 1 provides the definitions

of the three security concepts. Generally, our intention was to guide learners in answering three questions while dealing with each scenario:

- What are the possible attacks?
- Why does it encounter attacks?
- How can these attacks be prevented?

Table 1. The definition of security concepts and the corresponding focus questions

Security concept	Definition
Security Attack	It represents actions taken against the software case with the intention of doing harm.
Security Weakness	It represents bugs, flaws, vulnerabilities, and other errors in the software case.
Security Practice	It represents methods or mechanisms to mitigate security weakness to prevent security attacks.

Figure 3 illustrates the relationships between the concepts embedded in the proposed knowledge structure in the domain of software security. The knowledge structure provides the basis for the development of mental models in learning software security knowledge. As learners answer the what–why–how questions for each scenario, the relationships between the security concepts are emerging in their midst, and thus, their mental model expands.

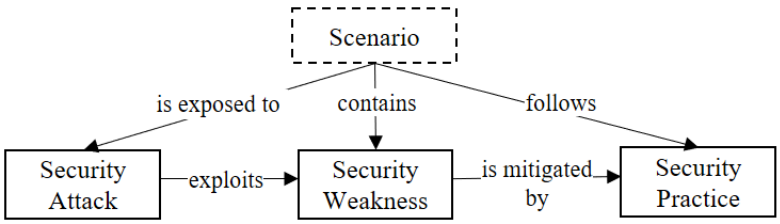


Figure 3. The relationship among security concepts of the knowledge structure

Moving from Concrete to Abstract Security Knowledge

Security Knowledge can be categorized as concrete or abstract facts, events, applications, conceptual descriptions, and principles. To help learners gain a more flexible understanding of the study concept in a range of situations with varying levels of abstraction, we organize security knowledge by blending abstract and concrete

perspectives; presenting it with a sequence from concrete to abstract. In our study, abstract knowledge refers to the conceptual security domain knowledge while the concrete knowledge relates to the contextualized scenario-specific security knowledge. Research has shown that presenting knowledge in both concrete and abstract terms are far more powerful than presenting either one in isolation (Pashler et al., 2007). Lave et al. (1991) also argued that abstract and generalized knowledge gains its power through the expert's ability to apply it in specific situations.

The used concrete-to-abstract approach in knowledge presentation differs from the traditional, where the concepts are of foremost importance and are usually explained first before concrete examples and applications are discussed. Figure 4 depicts the learning paths that are constructed by the proposed context-based approach. In such concrete-to-abstract knowledge presentation, learners discover meaningful relationships between practical functions and abstract knowledge in the context of real applications. The value of concrete representations has been frequently noted in education. Concrete materials can support abstract reasoning because they can be explicitly designed to promote true inferences from perceptual representations to abstract principles (Bassok, 1996). A method known as concreteness fading (Goldstone & Sakamoto, 2003) has the advantage of initially presenting concepts in a concrete fashion and then, over time, augmenting that initial presentation with progressively more abstract representations of the concepts. Abstract understanding is most effectively achieved through experience with perceptually rich, concrete representations (Goldstone & Son, 2005), while concrete materials make concepts real and therefore easily internalized (Kamina & Iyer, 2009). As long as the concrete knowledge and the underlying abstract explanation are understood by learners, learning transfers from one context to another will be more effective.

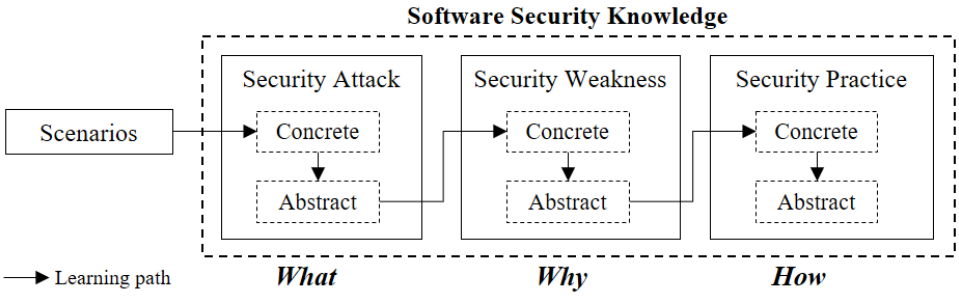


Figure 4. The constructed learning path based on the context-based approach

Study Method

To evaluate the proposed approach, a quasi-experiment with non-equivalent groups was designed and executed in the setting of a university learning environment. Our hypothesis in this study was:

Hypothesis: *The context-based approach to supporting students' software security learning yields better knowledge gain and learning satisfaction than the conventional learning approach.*

Two rounds of experiments with learning subjects related to *Web Security* were conducted with Bachelor students; each round lasted for about 70 minutes. According to the hypothesis, the variables in this experiment were defined as followings:

- Independent variables: The learning approaches (i.e., conventional vs. contextualized).
- Dependent variable: The security knowledge gain and learning satisfaction were measures providing insight into the effectiveness of the two approaches.

In this section, the sources of data, the tools used for data collection, the participants, and the experimental procedure are briefly outlined.

Participants

The participants were 42 Bachelor students from the fifth semester (third year), who were taking the "Software Security" course. The students were from two main study programs: Bachelor in IT Operations Information Security and Bachelor in Programming.

Treatments

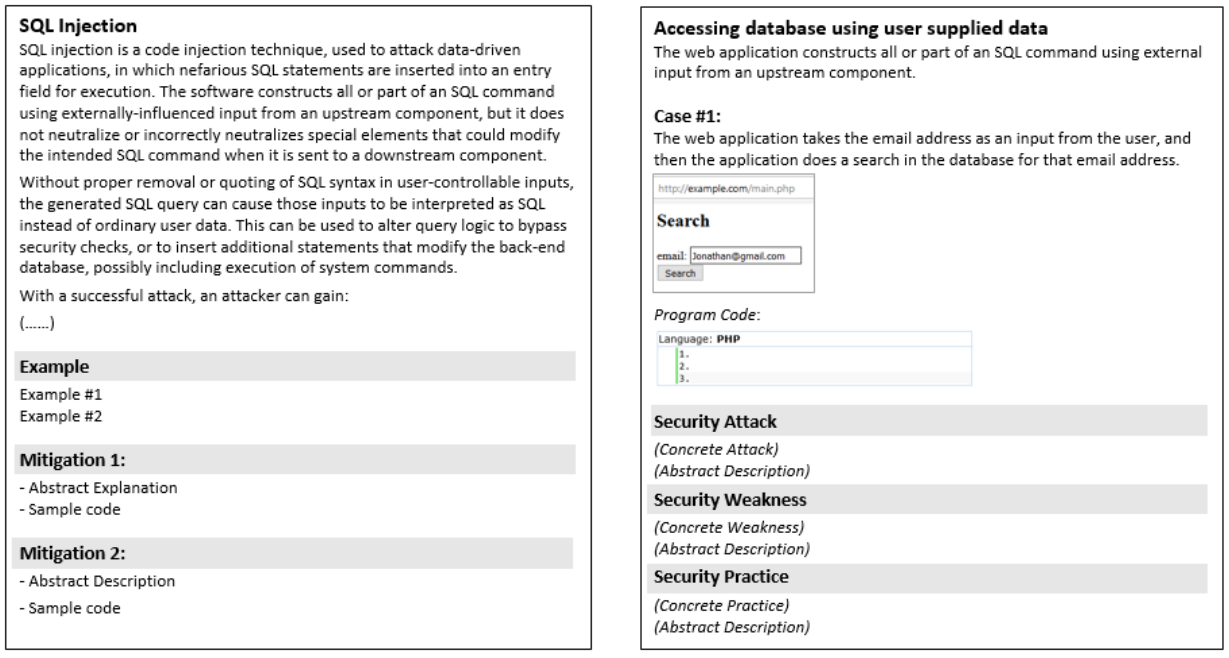
In this study, we designed two types of learning materials in a printed format as the experimental treatments, which were named type I and type II. The type I material used a conventional approach while type II adopted the proposed context-based approach to organizing software security knowledge. Regarding the learning subject, we used two common software vulnerabilities in web applications: SQLi and XSS. The materials were constructed using resources on the internet (e.g., OWASP[3] and CWE[4]) combined with the authors' teaching experience in the domain of software security. In the type I material, information was presented in the order of abstract to concrete. Conceptual knowledge about the vulnerability subject was described first, followed by examples with code fragments. Mitigations for the vulnerabilities were explained in the last section.

For the construction of the type II learning materials, we first set up the learning environment in a web application paradigm—an e-Store—using the LAMP[5] web service stack. For this specified context, the author developed a preliminary set of functionalities to operate a web-based e-Store application, including a login module, data input/output features, data processing, database access, and payment functions. Three critical application scenarios were created for each of the learning subjects within the scope of

[3] The Open Web Application Security Project (OWASP), an online community, produces freely available articles, methodologies, tools, and technologies in the field of web application security. https://www.owasp.org/
[4] Common Weakness Enumeration (CWE) is a universal online dictionary of weaknesses that have been found in computer software. https://cwe.mitre.org/
[5] LAMP is an open source web service stack that uses Linux as the operating system, Apache as the web server, MySQL as the relational database management system, and PHP as the object-oriented scripting language.

the e-Store functionalities. In the learning materials, functional features with the corresponding code fragments for each scenario were described and demonstrated in the beginning, followed by the security knowledge, which was organized based on the predefined knowledge structure (i.e., security attack, security weakness, and then security practice). Knowledge content for each security concept was presented in the order of concrete to abstract. All content demonstrating concrete knowledge was manipulated using the built application, including coding vulnerabilities, exploits, and code fixes.

Figure 5 shows the simplified view of the two types of learning materials in the subject of SQLi vulnerability. In terms of the type II material, three scenarios were introduced under an abstract functionality, "Accessing database using user-supplied data," which formed as the anchoring event for subsequently studying about relevant security knowledge.



(a) Type I material (b) Type II material

Figure 5. The simplified view of two learning materials for SQLi vulnerability

Data Collection

To collect data and measure the dependent variables, two types of instruments were used: pre- and post-tests and survey questionnaires. Pre and post-test sheets were developed to measure the learning gain (post-test/pre-test), in which items were created covering two types of security knowledge: theoretical and practical. The theoretical items focused on recalling and understanding conceptual security knowledge. The practical items required students to identify possible attacks in a given software context, mark coding errors in code fragments, and apply knowledge to different situations. The pre- and post-tests were similar except for the formulation of some questions, their order, and the answer options. Four test sheets (pre- and post-test for two rounds) were generated to assess the students' level of knowledge before and after the learning sessions. In each test sheet, there were 10 questions (6 theoretical and 4 practical), and the value for each question was five points.

We designed a survey questionnaire to collect students' perceptions of the two learning materials. Students were asked six questions for each type of learning material, which we used to measure the learning satisfaction factors, including interest creation, content fulfillment, learning efficiency, experience correlation, positive attitude, and personal satisfaction (Table 2). In this questionnaire, all respondents were required to choose the answer that reflected their own views and stance on the statements that were administered in accordance with a 5-point Likert scale, ranging from "strongly disagree" to "strongly agree."

Table 2. Questionnaire items for measuring learning satisfaction

Factor	**Question**
Interest Creation	I feel that the material is interesting when I get into it.
Content Fulfillment	The material provides knowledge content that fits my need precisely.
Learning Efficiency	The material helps me learn secure programming efficiently.
Experience Correlation	I could relate what I learned from the material to what I have already known or experienced before.
Positive Attitude	The material helps me foster a positive attitude towards learning about secure programming.
Personal Satisfaction	I find that at times studying the material gives me a feeling of personal satisfaction.

Experimental Procedure

The students were divided into two groups (group A and group B) after being seated in the classroom. They were first introduced to the main objectives of the experiment and informed of the procedure. Both rounds of experiments were performed with a similar experimental procedure. Table 3 shows the learning subjects arrangement and the dispatch rule of learning materials in each round/group. In the first round, students were given test sheets (pre- and post-test) and learning materials for the subject of SQLi. Students in group A studied type I learning material, while group B studied type II material. In round 2, the learning subject was changed to XSS, and we switched the type of learning material treated in the two groups. With the two-round experiment design, all students were able to experience both learning materials and thus the differences between the two. The major experiment steps in each round were as follows:

Step 1: Pre-test (15 minutes)

Step 2: Learning session (40 minutes)

Step 3: Post-test (15 minutes)

There was a 10-minute break between the two rounds. At the end of the second round, students completed the learning satisfaction questionnaire. This ended the experimental procedure.

Table 3. Learning materials dispatching rules

	Treatment	
	Round 1 (SQLi)	Round 2 (XSS)
Group A	Type I	Type II
Group B	Type II	Type I

Findings

In this section, we present the findings of the experiment, including an evaluation of the students' knowledge gain and learning satisfaction.

Knowledge Gain

The students' knowledge gain in the different types of materials was determined using a comparative means analysis. Table 4 presents the means analysis of the students' performance on the pre- and post-tests in each round of the experiment, including the mean scores and standard deviations. The results of the statistical analysis show that there was a positive knowledge gain (i.e., post-test to pre-test score) for both groups in both rounds. However, the group using type II materials had higher achievement levels than the group using type I materials, as shown in Figure 6.

Table 4. Comparative means analysis of students' performance on the pre- and post-tests

		Group A			Group B		
Round		N	Mean	SD	N	Mean	SD
1	Pre-test	20	26.75	5.20	22	24.32	5.19
	Post-test	20	29.50	6.90	22	33.86	4.86
2	Pre-test	20	21.75	8.78	22	20.00	9.26
	Post-test	20	26.25	6.90	22	30.91	8.54

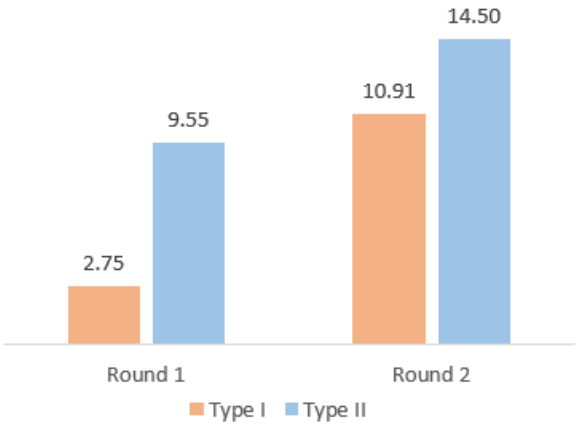


Figure 6. Knowledge gain for the two groups in each round of experiments

To determine whether there was a significant difference between the pre-test performances of group A and group B, an independent sample t-test was used. Table 5 shows the t-test analysis for the pre-test mean scores in the first round. The significance level (0.628) of Levine's test for equal variance was greater than 0.05, indicating "Equal variance assumed." Levine's test resulted in a "Sig. (2-tailed)" value of 0.137, which was above 0.05. Therefore, the null hypothesis of the independent sample t-test was rejected ($p > 0.05$), which implies that there were no significant differences between the two groups in terms of pre-test scores (i.e., the initial security knowledge) so that the significance of the knowledge gain can be concluded).

Table 5. Independent sample t-test results for pre-test scores in the first round

		Levine's Test		t-test				
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference
Pre-Test	Equal variances assumed	0.238	0.628	1.516	40	0.137	2.432	1.604
	Equal variances not assumed			1.516	39.601	0.138	2.432	1.605

Table 6 shows the independent sample t-test results in the first round for the post-test mean scores. Moreover, the difference between the post-test mean scores of the two groups is significant (2-tailed sig. = 0.02, $p < 0.05$). This indicates that our treatments resulted in a significant difference in security knowledge gain in the two groups of students.

Table 6. Independent sample t-test results for the post-test scores in the first round

		Levine's Test		t-test				
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference
Post-Test	Equal variances assumed	2.415	0.128	-2.413	40	0.020	-4.414	1.829
	Equal variances not assumed			-2.374	33.793	0.023	-4.414	1.859

We performed the same statistical analysis for the pre- and post-test scores in round 2 (Table 7). As can be seen in Table 7, there was also no significant difference in the pre-test scores in the two groups (2-tailed Sig. = 0.534, $p > 0.05$). The post-test 2-tailed Sig. was 0.032, thus achieving significant and indicating that the post-test score would also be affected by treatments in round 2.

Table 7. Independent sample t-test for pre- and post-test score in the second round

		Levene's Test		t-test				
		F	Sig.	T	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference
Pre-Test	Equal variances assumed	0.012	0.913	0.627	40	0.534	1.750	2.791
	Equal variances not assumed			0.629	39.921	0.533	1.750	2.784
Post-Test	Equal variances assumed	0.063	0.802	2.220	40	0.032	5.341	2.406
	Equal variances not assumed			2.243	39.431	0.031	5.341	2.381

Learning Satisfaction

The learning satisfaction for the two learning materials is represented as a radar chart with six axes (Figure 7). As depicted in the chart, the type II material had overall higher learning satisfaction mean scores than the type I materials in terms of the six satisfaction factors. Regarding the data series of the type II materials, the score of the six satisfaction factors were all above 4. Almost all of the responses regarding the type II were at least 3, and responses of 1 and 2 were rare. Of these, the mean scores of "Interest Creation" and "Experience Correlation" were the highest (4.33 and 4.29, respectively). In contrast, the scores of the two factors in the type I materials had the lowest mean scores (i.e., 2.81 and 2.83, respectively). The mean scores of the four other satisfaction factors evaluated for the type I materials were all approximately the same (3).

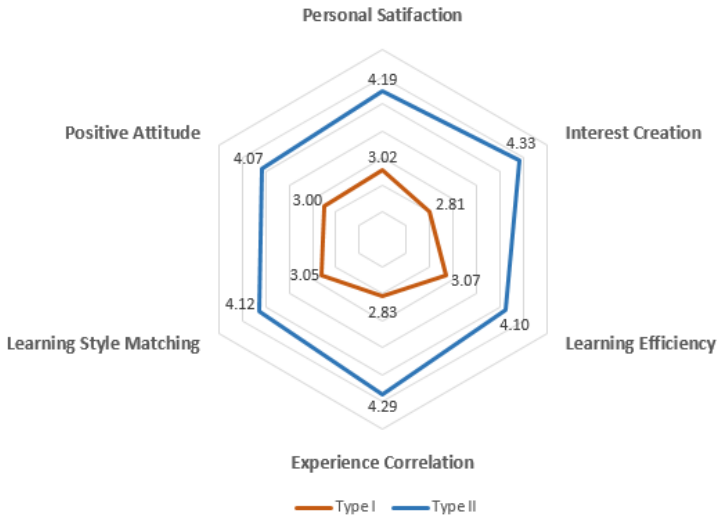


Figure 7. Radar diagram for learning satisfaction scores

Additional Findings

In this study, we were also interested in how the students performed with theoretical and practical questions when they were presented with the type II learning materials. According to Table 8, students performed better in the pre-test on theoretical questions than on practical ones in terms of hit rate (overall hit rate: 54.70% vs. 33.13%). After the type II materials were presented there was a knowledge gain in either the theoretical or practical questions. The average hit rates of both categories in the post-test reached the same level. In the first round, they fell to between 65% and 70%, while they were between 70% and 75% in the second round. Regarding the growth ratio of the mean scores from the pre-test to the post-test, it is clear that the students had better achievement with practical questions (110.29%) than with theoretical questions (28.74%).

Table 8. Comparative means analysis of students' performance on the pre- and post-tests

Round		Pre-test			Post-test			Growth Ratio
		N	Mean	Hit Rate	N	Mean	Hit Rate	
1	Theoretical	6	16.82	56.06%	6	20.00	66.67%	18.92%
	Practical	4	7.50	37.50%	4	13.86	69.32%	84.85%
2	Theoretical	6	16.00	53.33%	6	22.25	74.17%	39.06%
	Practical	4	5.75	28.75%	4	14.00	70.00%	143.48%
Sum	Theoretical	12	32.82	54.70%	12	42.25	70.42%	28.74%
	Practical	8	13.25	33.13%	8	27.86	69.66%	110.29%

Discussion

The objective of this study was to evaluate a context-based approach to improving learning about software security. A two-round pre-test/post-test experiment was used to measure the students' security knowledge gain, and a questionnaire was used to evaluate their learning satisfaction. The results of the pre-test/post-test experiment indicate an increase in the students' level of security knowledge for both the conventional and context-based approaches. According to the statistical t-test analysis, there was no significant difference between the two groups in terms of initial security knowledge; however, students using treatments with the context-based approach had significantly better knowledge gain than those using treatments with the conventional approach. The evaluation of the students' satisfaction with the two learning approaches supports our hypothesis, as the respondents showed higher learning satisfaction with the context-based knowledge approach than with conventional approaches.

As highlighted by the learning satisfaction analysis, a majority of students using conventional materials were unable to make connections between what they were learning about security and what they had been doing in programming. We argue that the way they process information and their motivation for learning is not supported by the conventional methods. Research has indicated that learning is most efficient when it is linked with the experience and prior knowledge that students bring to a given learning situation (Council, 2000, Leach et al., 2003); however, novice learners do not always make connections between new information and prior knowledge or everyday experiences in ways that are productive for learning (Land, 2000). In the context of software security learning, learners interpret the security knowledge they gain through a range of strongly held personal programming experiences. They often do not associate vulnerabilities with programs similar to what they were writing previously. Therefore,

establishing the relevance of learning materials before going into the details could provide a concrete foundation for the learning process.

Our approach attempts to place security learning in the context of real application scenarios, which serve as anchoring events and elicit the learners' memories and draw attention to software events and conditions. The results of our experiment show that this type of design keeps learners interested, motivated, and engaged in the learning experience. Since the given context is connected and relevant to their prior knowledge and life experiences in software development, security learning can then be related to a similar programming topic that they want to learn about or a problem to be solved. According to the results of the learning satisfaction survey, most students were very interested in studying type II materials and agreed that the materials could be correlated with their experiences. We believe this implies a direct effect on higher overall learning satisfaction, which motivates students to learn. The benefits of the contextualized approach can also be explained by the effective mechanism of intrinsic motivation, where a learner is drawn to engage in a task because it is perceived as interesting, enjoyable, and/or useful (Cordova & Lepper, 1996, Kozeracki, 2005, Dean & Dagostino, 2007).

In this study, we investigated how the contextualized approach affects students' learning performance in terms of answering theoretical and practical questions. The results show that type II materials can effectively support both abstract and concrete learning, and moreover, they provide a greater influence in terms of dealing with practical problems. Hence, a blend of concrete and abstract knowledge presentation can help learners gain a more flexible understanding of the study concept in a range of situations with varying levels of abstraction. Research has shown that presenting knowledge in both concrete and abstract terms are far more powerful than presenting either one in isolation (Pashler et al., 2007). Deductive reasoning is facilitated when the domain is familiar and

concrete rather than abstract (Wason & Shapiro, 1971). Our approach begins with the presentation of concrete information in a context familiar to students, which gradually leads to an abstract understanding. As long as the concrete knowledge and the underlying abstract explanation are understood by learners for a specific situation, learning transfers from one software context to another will be more effective.

Conclusion

In this paper, a context-based approach to presenting security knowledge is proposed for software security learning. This approach is composed of three main strategies. The first is to establish an application context to create a meaningful situation for learners, which is described by application paradigms, application functionalities, and scenarios. The design of the application context aims to activate the learner's prior knowledge of software programming and anchors the learning about security knowledge. The second strategy is to organize underlying security knowledge in a structured manner that can stimulate learners' mental models to support more efficient learning in the specified context. The third is to guide learners to engage with concrete knowledge before studying abstract knowledge. This strategy assists learners in discovering meaningful concepts and relationships between practical functions and abstract knowledge when working in this context. Furthermore, it helps them apply the knowledge in various other contexts.

The approach was evaluated through a controlled quasi-experiment with 42 Bachelor students. There were positive findings in terms of security knowledge gain and learning satisfaction when students studied learning materials that were constructed using the context-based approach. According to the results, the proposed approach provides a sounder basis for software security learning than conventional methods. It is recommended that curriculum developers of software security courses should use the context-based approach as one of the teaching strategies to improve students'

performance in security knowledge learning. In the future, we plan to promote this approach for teaching secure programming and to use it to build a web-based learning application. We believe that such an online learning environment would allow more learners' to benefit from the learning approach.

Reference

Apvrille, A. and M. Pourzandi (2005). "Secure software development by example." IEEE Security & Privacy **3**(4): 10-17.

Bassok, M. J. C. D. i. P. S. (1996). "Using content to interpret structure: Effects on analogical transfer." **5**(2): 54-58.

Bennett, J., F. Lubben and S. Hogarth (2007). "Bringing science to life: A synthesis of the research evidence on the effects of context-based and STS approaches to science teaching." Science education **91**(3): 347-370.

Brézillon, P. (1996). "Context in human-machine problem solving: A survey." Technical Report 96/29, LAFORIA **6**(1996): 029.

Brézillon, P. (2002). Modeling and using context: Past, present and future. Rapport de recherche interne LIP6. Paris.

Brézillon, P. (2003). "Making context explicit in communicating objects." Communicating with Smart Objects: Developing Technology for Usable Pervasive Computing Systems, Kogan Page, London.

Brézillon, P. and J.-C. Pomerol (1999). "Contextual knowledge sharing and cooperation in intelligent assistant systems." Le Travail Humain: 223-246.

Cognition Technology Group at Vanderbilt (1992). "Anchored instruction in science and mathematics: Theoretical basis, developmental projects, and initial research findings." Philosophy of science, cognitive psychology: 244-273.

Cognition Technology Group at Vanderbilt (1992). "The Jasper series as an example of anchored instruction: Theory, program description, and assessment data." Educational Psychologist **27**(3): 291-315.

Cooper, S. and S. Cunningham (2010). "Teaching computer science in context." Acm Inroads **1**(1): 5-8.

Cordova, D. I. and M. R. Lepper (1996). "Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice." Journal of educational psychology **88**(4): 715.

Council, N. R. (2000). How people learn: Brain, mind, experience, and school: Expanded edition, National Academies Press.

Craik, K. J. W. (1967). The nature of explanation, CUP Archive.

CVE. "Browse Vulnerabilities By Date." from https://www.cvedetails.com/browse-by-date.php.

Dean, R. J. and L. Dagostino (2007). "Motivational factors affecting advanced literacy learning of community college students." Community College Journal of Research Practice **31**(2): 149-161.

Dey, A. K. (2001). "Understanding and using context." Personal ubiquitous computing **5**(1): 4-7.

Diethelm, I., P. Hubwieser and R. Klaus (2012). Students, teachers and phenomena: educational reconstruction for computer science education. Proceedings of the 12th Koli Calling International Conference on Computing Education Research, ACM.

Dolmans, D. H., W. De Grave, I. H. Wolfhagen and C. P. Van Der Vleuten (2005). "Problem-based learning: Future challenges for educational practice and research." Medical education **39**(7): 732-741.

Felder, R. M. and L. K. J. E. e. Silverman (1988). "Learning and teaching styles in engineering education." **78**(7): 674-681.

Gentner, D. and A. L. Stevens (2014). Mental models, Psychology Press.

Goldkuhl, G. and E. Braf (2001). Contextual knowledge analysis-understanding knowledge and its relations to action and communication. Second European Conference on Knowledge Management Proceedings.

Goldstone, R. L. and Y. Sakamoto (2003). "The transfer of abstract principles governing complex adaptive systems." Cognitive psychology **46**(4): 414-466.

Goldstone, R. L. and J. Y. Son (2005). "The transfer of scientific principles using concrete and idealized simulations." The Journal of the Learning Sciences **14**(1): 69-110.

Guzdial, M. (2006). "Teaching computing for everyone." Journal of Computing Sciences in Colleges **21**(4): 6-6.

Guzdial, M. (2010). "Does contextualized computing education help?" ACM Inroads **1**(4): 4-6.

Jonassen, D. and S. Land (2012). Theoretical foundations of learning environments, Routledge.

Kamina, P. and N. N. Iyer (2009). "From concrete to abstract: Teaching for transfer of learning when using manipulatives." NERA Conference Proceedings 2009. 6. https://opencommons.uconn.edu/nera_2009/6.

Klemke, R. (2000). Context Framework-an Open Approach to Enhance Organisational Memory Systems with Context Modelling Techniques. PAKM.

Ko, A. J. and B. A. Myers (2008). Debugging reinvented: asking and answering why and why not questions about program behavior. Proceedings of the 30th international conference on Software engineering, ACM.

Kozeracki, C. A. (2005). "Preparing faculty to meet the needs of developmental students." New directions for community colleges **2005**(129): 39-49.

Land, S. M. (2000). "Cognitive requirements for learning with open-ended learning environments." Educational Technology Research and Development **48**(3): 61-78.

Lave, J., E. Wenger and E. Wenger (1991). Situated learning: Legitimate peripheral participation, Cambridge university press Cambridge.

Leach, J., P. J. S. Scott and Education (2003). "Individual and sociocultural views of learning in science education." **12**(1): 91-113.

McCaulley, M. H. (1976). "Psychological Types in Engineering: Implications for Teaching." Engineering Education **66**(7): 729-736.

McCaulley, M. H., E. Godleski, C. F. Yokomoto, L. Harrisberger and E. D. J. E. E. Sloan (1983). "Applications of Psychological type in engineering-education " **73**(5): 394-400.

McGraw, G. (2006). Software security: building security in, Addison-Wesley Professional.

Merrill, M. D. (2000). Knowledge objects and mental models. Advanced Learning Technologies, 2000. IWALT 2000. Proceedings. International Workshop on, IEEE.

Nonaka, I. and N. Konno (1998). "The concept of" ba": Building a foundation for knowledge creation." California management review **40**(3): 40-54.

Pashler, H., P. M. Bain, B. A. Bottge, A. Graesser, K. Koedinger, M. McDaniel and J. Metcalfe (2007). Organizing Instruction and Study to Improve Student Learning. . IES Practice Guide. NCER 2007-2004. National Center for Education Research.

Perin, D. (2011). "Facilitating student learning through contextualization: A review of evidence." Community College Review **39**(3): 268-295.

Perin, D. J. C. C. R. (2011). "Facilitating student learning through contextualization: A review of evidence." **39**(3): 268-295.

Predmore, S. R. (2005). "Putting it into Context." Techniques: Connecting education and careers **80**(1): 22-25.

Rivet, A. E. and J. Krajcik (2008). "Contextualizing instruction: Leveraging students' prior knowledge and experiences to foster understanding of middle school science." Journal of Research in Science Teaching: The Official Journal of the National Association for Research in Science Teaching **45**(1): 79-100.

Rouse, W. B. and N. M. Morris (1986). "On looking into the black box: Prospects and limits in the search for mental models." Psychological bulletin **100**(3): 349.

Seel, N. M., S. Al-Diban and P. Blumschein (2000). Mental models & instructional planning. Integrated and holistic perspectives on learning, instruction and technology, Springer**:** 129-158.

Sherwood, R. D., C. K. Kinzer, J. D. Bransford and J. J. J. J. o. R. i. S. T. Franks (1987). "Some benefits of creating macro-contexts for science instruction: Initial findings." **24**(5): 417-435.

Wason, P. C. and D. Shapiro (1971). "Natural and contrived experience in a reasoning problem." The Quarterly Journal of Experimental Psychology **23**(1): 63-71.