SecSDM: A Usable Tool to Support IT Undergraduate Students in Secure Software Development

L. Futcher and R. von Solms

Nelson Mandela Metropolitan University, Port Elizabeth, South Africa
e-mail : lynn.futcher@nmmu.ac.za; rossouw.vonsolms@nmmu.ac.za

Abstract

Many IT undergraduate programs neglect to address the importance of integrating information security into the software development lifecycle. SecSDM is an integrated, risk-based methodology for supporting IT undergraduate students in secure software development. A software tool, based on the SecSDM methodology, has been developed to provide a means by which to apply this methodology to software development projects. However, from a developer's perspective, any such software tool needs to be usable. This means that such a tool should have good utility, be effective to use, efficient to use, safe to use, easy to learn, easy to remember and satisfying to use. This paper provides an overview of the SecSDM methodology and presents the results of a user satisfaction survey relating to the SecSDM software tool.

Keywords

SecSDM, secure software development, risk management, user satisfaction

1 Introduction

Technological advancements and the rapid increase in the use of the Internet by organizations and businesses across the globe have had a huge impact on information security. This, in turn, has resulted in major challenges for the software development industry.

Over the past decade there has been an increase in the number of security incidents reported. A substantial percentage of these incidents are the result of inadequate consideration of security during the requirements analysis, the design, implementation and testing of software systems (Walden & Frank, 2006). Conklin and Dietrich (2007) further support this by stating that most cyber vulnerabilities can be traced back to defects in software. These defects are the result of bad design and poor development practices (Conklin & Dietrich, 2007).

Most software development methodologies do not take into consideration the risk issues associated with the information assets implicated, and typically add security as an afterthought, thereby neglecting to integrate security throughout the software development life cycle. This often results in the implementation of inappropriate security controls.

Software engineers need to learn to consider security when writing requirements and design specifications and when developing, testing and deploying software (Pothamsetty, 2005, p. 54). Furthermore, Burley and Bishop (2011) suggest that raising the security implications at each stage of the life cycle would make students more aware of, and more sensitive to, security considerations throughout the software development life cycle (Burley & Bishop, 2011).

Although many researchers are in agreement with this, currently very little has been done to provide a simple, practical, risk-based approach to integrating security into the early stages of the software development life cycle; and that could consistently support students in the development of secure software. For this reason, the secure software development methodology (SecSDM), as described in Section 2, has been developed. SecSDM is an integrated, risk-based approach to support IT undergraduate students in the development of secure software.

This methodology is based on various information security and software development standards, guidelines and best practices. These include ISO/IEC 27002 (2005), the international code of practice for information security management; the various risk-related guidelines, as determined by ISO/IEC 27005 (2008); NIST SP 800-14 (1996), which outlines generally accepted principles and practices for securing information technology systems; and ISO/IEC 7498-2 (1989), which provides the basis of information security in software systems through five basic security services, supported by eight security mechanisms. In addition, the major secure software development contributions of various other key role players were considered including Microsoft, Open Web Application Security Project (OWASP), Oracle and the Software Engineering Institute (SEI).

This paper provides an overview of the SecSDM methodology and presents the results of a user satisfaction survey of the SecSDM software tool which was developed to support IT undergraduate students in the application of this methodology to their software development projects.

2 An overview of SecSDM

This section describes an integrated, risk-based approach to support secure software development. This is achieved by presenting a secure software development methodology, SecSDM, which integrates security aspects throughout the software development life cycle. The following six principles were considered fundamental in the development of this methodology, namely:

- Security must be integrated throughout all phases of the software development life cycle;
- Security aspects need to be considered from the very beginning of the software development life cycle, i.e. from the investigation phase;
- A risk-based approach is required to ensure the implementation of appropriate security controls that are functional, effective, correct and safe to use;

- Any security control implemented must be related to a specified risk identified. Traceability back to such a risk is therefore required;
- A structured approach is required that transparently integrates security aspects, without adding additional overheads with respect to time, cost or expert security skills required; and
- The approach developed must be practical, easy to use and easy to understand.

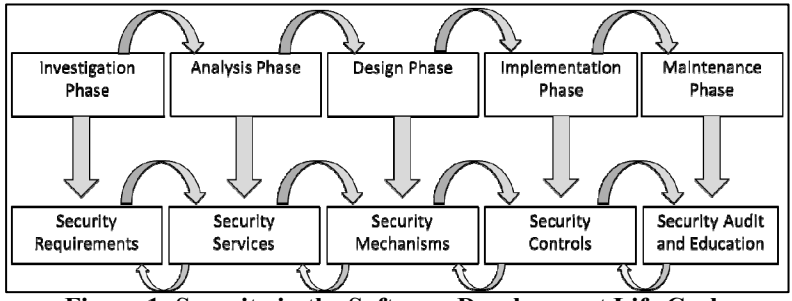

Figure 1: Security in the Software Development Life Cycle

In the literature studied, there exists strong support for integrating security throughout the software development life cycle, in order to minimize the risks associated with the information assets implicated. As depicted in Figure 1, the main security concerns to be addressed at each phase of the software development life cycle, according to SecSDM, are as follows:

- *Investigation Phase*: The output of this initial phase is a set of **security requirements** developed through a simple, structured risk-assessment exercise;
- *Analysis Phase*: During this phase, the **security services** are identified that satisfy the security requirements defined during the investigation phase;
- *Design Phase*: This phase determines how the security services will be implemented, by mapping them to specific **security mechanisms**;
- *Implementation Phase*: This phase identifies and implements the appropriate **security controls and components**; and
- *Maintenance Phase*: During this final phase, users need to be educated in **using the software application in a secure manner**.

SecSDM also ensures that all relevant security-related information is consolidated in a security report. This helps improve the auditability of the software application in question, since security-related decisions are traceable to the appropriate phase, as proposed by this secure software development approach.

A risk management approach, as described by Von Solms and Von Solms (2009) is integral to SecSDM. This approach requires that a detailed risk analysis be performed to identify the potential adverse business impacts of unwanted events, and the likelihood of their occurrence. The outcome of a detailed risk analysis can lead to the effective evaluation of risk, which is necessary in order to identify and implement

appropriate security controls and measures. This is essential for the development of secure software applications.

As depicted in Table 1, the ten steps of SecSDM are mapped directly to the risk management approach, as described by Von Solms and Von Solms (2009). From this table it is also clear that Steps 1 to 4 of SecSDM are mapped directly to risk analysis, while Steps 5 and 6 are mapped to risk evaluation. Similarly, Steps 7 to 10 are directly related to the treatment of such risks.

<table>
<tr><td rowspan="8">RISK MANAGEMENT</td><td rowspan="6">Risk Assessment</td><td rowspan="5">Risk Analysis</td><td rowspan="4">STEP 3: “Risk” Identification</td><td rowspan="2">Assets</td><td>Step 1a: Asset identification</td></tr>
<tr><td>Step 1b: Asset valuation</td></tr>
<tr><td rowspan="2">Threats</td><td>Step 2a: Threat identification</td></tr>
<tr><td>Step 2b: Threat assessment</td></tr>
<tr><td colspan="2">Vulnerabilities</td><td>Step 4: Vulnerability assessment</td></tr>
<tr><td rowspan="2">Risk Evaluation</td><td colspan="3">Step 5: Determine risk value or size</td></tr>
<tr><td colspan="3">Step 6: Prioritize risks</td></tr>
<tr><td rowspan="2">Risk Treatment</td><td colspan="4">Identify suitable controls
Step 7: Identify relevant security services
Step 8: Map security services to security mechanisms
Step 9: Summarize security services and security mechanisms required</td></tr>
<tr><td colspan="4">Implement identified controls
Step 10: Map security mechanisms to appropriate security controls and components</td></tr>
</table>

Table 1: Mapping of SecSDM Steps to Risk Management Approach

An important contribution of this methodology is that it does not simply focus on ‘*what’* needs to be done to support secure software development, but it also provides valuable guidance in terms of ‘*how’* this can be achieved. Such a simple, practical approach is always beneficial in the education of IT students. SecSDM provides a simple, easy-to-use and easy-to-understand approach to support secure software development by providing a set of repeatable and systematic steps to ensure that the set of security requirements generated is complete, consistent and easy to understand by all the stakeholders involved in the software development process. In addition, this methodology could help ensure that the security controls implemented will be functional, effective, correct and safe to use.

A further key contribution is the traceability of security requirements throughout the software development life cycle that is provided. This means that any security controls implemented can easily be traced back to a specific security requirement, based on a specific risk identified earlier in the software development life cycle.

3 Using SecSDM

SecSDM was initially implemented as a paper-based tool. However, there are certain drawbacks that are unavoidable when using such a paper-based system, including:

1. Cumbersome and time-consuming to use;
2. Essential steps in the process omitted;
3. Incorrect information captured;
4. Relies on the user recapturing essential information from previous steps;
5. Errors not easily detected;
6. Not easy to go back and make changes;
7. Does not result in a consolidated information security report.

In order to overcome these problems, a software tool, based on the SecSDM methodology was developed. The major goal of the SecSDM software tool is to alleviate all the above-mentioned drawbacks by having the application support the user in a usable and efficient manner.

In designing the software tool, a 'wizard-based' approach was taken to logically progress the user through the ten step process as determined by the SecSDM methodology. In addition, at each step of the application the user is provided with a task pane and an information pane, thereby providing the user with the necessary information relating to the particular step being carried out. A large portion of the interface was translated into background graphics to improve the performance of the application. Only the controls that users need to directly interact with are incorporated as actual controls.

Owing to space limitations, further details of this software tool lie outside the scope of this paper.

4 The User Satisfaction Survey

The usability of interactive products, including software tools and applications, refers to the extent to which such products have good utility and are effective to use, efficient to use, safe to use, easy to learn, easy to remember and satisfying to use - from a user's perspective. In this case, the users of the SecSDM software tool are software developers.

According to Xiao and Dasgupta (2002), '*user satisfaction is regarded as one of the most important measures of Information Systems success'*. The Questionnaire for User Interaction Satisfaction (QUIS) and the Software Usability Measurement Inventory (SUMI) are two well-known usability testing tools that gauge a user's satisfaction with using software applications. Many of the questions addressed by these tools are based on Jakob Nielsen's (1994) design principles, namely: visibility of system status; match between system and the real world; user control and freedom; consistency and standards; error prevention; recognition, rather than recall;

flexibility and efficiency of use; aesthetic and minimalist design; help user's recognize, diagnose and recover from errors; and help and documentation.

A user satisfaction survey was carried out on the SecSDM software tool. The purpose of the survey was to establish the extent to which the IT undergraduate students were satisfied with using the tool, and to gain valuable feedback that could be used to make improvements. The user satisfaction survey took the form of a paper-based questionnaire comprising 19 statements, as shown in Table 2.

Statement
1. Overall, I am satisfied with how easy it is to use this system.
2. It was simple to use this system.
3. I can effectively complete the assigned tasks using this system.
4. I am able to complete the assigned tasks quickly using this system.
5. I am able to efficiently complete the assigned tasks using this system.
6. I feel comfortable using this system.
7. It was easy to learn to use this system.
8. I believe I became productive quickly using this system.
9. The system gives error messages that clearly tell me how to fix problems.
10. Whenever I make a mistake using the system, I recovered easily and quickly.
11. The information (such as online help, on screen messages, and other documentation) provided with this system is clear.
12. It is easy to find the information I needed.
13. The help provided by the system is easy to understand.
14. The information is effective in helping me complete the tasks and scenarios.
15. The organisation of information on the system screens is clear.
16. The interface of this system is pleasant.
17. I like using the interface of this system.
18. This system has all the functions and capabilities I expect it to have.
19. Overall, I am satisfied with this system.

Table 2: SecSDM User Satisfaction Survey

Whereas statements 1 to 8 related to '*ease of use'*, statements 9 and 10 related to '*error messages'* and '*recovery from problems'*. Similarly, statements 11 to 15 related to '*online help and information provided',* and statements 16 to 19 referred to the '*general interface'*, '*system capabilities'* and '*overall user satisfaction'*.

The IT undergraduate project students were required to rate the extent to which they agreed to each of the 19 statements. A 5-point Likert scale was used for the ratings, where a rating of '*1*' indicated '*disagree'* and a rating of '*5'* indicated '*fully agree'*. A rating of *'3'* indicated that the students '*neither agreed nor disagreed'* with the statement. A '*not applicable'* option was also provided for cases in which students felt they were not in a position to respond to a specific statement.

The user satisfaction questionnaires were distributed amongst IT undergraduate students and completed during a practical class in a computer laboratory where they had further access to the SecSDM software tool. The questionnaires were then collated and the results captured in a Microsoft Excel spreadsheet. These were then

analyzed and interpreted. A total of 41 students, of the 56 registered for the software development project module, completed the user satisfaction survey, thereby representing a response rate of 73%. The results of the SecSDM user satisfaction survey and the associated interpretations are presented in the following section.

5 Results of User Satisfaction Survey

The results of the SecSDM user satisfaction survey are depicted in Table 3. From these results it is evident that the majority of respondents indicated that they either '*fully agreed'* (indicated by a *'5'*) or '*partially agreed*' (indicated by a *'4'*) with the 19 statements, as listed in Table 2. Statements 15, 16 and 17, which related primarily to the interface, measured the highest level of agreement with percentages of 78%, 90% and 95%, respectively.

This resulted in a high level of overall satisfaction with using SecSDM, as indicated for Statement 19, which recorded 80% of respondents in agreement with this statement. A further 73% of respondents were in agreement with Statement 3, which related to the effectiveness of SecSDM in completing the assigned tasks.

According to these results, the main areas of the SecSDM software tool requiring attention relate to error detection and recovery (Statements 9 and 10) and help and documentation (Statements 11, 12 and 13). Between 40% and 50% of the respondents were in full or partial agreement with these particular statements.

The average results indicate that the majority of the respondents were satisfied with the SecSDM software tool, with an average of 26% being in '*full agreement'*, having indicated a '*5*' on the Likert scale, and 37% being in '*partial agreement'*, having indicated a '*4*'. A further 25% were neither in agreement nor disagreement – being in the middle of the scale. On average, only 12% of the respondents indicated any level of disagreement.

Statement	1 Dis-agree	2	3	4	5 Fully Agree	N/A
1. Overall, I am satisfied with how easy it is to use this system	0%	5%	34%	39%	22%	0%
2. It was simple to use this system	0%	10%	24%	37%	29%	0%
3. I can effectively complete the assigned tasks using this system	0%	2%	22%	39%	34%	2%
4. I am able to complete the assigned tasks quickly using this system	0%	20%	20%	32%	27%	2%
5. I am able to efficiently complete the assigned tasks using this system	0%	7%	29%	27%	37%	0%
6. I feel comfortable using this system	0%	10%	29%	41%	20%	0%
7. It was easy to learn to use this system.	2%	7%	39%	34%	17%	0%
8. I believe I became productive quickly using this system.	2%	7%	27%	41%	20%	2%
9. The system gives error messages that clearly tell me how to fix problems.	12%	12%	29%	20%	22%	5%
10. Whenever I make a mistake using the system, I recovered easily and quickly.	7%	12%	32%	24%	22%	2%
11. The information (such as online help, on screen messages, and other documentation) provided with this system is clear.	12%	7%	37%	34%	7%	2%
12. It is easy to find the information I needed.	5%	15%	32%	29%	17%	2%
13. The help provided by the system is easy to understand.	7%	24%	15%	32%	15%	7%
14. The information is effective in helping me complete the tasks and scenarios.	2%	10%	27%	41%	20%	0%
15. The organisation of information on the system screens is clear.	0%	5%	17%	49%	29%	0%
16. The interface of this system is pleasant.	0%	0%	10%	34%	56%	0%
17. I like using the interface of this system.	0%	0%	5%	46%	49%	0%
18. This system has all the functions and capabilities I expect it to have.	0%	7%	29%	44%	20%	0%
19. Overall, I am satisfied with this system.	0%	2%	17%	56%	24%	0%
Average Percentages	**3%**	**9%**	**25%**	**37%**	**26%**	**1%**

Table 3: SecSDM User Satisfaction Survey Results

1	*'It was easy to identify the problems and risks we have never thought of in our project, so by using this software I am well aware of those risks'*
2	*'I was impressed with how easy they made it to generate a report'*
3	*'I find this system very user friendly and I think everyone can use it'*
4	*'I personally don't have dislikes I was totally impressed'*
5	*'This system was easy to use and understand but there is too much information at the information side'*
6	*'All in all it was excellent'*
7	*'This tool seems to be working pretty well, maybe just a few things to look at but I am totally satisfied'*
8	*'Great system it made our task very easy to finish'*
9	*'The interface looked really nice'*
10	*'Inserting my information and assigning risk values to them was relatively easy'*
11	*'The interface looks good, good colour usage'*
12	*'Overall I am very happy with the system it is better than doing it manually'*
13	*'Well done to the people who designed it'*

Table 4: SecSDM User Satisfaction Survey - Positive Comments

Tables 4 and 5 indicate the positive and negative comments recorded from the SecSDM user satisfaction survey carried out, respectively. The positive comments reflected in Table 4 reinforce the high level of satisfaction the students experienced in using the SecSDM software tool. Many of these comments related to the ease with which one could carry out the task of risk identification, and the general appeal of the interface.

An interesting observation is that 27% of the negative comments, as indicated in Table 5, related to a lack of understanding of the terminology used. This re-emphasizes the fact that the understanding of information security terms and concepts is fundamental to basic education in information security. These terms should be introduced early in the curriculum, and re-addressed throughout all the subsequent years of the CS/IS/IT qualification.

1	*'Key word explanation would be nice'*
2	*'The program should describe in greater detail how to identify your information assets'*
3	*'Check updating of risks if you click previous button'*
4	*'Why after moving past a certain point you can't go back and change certain fields'*
5	*'This system was easy to use and understand but there is too much information at the information side'*
6	*'I thought the security mechanisms for each risk would maybe be different or broken down and explained to us'*
7	*'The terminology is difficult to understand'*
8	*'It is good but must I had the English dictionary a few times as the terms used are not familiar'*
9	*'Had trouble generating a report'*
10	*'The system has too much text, the font is not big enough'*
11	*'Did not like the fact that I could not go back and change the assets'*

Table 5: SecSDM User Satisfaction Survey - Negative Comments

6 Conclusion

From the results of the SecSDM user satisfaction survey conducted, one may conclude that the SecSDM software tool supports the activities involved in developing secure software, by guiding the user through the process of risk identification and the selection of appropriate security mechanisms to help ensure the implementation of security controls that are functional, effective, correct and safe to use.

Although one cannot generalize by stating that SecSDM would guarantee the development of secure software, it is believed that SecSDM and the associated software tool form a good foundation to support further research in secure software development. It is envisaged that SecSDM could be extended to other educational institutions and to the software development industry. Although this methodology was evaluated in an academic environment, it is believed that it could also support the software development industry in the development of secure software. However, this would need to be verified by future research.

7 References

Burley, D., & Bishop, M. (2011). Summit on Education in Secure Software: Final Report. National Science Foundation.

Conklin, W. A., & Dietrich, G. (2007). Secure Software Engineering: A New Paradigm. Proceedings of the 40th Hawaii International Conference on System Sciences (pp. 1-6). IEEE.

ISO/IEC. (1989). ISO/IEC 7498-2. Information Processing Systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture. Switzerland: ISO/IEC.

ISO/IEC. (2005). ISO/IEC 27002. Information technology - Security techniques - Code of practice for information security management. Switzerland: ISO/IEC.

ISO/IEC. (2008). ISO/IEC 27005. Information technology - Security techniques - Information security risk management. Switzerland: ISO/IEC.

Nielsen, J. (1994). Heuristic Evaluation. New York: John Wiley & Sons.

NIST. (1996, September). SP800-14: Generally Accepted Principles and Practices for Securing Information Technology Systems. Retrieved March 23, 2012, from NIST - Computer Security Division - Computer Security Resource Center - Special Publications: http://csrc.nist.gov/publications/nistpubs/800-14/800-14.pdf

Pothamsetty, V. (2005). Where Security Education is Lacking. Proceedings from Information Security Curriculum Development (InfoSecCD) Conference (pp. 54-58). Kennesaw: ACM.

Von Solms, S., & Von Solms, R. (2009). Information Security Governance. New York: Springer.

Walden, J., & Frank, C. (2006). Secure Software Engineering Teaching Modules. Information Security Curriculum Development Conference (InfoSecCD) (pp. 19-23). Kennesaw: ACM.

Xiao, L., & Dasgupta, S. (2002). Measurement of User Satisfaction with Web-based Information Systems: An Empirical Study. Eighth Americas Conference on Information Systems, 1149-1155.