

# Raising Secure Coding Awareness for Software Developers in the Industry

Tiago Gasiba  
Siemens AG, München  
tiago.gasiba@siemens.com

Ulrike Lechner  
Universität der Bundeswehr München  
ulrike.lechner@unibw.de

**Abstract**—Many industrial IT security standards and policies mandate the usage of a secure coding methodology in the software development process. This implies two different aspects: first, secure coding must be based on a set of secure coding guidelines, and second software developers must be aware of these secure coding practices. On the one side, secure coding guidelines seems a bit like a black-art: while there exist abstract guidelines that are widely accepted, low-level secure coding guidelines for different programming languages are scarce.

On the other side, once a set of secure coding guidelines is chosen, a good methodology is needed to make them known by the people which should be using them, i.e. software developers.

Motivated both by the secure coding requirements from industry standards and also by the mandate to train staff on IT security by the global industry initiative "Charter of Trust", this paper presents an overview of important research questions on how to choose secure coding guidelines and on how to raise software developer awareness for secure coding using serious games.

**Index Terms**—security policy, secure coding, guidelines, IT security, industry standard, information systems, industry, serious games, capture-the-flag

## I. INTRODUCTION

The Charter of Trust [1] is a global initiative which is being undertaken by several leading companies to address the growing concerns related to IT Security of its products and services. In order to tackle IT Security issues at its root and early stages in product development, one of the points of this initiative addresses the topic of cybersecurity education and awareness [2].

This aspect is also mandated by several industry standards, to which companies are subject to compliance, such as 62.443 [3], 27k [4], NIST SP 800-39 [5], etc. As such, software developers need to be trained and familiar with how to develop, avoid security pit-falls and write secure code in the programming language being used for product development. The basis for this is a well defined and clear set of secure coding guidelines. These come in two flavors: abstract guidelines, such as OWASP [6], or programming language-specific such as MISRA-C [7], CERT SEI-C [8], CERT SEI-Java [9].

In order to tackle the issue of raising IT Security awareness of software developers in the industry, our vision is to use a serious game approach where the individual challenges are based on secure coding guidelines (SCG).

This work, based on our industry experience and observations, lays out some research questions that address both the

topic of selecting secure coding guidelines but also the topic on how to raise awareness about secure coding based on these guidelines.

Section II outlines the current state of the art. In Section III we propose a method to derive secure coding guidelines and also present our research questions. Finally Section IV presents preliminary results and future work.

## II. STATE OF THE ART

### A. Secure Coding Guidelines

Table I shows excerpts from three prominent industry standards, which mandate secure coding practices or even explicitly the usage of secure coding guidelines. The requirement gives no clear indication about which secure coding guidelines should be adopted - this can be understood in light of the fact that there is a lacking a general consensus and standardization of SCG.

TABLE I: Secure Coding Requirements from Standards

Standard	Requirement text
62443-4-1	[...] incorporate security coding [...]
27002	[...] secure coding guidelines for each programming language used [...]
NIST SP 800-39	[...] Information system security engineers employ ... secure coding techniques [...]

Our experience has shown that the quest for secure coding guidelines can result in (1) lack of SCG, (2) too many SCG or (3) conflicting SCG/recommendations. This diversity and lack of standardization leads to companies needing to define their own set of internal accepted secure coding guidelines. This results in a non-uniform and incoherent selection of SCG across the industry.

To the best of our knowledge, there is no previous work on how to systematically derive and define SCG (e.g. for a given programming language) and on raising awareness about SCG using serious games. In Section III we present a proposal for a possible methodology to derive SCG.

### B. IT Security Awareness Training

Software development in the industry is normally bound to a set of well established and existing programming languages [10]. It has been shown that there isn't really one programming language that is significantly more secure than any

another [11] - vulnerabilities appear across all programming languages.

Therefore, it makes sense to focus efforts on raising awareness of software developers on how to write secure code. According to Benenson [2], awareness can help to improve the understanding of the issues, to better identify the issues and to act accordingly to the issues. Furthermore Graziotin [12] has shown a correlation between developer happiness and source code quality.

One training methodology therefore that seems to be well suited is by using serious games [13], in particular if based on Capture-the-Flag (CTF).

### III. RESEARCH TOPICS

In the previous sections, we have briefly presented the importance of secure coding guidelines both to fulfill industry standards and policies and also as a basis for IT security awareness for software developers. Unfortunately not all programming languages have widely agreed secure coding guidelines, which leads to companies having to define their own. In the following, we propose a method to systematically derive secure coding guidelines. Furthermore, with the goal of raising secure coding awareness we present possible research questions to achieve this goal.

#### A. Systematic Derivation of Secure Coding Guidelines

Given a vulnerability database, such as [14], we propose a systematic method to derive secure coding guidelines comprising the following steps:

- 1) define a business impact metric (BIM) for vulnerabilities
- 2) compute the BIM for all vulnerabilities in the database
- 3) map vulnerabilities and BIM to language-specific rules
- 4) compile the set of rules into secure coding guidelines

The BIM is a company-specific metric which shall represent the perceived negative impact of the exploitation of the given vulnerability. This metric shall be aligned with business objectives and risk appetite [15] and can include parameters such as: impact score (e.g. based on estimated money loss), probability of occurrence, perceived ease of exploitation, etc.

The mapping of vulnerabilities to language-specific rules and constructs shall be done between IT security experts and software developers. At this stage, several language-specific recommendations could result from a single vulnerability. The last step is a codification step, which consolidates and abstracts all the derived recommendations into a catalog of secure coding guidelines.

The main advantage of this method is that, due to the usage of a metric, the resulting secure coding guidelines can be prioritized in terms of business importance. This leads to a natural categorization of the most important guidelines to focus on awareness training programs.

#### B. Secure Coding Awareness for Software Developers

Recently, there has been an increased interest on using serious games [13] to raise IT security awareness e.g. [16, 17, 18].

While the published work until now shows good indicators of the suitability of this approach, it has been (1) focused on a different target group than the one we wish to address, e.g. pentesters or security experts and (2) focused on general IT security awareness, e.g. email and password handling.

However, our target group are software developers for the industry and the content of the training is specific to secure coding. Nevertheless, we also hypothesize that an adapted serious games of the type CTF can also be effectively used to raise secure coding awareness of software developers. Our assumption is based on the positive indicators from similar work, but also on the following facts: (1) participants typically enjoy playing CTF games (Kees et al. [19]) and (2) happy developers write better code (Graziotin et al. [12]).

#### C. Research Questions

This short paper has briefly shown how important secure coding guidelines are for the industry and also for raising software developer awareness on the topic of secure coding. However, it does also raise some further important questions that need additional research. These questions include:

- Q1 What is the current state of usage of SCG across the industry?
- Q2 How can SCG be systematically derived?
- Q3 How to raise awareness about SCG for software developers in the industry by means of CTF serious games?

The first research question  $Q1$ , should allow us to validate the assumption that our reported experience is also shared among the industry. Question  $Q2$  would help in  $Q3$  when secure coding guidelines are missing as input to create a serious game. Due to the derivation of a business metric, it also allows to rank guidelines by importance to business. Motivated by the industry problem exemplified in this work,  $Q3$  tries to address it by means of designing a serious game.

### IV. PRELIMINARY RESULTS AND FUTURE RESEARCH

Currently ongoing investigations, based on a requirements engineering approach, intend to address the questions presented in Section III-C. The result aims at contributing on how to improve IT security awareness, in particular on secure coding topics, of software developers in the industry and, as a consequence, lead to improved quality of products and services.

Preliminary results [20] on the requirements for Capture-the-Flag challenge design give a positive indication that defensive-style game are appropriate for raising awareness about secure coding. Furthermore it confirms the happiness and satisfaction of the participants playing the game. Further preliminary research suggests that the presented methodology to derive secure coding guidelines can indeed be used as input to design defensive challenges and also to plan and prioritize a teaching curriculum.

Investigations which shall address the research questions above and also the architecture of the Capture-the-Flag serious game and player engagement are currently underway.

## REFERENCES

- [1] Siemens AG. The Charter of Trust Takes a Major Step Forward to Advance Cybersecurity. [Online]. Available: <https://www.siemens.com/press/en/feature/2018/corporate/2018-02-cybersecurity.php>
- [2] N. Hänsch and B. Zinaida, "Specifying IT Security Awareness," *25th International Workshop on Database and Expert Systems Applications*, pp. 326–330, 2014.
- [3] "Security for Industrial Automation and Control Systems - Part 4-1: Secure Product Development Lifecycle Requirements," International Electrotechnical Commission, Standard, 01 2018.
- [4] "ISO/IEC 27002:2013. Information Technology – Security Techniques – Code of Practice for Information Security Controls," International Organization for Standardization, Geneva, CH, Standard, 2013.
- [5] N. N. I. of Standards and T. J. T. F. T. Initiative, "Sp 800-39. Managing Information Security Risk: Organization, Mission, and Information System View," Gaithersburg, MD, United States, Tech. Rep., 2011.
- [6] OWASP OWASP Top 10. (17, June 2019). [Online]. Available: [https://www.owasp.org/images/7/72/OWASP\\_Top\\_10-2017\\_\(en\).pdf](https://www.owasp.org/images/7/72/OWASP_Top_10-2017_(en).pdf)
- [7] "Guidelines for the Use of the C Language in Critical Systems," Motor Industry Software Reliability Association, Nuneaton, Warwickshire, UK, Standard, 03 2012.
- [8] S. E. Institute. SEI CERT C Coding Standard. [Online]. Available: <https://wiki.sei.cmu.edu/confluence/display/c/SEI+CERT+C+Coding+Standard>
- [9] ——. SEI CERT Oracle Coding Standard for Java. [Online]. Available: <https://wiki.sei.cmu.edu/confluence/display/java/SEI+CERT+Oracle+Coding+Standard+for+Java>
- [10] T. S. BV. (2019) TIOBE Programming Community Index. Available at <https://www.tiobe.com/tiobe-index/>.
- [11] WhiteSource. What Are the Most Secure Programming Languages? [Online]. Available: <https://www.whitesourcesoftware.com/most-secure-programming-languages/>
- [12] D. Graziotin, F. Fagerholm, X. Wang, and P. Abrahamsson, "What Happens When Software Developers Are (Un)happy," *Journal of Systems and Software*, 2017.
- [13] R. Dörner, S. Göbel, W. Effelsberg, and J. Wiemeyer, *Serious Games: Foundations, Concepts and Practice*. Springer International Publishing, 2016.
- [14] M. Corporation. CVE details. [Online]. Available: <https://www.cvedetails.com/>
- [15] ISACA, *CISM Review Manual, 15th Edition*. Information Systems Audit and Control Association, 2016.
- [16] T. Awojana and T.-S. Chou, "Overview of Learning Cybersecurity Through Game Based Systems," in *2019 CIEC*. New Orleans, LA: Advances in Engineering Education, 2 2019, <https://peer.asee.org/31521>.
- [17] A. Rieb, T. Gurschler, and U. Lechner, "A Gamified Approach to Explore Techniques of Neutralization of Threat Actors in Cybercrime," 06 2017, pp. 87–103.
- [18] A. Rieb, "It-sicherheit: Cyberabwehr mit Hohem Spaßfaktor," *kma - Das Gesundheitswirtschaftsmagazin*, vol. 23, pp. 66–69, 07 2018.
- [19] K. Leune and S. J. P. Jr., "Using Capture-the-flag to Enhance the Effectiveness of Cybersecurity Education," *SIGITE'17*, pp. 47–52, 10 2017.
- [20] T. Gasiba, K. Beckers, S. Suppan, and F. Rezabek, "On the Requirements for Serious Games Geared Towards Software Developers in the Industry," in *submitted for publication: Conference on Requirements Engineering Conference*, 2019.