An Analysis Of Security Discussions In Stack Overflow

Mitchell Drummer
Division of Science And Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
drumm040@morris.umn.edu

ABSTRACT

In the past few years QnA websites like Stack Overflow having been increasing significantly in popularity. Simultaneously, programming a secure application is growing increasingly complex as both developers' tools and attackers' strategies are evolving. Thousands of developers visit Stack Overflow every day for coding help, but security topics are specifically problematic on the site. In this paper I explore three published research papers about Security discussions on SO, and our findings indicate important aspects of the unreliability of the security related posts on SO.

Keywords

Stack Overflow, Security, Java

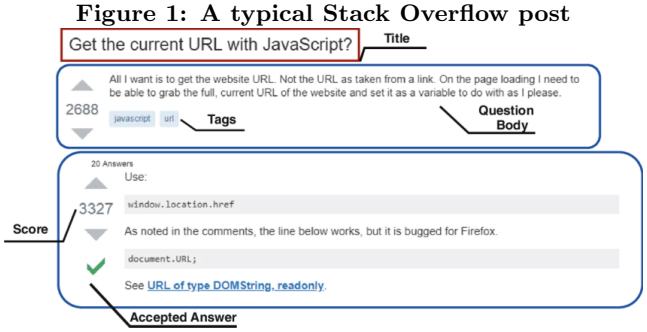
1. INTRODUCTION

In this paper I analyzed three papers related to how security is discussed on the website Stack Overflow. Stack Overflow(SO henceforth) is very popular among developers of any skill level, and since it is so commonly used, I thought I would research an aspect of the site that appears problematic: security. Since most users come to SO looking for solutions to errors in their code, users generally consider a question 'answered' if the suggested answer fixes their bug. Thus, code that works, isn't necessarily secure. Since SO is crowdsourced, this problem is further excacerbated by the fact many users may not be proficient in coding for security, as what is and isn't secure is more ambigous that just simply if the code runs. The research papers discussed also all focus excusively on Java security, so this allows me to make some general conclusions about that community. Data gathered from the three academic papers illustrate that:

1. A small, but significant portion of Java SO answers are insecure
2. The security community has a diverse set of users
3. A larger portion of SO answers are insecure.

Since we discuss how security is talked about on SO, I need to explain how the basics of the site work. This is important since I discuss certain aspects of the site's interface later in the paper, and also how users can only do certain actions based on their reputation. Stack Overflow was launched in 2008. It is a place for developers to ask and answer questions about coding. A user asks a question, then selects one answer as the "accepted" one, as shown in the example post in figure 1.

Figure 1: A typical Stack Overflow post



Users can upvote posts or comments, which gives the poster reputation points. If your answer, question, or comment was voted on it impacts your reputation as follows: if your answer is uploaded you get 10 reputation points. If your question is upvoted you earn five, if your comment is upvoted you earn two. This is ismportant since users are restricted from certain actions until they have a certain reputation score. There are four important reputation milestones to note: A user with 15 points can vote up posts. A user with 50 points can comment on others' posts. A user with 125 points can vote down posts. A user with 20,000 points gain the "trusted user" status, which allows them to edit or delete other people's posts (If those posts net a negative reputation).

The important part of this that should be emphasized is that users cannot comment on others' posts until they have been on the site for a while, which means that most comments will be by people who have asked or answered questions before. This also means that brand new users cannot comment, unlike some other crowdsourced question forums on the internet such as Quora for example.

So first I want to go over some other research that has some relevant data to what we are talking about, then we will discuss each of our papers. Our first paper is about security discussions concerning Java Library security and how discussions about them have insecure posts on Stack Overflow. When talking about them and we will delve into the comment section of the security community on the top 20 posts of all time and how users talk about Security in that section. We wil also talk about how code is shared on Stack Overflow and how often it is insecure or secure and

UMM CSci Senior Seminar Conference, May 2020 Morris, MN.

then we'll wrap up all the data together and talk about it and what all this data means together.

2. BACKGROUND

Additional research for this paper included the paper by Hossain Shahriar et al [5] . which described the different types of vulnerabilities found commonly in code, with solutions on how to avoid them. Their work was an excellent introduction to types of security vulnerabilities, especially for people who do not yet have experience coding for security. The knowledge I gained from their work helped me develop my goal for this paper: teaching other intermediate computer science students the value of security in coding, and why one should be thinking about it early on. Also relevant in understand security problems was Michael Gegick et al's [1] work on security vulnerabilities which helped me gain a key understanding of attack strategies of different types of attacks. Though it is a little old, it was still relevant and fit into the scope of this paper.

3. SECURE CODING PRACTICES IN JAVA: CHALLENGES AND VULNERABILITIES

> What are the common concerns in Java secure coding?

The research conducted by Meng et al. [4] looked at 503 Stack Overflow posts related to security, that were created from 2008 to 2016, and analyzed them. They discuss how accepted answers could sometimes include insecure code, and how the accepted answer would then be viewed by many others, causing more people to create insecure code.

3.1 Methods

All posts containing terms "Java" and "security" were gathered to create their initial pool of 22,195 posts. They then filtered less useful posts out, removed posts without code snippets, and discarded irrelevant posts.

They analyzed 503 Stack Overflow (SO) posts that were related to security and classified them into three main categories: Java Platform Security, Java EE Security, and Spring Security. There were 140 questions related to Java platform security and this category was further broken down into four categories: Cryptography(64 Questions), Access control(43 Questions), Secure Communication(31 Questions), and Other(2 Questions). There were 58 questions related to Java EE security, but this section was small enough that it was not broken into categories. There were 267 questions related to Spring Security and this category was further broken down into three categories: Authentication(225 Questions), Authorization(16 Questions), and Configurations(26 Questions).

3.2 Results

Their findings were that many of the accepted answers (the responses that the question selected as the correct answer) contained insecure code. Of the 503 posts analyzed, the following problems were found within the posted answers:
5 out of 12 posts about Spring Security's csrf() function had an insecure accepted answer
9 posts about SSL/TLS had an accepted answer to dangerously trust all certificates
3 out of 6 posts about hashing accepted vulnerable solutions
For these 17 vulnerable answers, the view count was 622,922 as of August 2017.

This is about 3% of all examined posts, but still remain significant as a result of their high view count. However, their search criteria limited the number of posts they analyzed, as they cut 22,195 posts down to 503 using them. Therefore, this number may be inaccurate in describing how often developers create and propagate vulnerable code.

3.3 Conclusion

More than half of all the examined questions were about using Spring Security, and Meng et al. noted about Spring that

> The challenges were due to incomplete documentation, as well as missing tool support for automatic configuration checking and converting.

Their finding was that Spring Security had many issues due to incomplete documentation which is why more than half the posts were about Spring Security. The researcher's analysis was that the incomplete documentation was the reason for most people asking these questions because otherwise they wouldn't have went to Stack Overflow to ask them. They would have just gone to the documentation.

Let's briefly look through the 17 posts that contain insecure code. nine out of ten of all the SSL/TLS (Secure Socket Layer and Transport Layer Security) questions were about bypassing it completely. If you see HTTPS at the beginning of a URL in your browser, that means it uses SSL/TLS, so the newer version TLS is the more secure one. If people talk about SSL, it is depreciated and it is no longer secure, even though it used to be secure a few years ago. There's people asking questions about it to this day which is interesting because it was insecure for quite a while now. Those posts were about bypassing it completely and 5 out of 12 of the cross-site request forgery posts were about disabling it completely. CSRF protects your website against cross-site request forgery attacks, so they were deciding to disable the security features that prevented people from doing that. It's interesting that all 5 people didn't know that the whole point of it was to prevent that. So the researchers summed this up as there are misleading indicators on a lot of these posts. People would accept the answer which signals to other users that it's a good answer even though it was insecure. A lack of knowledge by the poster is also partially to blame for the insecurity. In addition, if an answer is posted by a user with a lot of reputation, they're more likely to believe that answer which propagates the insecure code shared more often.

The most important finding of the data presented in their paper can best be summarized in their recommendations section:

> For Developers, conduct security testing to check whether the implemented features work as expected. Do not disable security checks (e.g., CSRF check) to implement a temporary fix in the testing or development environment. Be cautious when following SO's accepted or reputable answers to implement secure code because some of these solutions may be insecure and outdated. For SO administrators, they may consider adding

warnings to the posts with known vulnerable code, as these posts may mislead developers.

4. AN ANATOMY OF STACK OVERFLOW POSTS

The researchers [2] utilized data from their earlier study, where the top 20 questions of all time related to security were collected and analyzed. The researchers noted trends in answer, question, and comment sections of each of the question threads.. They mapped user activity between each post and determined users rarely interacted with different questions. They also discussed that the community around security specifically seems to be weak.

“”A slightly greater sense of security related activity can be seen by looking at an overview of information for Answer providers drawn from the wider site. Only one answer provider, EpicRainbow, identifies within their profile description as having an interest or expertise in security. However, for half of the answer providers, the top 3 highly voted tags associated with the answers suggest that other Stack Overflow users recognise and regard participation these developers make in posts that include security as a tag.”’

4.1 Methods

They gathered the top 20 security questions along with the comments section and answers. These 20 posts had the highest total up votes of all time for security questions. These posts were easily collected using SO’s filter feature to search all posts including the security tag, then filter by number of upvotes descending. The researchers then collected all the comments for these questions to use for their data. There are 250 unique users making 364 comments and it’s broken down as the following: 197 users left a single comment, 32 left two comments, 10 users left three comments, and 11 left more than three comments. They found that there was a low interconnectivity in the community; most just kept to one post. If they had written more than one comment, it was often in reply to someone in a conversation witth them for example.

4.2 Results

> How do developers on Stack Overflow engage with one another when dealing with issues related to security?

The researchers mapped a web of the interconnections between each of the 20 posts analyzed, and is best illustrated through their diagram in figure 2.

The takeaway from this data is that the most popular security answers on SO are not all answered, or asked by the same people, indicating each post to have a diverse amount of information.

Additionally, the researchers then looked at the nature of how users responded to others’ posts and comments. They noted four types of usual interactions on the set of posts they looked at. Pairs: A majority of users commented back and forth only to each other. Three-part exchange: users that left exactly two comments often participated in a comment structure that begins with a question or comment, another user responds with an answer, and the first user responds back to give thanks, or to acknowledge the other user’s response. Multi-part exchange: Often between two people,

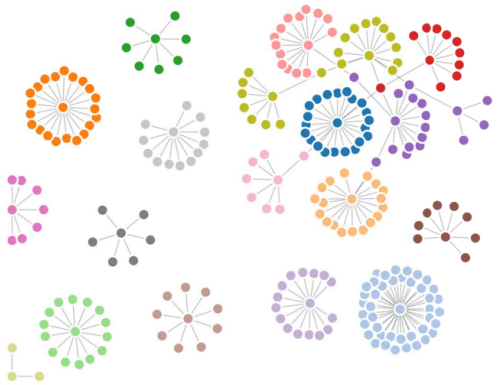


Figure 2: This image was included in Tamara Lopez et al’s original paper with the following caption: This image depicts commenting activity for answers. Each answer is represented by an individiual cluster. Dots around the center point of each cluster represent users who commented at least once within an answer stream. Within this set, answer streams include varying numbers of comments. Most commenting activity is isolated; few users comment on more than one stream.

this a series of questions and responses. Broadcasts: multiple developers all comment on a single topic.

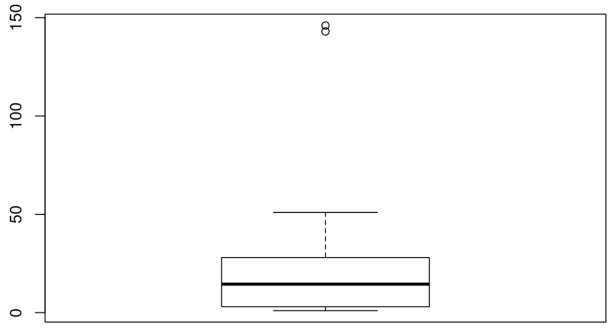


Figure 3: Number Of Questions Answered By Users Who Commented

This illustrates that a majority of commenters only answered a few questions about security, 75% of users had answered less than 19 questions about security, whereas two users answered 143 and 146 questions about security. This can be seen clearly in the boxplot shown in figure 3.

4.3 Conclusion

This illustrates that the majority of accounts only answered a few questions about Security in total. The data they had showed that 75% of all users in the posts answered less than 19 questions about security and the data was very spread out. There were also two outliers in the data: one user answered 243 questions and another answered 146 questions about security. Whereas most other users answered less than 19. The conclusion that they made was that there wasn’t really a community focused specifically on security where people in the comments had a specific interest in se-

curity. They found most people were just answering other questions on the site as well and came to this section not because they knew a lot about security or were interested in it, possibly because of the gamification of reputation on the site for example.

However, due to only looking at the top 20 security posts, this study says nothing on the makeup of the rest of the the less popular security questions on SO, so this data may not be representative of the whole security section on SO. The data still is useful though since the less upvotes a post gets, the less likely it will contain many comments, so looking at the comment sections of the most popular questions will give more data on interconnections than a post with few or no comments.

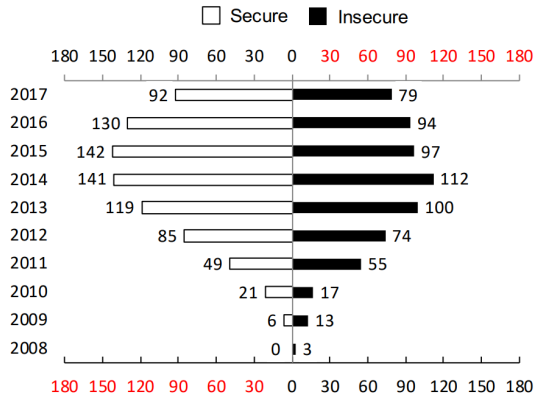


Fig. 4: The distribution of posts over during 2008-2017

5. HOW RELIABLE IS THE CROWDSOURCED KNOWLEDGE OF SECURITY IMPLEMENTATION?

How prevalent are insecure coding suggestions on SO?

Mengsu Chen et al. [3] performed an in-depth investigation of the popularity of all secure and insecure coding suggestions on Stack Overflow and the community activities around them.

5.1 Methods

They gathered 3121 posts with code Snippets included on Stack Overflow. Posts must have included code in any language to be included in the data. Once they collected the Snippets, they found the Snippets were repeated multiple times; users were copying the same code and posting it on two or more different questions or answers. They further categorized them into clone groups to analyze the data. They had 953 clone groups, which means that on average there's about three posts that had the same code snippet. They found that 587 of the groups out of the 953 were secure and 326 were insecure out of out of this group, which means 34.2% percent of the groups were insecure.

5.2 Results

In their paper, they collected answers on security related SO posts that contained code fragments and determined which posts included duplicated code. They found the following groups of code: Among the 953 clone groups, there were: 587 groups of duplicated secure code, 326 groups of similar insecure code, and 40 groups with a mixture of secure and insecure code snippets. covering 3121 total code snippets.

We can see how these clone groups are distributed by year and whether or not they are secure in Figure 4.

The data provided by the researchers showed that insecure posts had a higher average of upvotes, number of comments, favorites, and views. They state that this implies that insecure posts are more popular, and as such, users should not rely on these measures alone to determine whether or not a post is secure. This finding also indicates that at least more than half of user attention is focused on the site's insecure answers, which is a major problem for the security community if that is indeed true.

Another finding was that the site's trusted users (people who had more than 20,000 reputation) had 34% of their posts include vulnerable code. The researchers hypothesized that this was because users could gain the trusted status by posting in non-security areas, so users on average would have little more knowledge than the non-trusted users posting in the community. They also found that there was no correlation between how many times a code snippet was copied and its security. Just because a snippet was clone many times did not indicate that it was secure or not.

5.3 Conclusion

From this data, they also found that insecure posts on average are more popular receiving more upvotes views and comments, which is counterintuitive because you'd expect the ones that are secure to be more popular, but that was not the case as they found. And also further was that the trusted users in the site had 34% of their posts include vulnerable code. They found that a trusted user ironically cannot be trusted. Additionally there was no correlation between how many times a snippet was copied and how secure the post was. They also found, as the first paper did, a very large problem with the SSl/TLS category, where insecure posts dominated that category significantly. This allows the conclusion to be made that users still need help with SSL and TLS to this day.

6. COMBINING THE RESEARCH

So to recap our first paper had 503 posts, 17 of which were insecure, which is about 3%. Our second paper in the comment section Illustrated there was not a very clear community around security. It was gathered up of just general users. And for our third paper, out of 3121 posts thirteen hundred and nineteen were insecure, which is about 42 percent.

In the paper by Meng et al" Secure Coding Practices in Java: Challenges and Vulnerabilities" there were 17 insecure answers out of the 503 answers included, which was about 3%, and in Tamara Lopez et al's paper "An Anatomy of Security Conversations in Stack Overflow" 644 out of 1,429 answer posts were insecure which was about 45%. The great discrepancy between these two figures could be explaied by method of collection. Meng et al. manually categorized the 503 posts after some automatic filtering, while Lopez et al. gathered data by searching for duplicate code fragments. Since Lopez et al. had more posts in their sample, their data may be more reliable for creating an accurate figure on

how much insecure code is being spread. However I wish to point out that I have two data sets that seem to indicate opposite things. One data set has a very low percentage of insecurity, while the other is very high. This illustrates that we need more data before a sound conclusion on how severe the problem of security truly is on the site. Overall the research seems to indicate that there is at least some issues of security on SO, and I would like to share the following tips based on what I personally learned from all the data shared thus far:

Code included in posts on Stack Overflow are not always secure. Stack Overflow may be good for figuring out how to fix a problem with your code, but don't listen to everything they have to say about security. Be cautious if you're trying to implement something that has to do directly with security in the case that they might suggest a wrong solution.

Stack Overflow posts are not updated as security evolves because if something is found to be insecure people don't go back and update their post they made years ago on something about how to implement SSL. They're usually either just unaware that SSL was found to be insecure, or simply forgot their post existed in the first place. This could be a suggestion to SO in general: Moderators should be able to flag posts as outdated, or remove them entirely to stop the propagation of insecure coding practices.

Lots of these papers mentioned that secure socket layer was insecure and outdated and people had lots of trouble with implementing it in general, but a finding I had was that secure socket layer was outdated versus TLS. They both do relatively the same thing. In case any of you in the future are trying to implement https encryption, try to use TLS rather than SSL in case that ever comes up. The main problem on Stack Overflow is the fact that checking whether or not something you share on Stack Overflow is secure or not is harder to do than just checking whether something you share runs or not. Elsewhere on the site, users suggest code and the question asker accepts that answer if the code fixes their problem, and they tend to do the same with security suggestions. If someone suggests to trust all certificates, they're going to go and try that, and if it fixed the code, they might not think twice about what that does to the safety of the app as a whole. They just care about fixing one part, not realizing they destroyed the whole app. So my main advice to you here is to keep Security in the back of your mind forever. If you are ever coding something that has internet access, you may forget that many attackers exist, and will take advantage of your app given the chance. If you're using Wi-Fi in your app in the future, just make sure you are thinking about security. That's one of the main things for any programmers beginning to learn right now. If you ever come on Stack Overflow for anything involving security and you see something like trust all certificates or disabled authentication, that's a red flag to not take that advice.

7. CONCLUSION

Stack Overflow is a great place to go when you need help coding, but when it comes to help with security, you should take their advice with a grain of salt or you better look elsewhere.

8. REFERENCES

[1] M. Gegick and L. Williams. Matching attack patterns to security vulnerabilities in software-intensive system designs. In *Proceedings of the 2005 Workshop on Software Engineering for Secure Systems—Building Trustworthy Applications*, SESS '05, page 1–7, New York, NY, USA, 2005. Association for Computing Machinery.

[2] T. Lopez, T. Tun, A. Bandara, M. Levine, B. Nuseibeh, and H. Sharp. An anatomy of security conversations in stack overflow. In *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Society*, ICSE-SEIS '19, page 31–40. IEEE Press, 2019.

[3] 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 *Proceedings of the 1st International Workshop on Security Awareness from Design to Deployment*, SEAD '18, page 26–32, New York, NY, USA, 2018. Association for Computing Machinery.

[4] N. Meng, S. Nagy, D. D. Yao, W. Zhuang, and G. A. Argoty. Secure coding practices in java: Challenges and vulnerabilities. In *Proceedings of the 40th International Conference on Software Engineering*, ICSE '18, page 372–383, New York, NY, USA, 2018. Association for Computing Machinery.

[5] H. Shahriar and M. Zulkernine. Mitigating program security vulnerabilities: Approaches and challenges. *ACM Comput. Surv.*, 44(3), June 2012.