The impact of surface features on choice of (in)secure answers by Stackoverflow readers

Dirk van der Linden, Emma Williams, Joseph Hallett, and Awais Rashid

Abstract—Existing research has shown that developers will use StackOverflow to answer programming questions: but what draws them to one particular answer over any other? The choice of answer they select can mean the difference between a secure application and insecure one, as the quality of supposedly secure answers can vary. Prior work has studied people posting on Stack Overflow—a two-way communication between the original poster and the Stack Overflow community. Instead, we study the situation of one-way communication, where people only read a Stack Overflow thread without being actively involved in it, sometimes long after a thread has closed. We report on a mixed-method study including a controlled between-groups experiment and qualitative analysis of participants' rationale (N=1188), investigating whether explanation detail, answer scoring, accepted answer marks, as well as the security of the code snippet itself affect the answers participants accept. Our findings indicate that explanation detail affects what answers participants reading a thread select ($p<0.01$), while answer score and acceptance do not ($p>0.05$)—the *inverse* of what research has shown for those asking and answering questions. The qualitative analysis of participants' rationale further explains how several cognitive biases underpin these findings. Correspondence bias, in particular, plays an important role in instilling readers with a false sense of confidence in an answer through the *way it looks*, regardless of whether it works, is secure, or if the community agrees with it. As a result, we argue that StackOverflow's use as a knowledge base by people not actively involved in threads—when there is only one-way-communication—may inadvertently contribute to the spread of insecure code, as the community's voting mechanisms hold little power to deter them from answers.

Index Terms—Software Security; Stack Overflow; Human Factors; Rationale

1 Introduction

Copying and pasting code snippets from Stack Overflow is a well known, widespread, phenomenon among software developers [1]. Developers often copy-paste snippets without realizing the impact on security [2], [3]. This leads to rapidly spreading [4] less secure code [5], [6], [7] and inhibits developers' security thinking [8].

It is well understood why the original poster of a question on Stack Overflow selects the answer they do (cf. [9]) – being largely driven by answer scores to the extent of disregarding critical assessment of the security of a code snippet. But what of the much wider group of readers who use Stack Overflow as a knowledge base: discovering questions and answers long after their threads have gone silent? Prior work has focused on the posters actively engaged in two-way communication—the question askers and answerers—we instead ask *what drives Stack Overflow readers, who thus engage in one-way communication, to chose one answer over another in completed threads,* and, *to what extent can this be potentially manipulated by unscrupulous posters?*

Hence, we contribute the first large scale experimental work from the point of view of Stack Overflow *readers*. We pinpoint the differences in factors, compared to *posters*, that affect *reader's* choices of an answer and to what extent, if at all, security plays a part in that choice. From hereon we will refer to these two types of Stack Overflow demographics as 'readers' and 'posters'.

To understand what drives someone to chose an answer, we need to first understand how they evaluate an answer. Such evaluation operates as a heuristic information process [10], [11], involving multiple potentially concurrent cognitive strategies [12]. In particular, depending on the person, different *features* of the information are assessed in different order. People might first focus on *semantic* features of the information, assessing its correctness – looking directly at the code snippet and evaluating it, *if* they have the relevant domain knowledge. If they do not, they will resort to judging *surface* features, looking at how it is presented – its style and the explanation that accompanies it. They may further assess its *source* features by looking where it came from – who wrote the answer, and how the community has interacted with it.

We study what features affect Stack Overflow readers when selecting an answer, and with what rationale do they do so, by exploring the following research questions:

1) Do Stack Overflow readers select answers based on security of code snippets?
2) What features (semantic, surface, and source), if any, affect Stack Overflow readers' selection of answers?
3) What explanations underlie readers' answer choices?

To do so, we performed a between-group experiment (N=1188). We presented participants with two answers in the style of Stack Overflow (one secure, one insecure) varying *semantic* features (the code snippet's security, i.e., does it create a vulnerability), *surface* features (the level of detail each answer contained), and *source* features (the answer

- *D. van der Linden, J. Hallett, and A. Rashid are with the Bristol Cyber Security Group, University of Bristol, Bristol, UK. E-mail: dirk.vanderlinden@bristol.ac.uk*
- *E. Williams is with the School of Experimental Psychology, University of Bristol, Bristol, UK.*

Manuscript received September 2, 2019; revised Month d, yyyy.

scores or *accepted answer mark*). They were then asked to pick the answer they would follow and explain why.

The results show that participants chose answers, *secure and insecure alike*, primarily because of surface features: its explanation detail in particular. Correspondence bias mediated by surface features of the answer (e.g., style of its writing and code, perceived characteristics of its author) play a major role in Stack Overflow readers' answer selection. We make the following major contribution.

1) *Stack Overflow readers do not select answers based on security of code snippets.* The prevalence of participants' decision rationale which demonstrated reflection on the security of the code snippet (C_a) was so low as to hold no significance. It can thus not be said that readers, in general, select answers based on careful scrutiny of a code snippet's security. This may be indicative of a lack of relevant security knowledge on readers' part, forcing them to rely on surface features rather than semantic features.
2) *Readers select answers based on surface features – especially answer detail.* The detail in explanations accompanying code snippets affected readers' selection of answers significantly ($p<0.01$), while all other factors did not ($p>0.05$). The thematic analysis showed that they were driven by: (a) its level of detail, (b) its ability to teach them, (c) its provision of concrete step-by-step instructions, or (d) trust-related properties attributed to the answer's author that were inferred *from* the answer detail (e.g., perceived author expertise, knowledge or professionalism).
3) *Readers are effected by several cognitive biases.* The closed coding of participants' rationales and subsequent thematic analysis revealed that a variety of cognitive biases are at work, predominantly related to surface and source features of the answer. *Correspondence bias* is a particularly important one, leading participants to judge an answer's appropriateness by inferring information irrelevant to the code snippet itself from the answer's detail.
4) *The prevalence of correspondence bias is a major challenge for pro-security interventions reliant on warnings or nudges*. Effective warning text should come from a position of authority [13], yet, our analysis reveals that potentially dangerous explanations are persuasive precisely because readers already infer that its author is an authority on the topic.

2 Background & Related Work

2.1 Why do developers not care about security?

Intrinsic motivation for secure software has been found to translate into better attitude towards secure development [14]. But regardless of whether a programmer *wants* to write secure code, is the matter of whether they can. An important anti-motivation is developers' perceived lack of competence, typically arising through a lack of resources and support [14]. As a result, the "build or borrow" question [15] comes into play, with developers having to decide whether to overcome challenges themselves, or rely on outside resources like code snippets from Stack Overflow – with all the security implications this may hold [2], [3].

Our findings show that developers indeed rarely directly assess code snippets when deciding to use an answer, but rely on the answer explanation to assess its credibility.

2.2 What kind of insecurity is prevalent?

Fischer et al. [6] showed that insecure code snippets found on Stack Overflow covered a range of functionality such as insecure SSL/TLS use, (a)symmetric cryptography, secure random number generation, hashes, or signatures. Of these, especially insecure SSL/TLS use was widespread. This may be in large part by API documentation confusing developers and offering too little clear guidance on parameters and configuration [2]. Answers to such challenges often enabled vulnerabilities by advocating insecure workarounds, not contributing to an understanding of secure TLS use [3].

Our work is the first to experimentally investigate *why* these answers, even if containing insecure code, may remain attractive to Stack Overflow readers.

2.3 How is Stack Overflow used?

A significant body of work exists analyzing what Stack Overflow posters ask about and how questions are answered, investigating these questions predominantly through the mining of millions of posts followed by application of machine learning and other semi-automatic quantitative studies [16]. A key thing missing so far, that our paper investigates, is whether these effects manifest among readers of Stack Overflow not actively participating in threads. And, if so, to what extent, and most importantly, what explanatory mechanisms underlie them.

Re-use of code snippets is widespread (and insecure). Developers often ask for actionable instructions due to a lack of examples in documentation [17]. Abdalkareem et al. [18] found that the prevalence of re-use of code snippets originating from Stack Overflow was widespread among both junior and senior developers, and that software using such snippets on average tended to have a higher percentage of bugs after reusing such code. Wu et al. [19] investigated code-reuse from Stack Overflow, showing that many code snippets were either incomprehensive, of low quality, or required too much modification to be useful. Zhang et al. [20] studied obsolete code snippets, finding that most obsolete answers were already obsolete when they were posted, and few were ever updated. Other studies have similarly found that code snippet re-use has a negative impact on security [6], [7], one controlled experiment even showing that developers using solely Stack Overflow (as opposed to official documentation, development books, or free web searches) produced the least secure solutions [8].

The community is not always right (or secure). Gantayat et al. [9] classified 4.5 million Stack Overflow posts, and found that the majority (77.65%) of questions accepted by the question poster were those that received the most upvotes. This may be because community votes "lean towards short, concise answers that include external links and have a better readability score," and such votes further bias other community members to upvote the same answer [9]. Zhang et al. [21] analyzed a large dataset of Stack Overflow threads with code snippets, finding that many contained API use violations, and that such answers were often accepted as the best answer and/or upvoted by the community. Meng et al. further explain how these insecure code snippets may then proliferate because developers, perhaps naively, trust responses from highly upvoted posts, or individual accounts with high reputation in their community [7]. Yet, much work on identifying threads and answers on Stack Overflow relies on such community data. For example, Yao et al. [22] proposed an algorithm to detect high quality questions/answers, based entirely on answer scores as voted by the community. Yang et al. [23] analyzed parsability and executability of code snippets on Stack Overflow found in a thread's 'best' answer, concluding that larger snippets were more likely to be so. However, they similarly accepted 'best' as accepted answers, claiming the question poster was most fit to make such judgments.

Our experimental findings show that upvotes have less influence than answer details when selecting security-related answers– challenging views espoused in literature that insecure code snippets proliferate *because* of their upvotes.

Detail may be important. Nasehi et al. [24] noted that explanations accompanying code snippets are important for question askers, although they only investigated posts deemed 'good' quality because of upvotes. Treude and Robillard [25] further investigated the importance of explanations around code snippets. They found an even split between code snippets from Stack Overflow that participants understood without requiring additional explanation and code snippets that were not understandable as-is. The latter were deemed not understandable because they were incomplete, rather than lacking explanatory text.

Our findings emphasize and contrast these results, showing the importance of answer detail over any other features, including the code snippet('s understandability).

Advocating security is not always popular. There is some evidence for the existence of subcommunities focused on security practices on Stack Overflow [26], but advocating security is not necessarily popular. Meng et al. [7] showed instances of cyberbullying, where posters attempted to warn the question poster for some answers' security implications, and consequently had condescending comments directed at them. Such findings are consistent with studies of the dynamics of Stack Overflow and other Q&A websites, where influence gaming is a real phenomenon [27], and experiments have shown effective ways to promote post or answer visibility and manipulate poster engagement [28]. Moreover, Wang et al. [29] studied answer revision on Stack Overflow, finding that gamification stimulated posters to revise answers, although very active posters mostly made minor textual revisions, likely related to the high likelihood of revisions being reverted if a poster is very active.

Our findings show that community dynamics (i.e., how upvotes and answer acceptance are socially influenced) are of little concern to understanding readers, as community factors are unlikely to sway their answer selection.

3 The Experiment

To examine what features affect Stack Overflow 'readers' when selecting an answer, we conducted a between-groups experiment. We presented participants with a mock Stack Overflow thread (shown in Fig. 1) and asked them to select the answer they were most likely to follow, also capturing their confidence in their selection, and for what reason they chose that answer. Participants were randomly assigned to one of 36 groups, each of which had different manipulated features, as explained in Section 3.3. Table 1 shows all of these groups and the features that we manipulated in the Stack Overflow thread. For example, a group might have a highly upvoted long secure answer followed by a short insecure answer with fewer upvotes, but an accepted answer mark. Table 1 gives an overview of all the groups and their manipulations. We obtained approval from our ethics committee before any empirical work began. We did not capture any personal information from participants.

3.1 Participant Recruitment

We recruited participants via Prolific [30]. We required participants to self-identify as having *programming skills* as an inclusion criterion. No mention of security was made during the recruitment. We paid each participant £0.50 on Prolific for completion of the study – approximately £15 per hour. Power analysis using G*Power [31] indicated group N=33 was sufficient for acceptable power $\geq$ 0.8 with χ^2 for large effect sizes $\geq$ 0.5, requiring a total N=1188 for the study.

3.2 Task Materials

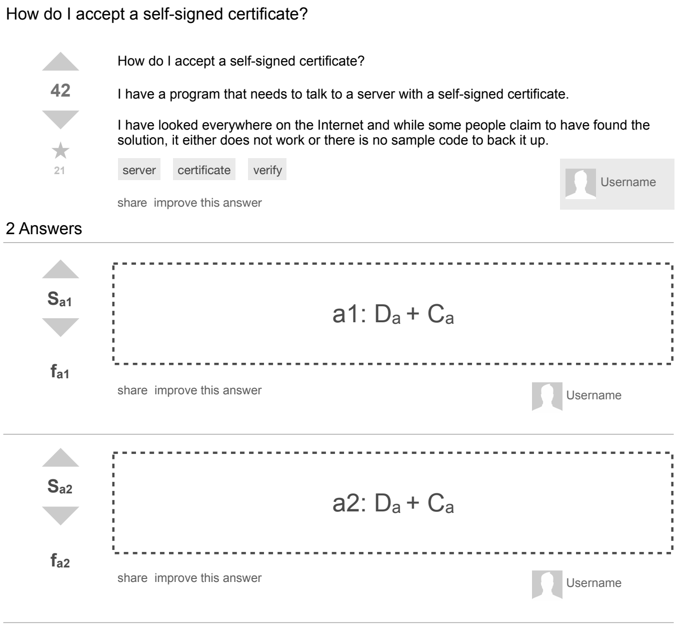
The Stack Overflow Thread. We built a parametrized Stack Overflow thread (see Fig. 1) with a question asking about accepting self-signed certificates—this topic solicits the most widespread insecure code snippets on Stack Overflow [6]. This thread is instantiated with two answers. Each answer contained an explanation (short or long) and code snippet (insecure or secure), leading to four possible answers (short and insecure, long and insecure, short and secure, long and secure). We did not include cases with both code snippets (in)secure, in order to focus on cases where a reader could be manipulated towards accepting (in)secure code.

Variables. The variables we focus on are Stack Overflow answers (a). Our study has several *independent variables* which we manipulate between groups. As shown in Fig. 1, each answer has a code snippet (C_a) which can be secure or insecure; an explanation (D_a) which may be detailed or undetailed. Moreover, each answer may have a numerical score (S_a) and/or a accepted answer mark (f_a). We measure how those independent variables affect our *dependent variable*: the answer a participant selects.

Hypotheses. To answer our research questions, we test a number of hypotheses describing the expected effect independent variables have on the dependent variable. Concretely, these are:

H1. Answer detail (D_a) affects answer selection.
H2. Answer score (S_a) affects answer selection.
H3. Answer acceptance (f_a) affects answer selection.
H4. Answer acceptance overrides answer score in answer selection.

The Thread Answers. To instantiate the Stack Overflow thread with answers, we created a secure code snippet and an insecure code snippet. Next, we created a short explanation and a detailed explanation for both those of



How do I accept a self-signed certificate?

How do I accept a self-signed certificate?

I have a program that needs to talk to a server with a self-signed certificate.

I have looked everywhere on the Internet and while some people claim to have found the solution, it either does not work or there is no sample code to back it up.

2 Answers

a1: $D_a + C_a$

a2: $D_a + C_a$

Fig. 1. Parametrized thread, variables shown in red.

those code snippets. We controlled the materials for their textual complexity, ensuring all were of comparable reading levels using Flesch-Kincaid readability tests.

Secure answer materials.
Short explanation: "Just add the self-signed certificate as an issuer."
Detailed explanation: "When checking a self-signed certificate, you need to check that the certificate they used is in the chain of trust for the site. They should make that certificate available somewhere and you should download and verify that it is correct. You'll need it in your program to verify the chain. Once you've got the certificate the steps are as follows: Verify that the site you're trying to connect to is the one you'd expect (i.e. that the hostname lines up). Add the certificate to the list of trusted issuers. Verify that every certificate in the chain was either signed by an issuer or signed by a certificate higher up in the chain. I've included code to do this below. Note that this overrides the default HTTPS connection mechanisms. If you need to connect to other sites you'll need to create, a new HTTPS.Connection for the self-signed one, and go back to the default for other sites."
Code snippet:

```
import HTTPS
import SSL
import X509CertificateChain

def site = "https://yourwebsite/"
def certificate = readFromFile("/thecert.cert")

def myHostnameCheck(String hostname):
        return (hostname == site)

defclass MySSLConnection(SSL.Connection):
        def checkChain(X509CertificateChain chain):
                for cert in chain:
                        if not this.checkCertificate(cert):
                                return false
                        return true

        def getIssuers():
                return [certificate]

HTTPS.Connection.setHostnameCheck(myHostnameCheck)
HTTPS.Connection.setSSLConnection(MySSLConnection)
```

Insecure answer materials.
Short explanation: "Just stick the code before you start the connection."
Detailed explanation: "Just stick the code in before you start the connection. Provided you know that the client is always going to talk to your server it is fine. RFC 2246 says you should do a whole bunch of extra checking to but this code will just accept your certificate. This will be fine in most cases. The code works by overriding the certificate checking mechanisms to just okay whatever it sees. Since the client and server checks will always say that it is okay, the connection will be set up and be encrypted with your certificate. A quick and easy way to implement HTTPS."
Code snippet:

```
import HTTPS
import SSL
import X509CertificateChain

def myHostnameCheck(String hostname):
        return true

defclass MySSLConnection(SSL.Connection):
        def checkChain(X509CertificateChain chain):
                return true
        def getIssuers():
                return []

HTTPS.Connection.setChecker(myHostnameCheck)
HTTPS.Connection.setSSLConnection(MySSLConnection)
```

3.3 Manipulations

Participants were assigned randomly to one of 36 groups across four manipulations. Each group (shown in Table 1) varied the features or their order of presentation.

Manipulation 1. Effect of answer detail. The only manipulation we made between groups was the level of detail that an answer had. Accounting for answer order, this led to four groups: (*1.1*) a long secure answer followed by a short insecure answer, (*1.2*) a short secure answer followed by a long insecure answer, (*1.3*) a short insecure answer followed by a long secure answer, (*1.4*) a long insecure answer followed by a short secure answer.

Manipulation 2. Effect of answer score. For each of the possible groups from manipulation one we manipulated the answer score. Accounting for answer order, this led to eight additional groups: each group from the first manipulation instantiated respectively with first answer score=32 and second answer score=7; and its inverse, first answer score=7 and second answer score=32.

Manipulation 3. Effect of accepted mark. For each of the possible groups from manipulation one we manipulated the accepted answer mark. Accounting for answer order, this led to eight additional groups: each group from the first manipulation instantiated respectively with the accepted answer = first answer; and its inverse, accepted answer = second answer.

Manipulation 4. Effect of accepted mark and answer score. For each of the possible groups from manipulation one we manipulated the answer score *and* accepted answer mark. Each group from the first manipulation was first instantiated with an answer score (32 or 7), and an accepted answer mark (answer one or two). Accounting for answer order, this led to 16 additional groups.

3.4 Data Collection and Analysis

Data collection procedure. Following informed consent, each participant was instructed as follows:

> "Assume that you were developing a program and had a problem with validating a self-signed certificate. After Googling your problem, you found an answered thread on Stack Overflow that addressed your problem. Please click on the answer you would follow in the below screenshot of the Stack Overflow thread."

Participants were then shown the Stack Overflow thread (Fig. 1) instantiated according to one of the 36 groups manipulations (Table 1). For example, participants assigned to group 4.1 saw a thread with a long secure answer followed by a short insecure answer, the first answer having a score of 32, and the second answer having a score of 7 and an accepted answer mark. Following the selection task we elicited a confidence score and a rationale by asking:

- How confident are you in the answer you chose [5pts Likert scale anchored with "not confident at all" – "extremely confident"]
- Why did you choose this answer? [open question]

The full questionnaire is shown in Appendix A. Raw data is available through our institutional repository [32].

Quantitative analysis. To ensure that results were meaningful in the sense of a group's answer selection not being similar to a randomly drawn sample, a binomial test was used to verify whether selection results(S=a1⊻a2, codified categorically as S=1⊻0) differed from chance (.5) levels. For our groups with N=33 this implies a non-chance range at p<0.05 for $S_{a1/2}$ lies within $12 \leq K \geq 21$ (resp. 36% and 64%). To assess whether groups differed significantly, we analyzed contingency tables with a chi-square (χ^2) test of independence to measure (dis)association between groups.

Qualitative analysis. To assess whether rationale aligned with the independent variables manipulated in groups, closed coding – marking rationales with pre-defined codes – for the independent variables was used (security, detail, score, acceptance). One author independently coded the rationale data, marking whether rationale was based on security of the code snippet (C_a), detail of the answer (D_a), score of the answer (S_a), or accepted mark (F_a). Another author coded a random 10% of the data, which was assessed through Cohen's Kappa for inter-rater reliability. Answer detail, score, and acceptance mark agreement all indicated 'very good' level of inter-rater reliability (resp. κ = 0.898, 0.856, 0.948). As there was a low number of rationales on security of the code snippets (5%), a second author coded all these rationales to establish inter-rater reliability, at 'perfect' level (κ=1). This was done to prevent a randomly selected subset of the data to contain very few security rationales, thereby skewing observed agreement towards expected agreement. To understand what explanations underlie readers' answer choices, thematic analysis [33], a structured form of qualitative research to discover meaningful patterns in data, was used to further analyze the elicited rationales, explained in more detail in Section 5. Two authors performed an open coding process where they independently coded the rationale data and subsequently categorized them into over-arching themes. Following this process, they then compared their results, integrating these into a shared codebook (see Appendix F). This codebook was then applied to the entire set of reflection data.

TABLE 1
Groups and manipulated variables.

Group	Answer Order	1st Answer Score	2nd Answer Score	Accepted Answer
	Manipulation 1: Effect of answer detail (D_a)			
1.1	long secure; short insecure	0	0	–
1.2	short secure; long insecure	0	0	–
1.3	long insecure; short secure	0	0	–
1.4	short insecure; long secure	0	0	–
	Manipulation 2: Effect of answer score (S_a)			
2.1	long secure; short insecure	32	7	–
2.2	long secure; short insecure	7	32	–
2.3	short secure; long insecure	32	7	–
2.4	short secure; long insecure	7	32	–
2.5	long insecure; short secure	32	7	–
2.6	long insecure; short secure	7	32	–
2.7	short insecure; long secure	32	7	–
2.8	short insecure; long secure	7	32	–
	Manipulation 3: Effect of accepted mark (f_a)			
3.1	long secure; short insecure	0	0	a1
3.2	long secure; short insecure	0	0	a2
3.3	short secure; long insecure	0	0	a1
3.4	short secure; long insecure	0	0	a2
3.5	long insecure; short secure	0	0	a1
3.6	long insecure; short secure	0	0	a2
3.7	short insecure; long secure	0	0	a1
3.8	short insecure; long secure	0	0	a2
	Manipulation 4: Effect of $S_a + f_a$			
4.1	long secure; short insecure	32	7	a1
4.2	long secure; short insecure	32	7	a2
4.3	long secure; short insecure	7	32	a1
4.4	long secure; short insecure	7	32	a2
4.5	short secure; long insecure	32	7	a1
4.6	short secure; long insecure	32	7	a2
4.7	short secure; long insecure	7	32	a1
4.8	short secure; long insecure	7	32	a2
4.9	long insecure; short secure	32	7	a1
4.10	long insecure; short secure	32	7	a2
4.11	long insecure; short secure	7	32	a1
4.12	long insecure; short secure	7	32	a2
4.13	short insecure; long secure	32	7	a1
4.14	short insecure; long secure	32	7	a2
4.15	short insecure; long secure	7	32	a1
4.16	short insecure; long secure	7	32	a2

3.5 Threats to validity

Internal validity. Constructs were carefully modeled after their real-world equivalents (using exact layout and graphical design from Stack Overflow), and textual explanations were controlled by ensuring similar readability. We did not find indications for text style being a confounding factor, nor did analysis of rationales indicate so, but this may differ with different participants. We specifically chose SSL/TLS use because it is a topic frequently discussed on Stack Overflow [34], known to be problematic for widespread posting of insecure code fragments [6], and posts enabling vulnerabilities by advocating insecure workarounds [3]. We chose answer scores so that both answers indicated community engaged of different levels, without an exaggerated skewing that might introduce demand characteristics. Difference in explanation length was modeled after highly visible real-world threads with both very short and long answer explanations. Using the Python textstat package, explanations fell in the 75.0–85.0 interval on the Flesch Reading Ease test, indicating fairly readable text at 6/7th grade level. Flesch-Kincaid readability test indicated a required 6th grade level to read (resp. 5.73 and 6.4 for long explanations), with short explanations necessarily indicating lower required grade levels of 4–5 due to reduced word count (resp. 4.96 and 3.65). We focused first on key variables known to affect posters (i.e., upvotes and acceptance) to investigate differences between posters and readers. Thus, we abstracted from other aspects potentially affecting answer selection such as user profiles (e.g., bias through username or pictures) and comments, which further research may specifically target.

External validity. These findings are focused on readers. We do not claim that they generalize to posters. Our sample has a Western bias, which may limit generalization to non-Western cultures. While the sample is also skewed towards men, this more likely represents a biased reality of software development. We avoided introducing security knowledge questions which could exaggerate pro-security answering. Moreover, it would be impossible to accurately contrast and compare self-reported security knowledge between participants, whether based on scales or familiarity with standards. This experiment covers one particular challenge, which may limit generalizability of the quantitative analysis. Based on analysis of the qualitative data, no confounding effects arose from familiarity with the materials (e.g., participants 'knowing' the right answer, or being unfamiliar with the topic) that could otherwise undermine external validity. We invite researchers to replicate the findings using our experimental setup, and extend it with e.g., additional knowledge measures and task materials representing different kinds of programming challenges.

Ecological validity. This study more closely resembles the dominant use of Stack Overflow by people simply reading threads, rather than the more limited subset active engagement in threads. Use of pseudo-code for the code snippets in the experiment may not represent exactly the kind of syntax readers would come across. We made this decision to ensure participants did not need experience with a specific language syntax and to control for different levels of familiarity with language syntax, constructing a pseudocode that was familiar to most programmers using Python syntax with Java idioms, coinciding with Stack Overflow's most popular languages. Moreover, this experiment did not allow for implementation of code, which may explain lower engagement with semantic features. Further research could explore to what extent attempts to implement code fragments potentially stimulate semantic and/or security considerations.

4 Quantitative Findings

Our analysis shows that *readers are distinct from those actively engaged in threads*, finding that (1) readers do not select answers based on security of code snippets, but (2) instead *strongly based on the explanation's amount of detail*–no matter whether the solution was secure or insecure, no matter whether it was upvoted or accepted by the community.

Demographics. Most (66%) participants were *passively* familiar with Stack Overflow: reading, but not actively using it to post answers or ask questions. 23% were non-users, while 11% were sometimes posters. Non-users were slightly less confident about their answer selection (avg. 2.8±1.1) than those who had, whether passively or actively (avg. 3.3±1) indicating a statistically significant difference in confidence (Mann Whitney U, respectively U=81279.5, $p<0.01$, Cliff's d=-0.24; and U=13550.5, $p<0.01$, Cliff's d=-0.28). Appendix C provides further demographic detail and a comparison to Stack Overflow demographics.

Answer selection. All answer selection results and Chi-square tests are shown in Appendices C–D. For brevity only statistically significant results are reported below.

To test Hypotheses 1–3, each manipulation varied one of the independent variables, while keeping others the same. To test Hypothesis 4, we varied two independent variables (S_a and f_a) together. For Hypothesis 1, we varied whether answer 1 or 2 had a detailed explanation. To find a significant effect of answer detail on answer selection regardless of answer order, independence should be shown in both cases.

H1. Answer detail affects answer selection:
Chi-square tests of independence revealed that the percentage of answer selection differed significantly (α=0.01) by answer detail regardless of whether the secure answer came first [χ^2 (1, N=33) = 27.27, $p<0.01$, ϕ = 0.6] or second [χ^2 (1, N=33) = 16.17, $p<0.01$, ϕ = 0.5]. In both of these cases the difference had a large ($\phi \geq 0.5$) effect size.
Conclusion: ***H1 is supported****.*

For Hypothesis 2, we took all four groups from the first manipulation and manipulated the answer score to be respectively 32, 7, and 7, 32. To find a significant effect, regardless of answer order or detail, independence should be shown across all four baseline group manipulations.

H2. Answer score affects answer selection:
Chi-square tests of independence revealed that percentage of answer selection only differed significantly (α=0.05) in one group when a short insecure answer came first with a medium effect size [χ^2 (1, N=33) = 6.2, $p<0.05$, ϕ = 0.3]. In all other tested groups (3 out of 4), difference in answer selection was statistically insignificant ($\alpha>0.05$).
Conclusion: ***H2 is only partially supported****.*

For Hypothesis 3, we took all four groups from the first manipulation and varied the accepted answer to be either answer 1 or 2. To find a significant effect, regardless of answer order or detail, independence should be shown across all manipulations of the four baseline groups.

H3. Answer acceptance affects answer selection: Chi-square tests of independence revealed that answer acceptance only differed significantly (α=0.05) in two groups with medium to small effect size when a long secure answer came first [χ^2 (1, N=33) = 6.99, p<0.05, ϕ = 0.3] or a short insecure answer came first [χ^2 (1, N=33) = 4.93, p<0.05, ϕ = 0.27]. In all other tested groups (2 out of 4), difference in answer selection was statistically insignificant (α>0.05).

Conclusion: ***H3 is only partially supported*.**

For Hypothesis 4, we took the groups from manipulation two and varied answer acceptance on top of the existing answer scores. To find a significant effect of f_a overruling S_a, independence should be shown across all manipulations.

H4. Answer acceptance overrides answer score: Chi-square tests of independence revealed that answer acceptance only different significantly ($\alpha \leq 0.05$) in three groups with low to medium effect size when an upvoted short secure answer came first [χ^2 (1, N=33) = 3.88, p<0.05, ϕ = 0.24], an upvoted short insecure answer came first [χ^2 (1, N=33) = 11.89, p<0.01, ϕ = 0.4], or a downvoted short insecure answer came first [χ^2 (1, N=33) = 12.44, p<0.01, ϕ = 0.4]. In all other tested groups (5 out of 8) difference in answer selection was statistically insignificant (α>0.05).

Conclusion: ***H4 is only partially supported*.**

We further checked whether answer selection differed between the 11% of participants who had actively used Stack Overflow, as they might be assumed to be more driven by upvotes and acceptance. A two-tailed Fisher's exact test found a difference between this subgroup and other participants only in two out of 16 groups (2.7 and 3.14; p=0.03), no other groups differed (p>0.3). The two identified differences did not lead to any difference for the hypotheses.

Answer rationale. Closed coding analysis of the elicited rationales showed two key findings: (1) participants by and large do *not* select answers based on security of code snippets, and (2), concurrent with the established support for H1 (answer detail affecting answer selection) answer detail was the dominant factor prevalent across all four manipulations (Fig. 2), differing significantly from chance levels for all manipulations (binomial test, p<0.05).

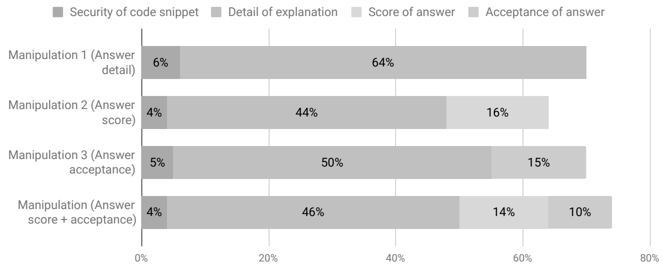


Fig. 2. Distribution of rationale by study.

Breakdown of rationales to familiarity with Stack Overflow indicated a similar distribution (Fig. 3), answer detail differing significantly from chance levels for all familiarity degrees (binomial test, p<0.05).

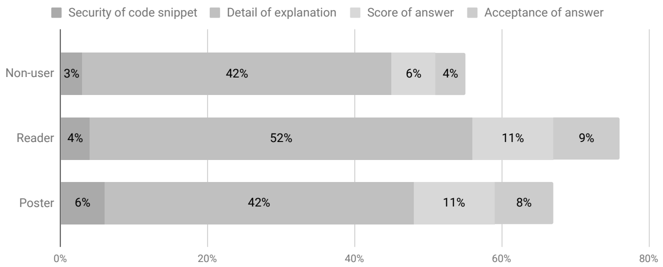


Fig. 3. Distribution of rationale by familiarity.

H1. Answer detail affects answer selection: Analysis of participant rationale showed that regardless of manipulation or participant familiarity with Stack Overflow, answer detail affected answer selection.

Conclusion: ***H1 is further supported.***

We further checked whether answer selection differed between the participants who expressed having some familiarity with Stack Overflow and those who did not. A two-tailed Mann Whitney U test comparing all rationales across manipulations 1–3 did not find a significant difference in the extent of rationale focusing on detail (U=527.5, z=−1.35, p=0.18, r=0.16), score (U=15.5, z=−1.68, p=0.09), nor for answer acceptance (U=13, z=−1.94, p=0.05).

We performed the same check for manipulation four, where both score and answer acceptance were manipulated at the same time. A two-tailed Mann Whitney U test did not find a significant difference in the extent of rationale focusing on score (U=88.5, z=−1.47, p=0.14), but did find so for answer acceptance (U=47.5, z=−3, r=0.53). Unfamiliar participants' rationale indicated reasoning about acceptance 4%(±0.08) of the time, while familiar participants did so 12%(±0.02) of the time. However, post-hoc power analysis using G*Power revealed only a statistical power of 0.76. More importantly, as the thematic analysis in Section 5 will show, many of these participants who expressed familiarity with Stack Overflow and focused on acceptance marks misinterpreted their semantics, interpreting them e.g., as implying code having been formally validated.

5 Qualitative findings

We next present the results of a thematic analysis of what explanations underlie readers' answer choices. We find that when, readers focus on the answer detail, they tend to focus on surface features of the answers, how the answer looks and feels, and critically, *over-interpret answers, reading more into them than strictly true*. Our analysis was driven by Braun and Clarke's approach [33] of systematically identifying and examining meaningful themes in data. This can be based on a pre-existing theoretical lens, or an entirely open inductive method. In this specific case, we focus on understanding the reasoning behind participants' decisions expressed in their rationales. Two authors coded the participants' rationales in an iterative process, meeting several times to discuss differences in coding and/or identification of the cognitive biases within the rationale data, which led to an agreed upon code book and labeling of features and cognitive biases across the identified themes[1].

1) by what *features* they assessed that theme, and,
2) what *cognitive biases* affecteded their reasoning.

Features – What they look at. When participants focused on something, heuristic information processes came into play, involving cognitive strategies that focus on different *features* to evaluate that information. The literature (e.g., [12], [10], [11]) shows that this typically involves three features which people fluidly move through depending on their knowledge and individual cognitive make-up [35], [36]. These typical three features map onto the independent variables we manipulated in the studies, namely:

1) *Semantic* features of the information (i.e., the code snippet itself, its correctness, accuracy, or, C_a)
2) *Surface* features, the way information is presented (i.e., the total post, explanation, its style and writing, or, D_a)
3) *Source* features of the information, such as where it came from (i.e, the Stack Overflow community and its voting mechanisms, or, S_a and f_a)

Cognitive biases – How they look at it. With further analysis through a lens of cognitive bias taxonomies (cf. [37], [38]), we identified a number of recurring cognitive biases evident in participants' reasoning. These explain the underlying mechanisms that drove people when selecting answers, guiding participants' decision making and rationalization processes – *stimulating participants to confidently choose both secure and insecure solutions alike*. In particular, we identified:

- *Correspondence bias*, choice affected by the tendency to draw inferences about information or a person which could be entirely explained by other factors (e.g., assuming an answer is right because its style inspires confidence).
- *Priming bias*: choice affected by the information provided focusing participants on that information (e.g., focusing exclusively on the exact steps given in an answer's solution).
- *Anchoring effects*: choice affected by the information first made apparent to the participant (e.g., going with the first answer provided).
- *Affinity bias*: choice affected a preference for things or people considered to be like the participant (e.g., assuming an answer is right because it feels like it was written by someone like you).
- *Conformity bias*: choice affected by whether others have accepted it (e.g., assuming an answer is right because everybody else says it is right).
- *Framing effect*: choice affected by whether options are presented as gains or losses (e.g., assuming an answer is right because it has an accepted answer mark while the other does not).

Table 2 gives an overview of these two aspects, showing which of the features and cognitive biases became salient in participants' rationales. Note that cognitive biases were identified within participants' rationales, and not necessarily shared by all participants in a theme. Participants' rationale following their passive, one-way engagement with a Stack Overflow thread primarily focused on *surface features*, in contrast to active engagement with Stack Overflow in literature, which primarily focuses on *source features*.

TABLE 2
Summary of what participants focused on, by what features they assessed it, and what cognitive biases were identified in their rationale when doing so.

	Features			**Cognitive Biases**					
***Overall* focus and specific focus**	Semantic	Surface	Source	Affinity bias	Anchoring effect	Conformity bias	Correspondence bias	Framing effect	Priming bias
Mainly focusing on semantic features									
I know the code snippet is secure	●								
Mainly focusing on surface features									
The author knows what they're talking about		●	●				●		
It teaches me how to do it		●					●		
The author comes across as wanting to teach		●	●				●		
The code snippet looks elegant		●					●		
It tells me step by step what to do		●							●
I am not affected by what answer came first		●	●		●				
It lets me just copy and paste		●		●					
Mainly focusing on source features									
The community is not always right			●			●			
The community might be right			●			●			
The answer was accepted			●			●		●	
The answer was most upvoted by the community			●			●			
I misunderstand the community mechanisms			●			●		●	

5.1 Focus on Semantic Features

Only few participants (≈5% as shown in Fig. 2) focused on semantic features, indicating that they possessed relevant domain knowledge to assess the code snippet itself.

5.1.1 I know the code snippet is secure

For the few participants who looked at the security of the code snippet, their domain knowledge came into play in their decision making process. These participants typically relied on security knowledge and best practices to make their decisions, such as one participant noting:

> "While the first answer looked like it had been better written, with a description along with the code, it looked to be overriding security checks, which is not a good idea. The second answer, while it didn't have a good description looked to have the better quality code without an obvious security flaw" (P4.12-20)

5.2 Focus on Surface Features

The largest number of participants (≈50% as shown in Fig. 2) focused on surface features, indicating that the level of detail in the answer enabled them to solve the problem. This was characterized by a focus on the answer itself, both

[1] When referring to quotes from participants, these will be identified by the number of their group, and the number of the participant within, e.g., P2.7–2.

the explanation and the code snippet. Frequently recurring were mentions that the extent of information instilled trust, such as e.g., participants noting that:

> "[. . .] because it had more information, which made it more worthy of my trust" (P2.7-2)

This worked to provide participants with confidence that they had selected the best answer, regardless of whether the code snippet was actually secure, as evidenced by e.g., participants rationalizing their selection of the insecure answer:

> 'It explained the theory behind the explanation which gives me confidence that it will be correct" (P4.6-9)

The focus on surface features discussed here enabled several cognitive biases, including: **correspondence bias**, **priming bias**, **anchoring effects**, and **affinity bias**. The most important underlying mechanism explaining this selection strategy is **correspondence bias**. In this context, the answer is seen as an indirect reflection of the person's disposition. Participants draw inferences about it being 'best' or 'correct' by virtue of properties such as its amount of detail or simply length by correlating this to the intention of the author. Several themes branched out into more specific aspects relating to the information, as discussed below.

5.2.1 The author knows what they're talking about.

Many participants made their decision reasoning about the author of the solution coming across as an *authority* on the topic, giving them confidence in the answer they selected. This was regardless of whether they selected secure or insecure solutions. As one participant noted:

> "I'm not familiar with this issue or the programming language so went by the explanation – my choice felt as if written by someone with more expertise" (P4.10-18)

While others chose insecurely for the same reasons:

> "I chose this answer because it was a much more detailed response which indicates the person may know a lot about the subject." (P4.9-23)

Correspondence bias is an important mechanism explaining this decision making. Participants first inferred attributes of the information's source – perceiving the author to " know a lot", or have "more expertise", which then was conflated with expected correctness of the solution. The problem here, as before, is that a lack of domain knowledge, combined with a lack of security knowledge, forces participants to defer from the *semantic* features of the answer to *surface* and *source* features, attributing what they perceive there as showing the answer's author holds a level of authority that must mean the information is credible.

5.2.2 It teaches me how to do it

This theme showed that some participants are driven by long-term goals rather than immediate satisfaction, wanting to learn how to overcome problems. For example, one participant noted that they selected an answer because:

> "[. . .] the answer was better suited to help me understand the problem rather than just fix it. That way I've gained knowledge on how to deal with the problem in the future." (P4.1-27)

However, participants similarly chose insecure answers to learn from, as evidenced here by another participant:

> "Theory behind the code is sound and to learn for next time" (P1.2-12)

An underlying mechanism here may similarly be **correspondence bias**, as participants typically emphasized perceived intention of the answer poster (teaching others with detailed explanation vs. just providing a solution) rather than the (in)correctness of the solution itself. This may lead to dangerous situations where insecure answers, merely by being written in a style that convinces readers they are meant to teach, provide them with the confidence that these answers *must* be correct, let alone secure.

5.2.3 The author comes across as wanting to teach.

This theme goes further than the above *It teaches me how to do it (5.2.2)* theme, as participants reasoned not about the ability of the information to teach them, but that the source of that information – its author, *wants* to help or teach them. The way information is presented spurs participants to reason about the personal characteristics of the answer's author, while selecting secure or insecure answers:

> "The poster seems more polite and has taken more trouble to answer the opening poster." (P2.7-3)

As above, **correspondence bias** is an important mechanism, as participants conflate perceived personality traits ('polite', 'encouraging'), which may not be accurate, with expected correctness of the solution, and then act upon that.

5.2.4 The code snippet looks elegant

Answer selection in this theme was driven by focus on how the code snippet looked – its surface features, associating coding style with other perceived properties such as correctness or security. For example, a participant selecting the insecure code snippet did so rationalizing on:

> "Very neat and 'clean' coding, doesn't rely on static values" (P4.8-20)

An underlying mechanism here may similarly be **correspondence bias**, as selection for a desired property is made on the basis of perceived properties that do not necessarily correlate to that (e.g., neat and clean coding is not necessarily correct or secure).

5.2.5 It tells me step by step what to do

Answer selection here was driven by the answer giving concrete, exact instructions purporting to solve the problem if they were followed. For example, participants reasoned:

> "There was enough clear cut steps for me to follow with step by step instructions." (P4.1-26)

This goes further than the higher-level general focus on surface features, where participants rationalize their choice in the amount of information. Rather, the structure of the answer comes into play, and guides their future behavior more strongly. **Priming bias** explains this theme succinctly – participants are influenced by what the answer poster says should be done (i.e., the steps), shaping their idea of what the solution should contain or do. The dangers of this bias are especially evident given some participants' explicit reasoning that they use Stack Overflow *because* they do not want to further research or think about the challenge:

> "Uncommented, raw code required me to do extra research that's not the point of Stack Overflow." (P3.4-13)

5.2.6 *I am not affected by what answer came first*

In some cases the initial information (i.e., the first answer) given affected the decision making process of the participant, or, vice versa, participants aware of such effects in their resolve to avoid making their decision based on the answer order. For example, one participant noted active reflection about answer order, confidently selecting the insecure solution simply because they attribute some meaning to the answer order:

> "Second answer is most of the time the answer im looking for." (P4.5-11).

Another participant reflected on the pitfalls of this, noting that:

> "[I did not chose the first answer because] the poster marked the first one that works for him. But the best answer could be answered weeks later." (P4.14-26)

An underlying mechanism here may be **anchoring effects** – the information first presented to participants factors significantly into their decision making, whether positively (accepting the first answer because it was first), or negatively (rejecting the first answer because it was first).

5.2.7 *It lets me just copy and paste*

The well known trope of developers 'just' copy and pasting from Stack Overflow appeared to some extent as well, with participants deciding what answer to select based on how straightforward the code snippet was to re-use. This led participants to select both secure and insecure solutions alike, with one participant selecting a secure answer because:

> "In my opinion the code in the first answer is much easier to introduce and use" (P2.3-33)

While another participant selected an insecure answer:

> "It seemed like the easiest solution. When I want go get something to work and I've been on it for hours, I don't care why it works, just that it works." (P4.15-33)

We found **affinity bias** to explain this decision making, as some participants showed clear preference for solutions provided by others perceived to be like them, sharing some kind of shared identity (e.g., 'programmers'). Participants contextualized their rationale for wanting brief solutions with little explanation by their status "as coders", thereby preferring answers conforming to what they would expect from others in their social group.

5.3 Focus on Source Features

A much lower number of participants (≈15% as shown in Fig. 2) focused on source features. This was characterized by participants deferring from judging an answer themselves to *conforming* with what the wider community says, such as:

> "I myself wouldn't know which is a more prudent choice, I would defer to collective knowledge and trust that the people upvoting have tried the solution and found that it works." (P2.4-3)

The focus on source features discussed here enabled two main cognitive biases: **conformity bias** and **framing effects**. Most important here is **conformity bias**. In this context, participants defer to collective knowledge, some themes showing this effect strengthened through **framing** of answer scores and accepted marks, in line with the well-established Asch effect [39].

5.3.1 *The community is not always right.*

In some rare cases domain knowledge or, yet again, stronger **correspondence bias** reduced conformity to the widely accepted answer. One participant noted so:

> "[the poster who] explained seemed to have knowledge in the matter and was more complete than the verified response." (P3.7-2)

In terms of the Asch effect this can be explained that the appearance of dissenting factors (e.g., where a perceived authority figure dissents from the majority opinion) reduces conformity to the majority opinion.

5.3.2 *The community might be right.*

A more nuanced theme followed as well, with participants explicitly reflecting on their decision making process and how their certainty was reduced by the appearance of community mechanisms. For example, one participant selected the secure solution, but was thrown off by an answer acceptance to the insecure solution:

> "It seemed more thorough. Although the first answer had a check mark in front of it, so that threw me off a little bit." (P3.7-11)

Other participants chose insecure solutions still unsure whether they should have conformed to majority opinion:

> "The answer looked like it was complete, there was an explanation provided with the code. However, the answer had only 3 'likes', while the other answer had 32. This is why i'm not entirely sure." (P4.11-31)

Similar to above, the appearance of dissenting factors (e.g., mismatched answer acceptance) reduces **conformity bias**, allowing for other mechanisms to overtake the decision making process, such as the two examples above showing participants based their decisions on the perceived completeness and level of detail instead.

5.3.3 *The answer was accepted.*

Participants were affected by the appearance of "accepted answer" marks, whether placed at secure or insecure solutions, reasoning these must be the solution:

> "As it had a green tick against it which I believe to be put on there by the original poster to say that this response solved their problem." (P3.3-31)

The use of the answer acceptance mark here shows a **framing effect** that affects participants' decision making, as it places some information in a positive light ('accepted'), while the other information remains neutral or even negative ('not accepted'). This is a concern, as participants confidently selected insecure solutions because they were framed in a positive way by the community mechanisms: "[I chose this answer because of the] Green tick. As this is original posters choice." (P3.4-25)

5.3.4 The answer was most upvoted by the community.

Both secure and insecure solutions alike were selected by participants based on credibility attributed to the score:

> "More users have credited this user through the up-/down vote system. Therefore upon first viewing this answer looks like it would have more credibility than the one below." (P2.3-26)

These decisions are a classical example of **conformity bias**, where the majority acceptance of an answer will spur participants trust and accept in majority opinion.

5.3.5 I misunderstand the community mechanisms.

A nuance of the two themes above, several participants showed that their decision making was influenced by an interpretation of community mechanisms that does not accurately reflect how these mechanisms work, or how much trust can be reasonably placed in them. For example, some participants confidently selected the insecure solution assuming answer acceptance meant the community as a whole judged it, rather than the original asker of the question:

> "It got the check sign, which means it's the generally accepted and good answer." (P3.2-12)

Similarly, participants selected insecure solutions based on an interpretation of upvotes implying that all those who upvoted had actually *used* the code snippet:

> "Most upvotes, i would assume the code in that answer would work better as more people have used it" (P2.2-9)

This, perhaps overly positive, view of the Stack Overflow community mechanisms can be best described by a participant who chose the secure solution, because

> "[. . .] the programmer community can feel confident that the best solution has been peer reviewed." (P4.1-9)

These misinterpretations of community mechanisms further strengthened the **conformance** and **framing** effects already found in their respective themes.

6 Discussion and Takeaways

6.1 How do the findings relate to other work?

The general finding that, regardless of reason, participants selected secure and insecure solution alike complements research that has shown that insecure solutions are just as likely to be upvoted and accepted by Stack Overflow posters as secure solutions (cf. [21]). Our findings complement work detailing the importance of explanations accompanying code snippets [24], showing this importance likely holds for readers and posters, as well as having identified a number of explanatory mechanisms that show *why* these explanations are perceived as important by readers (Table 2). Our findings contrast with related work that has investigated readers. Treude and Robillard [25] found that code snippets deemed not understandable by readers were judged so because they were considered incomplete, rather than because they lacked explanation. Our findings contrast this, as code snippets deemed not understandable by readers were highlighted as lacking a detailed explanation, rather than a need for more code. Indeed, the latter is likely reliant upon semantics and domain knowledge that readers often do not exhibit (cf . Figures 2–3). More importantly, our work contrasts with the general view of the importance of community mechanisms. Whilst some work found and/or proposed identifying high quality posts and answers based on upvotes (cf. [7]), our findings instead show that, for readers, this may have less persuasive power.

6.2 Why are 'readers' and 'posters' different?

As shown in Table 2, we found that 'reader' strategies for selecting answers are characterized by assessing aspects related to the surface features of information and its source, which are mediated by *correspondence biases*. In stark contrast, posters, as evidenced by related work, is characterized by a focus on the medium, using strategies assessing source features, and mediated by *conformity biases*. Whereas active participation in a community brings with it social pressures (cf. Meng et al. [7]'s description of the cyberbullying of security-minded replies), which pressure people to conform in order to become, and remain, part of the social in-group, only reading the community's generated knowledge brings no such considerations. Thus, in the absence of relevant semantics or domain knowledge, there is a shift of assessing credibility from relying on source features to assessing the information and its related surface features. In general, readers demonstrate a tendency to default to trusting information, known as the 'truth bias' [40] (albeit mediated by beliefs that they hold towards the risk of information technology in general [41], [42]), thus making it less likely that they will critically assess answers and explanatory text for potential adversarial or otherwise malicious elements, a process that may take substantial time and mental resource. Given that the amount of cognitive effort that a person is willing to expend to process answer-related information is likely to differ according to their attitudinal commitment [43] to its source [44] (i.e., the answer's author), the correspondence biases we identified are particularly important. They show a key component of such attitudinal commitment: normative, or emotional attachment to the answer and its author [45] (cf. the identified themes in Section 5.2). Thus, attention to the implications of this shift for the spread of insecure solutions as readers rely on surface features are particularly critical.

6.3 What are the implications of this difference?

Recent work has shown that scenarios invoking heuristic information processing strategies increase susceptibility to persuasive elements within fraudulent messages compared to more systematic processing conditions [46]. The process by which Stack Overflow readers determine what answer to select operates as such a process (see Sec. 5) by alternatively assessing the answers' semantic, surface, or source. It should thus come as no surprise that answer detail was capable of inspiring confidence in insecure solutions.

The detection of obsolete and insecure code snippets is an important part of ensuring that their spread is limited (cf [20]). However, as the readers themselves are the last line of defense, warning messages to help them make secure decisions are an effective complementary strategy [47]. Such warning messages could be technically grounded in the creation of classifiers on the basis of a wider corpus of persuasive explanations, as has been done for, e.g., identification of scam emails [48]. Stack Overflow readers could be offered such support by, e.g., browser plugins providing additional information detailing when and where given explanations are written in a way that may trigger some of the biases we identified – for example, when an explanation is likely to instill a false sense of confidence in its author, or when its focus is likely to prime someone into ignoring other factors. However, such support needs to be carefully designed from a usability point of view so as not to overburden its users with additional information to consider.

Lack of domain knowledge (whether functional or security-related) in regards to a programming problem may cause Stack Overflow readers to defer to features of the answer and its perceived author to assess its credibility. Warnings could play a role in nudging readers to express some skepticism towards it by overcoming their inherent truth bias. This is important because recent work investigating the persuasiveness of phishing emails [49] found that "without a reason to doubt the legitimacy of an email, [people] may then simply defer to assuming that the communication is likely to be genuine." A challenge here is that research has shown that effective warning text should include clear descriptions of potential negative outcomes, and it should come from a position of authority [13]. Yet, the themes we identified show that potentially dangerous explanations are perceived as persuasive precisely because readers infer that its author has authority on the topic; potentially leading such warnings into a 'he said / she said' scenario that is unlikely to definitively convince readers whom to trust.

6.4 How do we move forward in understanding what drives Stack Overflow readers?

We propose that further work should focus on establishing patterns of persuasive writing styles that feed into correspondence bias during answer selection. Moreover, gender effects reducing women's involvement on Stack Overflow have been studied [50] and may also factor into readers' acceptance of answers. We controlled for any effects from profile information of answer posters (e.g., profile picture, username), but even so, .04% of male participants used gendered language while referring to the answer author (e.g., "that gave the impression he knew what he was doing" (P4.15-28)), while only .006% of female participants did so. Further work can additionally explore whether gender markers of answer authors may further contribute to answer (de)selection through, e.g., correspondence and affinity bias.

7 Conclusion

In this paper, we conducted the first large scale (n=1188) between-groups experiment of Stack Overflow readers. We found statistically significant support for answer detail affecting what answers (and code snippets) readers will select ($p<0.01$). Critically, this means that *readers of Stack Overflow will select secure and insecure answer alike based on the amount of explanatory detail provided – even if incorrect, regardless of whether the community has attempted to sway them from it.*

Through a thematic analysis focused on understanding the underlying reasons, we found that correspondence bias mediated by surface features of the answer (e.g., style of its writing and code, perceived characteristics of its author) plays a major role in readers' answer selection. *Readers place so much stock in an answer's explanation detail because they perceive such answers to come from an authoritative source – regardless of whether that is true at all.*

These findings hold important implications for the widespread use of Stack Overflow as a databank of reusable code snippets. While there are many people actively engaged in Stack Overflow threads, *developers who just read threads and use code snippets represent a far larger demographic that needs support to not inadvertently use insecure code snippets.*

Guidelines for Reading & Using Stack Overflow

It is difficult to assess what code fragments are safe to take. Consider these guidelines below to guide your decision-making:

- Realize not everyone posting an answer is your friend—there may be unintentional or malicious poor code posted. *Action for users:* utilize plugins warning of dangerous code [51], [52].
- Don't be swayed just by more detail that makes it look as if a poster knows what they are talking about—it may sway you for irrelevant reasons. *Action for researchers*: Build classifiers for 'persuasive' explanation text which could inform similar plugins for 'dangerous explanations'.
- Make sure you understand the visual semantics well—a green acceptance mark does *not* mean the code has been validated or proven to work well, securely, or at all.

Acknowledgments

This work is partially supported by EPSRC grant EP/P011799/1, Why Johnny doesn't write secure software? Secure software development by the masses.

References

[1] S. Baltes, R. Kiefer, and S. Diehl, "Attribution required: Stack overflow code snippets in github projects," in *ICSE Companion*. IEEE Press, 2017, pp. 161–163.

[2] M. Georgiev, S. Iyengar, S. Jana, R. Anubhai, D. Boneh, and V. Shmatikov, "The most dangerous code in the world: validating ssl certificates in non-browser software," in *CCS*. ACM, 2012, pp. 38–49.

[3] S. Fahl, M. Harbach, H. Perl, M. Koetter, and M. Smith, "Rethinking ssl development in an appified world," in *CCS*. ACM, 2013, pp. 49–60.

[4] H. Imai and A. Kanaoka, "Time series analysis of copy-and-paste impact on android application security," in *AsiaJCIS*. IEEE, 2018, pp. 15–22.

[5] Y. Acar, M. Backes, S. Fahl, D. Kim, M. Mazurek, and C. Stransky, "How internet resources might be helping you develop faster but less securely," *IEEE Security Privacy*, vol. 15, no. 2, pp. 50–60, 2017.

[6] F. Fischer *et al.*, "Stack overflow considered harmful? the impact of copy&paste on android application security," in *Symp. S&P'17*. IEEE, 2017, pp. 121–136.

[7] N. Meng, S. Nagy, D. Yao, W. Zhuang, and G. Arango-Argoty, "Secure coding practices in java: Challenges and vulnerabilities," in *ICSE*. IEEE, 2018, pp. 372–383.

[8] Y. Acar *et al.*, "You get where you're looking for: The impact of information sources on code security," in *Symp. S&P'16*. IEEE, 2016, pp. 289–305.

[9] N. Gantayat, P. Dhoolia, R. Padhye, S. Mani, and V. S. Sinha, "The synergy between voting and acceptance of answers on stackoverflow, or the lack thereof," in *MSR*. IEEE, 2015, pp. 406–409.

[10] T. Lucassen, R. Muilwijk, M. L. Noordzij, and J. M. Schraagen, "Topic familiarity and information skills in online credibility evaluation," *J. Assoc. Inf. Sci. Technol.*, vol. 64, no. 2, pp. 254–264, 2013.

[11] T. Lucassen and J. M. Schraagen, "Factual accuracy and trust in information: The role of expertise," *J. Assoc. Inf. Sci. Technol*, vol. 62, no. 7, pp. 1232–1242, 2011.

[12] E. Sillence, P. Briggs, P. Harris, and L. Fishwick, "A framework for understanding trust factors in web-based health advice," *Int. J. Hum-Comput. St.*, vol. 64, no. 8, pp. 697–713, 2006.

[13] D. Modic and R. Anderson, "Reading this may harm your computer: The psychology of malware warnings," *Comput. Hum. Behav*, vol. 41, pp. 71–79, 2014.

[14] H. Assal and S. Chiasson, "Motivations and amotivations for software security – preliminary results," in *WSIW*, 2018.

[15] J. Brandt, M. Dontcheva, M. Weskamp, and S. R. Klemmer, "Example-centric programming: integrating web search into the development environment," in *CHI*. ACM, 2010, pp. 513–522.

[16] A. Ahmad, C. Feng, S. Ge, and A. Yousif, "A survey on mining stack overflow: question and answering (q&a) community," *Data. Technol. Appl.*, vol. 52, no. 2, pp. 190–247, 2018.

[17] C. Rosen and E. Shihab, "What are mobile developers asking about? a large scale study using stack overflow," *Empirical Software Engineering*, vol. 21, no. 3, pp. 1192–1223, 2016.

[18] R. Abdalkareem, E. Shihab, and J. Rilling, "On code reuse from stackoverflow: An exploratory study on android apps," *Information and Software Technology*, vol. 88, pp. 148–158, 2017.

[19] Y. Wu, S. Wang, C.-P. Bezemer, and K. Inoue, "How do developers utilize source code from stack overflow?" *Empirical Software Engineering*, pp. 1–37, 2018.

[20] H. Zhang, S. Wang, T.-H. P. Chen, Y. Zou, and A. E. Hassan, "An empirical study of obsolete answers on stack overflow," *IEEE Transactions on Software Engineering*, 2019.

[21] T. Zhang, G. Upadhyaya, A. Reinhardt, H. Rajan, and M. Kim, "Are code examples on an online q&a forum reliable?: a study of api misuse on stack overflow," in *ICSE*. IEEE, 2018, pp. 886–896.

[22] Y. Yao, H. Tong, T. Xie, L. Akoglu, F. Xu, and J. Lu, "Detecting high-quality posts in community question answering sites," *Information Sciences*, vol. 302, pp. 70–82, 2015.

[23] D. Yang, A. Hussain, and C. V. Lopes, "From query to usable code: an analysis of stack overflow code snippets," in *MSR*. ACM, 2016, pp. 391–402.

[24] S. M. Nasehi, J. Sillito, F. Maurer, and C. Burns, "What makes a good code example?: A study of programming q&a in stackoverflow," in *ICSM*. IEEE, 2012, pp. 25–34.

[25] C. Treude and M. P. Robillard, "Understanding stack overflow code fragments," in *ICSME*. IEEE, 2017, pp. 509–513.

[26] T. Lopez, T. T. Tun, A. Bandara, M. Levine, B. Nuseibeh, and H. Sharp, "An investigation of security conversations in stack overflow: perceptions of security and community involvement," in *Proc. SEAD*. ACM, 2018, pp. 26–32.

[27] A. Bosu, C. S. Corley, D. Heaton, D. Chatterji, J. C. Carver, and N. A. Kraft, "Building reputation in stackoverflow: an empirical investigation," in *MSR*. IEEE, 2013, pp. 89–92.

[28] M. Carman, M. Koerber, J. Li, K.-K. R. Choo, and H. Ashman, "Manipulating visibility of political and apolitical threads on reddit via score boosting," in *TrustCom 2018*, 2018, pp. 184–190.

[29] S. Wang, T.-H. P. Chen, and A. E. Hassan, "How do users revise answers on technical q&a websites? a case study on stack overflow," *IEEE Transactions on Software Engineering*, 2018.

[30] S. Palan and C. Schitter, "Prolific.ac–a subject pool for online experiments," *J Behav Exp Finance*, vol. 17, pp. 22–27, 2018.

[31] F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner, "G* power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences," *Behavior research methods*, vol. 39, no. 2, pp. 175–191, 2007.

[32] van der Linden, Dirk and others, "Raw Data Repository for "The impact of surface features on choice of (in)secure answers by Stackoverflow readers"'," https://bit.ly/3cQ4xjj, 2020, online; accessed 10 January 2020.

[33] V. Braun and V. Clarke, "Using thematic analysis in psychology," *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[34] N. Patnaik, J. Hallett, and A. Rashid, "Usability smells: An analysis of developers struggle with crypto libraries," in *SOUPS*, 2019.

[35] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *CHI*. ACM, 2006, pp. 581–590.

[36] B. J. Fogg, C. Soohoo, D. R. Danielson, L. Marable, J. Stanford, and E. R. Tauber, "How do users evaluate the credibility of web sites?: a study with over 2,500 participants," in *DUX*. ACM, 2003, pp. 1–15.

[37] C. R. Carter, L. Kaufmann, and A. Michel, "Behavioral supply management: a taxonomy of judgment and decision-making biases," *International Journal of Physical Distribution & Logistics Management*, vol. 37, no. 8, pp. 631–669, 2007.

[38] E. Dimara, S. Franconeri, C. Plaisant, A. Bezerianos, and P. Dragicevic, "A task-based taxonomy of cognitive biases for information visualization," *IEEE Trans. Vis. Comput. Graph*, 2018.

[39] S. E. Asch, "Studies of independence and conformity: I. a minority of one against a unanimous majority." *Psychological monographs: General and applied*, vol. 70, no. 9, p. 1, 1956.

[40] C. F. Bond Jr and B. M. DePaulo, "Accuracy of deception judgments," *Pers. Soc. Psychol. Rev*, vol. 10, no. 3, pp. 214–234, 2006.

[41] Y. D. Wang and H. H. Emurian, "An overview of online trust: Concepts, elements, and implications," *Comput. Hum. Behav*, vol. 21, no. 1, pp. 105–125, 2005.

[42] C. L. Corritore, B. Kracher, and S. Wiedenbeck, "On-line trust: concepts, evolving themes, a model," *International journal of human-computer studies*, vol. 58, no. 6, pp. 737–758, 2003.

[43] N. J. Allen and J. P. Meyer, "The measurement and antecedents of affective, continuance and normative commitment to the organization," *J. Occup. Organ. Psychol*, vol. 63, no. 1, pp. 1–18, 1990.

[44] A. Vishwanath, "Habitual facebook use and its impact on getting deceived on social media," *Journal of Computer-Mediated Communication*, vol. 20, no. 1, pp. 83–98, 2014.

[45] M. Workman, "Wisecrackers: A theory-grounded investigation of phishing and pretext social engineering threats to information security," *J. Assoc. Inf. Sci. Technol*, vol. 59, no. 4, pp. 662–674, 2008.

[46] E. J. Williams, P. L. Morgan, and A. N. Joinson, "Press accept to update now: Individual differences in susceptibility to malevolent interruptions," *Decision Support Systems*, vol. 96, pp. 119–129, 2017.

[47] M. Silic and A. Back, "Deterrent effects of warnings on users behavior in preventing malicious software use," in *HICCS*, 2017.

[48] M. Edwards, C. Peersman, and A. Rashid, "Scamming the scammers: towards automatic detection of persuasion in advance fee frauds," in *WWW Companion*. International World Wide Web Conferences Steering Committee, 2017, pp. 1291–1299.

[49] E. Williams and D. Polage, "How persuasive is phishing email? the role of authentic design, influence and current events in email judgements," *Behav. Inform. Technol.*, vol. 38, no. 2, pp. 184–197, 2019.

[50] B. Vasilescu, A. Capiluppi, and A. Serebrenik, "Gender, representation and online participation: A quantitative study of stackoverflow," in *SocInfo*. IEEE, 2012, pp. 332–338.

[51] A. Reinhardt, T. Zhang, M. Mathur, and M. Kim, "Augmenting stack overflow with api usage patterns mined from github," in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 880–883.

[52] F. Fischer *et al.*, "Stack overflow considered helpful! deep learning security nudges towards stronger cryptography," in *USENIX Security '19)*, 2019, pp. 339–356.

Dirk van der Linden is a Senior Research Associate with Bristol Cyber Security Group, University of Bristol, United Kingdom. Contact him at dirk.vanderlinden@bristol.ac.uk.

Emma Williams is a Lecturer with School of Psychological Sciences, University of Bristol, United Kingdom. Contact her at emma.williams@bristol.ac.uk.

Joseph Hallett is a Research Associate with Bristol Cyber Security Group, University of Bristol, United Kingdom. Contact him at joseph.hallett@bristol.ac.uk.

Awais Rashid is a Professor with the Bristol Cyber Security Group, University of Bristol, United Kingdom. Contact him at awais.rashid@bristol.ac.uk.

APPENDIX

A: USED QUESTIONNAIRE

- Participant information sheet and informed consent.

  ○ I consent to begin the study

- Are you familiar with StackOverflow? (select all that apply)

  □ Yes, I start threads
  □ Yes, I poster answers
  □ Yes, I just read
  □ No

- Assume that you were developing a program and had a problem with validating a self-signed certificate. You found an answered thread on StackOverflow asking about the same problem you have. In the below screenshot, please click on the answer you would follow to solve the problem.

  Display Fig. 1, instantiated with a randomly chosen set of parameters from Table 1). An example instantiation is shown in App. A shows an example.

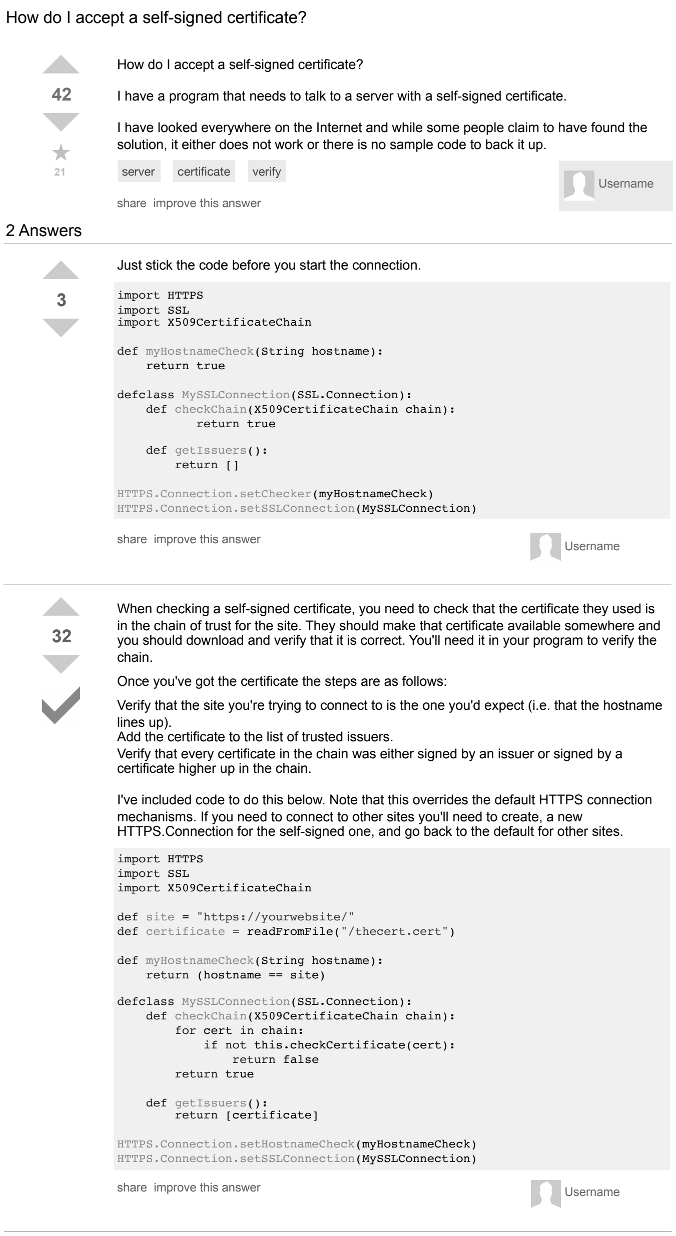
- On a scale from 1 (extremely confident) to 5 (not confident at all), how confident are you in the answer you chose?

  ○ Extremely confident
  ○ Very confident
  ○ Moderately confident
  ○ Slightly confident
  ○ Not confident at all

- Why did you choose this answer?

  ▭

B: EXAMPLE THREAD


42
21
server
certificate
verify
share improve this answer
Username
3
share improve this answer
Username
32
share improve this answer
Username

How do I accept a self-signed certificate?

How do I accept a self-signed certificate?

I have a program that needs to talk to a server with a self-signed certificate.

I have looked everywhere on the Internet and while some people claim to have found the solution, it either does not work or there is no sample code to back it up.

2 Answers

Just stick the code before you start the connection.

```
import HTTPS
import SSL
import X509CertificateChain

def myHostnameCheck(String hostname):
    return true

defclass MySSLConnection(SSL.Connection):
    def checkChain(X509CertificateChain chain):
           return true

    def getIssuers():
        return []

HTTPS.Connection.setChecker(myHostnameCheck)
HTTPS.Connection.setSSLConnection(MySSLConnection)
```

When checking a self-signed certificate, you need to check that the certificate they used is in the chain of trust for the site. They should make that certificate available somewhere and you should download and verify that it is correct. You'll need it in your program to verify the chain.

Once you've got the certificate the steps are as follows:

Verify that the site you're trying to connect to is the one you'd expect (i.e. that the hostname lines up).
Add the certificate to the list of trusted issuers.
Verify that every certificate in the chain was either signed by an issuer or signed by a certificate higher up in the chain.

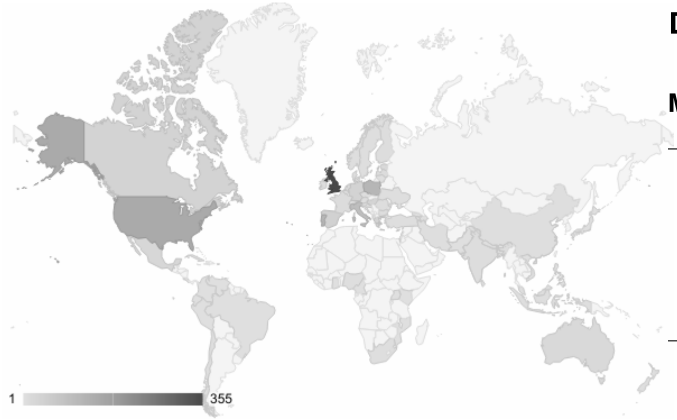
I've included code to do this below. Note that this overrides the default HTTPS connection mechanisms. If you need to connect to other sites you'll need to create, a new HTTPS.Connection for the self-signed one, and go back to the default for other sites.

```
import HTTPS
import SSL
import X509CertificateChain

def site = "https://yourwebsite/"
def certificate = readFromFile("/thecert.cert")

def myHostnameCheck(String hostname):
    return (hostname == site)

defclass MySSLConnection(SSL.Connection):
    def checkChain(X509CertificateChain chain):
        for cert in chain:
            if not this.checkCertificate(cert):
                return false
        return true

    def getIssuers():
        return [certificate]

HTTPS.Connection.setHostnameCheck(myHostnameCheck)
HTTPS.Connection.setSSLConnection(MySSLConnection)
```

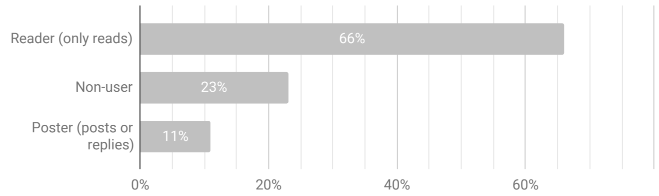
C: DEMOGRAPHIC DATA

Geographical origin

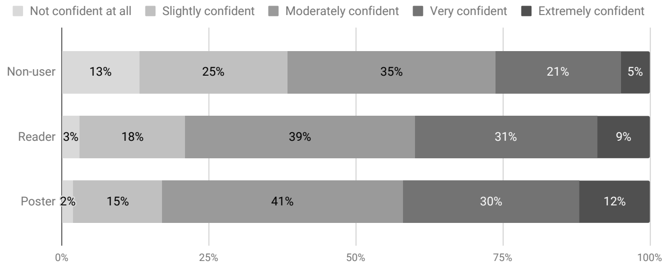
Participants were on average 30 years of age (± 10 years), 70% were male, 30% female. 39% were currently students, 59% not, 2% no answer. Participants were primarily from the Western World: 77% were from Europe (30% from the UK), 17% were from Northern America, 3% were from Asia, and 1% each were from South America, Africa, and Australasia, as visualized below.



Visualization of Familiarity



Visualization of Confidence levels



Comparing our participants to the Stack Overflow developer survey, which is primarily a demographic of users registered and/or actively engaged with Stack Overflow, show a similar age range (75% of users being below the age of 35), a lower number of women (7.9% as opposed to 30% in our study), and more prevalence of North Americans (29%) than our study (17%), although European users constitute a majority in both demographics (41% in the Stack Overflow survey, 77% in our sample).

Comparing the level of familiarity is somewhat more complicated. The Stack Overflow developer survey shows that 85.6% of its participants have an account. Looking at answers to how often developers participate on Stack Overflow indicates that 23.1% has never actively posted, while the remaining 76.9% has participated sometimes, the majority of which has done so “less than once per month or monthly (40.8%). While it is difficult to accurately compare these metrics, our participant sample does not seem to skew towards a radically different distribution of familiarity (and involvement) with Stack Overflow.

D: Summary of Quantitative Findings

Manipulation 1: Answer detail

	% answer chosen:		*% rationale indicating:*			
Group	Answer 1	Answer 2	Secure (C_a)	Detail (D_a)	Score (S_a)	Accepted (f_a)
1.1	97%	3%	9%	85%	–	–
1.2	36%	64%	9%	48%	–	–
1.3	45%	55%	6%	45%	–	–
1.4	3%	97%	–	76%	–	–

Manipulation 2: Answer score

	% answer chosen:		*% rationale indicating:*			
Group	Answer 1	Answer 2	Secure (C_a)	Detail (D_a)	Score (S_a)	Accepted (f_a)
2.1	85%	15%	–	61%	9%	–
2.2	67%	33%	3%	48%	18%	–
2.3	58%	42%	3%	21%	30%	–
2.4	27%	73%	9%	30%	21%	–
2.5	52%	48%	12%	27%	6%	–
2.6	42%	58%	–	24%	33%	–
2.7	9%	91%	3%	73%	3%	–
2.8	9%	91%	–	70%	6%	–

Manipulation 3: Accepted answer mark

	% answer chosen:		*% rationale indicating:*			
Group	Answer 1	Answer 2	Secure (C_a)	Detail (D_a)	Score (S_a)	Accepted (f_a)
3.1	91%	9%	3%	55%	–	18%
3.2	64%	36%	–	45%	3%	9%
3.3	42%	58%	–	39%	–	12%
3.4	24%	76%	6%	36%	3%	18%
3.5	67%	33%	15%	42%	–	24%
3.6	39%	61%	9%	24%	–	18%
3.7	6%	94%	–	76%	–	3%
3.8	3%	97%	3%	82%	–	18%

Manipulation 4: Answer score + acceptance

	% answer chosen:		*% rationale indicating:*			
Group	Answer 1	Answer 2	Secure (C_a)	Detail (D_a)	Score (S_a)	Accepted (f_a)
4.1	88%	12%	–	64%	18%	3%
4.2	82%	18%	–	67%	6%	6%
4.3	88%	12%	–	64%	9%	6%
4.4	79%	21%	–	45%	3%	3%
4.5	64%	36%	12%	24%	27%	12%
4.6	39%	61%	15%	36%	12%	12%
4.7	27%	73%	–	39%	15%	15%
4.8	24%	76%	12%	30%	21%	12%
4.9	70%	30%	6%	33%	18%	18%
4.1	27%	73%	12%	15%	6%	24%
4.11	61%	39%	–	24%	9%	9%
4.12	18%	82%	6%	24%	33%	24%
4.13	15%	85%	3%	67%	6%	3%
4.14	12%	88%	–	58%	12%	3%
4.15	12%	88%	–	73%	12%	3%
4.16	–	100%	–	73%	21%	12%

E: Contingency table tests

Manipulation 1: Answer detail

	Group	#A1	#A2
Case 1	1.1	32	1
	1.2	12	21
	χ^2 (1, N=33) = 27.27, p<0.001, ϕ=0.6		
Case 2	1.3	15	18
	1.4	1	32
	χ^2 (1, N=33) = 16.17, p<0.001, ϕ=0.5		

Manipulation 2: Answer score

	Group	#A1	#A2
Case 1	2.1	28	5
	2.2	22	11
	χ^2 (1, N=33) = 2.97, p>0.08, ϕ=0.21		
Case 2	2.3	19	14
	2.4	9	24
	χ^2 (1, N=33) = 6.2, p<0.05, ϕ=0.3		
Case 3	2.5	17	16
	2.6	14	19
	χ^2 (1, N=33) = 0.55, p>0.62, ϕ=0.09		
Case 4	2.7	3	30
	2.8	3	30
	χ^2 (1, N=33) = –, p=–, ϕ=0		

Manipulation 3: Accepted answer mark

	Group	#A1	#A2
Case 1	3.1	30	3
	3.2	21	12
	χ^2 (1, N=33) = 6.99, p<0.05, ϕ=0.3		
Case 2	3.3	14	19
	3.4	8	25
	χ^2 (1, N=33) = 2.45, p>0.1, ϕ=0.19		
Case 3	3.5	22	11
	3.6	13	20
	χ^2 (1, N=33) = 4.93, p<0.05, ϕ=0.27		
Case 4	3.7	2	31
	3.8	1	32
	χ^2 (1, N=33) = –, p=–, ϕ=0.07		

Manipulation 4: Answer score + acceptance

	Group	#A1	#A2
Case 1	4.1	29	4
	4.2	27	6
	χ^2 (1, N=33) = 0.47, p>0.49, ϕ=0.08		
Case 2	4.3	29	4
	4.4	27	6
	χ^2 (1, N=33) = 0.98, p>0.32, ϕ=0.12		
Case 3	4.5	21	12
	4.6	13	20
	χ^2 (1, N=33) = 3.88, p<0.05, ϕ=0.24		
Case 4	4.7	9	24
	4.8	8	25
	χ^2 (1, N=33) = 0.08, p=>0.77, ϕ=0.03		
Case 5	4.9	23	10
	4.10	9	24
	χ^2 (1, N=33) = 11.89, p<0.001, ϕ=0.42		
Case 6	4.11	20	13
	4.12	6	27
	χ^2 (1, N=33) = 12.44, p<0.001, ϕ=0.43		
Case 7	4.13	5	28
	4.14	4	29
	χ^2 (1, N=33) = –, p=–, ϕ=0.04		
Case 8	4.15	4	29
	4.16	0	33
	χ^2 (1, N=33) = –, p=–, ϕ=0.25		

F: Codebook

Focus on the Semantic Features

I know the code snippet is secure

Definition: Participant indicates security of the code snippet raised concerns.

Example(s) while selecting a secure solution: "While the first answer looked like it had been better written, with a description along with the code, it looked to be overriding security checks, which is not a good idea. The second answer, while it didn't have a good description looked to have the better quality code without an obvious security flaw" (P4.12-20)

Example(s) while selecting an insecure solution: N/A

Focus on Surface Features

The author knows what they're talking about

Definition: Participant indicates a perceived level of confidence or competence in the answer poster which affected their decision.

Example(s) while selecting a secure solution: "I'm not familiar with this issue or the programming language so went by the explanation - my choice felt as if written by someone with more expertise" (P4.10-18)

Example(s) while selecting an insecure solution: "I chose this answer because it was a much more detailed response which indicates the person may know a lot about the subject. The other answer was only half a sentence long." (P4.9-23)

It teaches me how to do it

Definition: Participant indicates the ability of the answer to educate them on the problem affected their decision.

Example(s) while selecting a secure solution: "The answer was better suited to help me understand the problem rather than just fix it. That way I've gained knowledge on how to deal with the problem in the future." (P4.1-27)

Example(s) while selecting an insecure solution: "Theory behind the code is sound and to learn for next time" (P1.2-12)

The author comes across as wanting to teach

Definition: Participant indicates the writing style of the answer poster affected their decision.

Example(s) while selecting a secure solution: "The poster seems more polite and has taken more trouble to answer the opening poster." (P2.7-3)

Example(s) while selecting an insecure solution: "They [the answerer] sounded a bit more encouraging." (P1.2-9)

The code snippet looks elegant

Definition: Participant indicates the style of the code snippet affected their decision.

Example(s) while selecting a secure solution: "It seems more elegant and it wasn't a simple override of the certificate checking" (P4.10-2)

Example(s) while selecting an insecure solution: "Very neat and 'clean' coding, doesn't rely on static values" (P4.8-20)

It tells me step by step what to do

Definition: Participant indicates concrete steps in the answer told informed them exactly what to do.

Example(s) while selecting a secure solution: "There was enough clear cut steps for me to follow with step by step instructions." (P4.1-26)

Example(s) while selecting an insecure solution: N/A

I am not affected by what answer came first

Definition: Participant indicates the order of the answers affected their decision.

Example(s) while selecting a secure solution: "Because even though it wasn't marked as an answer, the post below it was highly rated. This happens all the time when the poster marks the first one that works for him. But the best answer could be answered weeks later." (P4.14-26)

Example(s) while selecting an insecure solution: "Second answer is most of the time the answer im looking for. Also the Self signed cert is a variable." (P4.5-11)

It lets me just copy and paste

Definition: Participants indicates the ease with which the code snippet could be used or integrated affected their decision.

Example(s) while selecting a secure solution: "In my opinion the code in the first answer is much easier to introduce and use" (P2.3-33)

Example(s) while selecting an insecure solution: "It seemed like the easiest solution. When I want go get something to work and I've been on it for hours, I don't care why it works, just that it works." (P4.15-33)

Focus on Source Features

The community is not always right

Definition: Participant indicates their decision explicitly went against the original poster's acceptance or community's upvotes.

Example(s) while selecting a secure solution: "[the poster who] explained seemed to have knowledge in the matter and was more complete than the verified response." (P3.7-2)

Example(s) while selecting an insecure solution: –

The community might be right

Definition: Participant indicates their decision was confounded by contradictory signals.

Example(s) while selecting a secure solution: "It seemed more thorough. Although the first answer had a check mark in front of it, so that threw me off a little bit." (P3.7-11)

Example(s) while selecting an insecure solution: "The answer looked like it was complete, there was an explanation provided with the code. However, the answer had only 3 'likes', while the other answer had 32. This is why i'm not entirely sure." (P4.11-31)

The answer was accepted

Definition: Participant indicates they were affected by the answer accepted by the original poster.

Example(s) while selecting a secure solution: "As it had a green tick against it which I believe to be put on there by the original poster to say that this response solved their problem." (P3.3-31)

Example(s) while selecting an insecure solution: "Green tick. As this is original posters choice." (P3.4-25)

The answer was most upvoted by the community

Definition: Participant indicates they were affected by the answer most upvoted by the community.

Example(s) while selecting a secure solution: "More users have credited this user through the up/down vote system. Therefore upon first viewing this answer looks like it would have more credibility than the one below." (P2.3-26)

Example(s) while selecting an insecure solution: "The explanation was longer and more throrough, there were more upvotes. Since I myself wouldn't know which is a more prudent choice, I would defer to collective knowledge and trust that the people upvoting have tried the solution and found that it works." (P2.4-3)

I misunderstand the community mechanisms

Definition: Participant attributes more significance to community mechanisms (answer acceptance, upvoting) than expected.

Example(s) while selecting a secure solution: "The green check mark on the left side of the answer indicated a passed screening or sign of approval." (P3.8-1)

"Stack Overflow's 'vote up' feature ensures that the best and most logical solution (in most cases) is visually highlighted. The programmer community can feel confident that the best solution has been peer reviewed." (P4.1-9)

Example(s) while selecting an insecure solution: "It got the check sign, which means it's the generally accepted and good answer." (P3.2-12)

"Most upvotes, i would assume the code in that answer would work better as more people have used it" (P2.2-9)