Taking the Middle Path: Learning about Security through Online Social Interaction

Tamara Lopez, Thein T. Thun, Arosha Bandara, Mark Levine, Bashar Nuseibeh and Helen Sharp

Abstract— As software-intensive digital systems become an integral part of modern life, ensuring that these systems are developed to satisfy security and privacy requirements is an increasingly important societal concern. Integrating security into software development involves more than learning security principles or applying techniques. Security in practice is shaped through experience. It can be integrated into software development by following a middle path, through which developers draw together formal knowledge and software development techniques. Social interactions facilitate this process. This article recommends four strategies developers can use to maximise security in practice using online communities like Stack Overflow, including approaching security from within specific tasks, critically assessing content in posts, actively participating, and bringing online information into real-world situations.

Index Terms—social learning techniques, support for security, software construction

1 Introduction

Fig 1. *Secure coding is shaped by experience, integrated into software development through a* middle path, *by which developers draw formal knowledge and software development techniques into practices that can be applied.* (cf. the bee in Bacon's *Novum Organum, I.95, trans. 1858.* https://archive.org/details/workscollecteda16spedgoog/page/n104).

Software-intensive systems are now an integral part of many aspects of daily life. Worries about how to secure these systems are growing, obligating governments to extend the provision of security for their citizens to include cybersecurity [1]. The potential personal, reputational and monetary costs of security breaches are high. To counter this, regulations such as the General Data Protection Regulation (GDPR) in Europe impose large financial penalties for breaches. In short, security has become a primary concern when developing systems that collect, process and store personal data. To address the need for security in software, there is a growing list of cybersecurity tools, guidance, training materials and case studies. Unfortunately, the number of breaches seems to be undiminished. Indeed, many recent breaches, such as those experienced by Equifax [2], exploit known vulnerabilities in software systems.

In actions taken at the desk, developers are responsible for introducing and maintaining security mechanisms in the code they write [3]. The

- *Tamara Lopez is with the School of Computing and Communications, The Open University, Milton Keynes, UK. E-mail: tamara.lopez@open.ac.uk.*
- *Thein T. Tun is with the School of Computing and Communications, The Open University, Milton Keynes, UK. E-mail: thein.tun@open.ac.uk.*
- *Arosha Bandara is with the School of Computing and Communications, The Open University, Milton Keynes, UK. E-mail: arosha.bandara@open.ac.uk.*
- *Mark Levine is with the University of Exeter, Exeter, UK. E-mail: M.Levine@exeter.ac.uk.*
- *Bashar Nuseibeh is with the School of Computing and Communications, The Open University, Milton Keynes, UK. E-mail: bashar.nuseibeh@open.ac.uk.*
- *Helen Sharp is with the School of Computing and Communications, The Open University, Milton Keynes, Uk. E-mail: helen.sharp@open.ac.uk.*

xxxx-xxxx/0x/$xx.00 © 200x IEEE Published by the IEEE Computer Society

expectation is that developers will consider security while undertaking tasks at all stages of the software development lifecycle.

However, "good security" involves more than a set of principles or formal techniques. Developers must also understand how particular techniques provide assurances that the software is secure, and meet policies within their companies about what is needed to be secure [3]. Everyday security in software development is technical, but it is also social, built upon relationships between people that engender trust [4]. Peer interaction, experience of security failures, and an awareness about the impact of security failures on people's well-being influence the decisions individuals make about whether or not to be secure on the job [5].

Our larger research program is investigating ways to initiate and sustain secure software development practice. In a series of studies, we have shown how developers talk about security in Stack Overflow posts given the "security" tag [7][8], and examined how security figures into practice within office settings [9]. The approach taken in these studies does not assess the quality of the information about security developers provide to one another, but rather seeks to understand more about how they engage with and guide one another in practice.

Based on our findings, this article recommends four strategies to maximise developers' learning within online communities. To do this, we recommend they follow a ***middle path***, using social interactions to draw together formal security principles and software development techniques. In this way, developers can bolster experience, solidifying practices that can be applied at the desk when circumstances require.

2 Security practice in Professional Software Development Environments

Within software engineering, security is commonly considered in terms of a particular mindset that encourages developers to think like attackers, imagining how a system might be exploited [10], and to acknowledge and address vulnerabilities within their code. However, many software developers *do not* consistently and accurately make use of the established security practices and technologies that are available. Studies indicate that this may be because other pressures and demands in the workplace push security down the list of priorities [11], or that developers have a lack of knowledge or awareness [12]. In fact, although advice given to developers is abundant and there are many tools available for making code secure, the tools can be difficult to use, and the security messages and warnings can overwhelm developers [12].

Our interviews taken with engineers in office environments suggest that developers who are not in specialist cyber security teams often understand, in principle, how to counter common vulnerabilities, and have an awareness of the need to protect information for companies and for users of software [9]. But at the same time, and in agreement with other findings [11], these developers report that security is not at the forefront of their daily activity or decision making. Security-related activity within code is primarily driven by business or regulatory requirements and developers often rely on these events to bring security to the forefront of attention.

Security among these engineers is not upheld by individuals who champion secure practices at every step of the software lifecycle, but is instead sustained through relational practices [4] that form a collective responsibility. Developers trust in the security frameworks and infrastructure put in place by others in their company. They rely upon guidance from team members to gain experience in applying security techniques within their own code and local environments.

Contrary to suggestions that developers are inattentive, or lack knowledge [12], our findings suggest that developers possess a degree of awareness and knowledge, but may not have control over making security decisions or prioritising security activity in their environments. As a result, they may also lack opportunities to develop the experiential knowledge needed to apply security principles or techniques within daily practice.

One way that developers augment their own experience is through social connections made with other practitioners [6]. Our studies have identified two features of software development culture that support learning about security: everyday conversation, and personal networks of practice.

2.1 Everyday Conversation

Developers today, like much of society, are often active participants in online development communities, and supported by a range of communication tools. Software development has always been social, embedded within a rich community and culture of practice [6]. Conversation is integral to this culture. Developers share stories with one another, but also construct narratives in the midst of tasks. This kind of verbal exchange develops confidence, provides a space to trade information and a springboard for learning. It is, in part, through these everyday social interactions that materials gathered through experience, and knowledge gained in formal training and education, are transformed into practices that can be used [13].

2.2 Networks of Practice

The interactions and conversations we have observed within office settings [9] and comment streams on Stack Overflow [7], [8] suggest that secure coding practice is supported in both kinds of environment through personal networks of practice [14] that operate *within* larger environments. Online, in websites like Stack Overflow, such networks operate within the comment streams attached to question and answer posts. Within office environments, the networks operate within the structure of the larger software engineering department. In both cases, the networks are maintained through personal connections [14].

Technical and theoretical information about security is shared in these connections through focused exchanges made between individuals. Likewise, it is within these exchanges that developers have an opportunity to make statements about security that reflect personal values and attitudes like responsibility, trust, and fear [7]. All three elements are key to promoting and supporting secure coding practice.

In the next section, recommendations for developers suggest how to get the most from online sources to learn about security.

3 How To Learn About Security Online

Stack Overflow is one question and answer website in which developers can ask questions about programming problems they are solving, and get answers. In operation since 2008, the website has grown into a definitive *community of practice* [14], a social space in which participants form a partnership around the shared need to solve programming problems.

This section synthesises observations from our studies into a series of recommendations to help professionals leverage online communities like Stack Overflow to develop their own network of practice for learning about security. Though the recommendations reflect patterns of use that were observed among users of that site, they may also apply in other discussion-based online environments.

3.1 Start with a specific programming task

Developers can effectively approach the topic of security in online sites within the context of specific tasks they need to complete. Our findings suggest that developers on Stack Overflow do this, expanding their security awareness and understanding incrementally, by exploring implications in the context of technologies and skills that are familiar. For example, a developer may use a general question about secure password storage as a space to examine different features of a language API. In asking a follow-up question about the API, the developer is able to learn more about how the language works, and at the same time gathers information about secure information storage.

3.2 Look for tended posts and comment streams

Security is dynamic. It takes time to learn and understand good practices, and the shifting landscape of threats requires ongoing attention to ensure that mechanisms remain effective and up-to-

date. Security tagged posts on Stack Overflow reflect this, and often remain active for months or even years after an answer is accepted [8]. Ongoing activity may be primarily curatorial—links might be kept up-to-date or added to dictionary entries or other documents, or the language of the question and answer posts might be refined for clarity.

However, in other cases, developers collectively ***tend to*** *the content within* posts and comment streams. Answers are developed over time to include different scenarios of application, to consider new developments around an issue, or to develop the argument for an answer. Changes to answer posts may also integrate information from comments, suggesting that interactions are producing new, relevant information or a new perspective.

The comment streams for tended threads in security tagged posts exhibit several distinct characteristics.

1. ***Information trading.*** Tended threads show evidence that information is freely traded even after significant gaps in time. These exchanges trade "small" pieces of information that serve an immediate need for the participants.
2. ***Broadcasts*** situate security problems in time, giving updates about technologies or libraries or software company activity. Sometimes this information is added to answer or question posts, but in other instances, the comment thread becomes the information store.
3. ***Related Works*.** Explanations are supported within exchanges using links to other information, including related answers, articles and documentation.

3.3 Join in by lending a hand or asking for help

Personal connections support secure coding practice. Upvotes are not enough. Developers in the field [9] indicate that while community voting activity in online sources is helpful for isolating promising information, subsequent problem-solving of security problems often requires additional support in the form of personal connections made between individuals.

Security problems, like other problem solving that developers undertake at the desk reflect individual needs in the moment that are shaped by personal knowledge, the context of work, and the technologies at play [16]. Support is given within comments written by the author of accepted answers, and by other users that have particular knowledge of a language or technical aspect of security.

Developers can use exchanges to:

1. **Give and receive focused, non-judgemental assistance**. Interactions can be used to provide or ask for information, clarification or corrections and confirm understanding.
2. **Associate technology facts with security problems.** This linking is often material, for example in associating small details about how a language works with an equally small feature of security.
3. **Situate technical advice in the broader security landscape**, either using anecdotes to explain how attackers use particular technologies, or by explaining broader attack scenarios.

3.4 Bring online security conversations into local practice

Social interactions online about security can be used to strengthen local networks of practice. Developers indicate that they prefer in the first instance to draw on the support of colleagues before turning to online sources, but problem solving often involves an interplay of online and real-world interactions. Though studies have shown that a high proportion of vulnerabilities are introduced to software from code snippets that are copied and pasted from internet sites [15], developers observed in other contexts indicate that it is often not possible to use solutions found online in this way. It is more common for the information found online, including code snippets, to require moulding to the local software environment, to the functional requirements, or to the particular requirements for security in a given context [16]. This moulding process is often also socially mediated, as information found online is shared with colleagues, talked over and assessed.

4 Conclusion

Developers can actively increase their experience with security by expanding their support network within online discussions. In writing and developing answer posts, and dropping in on comment streams to share information and receive help, developers foster and promote security as a part of daily practice. Informal learning through social interaction is the ***middle path*** for security practice, providing a way for developers to give practical shape to security knowledge, enabling it to be called to mind and applied as circumstances require.

As a forum for exchanging information, sites like Stack Overflow are relevant resources for supporting informal learning, allowing developers to thoughtfully connect with and tend to security problems. In so doing, they are also able to develop cababilities for writing secure code that include new skills and knowledge, but also new perspectives and ideas about how to go about securing software.

Acknowledgment

We thank the users of Stack Overflow, and the developers who have participated in our field studies. Supported by the National Cyber Security Centre (NCSC). Additional support provided by SFI and EPSRC.

References

[1] Cyber Resilience: Playbook for Public- Private Collaboration," World Economic Forum, Tech. Rep. [Online]. Available: http://reports.weforum.org/cyber-resilience/

[2] H. Berghel, "Equifax and the Latest Round of Identity Theft Roulette," *Computer*, vol. 50, no. 12, pp. 72–76, Dec. 2017.

[3] R. Anderson, "Chapter 1. What is Security Engineering?" in *Security engineering*. John Wiley & Sons, pp. 3-15, 2008.

[4] L. Coles-Kemp, and R.R. Hansen, "Walking the Line: The Everyday Security Ties that Bind," In T. Tryfonas (Ed.), *Human Aspects of Information Security, Privacy and Trust* , pp. 464–480, . Springer International Publishing, 2017.

[5] C. Posey, T. L. Roberts, P. B. Lowry, and R. T. Hightower, "Bridging the divide: A qualitative comparison of information security thought patterns between information security professionals and ordinary organizational insiders," *Information & Management*, vol. 51, no. 5, pp. 551–567, Jul.2014.

[6] H. Sharp, H. Robinson, and M. Woodman, "Software engineering: Community and culture," *IEEE Software*, vol. 17, no. 1, pp. 40–47, Jan.2000.

[7] T. Lopez, T.T. Tun, A. Bandara, B. Nuseibeh, H. Sharp, and M. Levine (2018). "An Investigation of Security Conversations in Stack Overflow: Perceptions of Security and Community Involvement." *1st International Workshop on Security Awareness from Design to Deployment, International Conference of Software Engineering*, 2018. Gothenburg, Sweden, 27 May, 2018.

[8] T. Lopez, T.T. Tun, A. Bandara, B. Nuseibeh, H. Sharp, and M. Levine (2019) "An Anatomy of Security Conversations in Stack Overflow." *Software Engineering in Society, International Conference of Software Engineering,* 2019. Montréal, Canada, May 25 - June 1, 2019.

[9] T. Lopez, H. Sharp, T.T. Tun, A. Bandara , M. Levine, and B. Nuseibeh. (2019) 'Hopefully We Are Mostly Secure': Views on Secure Code in Professional Practice', *12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE), International Conference of Software Engineering, 2019*. Montréal, Canada, May 27, 2019.

[10] "The Security Mindset - Schneier on Security." [Online]. Available: https://www.schneier.com/blog/archives/2008/03/the security mi 1.html

[11] H. Assal and S. Chiasson, "Security in the software development lifecycle," in *Fourteenth Symposium on Usable Privacy and Security ({SOUPS} 2018)*, 2018, pp. 281–296.

[12] Y. Acar, S. Fahl, and M. L. Mazurek, "You are not your developer, either: A research agenda for usable security and privacy research beyond end users," in *Cybersecurity Development (SecDev), IEEE*. IEEE, 2016, pp. 3–8.

[13] M. Eraut, "Informal learning in the workplace," *Studies in Continuing Education*, vol. 26, no. 2, pp. 247–273, Jul. 2004.

[14] E. Wenger, B. Traynor, and M. de Laat, "Promoting and assessing value creation in communities and networks: a conceptual framework," Open University of the Netherlands, Ruud de Moor Centrum, Rapport 18, 2011.

[15] Y. Acar, M. Backes, S. Fahl, D. Kim, M. L. Mazurek, and C. Stransky, "You Get Where You're Looking For: The Impact Of Information Sources on Code Security," in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 289–305.

[16] T Lopez, "Error Detection and Recovery in Software Development." PhD Thesis, the Open University, 2016.