Motivations and Amotivations for Software Security

Preliminary Results

Hala Assal
School of Computer Science
Carleton University
Ottawa, ON, Canada
HalaAssal@scs.carleton.ca

Sonia Chiasson
School of Computer Science
Carleton University
Ottawa, ON, Canada
chiasson@scs.carleton.ca

ABSTRACT

We conducted an interview study with software developers to explore factors influencing their motivation towards security. We identified both *intrinsic* and *extrinsic* motivations, as well as different factors that led participants to lack motivation towards software security. We found that when the motivation stems from the developer, rather than external factors, our participants exhibited better attitudes towards software security. We discuss each of the identified (a)motivations and the importance of transforming motivations to be internally-driven by developers.

1. INTRODUCTION

Developers are not necessarily security experts, however, end-users expect them to develop secure applications. Security initiatives and tools have been proposed to support the integration of security in the Software Development Lifecycle (SDLC) (*e.g.* [4, 7, 9, 22]). Despite these efforts, vulnerabilities persist [16] and even extend to applications that were not considered security-sensitive [14, 17]. Conflicting opinions argue the reason might be because security guidelines do not exist or are not mandated by the companies [29, 32, 33], that developers lack security knowledge [5, 31], or that developers might lack the ability [16] or proper expertise [6] to identify vulnerabilities despite having security knowledge.

Recently, usable security research focused on developers and the human factors of software security [13]. For example, Acar *et al.* [2] developed a research agenda that focuses on proposing and improving security tools and methodologies, as well as understanding how developers' view and deal with software security. Our previous work [3] explored software security practices in real-life and identified factors that may influence these practices. In this paper, we explore factors that motivate (or deter) developers from addressing software security. One of the problematic properties of security is *the unmotivated user property* [28]. This concept also applies to software developers—security is rarely their primary objective [2, 13]. From our study, we also found several factors that may induce developers' amotivation towards security, despite their knowledge and belief of its importance. Thus, besides supporting developers technically with security tools and libraries, our data shows the importance of internalizing software security and acting with volition towards it.

Copyright is held by the author/owner. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee.
USENIX Symposium on Usable Privacy and Security (SOUPS) 2018.
August 12–14, 2018, Baltimore, MD, USA.

2. RELATED WORK

Ensuring application security is not a trivial task, especially for developers who are often mistaken as security experts [2, 13]. Different approaches have been proposed to support developers in their security tasks, including using machine learning to assist in the discovery of vulnerabilities [15, 25], supporting developers' information needs during vulnerability analysis [22], and improving the usability of Application Programming Interfaces (APIs) [1, 31].

Baca *et al.* [6] suggested that to achieve best results in analyzing security vulnerabilities, developers need to gain practical experience in using Static-code Analysis Tools (SATs) with a focus on security aspects. Oliveira *et al.* [16] recommended in-context security education. Thomas *et al.* [24] also recommends providing training opportunities that target the specific security issues that developers encounter in their code, and tailoring security training to address developers' weak security knowledge spots.

Previous work investigated factors that influence the adoption of new security tools [29, 32, 33]. The development company's policies and the overall company culture towards security were found to be among the main deciding factors in motivating security in development and encouraging developers' decision to adopt new security tools [32, 33]. The domain and context of use of the application was another prominent factor in adopting security tools [32, 33]. In addition, some developers are reluctant to use security tools because they are complex and require special security knowledge [29, 32]. For many, installing and learning how to use and interpret the output of a new tool was too steep a cost that sometimes outweighs the benefits [29]. In addition, through a survey with information system professionals (*e.g.*, developers, analysts, managers), Woon and Kankanhalli [30] found that participants' perception of the usefulness of security to their applications influences their intentions to practice secure development.

This work focuses on factors that influence the adoption of security processes in general. In this paper, we explore developers' motivations and amotivations to software security.

3. METHODOLOGY

We designed a semi-structured interview study to explore developers' motivation to software security and received IRB

clearance. We recruited 13 participants through posting on development forums and relevant social media groups, and announcing the study to professional acquaintances. Each participant received a $20 Amazon gift card as compensation. Each interview lasted approximately one hour, was audio recorded, and later transcribed for analysis. Data collection was done in 3 waves, each followed by preliminary analysis and preliminary conclusions [12]. We followed Glaser and Strauss's [12] recommendation by concluding recruitment on saturation (*i.e.*, when new data collection does not add new themes or insights to the analysis). Other aspects of these interviews were published separately [3], but this paper focuses on different analysis.

All participants hold university degrees which included courses in software programming, and are currently employed in development with an average of 9.35 years of development experience (*median* = 8). Our dataset included participants developing different application types: web applications and services (*e.g.*, e-finance, online productivity, online booking, and social networking), embedded software, kernels, design and engineering software, support utilities, and information management and support systems.

Analysis. We used Grounded Theory methodology [23] to analyze our interviews. We did not start with a preconceived theory, rather we started by exploring the data to offer insights and enhance understanding of the phenomenon under study. We used Atlas.ti to code our interviews, which resulted in a total of 170 open codes.

4. RESULTS

We identified *both* intrinsic and extrinsic motivations to software security. Intrinsic motivation is when an activity is voluntarily performed for the pleasure and enjoyment it causes. Intrinsic motivations are driven by humans' "inherent tendency to seek out novelty and challenges, to extend and exercise one's capacities, to explore, and to learn." [19]. In contrast, extrinsic motivation is when a person is engaged in an activity for outcomes separate from those innate to the activity itself [10]. In addition, we identified amotivations to software security, where the person lacks motivation to act (they do not act at all or act without intent) [19].

4.1 Motivations to software security

Intrinsic motivation. The only intrinsic motivation to security in our data was "*self-improvement*", where the developer challenges one-self to write secure code. For example, P1 said, *"Sometimes I will have the challenge, that 'okay, this time I'm going to submit [my code] for a review where nobody will give me a comment'."*

On the other hand, we found several extrinsic motivations to software security that vary in the degree of autonomy—whether the motivation is internal to the developer, or driven by external factors [10].

Internally-driven extrinsic motivation. *Professional responsibility* and *concern for users* are two extrinsic motivations, where the action is not performed for its inherent enjoyment, but rather to fulfill what the developer views as their responsibility to their profession and to safeguard users' privacy and security. For example, P3 said, *"I would not feel comfortable with basically having something used by end users that I didn't feel was secure, or I didn't feel respective of privacy, umm so I would try very hard to not compromise on that."*

In addition, our analysis shows that *understanding the implications* of ignoring or dismissing security, increased security awareness and motivated participants' teams to integrate security in their SDLCs. This was especially true when the understanding came through practical examples of how the developer's code could lead to a security issue or through experiencing a real security issue at work. P4 explained, *"I know for me personally when I realized just how catastrophic something could be, just by making a simple mistake, or not even a simple mistake, just overlooking something simple, it changes your focus."*

Caring about the *company reputation* and recognizing how it could be negatively affected in case of a security breach is another motivator. Moreover, when the whole project team is responsible for security, as opposed to singling out a specific entity, our participants recognized that as part of the team they should participate in this *shared responsibility*. P10 explained, *"[If we find a vulnerability,] we try not to say, 'you personally are responsible for causing this vulnerability'. I mean, it's a team effort, people looked at that code and they passed on it too, then it's shared, really."* We found evidence in our data that this behaviour could have a snowball effect and lead to motivating more team-members to recognize the importance of considering security as their colleagues do (*induced initiative*). P7 said, *"When you see your colleagues actually spending time on something, you might think that 'well, it's something that's worth spending time on', but if you worked in a company that nobody just touches security then you might not be motivated that much."*

Externally-driven extrinsic motivation. We identified security motivations that are driven by external factors, such as receiving rewards and avoiding punishment. Our analysis shows that addressing security can be driven by the desire to being recognized as the security expert or receiving acknowledgement, or maintaining self-esteem and self-worth (*prestige*). In addition, receiving rewards in the form of *career advancement* is another external motivation for security. We also found three motivations that are driven by the desire to avoid negative consequences of the lack of security: an overseeing entity finding non-compliance with regulations (*audit fear*), losing marketshare or market value in case of a security breach (*business loss*), and being monitored and pressured by superiors (*pressure*). P1 explained, *"If they find a security issue, then you will be in trouble. Everybody will be at your back, and you have to fix it as soon as possible."*

4.2 Software security amotivations

We also explored amotivations for software security; why security is deferred or dismissed.

Perceived lack of competence. Our analysis revealed that the *lack of resources* and the *lack of support* are two factors that led to a perceived lack of competence to address software security. Some participants indicated that they do not have the necessary budget, time, people-power, or expertise, to properly address security in their SDLC. For example, P12 said, *"We don't have that much manpower to explicitly test security vulnerabilities, [..] we don't have those kind of resources. But ideally if we did have [a big] company, I would have a team dedicated to find exploits. But unfortu-*

nately we don't. We also found that this lack of trust in their ability to address security occurs when there is no security plan in place, when security tools are nonexistent or lacking, and when developers are unaware of such tools' availability.

Lack of interest, relevance, or value. The other type of amotivation comes from the lack of interest, relevance, or value of performing security tasks. The lack of relevance could happen when security is not considered one of the developer's everyday duties (*not my responsibility*), or when security is viewed as another entity's responsibility (*security is handled elsewhere*), such as another team or teammember. Our analysis shows that when this is the general attitude in a team, it could have detrimental effects such as *induced passiveness*. For example, even though P9 believed in the importance of addressing security, he became amotivated towards it and prefers to focus on his 'more valuable' existing duties. He said, *"I don't really trust [my team members] to run any kind of, like, source code scanners or anything like that. I know I'm certainly not going to."*

Additionally, our analysis revealed reasons why security efforts lack value for some teams as indicated by participants in our dataset. First, we found that some of our teams suffer from the optimistic bias [18, 27], thinking that attackers would not be interested in their applications, or that they are not a big enough company to be a target for attacks. Thus, as they see *no perceived risk* and security efforts lack value. We also found that when there are no perceived negative consequence to the individuals or to the business from the lack of security in the SDLC (*no perceived loss*), then security efforts lack value. For example, when developer are not held responsible for security issues found in their code, they would rather spend their time on aspects for which they will be held responsible. P7 explained, *"[If] I made a bad security decision, nobody would blame me as much as if I made a decision that lead to a [non-security] bug in the system. So the priority of security is definitely lower than introducing bugs in the system."* Moreover, as different tasks compete for resources (the developer's time in the previous quote), when security has no perceived value, those deemed more valuable are prioritized.

Defiance/Resistance to influence. The final amotivation we identified is *inflexibility*. We found that some developers would work around security, not because it is difficult to comply, but rather because it conflicts with their perception of the proper way of coding, or it conflict with how they are used to writing code. P9 explained how one of his team members resists using a framework in the proper way, despite having *"gotten into so many arguments"* (P9) with his manager, *"I can tell he is very self-absorbed with his own thoughts, and he thinks that what he says is somehow the truth, even if it doesn't necessarily pan out that way".*

5. DISCUSSION

Several factors lead to the types of amotivation identified in our data, such as the optimism bias—thinking that the applications are safe from the adverse consequences of lack of security. It could also arise from workplace dynamics, *e.g.*, in a team where secure coding and security tasks are resisted, a developer may feel that her efforts towards software security alienates her from the team, and with no expectation of reward, she may lose motivation to go the extra-mile. It could also induce a feeling of helplessness [26] based on disbelief that her focus on security could change the course of events, given that the majority of the application was not built with security in mind.

On the other hand, in teams where security was in the company culture and support for security tasks was available, developers were more motivated to focus on software security. This could be because they feel competent to perform their security tasks, especially when support for such tasks and learning opportunities are available. It could also be because, in such teams, secure coding behaviour increases the developers' relatedness to their teams, *e.g.*, by feeling they are connected to the culture and contribute to the team. Consequently, rather than performing security tasks purely to follow mandates, developer internalize such tasks, accept them, and experience volition to act.

In fact, we found that even in cases where security tasks were mandatory, motivation to act often arises from reasons other than the mandate. Although it may be a first step to motivate security, mandating security tasks should be accompanied by improving the morale when it comes to security through adopting a security culture, supporting developers in these tasks and providing positive encouragement, and allowing developers and teams to see the value of such tasks and identify with them. This facilitates internalization of security, which has a significant positive effect on persistence and performance [19].

6. CONCLUDING REMARKS

Finding the best way to motivate developers is not a trivial task. External rewards and punishment may help induce external motivation. However, previous research found that these have a detrimental effect on intrinsic motivation as it shifts the perceived locus of causality from internal to external. In addition, research in the education domain found that tangible rewards negatively influence conceptual learning and problem solving [11]. Other research hypothesizes that engagement-contingent and non-tangible rewards may avoid the externalization of intrinsic motivations [8, 19]. However, research in this area is inconclusive [11]. Moreover, the effect of reward (or punishment) contingencies on internalizing and accepting activities is unclear [21].

With all these uncertainties and potential negative effects that reward contingencies may have on motivation and performance, we highlight the need for future research focusing on the long-term effect of reward (and punishment) contingencies on intrinsic motivation and the internalization of software security. In addition, research recommendations for incentivizing developers through tangible rewards should be re-assessed based on their long-term effects.

Previous research demonstrated the importance of internalizing external motivation as it leads to improved performance and the ability to learn [19, 20]. As a continuation to the work presented herein, we will focus our analysis on the process of internalizing software security to understand factors that influence developers' internalization of software security activities and how it can be supported.

7. ACKNOWLEDGMENTS

H. Assal acknowledges her NSERC Postgraduate Scholarship (PGS-D). S. Chiasson acknowledges funding from NSERC for her Canada Research Chair and Discovery grants.

8. REFERENCES

[1] Y. Acar, M. Backes, S. Fahl, S. Garfinkel, D. Kim, M. L. Mazurek, and C. Stransky. Comparing the Usability of Cryptographic APIs. In *IEEE Symposium on Security and Privacy*, 2017.

[2] 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 *IEEE Cybersecurity Development*, 2016.

[3] H. Assal and S. Chiasson. Security in the Software Development Lifecycle. In *Symp. on Usable Privacy and Security (SOUPS)*. USENIX Association, 2018.

[4] H. Assal, S. Chiasson, and R. Biddle. Cesar: Visual representation of source code vulnerabilities. In *IEEE Symp. on Visualization for Cyber Security*, 2016.

[5] B. K. Marshall. Passwords Found in the Wild for January 2013. `http://blog.passwordresearch.com/2013/02/`. [Accessed April-2017].

[6] D. Baca, K. Petersen, B. Carlsson, and L. Lundberg. Static Code Analysis to Detect Software Security Vulnerabilities - Does Experience Matter? In *Int. Conf. on Availability, Reliability and Security*, 2009.

[7] M. Backes, K. Rieck, M. Skoruppa, B. Stock, and F. Yamaguchi. Efficient and Flexible Discovery of PHP Application Vulnerabilities. In *IEEE European Symp. on Security and Privacy*, 2017.

[8] G. G. Bear, J. C. Slaughter, L. S. Mantz, and E. Farley-Ripple. Rewards, praise, and punitive consequences: Relations with intrinsic and extrinsic motivation. *Teaching and Teacher Education*, 65, 2017.

[9] B. Chess and G. McGraw. Static Analysis for Security. *IEEE Security & Privacy*, 2(6):76–79, 2004.

[10] E. Deci and R. M. Ryan. *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer US, 1 edition, 1985.

[11] M. Gagné and E. L. Deci. Self-determination theory and work motivation. *Journal of Organizational Behavior*, 26(4).

[12] B. G. Glaser and A. L. Strauss. *The discovery of grounded theory: strategies for qualitative research*. Aldine, 1967.

[13] M. Green and M. Smith. Developers are Not the Enemy!: The Need for Usable Security APIs. *IEEE Security Privacy*, 14(5), 2016.

[14] A. Greenberg. Hackers Remotely Kill a Jeep on the Highway—With Me in It. `https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/`, 2015. [Accessed May-2017].

[15] G. Grieco, G. L. Grinblat, L. Uzal, S. Rawat, J. Feist, and L. Mounier. Toward Large-Scale Vulnerability Discovery Using Machine Learning. In *ACM Conf. on Data and Application Security and Privacy*, 2016.

[16] D. Oliveira, M. Rosenthal, N. Morin, K.-C. Yeh, J. Cappos, and Y. Zhuang. It's the Psychology Stupid: How Heuristics Explain Software Vulnerabilities and How Priming Can Illuminate Developer's Blind Spots. In *Annual Computer Security Applications Conf.*, 2014.

[17] J. Radcliffe. Hacking Medical Devices for Fun and Insulin: Breaking the Human SCADA System. `https://media.blackhat.com/bh-us-11/Radcliffe/BH_US_11_Radcliffe_Hacking_Medical_Devices_WP.pdf`, 2011. [Accessed Feb-2017].

[18] H.-S. Rhee, Y. U. Ryu, and C.-T. Kim. Unrealistic optimism on information security management. *Computers & Security*, 31(2):221–232, 2012.

[19] R. M. Ryan and E. L. Deci. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1):68, 2000.

[20] R. M. Ryan and E. L. Deci. *Self-determination theory: Basic psychological needs in motivation, development, and wellness*. Guilford Publications, 2017.

[21] M. Selart, T. Nordström, B. Kuvaas, and K. Takemura. Effects of Reward on Self-regulation, Intrinsic Motivation and Creativity. *Scandinavian Journal of Educational Research*, 52(5):439–458, 2008.

[22] J. Smith, B. Johnson, E. Murphy-Hill, B. Chu, and H. R. Lipford. Questions Developers Ask While Diagnosing Potential Security Vulnerabilities with Static Analysis. In *Joint Meeting on Foundations of Software Engineering*. ACM, 2015.

[23] A. L. Strauss and J. M. Corbin. *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, Inc., 1998.

[24] T. W. Thomas, M. Tabassum, B. Chu, and H. Lipford. Security During Application Development: An Application Security Expert Perspective. In *CHI Conf. on Human Factors in Computing Systems*, 2018.

[25] O. Tripp, S. Guarnieri, M. Pistoia, and A. Aravkin. ALETHEIA: Improving the Usability of Static Security Analysis. In *ACM SIGSAC Conf. on Computer and Communications Security*, 2014.

[26] R. J. Vallerand and R. Blssonnette. Intrinsic, Extrinsic, and Amotivational Styles as Predictors of Behavior: A Prospective Study. *Journal of Personality*, 60(3):599–620.

[27] N. D. Weinstein and W. M. Klein. Unrealistic Optimism: Present and Future. *Journal of Social and Clinical Psychology*, 15(1):1–8, 2017/08/12 1996.

[28] A. Whitten and J. D. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *USENIX Security Symposium*, volume 348, 1999.

[29] J. Witschey, S. Xiao, and E. Murphy-Hill. Technical and Personal Factors Influencing Developers' Adoption of Security Tools. In *ACM Workshop on Security Information Workers*, 2014.

[30] I. M. Woon and A. Kankanhalli. Investigation of IS professionals' intention to practise secure development of applications. *Int. Journal of Human-Computer Studies*, 2007.

[31] G. Wurster and P. C. van Oorschot. The Developer is the Enemy. In *New Security Paradigms Workshop*. ACM, 2008.

[32] S. Xiao, J. Witschey, and E. Murphy-Hill. Social Influences on Secure Development Tool Adoption: Why Security Tools Spread. In *ACM Conf. on Computer Supported Cooperative Work & Social Computing*, 2014.

[33] J. Xie, H. R. Lipford, and B. Chu. Why do programmers make security errors? In *IEEE Symp. on Visual Languages and Human-Centric Computing*, 2011.