Proceedings of the Eleventh International Symposium on
Human Aspects of Information Security & Assurance (HAISA 2017)

Secure Coding Practices in the Software Development Capstone Project

V.S. Mdunyelwa, J.F. van Niekerk and L.A. Futcher

Centre for Research in Information and Cyber Security, Nelson Mandela University, Port Elizabeth, South Africa
e-mail: {s211117803, Johan.VanNiekerk, Lynn.Futcher}@nmmu.ac.za

Abstract

Web applications play an important role in many organisations, but could also expose these organisations to cyber security risks. Organisations use a variety of cyber security controls to mitigate risks. Currently, most organisational security spending focus on reducing network security related risks. However, most attacks focuses on vulnerabilities existing at the web application layer. Security breaches in web applications are mostly caused by programmers' failure to adhere to secure coding practices, such as those recommended by the Open Web Application Security Project. The purpose of this paper is to determine whether software development students know about secure coding practices and whether they implement them when developing web applications as part of their capstone projects.

Keywords

Secure Coding, Web Application Security, OWASP, Capstone Project, Knowledge, Behaviour

1. Introduction

Nowadays many organisations use information to achieve their daily activities. Both the private and public sectors rely on information systems to carry out their key operational activities. Previously, many software applications were desktop applications which could be used on standalone computers. The arrival of the internet has seen web application development gain a significant amount of attention. Media players and word processors are typical examples of desktop applications, whilst internet banking and online shopping carts on e-commerce websites can be considered as web applications (Jeff 2017).

Desktop applications are usually installed on a single computer for a specific task, whereas web applications are made available on servers (Jeff 2017).Thus, web applications are exposed to a number of threats on the internet that desktop applications may not be exposed to. In addition, web applications often handle very sensitive data, used for carrying out critical tasks such as banking, online shopping and online tax filing (Deepa & Thilagam 2016). These applications are trusted by billions of users for performing such daily activities.

Web applications have received attention from both academia and industry to initiate some defence mechanisms to protect them from security threats (Deepa & Thilagam 2016). Handling risks related to the security of web applications is a major challenge for many organisations. To this effort, information security has become one of the top managerial priorities in many organisations (Benbasat 2010). The need for securing organisation's network resources has increased over the past decade. Many organisations address this need by including costs for firewalls and other network security controls in their budgets. However, 75% of all attacks that are on the internet are executed through the application layer of the OSI Model (Customs Solutions Group 2012). This indicates that more focus needs to be put on securing web applications.

More than 75% of web applications have vulnerabilities (Chandrasekar et al. 2017). Many of these web applications have common vulnerabilities which can be easily corrected (Zhu et al. 2014). Addressing many of these web application vulnerabilities is relatively straightforward through introducing secure coding practices. Academic institutions may teach proper programming, but because of the limited resources they only focus on the correctness of code and end up overlooking security in the code (Bishop & Orvis 2006). This oversight in addressing security issues is felt most during the maintenance phase of the Software Development Life Cycle, which is costly (Customs Solutions Group 2012). Therefore, if academic institutions would integrate secure coding practices in undergraduate software development courses, students would be more likely to implement them and in turn help improve security in web applications in industry.

Human aspects in information and cyber security usually consists of two major elements, namely knowledge-based and behavioural elements. Knowledge deals with the underlying requisite knowledge that would allow a person to behave in accordance to security guidelines, whilst the behavioural elements deal with everything else which if one can assume requisite knowledge, would actually make them apply it (Van Niekerk & Von Solms 2010). It is possible for a software development student to have the requisite knowledge and not behave appropriately, but it is unlikely for someone to behave appropriately if they do not have the requisite knowledge.

The purpose of this paper is firstly to demonstrate that software development students generally do not adhere to secure coding practices when performing their capstone projects. Secondly, it tries to determine whether the lack of adherence is caused by purely behavioural aspects or an underlying lack of knowledge. It therefore focuses on both knowledge and behaviour relating to secure coding practices. The next section discusses secure coding practices according to the Open Web Application Security Project (OWASP) guidelines. It further discusses what the Association for Computing Machinery (ACM) curricular guidelines for Information Technology states in terms of secure coding practices in Section 3. Section 4 presents the sample and setting, while Section 5 addresses the behavioural aspects of students and Section 6 focuses on the knowledge of students. Future work and conclusion follows in Section 7 and 8 respectively.

2. Secure coding practices

The secure processing of information consists of the implementation of various controls in order to reduce risks caused by a specific security vulnerability (von Solms & van Niekerk 2013). In the case of web applications there are various types of controls that could play a role in protecting information. The types of controls are:

- Physical controls, for example, providing a lock on the computing facility.
- Technical controls, for example, an authentication mechanism.
- Operational controls deal with human behaviour, for example, having to use the lock to prevent access to the computing facility or requiring authentication before allowing access to computing resources.

As demonstrated above, if these control types are not collectively addressed, it could lead to security breaches. Some of the controls may deal with end-user behaviour, while others deal with the behaviour of Information Technology staff in relation to configuring technical controls (Chung et al. 2014). This includes programming staff who develop web-based applications and who should adhere to secure coding practices to mitigate risks associated with web application security. While the technical control would be a secure coding practice, an operational control would be a policy and procedures for a programmer to adhere to secure coding practices.

In the case of web application security, one of the best sources for securing web applications is the Open Web Application Security Project (OWASP) (owasp.org). OWASP is a non-profit organisation with a goal to improve the security of web applications. This organisation focuses on making security visible, so that individuals and organisations such as governments, companies and educational institutions have more knowledge when developing web applications (OWASP 2017).

OWASP focuses on many development security platforms. This research study uses OWASP security practices which apply to the Microsoft .NET framework. There are several ways in which programmers could introduce security flaws when using the .NET framework especially when working at the data access layer.

The controls OWASP proposes are technical measures, while the programmers themselves having to adhere to them would be an operational measure. Operational controls fail when the human does not adhere to them. Humans will often not adhere to such a control due to the lack of knowledge or because there is no procedural requirement forcing them to adhere to the control. Therefore, programmers should be educated on secure coding practices to ensure that they are aware of the vulnerabilities that may otherwise be missed or realised at a stage where it might be too late. This behaviour should also be audited to ensure that programmers adhere to such secure coding practices.

Table 1 presents a sample set of secure coding practices extracted from the OWASP guidelines. Those selected are specifically for the Windows environment used in the .NET framework. In addition, these secure coding practices were chosen because they are specific to the data access layer where attackers could easily access information if it is not secure.

SP	**OWASP secure coding practices**
SP1	Use Parameterised SQL commands for all data access, without exception.
SP2	Do not use SQL command with string parameter made up of a concatenated SQL string.
SP3	Whitelist allowable values coming from the user. Use Enums, TryParse of lookups to ensure that the data coming from the user is as expected.
SP4	Apply the Principle of Least Privilege when setting up the Database of your choice.
SP5	When using SQL Server, prefer integrated authentication over SQL authentication.
SP6	Using stored procedures is the most effective way to counter the SQL injection vulnerability.
SP7	Encrypt sensitive data in the database including connection strings.
SP8	Connection strings should be based in a configuration file.
SP9	Never write your own encryption.

Table 1: OWASP secure coding practices

Since this research focuses on the knowledge and behaviour of software development students regarding secure coding practices, it is important to consider curricular guidelines. These are discussed in the next section.

3. Curricular guidelines within the software development curriculum.

Traditionally, most software developers are educated in tertiary institutions. Tertiary institutions follow various curricula to determine what to teach. According to the ACM curricular guidelines for Information Technology (Lunt et al. 2017), the focus of a software development curriculum is to educate students with programming and database fundamentals. During the students' final year it is important for them to consolidate and integrate all the skills gained throughout the qualification. Further, the ACM curricular guidelines state that security should be integrated into the software development life cycle throughout the curriculum. This means that secure coding practices should be taught from the early stages in the curriculum.

Capstone experiences are projects that are performed during the final year of a qualification, mostly in engineering qualifications (Lunt et al. 2017). However, the capstone experience has gained a lot of support in the computing discipline. Capstone projects normally come with 3 major elements, namely; 1) working in teams of 3 to 5; 2) choosing a real-world problem to solve using the project and the project must be integrative and students who are engaged in this experience should have completed most of the curriculum before attempting this project (Lunt et al. 2017). Capstone projects are therefore the best place to determine whether secure coding practices are being taught and applied during the software development life cycle.

4. Sample and setting

This study was conducted within a comprehensive tertiary institution in South Africa. Comprehensive institutions offer both academic degrees as well as vocational qualifications. In this case, the sample was drawn from students registered for the National Diploma: Software Development. The National Diploma: Software Development is a vocational qualification focusing on providing prospective software developers with the requisite skills for professional work. Students from this program were selected as a research sample for this study. The sample was selected both *purposively* and because it was *convenient*.

They were selected *purposively* in the sense that these students were selected specifically because the researchers believe that they should have requisite web application development skills. They are a *convenient* sample because the researchers have access to these specific students. This research project received ethics approval from the university research ethics committee (Ref H15-ENG-ITE-009).

The Software Development National Diploma focuses on software development skills and therefore has a large portion of the curriculum dedicated to the development of web applications. Subjects include Internet Programming and Development Software as part of the curriculum. The department where the students reside is well known for research in information security. It should therefore be reasonable to assume that these students have some form of secure coding knowledge. To a certain extent this qualification has also been modelled according to the ACM curricular guidelines for Information Technology (Lunt et al. 2017). In the National Diploma: Software Development, final year students are expected to develop a capstone project as described in the ACM curriculum (Lunt et al. 2017).

This capstone project is a software development orientated project which consolidates and integrates all the skills gained throughout the software development qualification. The researchers therefore believe that this capstone project is an ideal place to determine whether students adhere to the secure coding practices for the development of secure web applications as proposed by OWASP.

Students at the university within this qualification are trained specifically to develop software in a Windows environment using the .NET framework. The students therefore, as a minimum, if they were taught the appropriate secure web application principles should be aware of the OWASP guidelines for the .NET framework.

5. Behaviour regarding secure coding practices

The first phase in this research sets out to determine whether students doing their capstone project actually adhere to the secure coding practices as recommended by OWASP. In order to determine the degree of behavioural compliance, the researchers assume that the students would already have all the requisite knowledge regarding secure coding practices. Therefore, this research could focus only on whether or not completed capstone project implemented these practices. Since the capstone projects are carried out after all formal programming education subjects have been completed in the curriculum, this assumption was deemed appropriate. For this purpose, the researchers focuses specifically on completed projects from a prior year.

5.1. Data generating instruments for behaviour

The questions in Table 2 relate to the OWASP secure practices (SP1 to SP9) previously illustrated in Table 1. These questions were used to create an instrument in the form of a checklist that was used to perform a code review for final year capstone projects submitted during the 2015 period. All web-based application projects were included in this code review. The data used in this study was specifically for 2015, mainly because data from 2017 is incomplete and will only be available at the end of the year. 2016 proved to be an unfavourable year for South African university students, since there were wide spread student protests during the year which disrupted classes and student projects. From the 2015 data, fifteen projects were included in this study and every web form was reviewed and all data was recorded on the instrument as shown in Table 2. The code review was conducted to determine whether students' adhere to secure coding practices when developing their capstone projects.

SP	**Questions asked based on OWASP practices**
SP1	Are they making use of parameterised SQL commands for all data access?
SP2	Do they make use of concatenated strings in the queries?
SP3	Are all input fields validated?
SP4	Do they make use of the Principle of Least Privilege when setting up their databases?
SP5	Do they use integrated authentication or do they use SQL authentication?
SP6	Do they use stored procedure for their queries?
SP7	Are they encrypting their connection strings?
SP8	Does the connection string only appear once in the web.config file?
SP9	Is all the sensitive data being encrypted using the OWASP recommended methods?

Table 2: Checklist to use for code review

The code review was conducted by the primary researcher using the checklist provided in Table 2. All results were captured in a database for ease of analysis.

5.2. Data collection and analysis

Table 3 shows the results of the code review for each project (P1 to P15) and every secure coding practice (SP1 to SP9). Each project consisted of a number of webforms. The percentage to which a project adhered to a secure coding practice was calculated as the ratio between the number of webforms that should have used the practice and those that actually did use the practice. For example, Project P12 had 19 forms and only 9 of those accessed the database, of those that accessed the database, only 3 forms used stored procedures as in SP6, therefore the percentage would calculate as 3/9*100=33%.

Proj#	**SP1**	**SP2**	**SP3**	**SP4**	**SP5**	**SP6**	**SP7**	**SP8**	**SP9**
P1	100%	100%	100%	100%	0%	85%	0%	100%	28%
P2	84%	100%	100%	100%	0%	100%	0%	65%	15%
P3	100%	100%	97%	100%	0%	27%	0%	42%	15%
P4	100%	100%	100%	100%	0%	0%	0%	100%	12%
P5	50%	50%	6%	100%	0%	50%	0%	0%	40%
P6	90%	80%	70%	81%	0%	13%	0%	81%	60%
P7	93%	93%	100%	92%	0%	67%	0%	57%	40%
P8	100%	100%	100%	0%	0%	27%	0%	100%	0%
P9	100%	100%	100%	55%	0%	45%	0%	100%	12%
P10	100%	100%	100%	5%	0%	40%	0%	100%	5%
P11	100%	100%	85%	0%	0%	57%	0%	100%	40%
P12	85%	85%	100%	0%	100%	33%	0%	87%	57%
P13	73%	70%	40%	0%	100%	13%	0%	53%	26%
P14	12%	27%	0%	9%	100%	0%	0%	36%	40%
P15	40%	57%	14%	92%	100%	0%	0%	35%	38%
Avg.	82%	84%	74%	56%	27%	37%	0%	70%	29%

Table 3: Adherence to secure coding practices

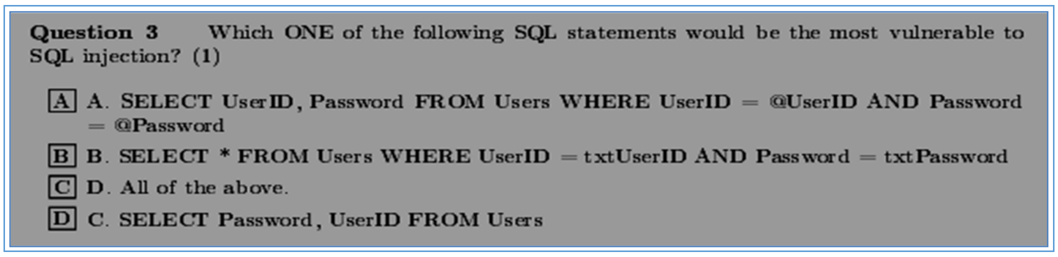
As shown in Table 3 above, there was generally a lack of adherence to secure coding practices. Where adherence did occur it was often inconsistent. For example, SP5 was about using Windows authentication. 11 of the projects did not use Windows authentication, however 4 of them did use the desired authentication model. This clearly indicates that some of them knew how to do this. This is probably a behavioural issue caused by a lack of knowledge as to **why** they should choose one authentication model above the other. As another example, SP7 deals with the encryption of a connection string. None of the projects encrypted their connection strings. However, it was determined that the capstone project requirement is that they should not encrypt the connection strings because values used in the connection string are used as part of the project evaluation. In this case, additional measures should be taken to make sure that students do know how to encrypt these connection strings. SP9 deals with the Principle of Least Privilege. Students are required to apply this principle. The fact that it is partly are adhered to it shows that they either cannot use it or they do not know why they should use it. Therefore, the lack of behavioural compliance might in fact be caused by an underlying lack of knowledge. The next phase of the research focused on determining whether current capstone students have this requisite knowledge.

6. Knowledge regarding secure coding practices

This phase of the research set out to determine whether students doing their capstone projects have the requisite knowledge regarding secure coding practices as recommended by OWASP.

6.1. Data generating instruments for knowledge

The data generating instrument for determining the students' knowledge regarding secure coding practices took the form of a questionnaire. The questionnaire was structured according to the OWASP secure coding practices as shown in Table 1. Questions for each SP were constructed to determine whether students would be able to apply the requisite knowledge. Due to space limitations, the full set of questions cannot be depicted in their entirety, as many of these questions contained code snippets and schematic representations as part of phrasing of the question. Figure 1 shows an example of a question asked to determine students' knowledge.



Question 3 Which ONE of the following SQL statements would be the most vulnerable to SQL injection? (1)

A A. SELECT UserID, Password FROM Users WHERE UserID = @UserID AND Password = @Password

B B. SELECT * FROM Users WHERE UserID = txtUserID AND Password = txtPassword

C D. All of the above.

D C. SELECT Password, UserID FROM Users

Figure 1: Example question for knowledge questionnaire

The questions were multiple choice questions where students were required to select the answer that best answered the question. These results were collected and analysed as discussed in the next section.

6.2. Data collection and analysis

In the case of secure coding practices, students should at the very least be able to apply the requisite knowledge. The questions were thus structured appropriately. The questionnaire was administered during a 2017 project lecture where 86 students were present. The lecturer for the capstone project was not present. The researchers briefly provided students with the context of the study before distributing the paper-based questionnaire for completion.

The questionnaire was marked by an automated computer marking system. Table 4 shows the consolidated results of the questions for each SP by showing the percentage of correct answers, and the averages from the behavioural instrument.

SP	**SP1**	**SP2**	**SP3**	**SP4**	**SP5**	**SP6**	**SP7**	**SP8**	**SP9**
K	66%	36%	30%	36%	17%	21%	11%	3%	1%
B	82%	84%	74%	56%	27%	37%	0%	70%	29%

Table 4: Underlying knowledge related to the ability to apply secure coding practices

If one considers the results of both parts of the study collectively as demonstrated in Table 4, in some cases students do have both the behavioural and knowledge aspects. However, in most cases they performed poorly. For example, in SP4, Principle of

Least Privilege, some students did not adhere to this practice, and this is important in terms of web application security. From SP4, it is evident that students also did not have all the requisite knowledge as to how to use this and/or why the Principle of Least Privilege should be used. Another interesting observation is when looking at SP1, students generally behaved well and only a few did not comply with this secure practice. In many cases, the level of Behavioural adherence is higher than the actual level of the underlying security knowledge. This could quite possibly be an indication that lectures are teaching students the appropriate ways of using the code without linking it to the underlying security context. This still poses a long-term risk because if the programmers do not understand why they are doing something, they might still disregard it in future implementations. Also by looking at SP1, these students generally have the knowledge about the use of SQL commands but do not perform accordingly because they possibly do not know why they should adhere to secure practices. Therefore, it can be seen that these students lack both behavioural compliance and the requisite knowledge.

7. Future work

Future work will focus on both the knowledge and the behavioural aspects of compliance to secure coding best practices. In order to address the lack of knowledge, the primary researcher will create an educational intervention in the form of a blended learning program. It is the intention that students who complete this educational intervention will be tested again using the same instrument as used during this research to provide the researcher with pre and post test data.

Addressing the lack of underlying knowledge might have an impact on behavioural compliance. However, in order to further encourage behavioural compliance an instrument in the form of a checklist will form part of formal project documentation for future capstone project students.

8. Conclusion

We live in a world where information is important. Therefore, none adherence to secure coding practices by developers puts information at risk. These programmers cannot adhere if they are not educated regarding the secure coding best practices in the first place. This research has shown that such a lack in knowledge exists among current software development students at a specific South African university. More attention therefore needs to be placed on educating students in this regard during the course of their studies.

9. Acknowledgements

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the authors, and are not necessarily to be attributed to the NRF.

10. References

Benbasat, I., 2010. S PECIAL I SSUE I NFORMATION S ECURITY P OLICY C OMPLIANCE : A N E MPIRICAL S TUDY OF R ATIONALITY -B ASED B ELIEFS. , 34(3), pp.523–548.

Bishop, M. & Orvis, B.J., 2006. A Clinic to Teach Good Programming Practices. *Proceedings of the 10th Colloquium for Information Systems Security Education*, pp.168–174. Available at: http://nob.cs.ucdavis.edu/~bishop/papers/2006-cisse-2/clinic.pdf.

Chandrasekar, K., Cleary, G., Cox, O. and O Gorman, B., 2017. *Internet Security Threat Report*, Available at: https://www.symantec.com/security-center/threat-report.

Chung, S., Hansel, L., Bai, Y., Moore, E., Taylor, C., Crosby, M., Heller, R., Popovsky, V. and Endicott-Popovsky, B., 2014. What approaches work best for teaching secure coding practices? *2014 HUIC Education & STEM Conference*.

Customs Solutions Group, 2012. A CISO ' s Guide to Application Security. Available at: http://h30528.www3.hp.com/Security/CISOGuideToApplicationSecurity.pdf.

Deepa, G. & Thilagam, P.S., 2016. Securing web applications from injection and logic vulnerabilities: Approaches and challenges. *Information and Software Technology*, 74, pp.160–180. Available at: http://dx.doi.org/10.1016/j.infsof.2016.02.005.

ISO/IEC, 2013. Sans 27002 2013.

Jeff, S., 2017. No Title. Available at: http://www.streetdirectory.com/travel_guide/114448/programming/desktop_applications_vs_web_applications.html [Accessed August 28, 2017].

Lunt, B., Sabin, M., Hala, A., Impagliazzo, J. and Zhang, M., 2017. Information Technology Curricula 2017. , pp.1–92.

Van Niekerk, J.F. & Von Solms, R., 2010. Information security culture: A management perspective. *Computers and Security*, 29(4), pp.476–486. Available at: http://dx.doi.org/10.1016/j.cose.2009.10.005.

OWASP, 2017. No Title. Available at: https://www.owasp.org/index.php/Main_Page [Accessed August 8, 2017].

von Solms, R. & van Niekerk, J., 2013. From information security to cyber security. *Computers and Security*, 38, pp.97–102. Available at: http://www.sciencedirect.com.ezproxy.unisabana.edu.co/science/article/pii/S0167404813000801.

Zhu, J., Xie, J., Lipford, H. & Chu, B., 2014. Supporting secure programming in web applications through interactive static analysis. *Journal of Advanced Research*, 5(4), pp.449–462. Available at: http://dx.doi.org/10.1016/j.jare.2013.11.006.