# An Investigation of Online Reverse Engineering Community Discussions in the Context of Ghidra

Daniel Votipka
*Tufts University*
*dvotipka@cs.tufts.edu*

Mary Nicole Punzalan, Seth M. Rabin, Yla Tausczik
and Michelle L. Mazurek
*University of Maryland*
*{mpunzala,srabin,ylatau,mmazurek}@umd.edu*

***Abstract*—Reverse engineering is a complex task. As with many other expert tasks, reverse engineers rely on colleagues and the broader reverse engineering community to provide guidance and develop knowledge necessary to achieve their goals. For example, it is common for reverse engineers to reach out for help to understand and effectively use new tools. Thus far, however, there has been limited investigation of the way knowledge is developed in this community and new tools are adopted. This paper takes a first step toward understanding reverse engineering community dynamics around tool adoption, using the release of the National Security Agency's Ghidra reverse engineering framework as a point of focus. In this paper, we review discussions about Ghidra to identify what features reverse engineers are most interested in, how reverse engineers develop knowledge about Ghidra together online, and whether these dynamics differ between forums.**

**In total, we analyze 1590 reverse engineering discussions between 688 reverse engineers over 3 forums (i.e., Twitter, Reddit, and StackExchange). Our results suggest reverse engineers are most interested in features that allow them to customize Ghidra. We also observe limited evidence of collective sensemaking on the forums, with few reverse engineers participating in multiple discussions threads and most acting as either knowledge producers or consumers. Finally, we found that the forums operated similarly, but Twitter was most often used to announce information (e.g., tutorial links, tool overviews, vulnerabilities in Ghidra) and reverse engineers used StackExchange mostly to get support for specific problems. Reddit acted as a middle option. Based on these results, we make recommendations to improve reverse engineering tool development, improve community participation during adoption, and suggest directions for future work.**

***Index Terms*—reverse engineering, sensemaking, documentation, online forums**

## 1. Introduction

Successful performance of several cybersecurity functions, including vulnerability discovery and malware analysis, requires security professionals to be proficient in software reverse engineering (We refer to this task as reverse engineering and its practitioners as reverse engineers, for brevity) [1], [2], [3, pg. 5-7].

Reverse engineering typically requires significant knowledge and experience, due to the complex and time-consuming nature of the task. [4], [5]. For example, Yakdan et al. found reverse engineers required 39 minutes on average to answer common malware-analysis questions on small (less than 150 lines of code) decompiled code snippets [5].

As is the case for most expert tasks, reverse engineers rely on support from colleagues and the broader reverse engineering community to answer questions and develop expertise [1]. A common example of this community-supported knowledge development occurs as reverse engineers attempt to adopt new tools. Significant effort by academia, industry, and the practitioners themselves, has gone into simplifying the reverse engineering process through tool support [5]–[24]. However, due to the ad-hoc development of many of these tools and—like much other modern software—limited documentation support, reverse engineers regularly reach out to others for advice and perspective. This includes answering questions regarding for which specific tasks a new tool would be beneficial to adopt and how to use the new tool, often through online information-sharing platforms like StackExchange (SE) and Twitter. This community discussion is also essential for reverse engineering tool extension, as many tools are designed with the power-user in mind, due to the complexity and variability of reverse engineering. That is, the tool developer provides a basic foundation, but allows the community to personalize the tool to their needs through customization.

While this community aspect of reverse engineering tools has a significant impact on tool adoption and use, there has been little work investigating the community's dynamics. With a better understanding of the community, we believe better tools could be developed to fit expressed user needs, tool developers could more effectively engage with their user base, and users could be more effectively directed to support.

A possible approach to answering this question would be to look to similar research into the software engineering community's online behaviors. Like reverse engineers, software engineers also perform complex tasks (e.g., programming) and rely on online resources and Q&A sites for support and sensemaking [25]–[27], but significant research has been conducted looking at how software engineers work together online [28]–[34]. However, we believe reverse engineers are distinct in several ways that could impact community behaviors and necessitate independent research. For example, the reverse engineering community is smaller and typically more privacy and security conscious. This could make reverse engineers more isolated and cautious or more tightly connected and cooperative, because of the community's size and shared interest. Additionally, there are fewer tools and languages used, potentially reducing the community segmentation common in software engineering.

In this paper, we seek to understand the reverse engineering community's adoption of a new tool by focusing on online communication around a unique event of interest, the National Security Agency's (NSA) release of Ghidra [35]. Ghidra is a software reverse engineering framework including a decompiler, disassembler, and debugger, all of which can be extended through Ghidra's scripting support. Ghidra's release is unique in that it became available after years of internal development, use, and revisions at the NSA, as opposed to a traditional release process which typically adds only a few features at a time with minimal real-user testing. Ghidra is also an open-source product, providing capabilities equivalent to its expensive competitors (e.g., IDA Pro). Finally, Ghidra's affiliation with the NSA's mystique as a shadowy hacking organization also likely piqued many reverse engineers' interest.

Perhaps because of all these factors, Ghidra's release did draw significant attention from the community in the form of talks at popular security conferences [36]–[40] and press attention [41]–[50]. In addition, because of the amount of attention and extensive set of features, we expected Ghidra's release would produce significant tool discussion in online forums, providing a unique window into community tool adoption dynamics.

To explore these community dynamics, we analyze 590 reverse engineering discussions between 688 reverse engineers from three forums (i.e., Twitter, Reddit, and SE) collected over a six-month time period starting at Ghidra's release announcement (two weeks prior to its actual release). Through our analysis, we sought to answer the following research questions:

> **RQ1**. What tool features are reverse engineers most interested in? Do forums' discussions cover all tool features?

Ghidra is a broad framework providing many common features required by reverse engineers. The discussion around its release offers potential insights into the features most important to reverse engineers, indicating the features which should be prioritized in future tool development.

Conversely, we sought to understand whether users rely on the community to provide sufficient support for all tool feature-related questions. If not, is there any bias in the features questions receiving responses? Answering this question can help prioritize tool developer documentation effort to fill gaps in community support, similar to prior work studying documentation for API support [32], [33].

> **RQ2**. How is knowledge about a new tool shared and developed among reverse engineers?

Our second question aims to better understand the reverse engineering community itself. How is the community organized and how do participants behave when considering a new tool? Is it dominated by a few experts or is knowledge developed collectively through a broad set of members? Additionally, is the community open and welcoming to anyone who wants to provide input on the new tool? Understanding these dynamics can guide developer interactions with key players in the community and suggest ways to make the community more inclusive. To answer this question, we draw on methods and observations from studies of other knowledge-sharing networks [51], [52].

> **RQ3**. How do the specific forums used impact community adoption behaviors?

Finally, there are several online forums reverse engineers use to discuss their tools, ask questions, and share their opinions. The uniqueness of Ghidra's release provides us the opportunity to compare community tool adoption behaviors across forums as discussions on each began at the same time. Therefore, our final question seeks to identify any characteristics unique to each forum and how they work collectively to help the community investigate a new tool.

We found that reverse engineers discuss a broad variety of Ghidra's features, but mostly focus on features which allow them to customize Ghidra and on Ghidra's decompiler. When asking questions, most problems posed by reverse engineers are answered across all forums, though more active reverse engineers are more likely to have their questions answered.

Considering the way knowledge about a new tool is formed, there is limited evidence of collective sensemaking. Instead, most reverse engineers either ask or answer questions, with few participating in both sides of knowledge sharing process. Further, few reverse engineers participated in multiple discussion threads, and very few participated in multiple forums, though this is an under-approximation as our process for cross-forum account matching is limited. This limited interaction contrasts with behaviors observed in the software engineering community where participation and back-and-forth conversations were common [30], [53], [54]. This may indicate that the community, at least in the adoption phase, is not welcoming. However, we did not observe a large number of negative comments and they were mostly made by less active reverse engineers.

Finally, we observed that discussions across forums generally were similar, though separate, as few reverse engineers participated in multiple forums. The biggest difference between forums was related to the type of discussions. Twitter was most often used as a platform to announce information (e.g., links to tutorials, vulnerabilities identified in Ghidra). Whereas, SE was almost exclusively used by reverse engineers to ask specific questions, which was expected as SE was designed specifically as a Q&A platform. Reddit evenly mixed these two types of discussions. Additionally, conversations on Twitter were most likely to be isolated, i.e., few conversations shared participants.

From these findings, we distill recommendations to improve tool development, support developer interaction with the community around adoption, and guide future research.

## 2. Related Work

While little work has investigated the reverse engineering community, there has been a growing body of research aimed at understanding their processes and many researchers have studied a related community—software engineers. In this section, we review related work in these two areas.

### 2.1. Understanding reverse engineers and their needs

Multiple qualitative studies have sought to understand the processes and mental models central to reverse engineering. In our prior work, we compared experts' and novices' descriptions of their reverse engineering process through interviews, finding both groups followed roughly the same steps [1]. Multiple studies improved on this work by observing reverse engineers in action [55]–[57] or reviewing artifacts they generated [2], identifying specific questions reverse engineers ask and strategies followed. Cowley interviewed 10 reverse engineers, identifying necessary skills and professional development levels. Finally, Nosco et al. considered the processes of reverse engineering teams, comparing different strategies for coordination [58].

Other work has considered the tools (both technical and organizational) reverse engineers use. Fang et al. surveyed reverse engineers to identify the types of automation they use, finding that they preferred dynamic over static analyses [59], [60]. Summers et al. investigated how reverse engineers deal with ambiguity in their reviews, observing that they rely on discussions with others and visualization techniques (i.e., mapping system semantics on a whiteboard).

Our research builds on this work broadening the scope beyond the individual or team to consider the needs and dynamics of the broader reverse engineering community, focusing on tool adoption.

### 2.2. Software engineering and forums

Software engineering is very similar to reverse engineering. In fact, many software engineers perform reverse engineering tasks when trying to comprehend code written by other engineers for debugging, maintenance, or modification purposes. However, in contrast, there has been significant research studying software engineers' use of forums, much of which has aligned with our main research questions.

**Software engineers as "social engineers".** Across several surveys and interviews, researchers have found software engineers commonly rely on forums, such as mailing lists and Q&A platforms (specifically SE and Stack Overflow (SO)), for asking questions and identifying solutions, both generally [25]–[27] and for security-specific issues [61]–[65]. Storey et al. coined this phenomena the rise of the "social engineer" who actively participates in online documentation of crowdsourced socio-technical content [53].

In addition to common Q&A sites, other work has found that software engineers engage and share knowledge through micro-blogging sites like Twitter [28]–[30]. Singer found that software engineers use Twitter for sharing and obtaining the latest news and staying up-to-date with relevant technology [66]. However, researchers also noted Twitter was not an ideal medium for knowledge sharing as software engineers struggle to sift through irrelevant tweets. Additionally, engineers may find it hard to maintain a relevant network of users to follow as people may switch topics of focus or stop providing relevant information, meaning the engineer would need to seek out other relevant users to follow.

**Forums as crowd documentation (RQ1).** With the increasing social nature of software engineers, other work has investigated the types of features most commonly asked about on forums and whether they provide sufficient answers to those questions. Several studies have considered the types of questions asked on SO and identified trends in which are most commonly answered, regarding general software development [67]–[74] and security [75], [76] and privacy [77] issues. MacLeod et al. looked at the types of information software engineers share using YouTube videos, finding that videos are useful for sharing knowledge and demonstrating experiences [31]. Parnin and Treude [32] reviewed blog posts about the jQuery API to understand whether all functions of the API were covered. Perhaps the most in-depth study of forums as crowd documentation was completed by Parnin et al. who reviewed function coverage for three APIs (GWT, Android, and Java) on SO [33]. They found that the crowd can provide good coverage given enough time and provide several more examples per function than traditional documentation. However, the crowd may need to be directed toward less popular topics (e.g., accessibility and DRM).

**Community dynamics in forums (RQ2).** Other researchers have investigated how software engineers interact on forums and the community dynamics of knowledge sharing. Prior work has looked at follower relationships on Twitter [29] and GitHub [78], observing how knowledge is shared between individuals. Bougie et al. compared Twitter behaviors of three communities of software engineers to the broader Twitter user base [30]. They found that software engineers were more likely to have back-and-forth conversations instead of simply making announcements with little interaction. Lopez et al. observed similar behaviors when reviewing conversations between developers about security issues [54]. Specifically, they found developers use SO to actively connect to others, help each other find solutions, and foster and share knowledge. Finally, Posnett et al. looked at software engineer behaviors in SE to understand whether expertise—as measured by number of questions answered—develops over time in the community [34]. They found that user expertise does not increase with time spent in the community; experts join the community as experts, and provide good answers from the beginning.

**Comparison of forums (RQ3).** Finally, prior work in software engineering has considered differences between community behaviors across forums. Vasilescu et al. considered the relationship in developer behaviors between Github and SO, looking at whether question asking behaviors on SO are coordinated with development behaviors in Github [79]. Squire observed four case-study development projects as they transitioned from mailing lists to SO as their primary vector for Q&A support. Squire compared question response time and software engineer participation between the two forums, finding both metrics improved after moving to SO—even though two projects chose to move back to mailing lists [80]. Similarly, Vasilescu et al. looked at differences in discussion participation about the R programming language between SE and a related mailing list (i.e., r-help). They found software engineers were generally more active on SE, though the most active individuals were active in both forums [81].

Our research builds on this prior body of work from software engineering, borrowing and adapting methods to address differences between software engineers and reverse engineers. For example, while much of the prior work on software engineers studies their use of APIs, reverse engineers generally rely on tools like Ghidra which provide both a base set of functionality and an API for feature extension. This less-defined feature set required us to modify tagging approaches used in prior work. Additionally, due to the unique nature of Ghidra's release, we are able to perform a direct comparison across a broader set of forums than prior work in software engineering.

# 3. Methods

To identify the reverse engineering tool features reverse engineers are most interested in and understand the tool adoption dynamics of the reverse engineering community within and across forums, we sought to collect all Ghidra-related forum discussions over a six-month period starting two weeks prior to Ghidra's initial release, March 5th, 2019, and ending on 31 August, 2019. The starting date reflects when Ghidra's release was announced; this allows us to include discussion of the upcoming release. In total, we collected 3529 Ghidra discussion threads from Twitter, Reddit, and SE. We performed a rigorous *iterative qualitative coding* [82, pg. 101-122] of a random sample of 343 threads[1], including 1590 conversational actions and 688 unique authors. Our study was reviewed and approved by our universities' Institutional Review Boards.

This section describes how we collected forum threads, our codebook development, and the quantitative analyses we performed on the coded data.

## 3.1. Data Collection

As a first step, we sought to collect all reverse engineering discussions of Ghidra across several popular online forums. We began by performing preliminary searches of forums considered in prior work on software engineering. This review included Twitter and SE. We also considered Reddit because some reverse engineers reported getting information from this forum in our prior studies [1].

As we performed our preliminary review of these forums, we observed participants advertising two public Slack workspaces for further discussion. Unfortunately, we were not able to get organizer support to collect data from one of the workspace and participation was very limited in the other workspace (i.e., 27 participants with two making up 63% of all conversations).

Table 1 gives the final list of forums from which we collected Ghidra discussion threads. For each forum, we collected all threads over our time period of interest (i.e., 5 March to 31 August 2019) which were tagged #ghidra or contained the keyword "ghidra," indicating some textual relationship with the Ghidra tool. We also included all tweets by and mentions of the @Ghidra_RE account on Twitter. This account was created shortly after the public announcement of Ghidra's release and indicated that it would be a hub for sharing information about the tool, with its bio stating "Here you can get your ghidra tips and updates." It was presumed at the time that this was an official Ghidra account created by the Ghidra developers and was treated as such by the community. However, it was determined after data collection that the account was not directly affiliated with the NSA's Ghidra team, though it continues to provide useful information about the project. There is no official Twitter account associated with the Ghidra developers. Tweets were collected using the basic Twitter API, which gives the most recent tweets from all the tweets containing the target hashtag, account, and keyword.

In Reddit, we also collected all posts on the ghidra subreddit (i.e., r/ghidra), which claims to be a "Community dedicated to discussion about the National Security Agency's reverse engineering framework, Ghidra."

Using this search, we identified 3464 threads on Twitter, 65 on Reddit, and 53 on SE. Due to the amount of time required to manually review and code each thread, we chose to perform a complete review of the Reddit and SE threads, but reviewed a random sample of about 8% (273) of Twitter threads.

We included 338 of 391 (86%) total threads across all forums that specifically included direct discussions of Ghidra's features. The remaining 14% of threads were excluded for several reasons. Nine threads were removed because they were actually discussions about the three-headed dragon from Godzilla mythology [83] [2]. Many other threads (N=8) discussed the QAnon conspiracy theory. These conspiracy theorists believe—or, at a minimum, post online—that Ghidra was made public through the work of Q to allow ordinary citizens to combat the "deep state." We also removed 12 threads that simply aggregated information from other threads (e.g., posting statistics for the number of times the #ghidra tag was used) and 7 threads which mentioned Ghidra in passing, but did not include any specific details about Ghidra features (e.g., "IDA Pro, BinaryNinja, and Ghidra are reverse engineering tools"). Finally, we removed 17 threads whose content had been deleted by the authors or the forum by the time of our review.

Finally, to allow for comparison across forums, we attempted to link account names for authors who participated in multiple forums under different aliases. For this, we manually reviewed the account profile for each author in our sample, looking for any links to accounts on other forums. Whenever alternate aliases were identified, we linked the two aliases by replacing both with a single unique author ID. Also, we assumed any accounts with matching handles indicated they were owned by the same person. This could possibly lead to incorrect matches. However, we expect the bias in account matching to be more toward recall, i.e., missing that two handles are owned by the same person, instead of precision, i.e., accidentally assuming the same handle means the same person. In our review of reverse engineering accounts we found they commonly used the same handle across forums and generally used unique names to identify themselves. Also, due to the small number of cases where matches were

[1] All threads from Reddit and SE and a random sample of about 8% of Twitter threads. See Section 3.1 below for more details.

[2] Ghidra is named after the monster from Godzilla, but luckily, the NSA developer who named Ghidra did not know the correct spelling, simplifying our disambiguation problem.

| Forum | Search Methods | Collected / Related / Analyzed | Convo. Acts[1] | Authors |
|---|---|---|---|---|
| Twitter | #ghidra tag, @Ghidra_RE tweets and mentions, "ghidra" keyword | 3455/ 3412[2]/ 230 | 842 | 443 |
| Reddit | #ghidra tag, r/ghidra posts, "ghidra" keyword | 65/62/62 | 317 | 116 |
| SE | #ghidra tag, "ghidra" keyword | 53/51/51 | 431 | 136 |
| Total | – | 3573/ 3525/ 343 | 1590 | 688[3] |

[1] The number of comments or replies in each discussion thread.
[2] Unrelated threads were only removed after randomly sampling, so this is an overestimate.
[3] Some authors participate in multiple forums.

TABLE 1: Forum data collection search methods and statistics

made only based on account name matches (N=3), any error due to this over-approximation is inherently limited.

## 3.2. Qualitative Coding

After collecting all relevant threads, we qualitatively coded each across three dimensions: features discussed, conversational actions, and sensemaking characteristics. Table 2 summarizes each dimension and each full codebook is given in Appendix A.

For each dimension, we began with an initial codebook taken from similar prior work or based on relevant pre-existing categorizations. We discuss each codebook in detail below. Two researchers independently coded threads in groups of 30 using the initial codebooks and allowing additional codes to emerge from the data. After completing a batch of threads, the researchers met to compare codes, resolve disagreements, and update the codebooks as necessary (re-coding previously coded threads). When comparing codes, we calculated Krippendorff's alpha ( $\alpha$ ) using the ReCal2 software package [85] to measure inter-coder reliability. We chose to use Krippendorff's alpha since it is a conservative measure which accounts for chance agreements [86]. This process was repeated seven times until a reasonable level of reliability was reached for each variable—Krippendorff recommends the threshold $\alpha > 0.80$ [86]. The remaining threads were divided between the two researchers and coded independently by a single researcher.

**Features (RQ1).** For our first research question, we needed to determine the features in Ghidra discussed in each thread. Prior work in feature coverage has typically focused on APIs, leading researchers to search for function names in the discussion [32], [33]. However, because Ghidra is a framework of tools and extensible components, this approach is not possible. Fortunately, the owners of the official Ghidra GitHub repository provide a thorough list of feature tags which they use to indicate the relevant feature for each bug submission [84]. We used this list as a starting point for developing our feature codebook. To determine specific definitions for each feature (given in Appendix A), we reviewed the official Ghidra documentation,[3] public talks given by the Ghidra developers [87], [88], and the content of the issues tagged on the Ghidra github page. Additionally, two external Ghidra experts reviewed our feature definitions and confirmed they matched their understanding of the tool and covered the breadth of Ghidra's features.

In addition to features covered in the GitHub tag list, we allowed labels beyond Ghidra-specific features to emerge from the data. This included comparisons to other reverse engineering frameworks, presentations of educational materials, and discussions of Ghidra vulnerabilities.

Finally, we performed *axial coding* of the resulting features, grouping them into related feature areas [82, pg. 123-142]. This grouping was performed to support later quantitative analysis of higher-level themes in the types of features reverse engineers consider most often. Definitions for each feature area are given in Appendix A.

**Conversational Acts (RQ2).** To determine how knowledge is shared between reverse engineers, we also needed to consider *how* discussions are carried out. Because the dynamics of conversational actions have been well studied previously and are not specific to Ghidra, we chose to rely on previously existing codebooks. Specifically, we used the codebook for thread types developed by Mamykina et al. from their review of communication within an online diabetes community called TuDiabetes [51] and the discourse acts codebook was taken from Zhang et al.'s categorization of general online discussions developed based on over 9,000 threads [52].

Using the process demonstrated by Mamykina et al, we sought to identify the main reason for the thread (e.g., for the initiating author to ask for help understanding a particular topic) [51]. This allows us to identify how the forums are used (e.g., Q&A support or personal promotion). We also coded the type of discourse action for each comment in the thread, using the codebook established by Zhang et al. [52]. This presents an indicator for how knowledge is developed (i.e., whether authors build off of prior responses) and the openness of the forum (i.e., whether comments or questions attract negative responses).

**Sensemaking (RQ2).** Finally, we sought to directly measure whether elements of collective sensemaking were observed in each thread. Collective sensemaking occurs when community members share information, building meaning and knowledge structures together [89]. Again, we relied on prior work, taking our three levels of sensemaking from Mamykina et al. [51]. At the most basic level, we considered whether an author appeared to reflect on and react to a prior statement by another author, indicating information had flowed from one author to another. Next, if we observed two authors responding back-and-forth to each other, we considered this lateral engagement, indicating a two-way information flow between authors.

[3] https://github.com/NationalSecurityAgency/ghidra/tree/master/GhidraDocs

| Dimension | Variable | Description | Examples | Source | $\alpha$ |
|---|---|---|---|---|---|
| Feature | Feature Discussed | Ghidra functionality discussed in the thread | GUI, Extensions, Processor | [84] | 0.80 |
| Conversational Actions | Thread type | Type of conversation initiated | Personal issue, Announcement, Opinion poll | [51] | 0.86 |
| | Discourse Act | Type of the discourse for each comment in the thread | Question, Answer, Reframing the problem | [52] | 0.80 |
| Sensemaking | Level of Collective Sensemaking | What level of sensemaking is evident in the thread | Reaction, Lateral engagement, Idea transformation | [51] | 0.88 |

TABLE 2: Summary of qualitative coding dimensions

Finally, we marked a thread as having the highest level of collective sensemaking if we observed a change in an author's perception of the topic, indicating a transformation of knowledge derived from the discussion.

## 3.3. Quantitative analysis

With the resulting coded data, we performed regressions to understand the correlation of feature type, forum, and author expertise on several outcomes relevant to our research questions. In each regression, we used a model appropriate for the type of the outcome variable. For example, when considering the number of times each Ghidra feature is discussed, we used a poisson regression (appropriate for count data [90, 67-106]). For each regression, we include as potential explanatory variables the forum, as well as a binary variable indicating whether the commenting author was an answerer. We define an answerer here as any author in our dataset who we observed answer more than two questions (21% of all authors) constructively (i.e., only responses that provided additional information to the asker). This threshold was the natural choice as it was the elbow of the distribution curve for author participation [91]. Prior work has used a similar cutoff for distinguishing between author activity levels [33], [79].

Recent interviews and surveys of reverse engineers suggests they place importance on the ability to customize their tools [57]. To test whether this preference is actually displayed in their tool discussions, we added a binary explanatory variable to our regressions indicating whether the associated feature is a current tool or customization.

For each regression, we included all the previously stated explanatory variables in our initial regression model along with every possible two-way interaction between variables. From all possible variable and interaction combinations, we selected the model with minimum Bayesian Information Criteria—a standard metric for model fit [92].

## 3.4. Network graphs

To measure the knowledge sharing dynamics around tool adoption in the reverse engineering community, we generated a series of network graphs at varying relational levels based on observed communication patterns across forums — a common method for understanding community dynamics in social networks [78], [93]. At the lowest level of granularity, we created a social network graph with all authors as nodes, adding directed edges between authors providing insights, perspectives, and information to others participating in the thread.

Next, we considered thread-level communications for each forum. Each node in our thread-level graph represented a discussion thread. To understand how information was shared across threads, edges were drawn between nodes that shared authors. For example, if *thread1* was a discussion between authors *A* and *B* and *thread2* was a discussion between authors *B* and *C*, we would add an edge between *thread1* and *thread2* because author *B* contributed to both and created a bridge of knowledge between discussions. Additionally, we maintained an attribute for each node indicating the Ghidra feature discussed to determine whether the same authors were discussing the same feature areas.

Finally, we built a forum-level graph, merging the thread-level graphs from each forum. Each node in this graph indicated a cluster of threads. A thread cluster indicated a set of connected threads from the prior graph. Again, edges were added to indicate shared authors between clusters. The purpose of the forum-level graph is to identify the set of authors participating in multiple forums.

## 3.5. Limitations

There are several limitations inherent to our methodology. First, the picture of the reverse engineering community presented in our data represents only a snapshot in time, specific to the adoption of a new tool. We only consider discussions around a single tool over a limited time window, and we have likely missed discussions of Ghidra in other forums. We believe our approach provides a sufficiently broad view of the community's tool adoption discussions, as we analyze threads from more forums than much of the prior literature. Additionally, Ghidra's unique characteristics as a fully-featured and highly publicized tool on release likely offers the ideal setting to investigate tool adoption discussions, drawing much interest from the reverse engineering community who have little pre-existing knowledge about the tool and presenting a wide variety of topics to discuss. Conversely, Ghidra itself is structurally similar to other reverse engineering frameworks (IDA, BinaryNinja, Radare), mean the specific topics discussed likely generalize to other, similar tools. In fact, we observed that much discussion regarded how to transfer knowledge from other frameworks to Ghidra, indicating what features reverse engineers care about in other frameworks. Future work should consider later time periods to understand knowledge sharing dynamics once a more established community of interest around the tool is developed and adoption is not the primary concern. Similarly, additional work should investigate whether community dynamics change when reverse engineers use less public forums, such as the Slack workspace we were not permitted to collect data in.

Our network graphing procedure is an under-approximation of information sharing in the network. We only consider discussion participants, since we are unable to identify other reverse engineers who might view the thread, take in the information, but not comment. Additionally, our account-linking procedure likely underestimates the number of authors with accounts on multiple forums. While authors commonly shared their Twitter and SE account names on their websites, very few made their Reddit account names public. Therefore, we expect more knowledge overlap exists between threads and forums than we identified. However, because collecting more accurate information is only possible by contacting authors directly and requesting they provide all their associated aliases and threads visited, or other privacy intrusive methods, we believe our approach provides a reasonable level of accuracy for initial analysis, given the tradeoff.

## 4. Results

In this section, we present results related to our three research questions. We begin with a discussion of the Ghidra features observed in the studied forums (RQ1) and then cover the observed dynamics of knowledge sharing between reverse engineers (RQ2). Throughout, we discuss differences in findings between forums where appropriate (RQ3).

### 4.1. Features Discussed (RQ1)

We began our analysis by coding the features discussed in our sample of 338 threads across all forums. Using the Ghidra github project features list along with additional topics which emerged from the data (See Appendix A), we coded 428 unique features discussed (each thread could potentially cover multiple features). Next, we grouped these features into 10 related areas, including both Ghidra features and other topics which were not feature-specific (e.g., framework overviews, Ghidra vulnerabilities, documentation questions, etc.). Additionally, because Ghidra—like most other reverse engineering frameworks—offers a large suite of features and the ability to extend its functionality, we further divided feature-specific areas into two categories: current tools and customization. Table 3 presents these 10 groups divided into the two feature-specific categories and a third general category of discussions that were not feature specific (i.e., *Other*), along with the number of times they were discussed in our dataset and the percentage of threads in each forum in which they were discussed.

**Regression analysis.** To understand whether certain Ghidra features were discussed more often depending on their type, we performed a poisson regression analysis (appropriate for count data [90, 67-106]) to identify how the specific forum, feature type, and expertise of initial author correlate with the number of times a feature was discussed. Using the model selection process described in Section 3.3, the final selected regression is given in Table 4. Note, because we are interested in which Ghidra features are discussed, we remove from our analysis topics that are not feature-specific (i.e., features categorized as *Other*). Based on an *a priori* power analysis with our given sample number, our regression should be able to identify small effects (> 0.15) [94] at a significance level ($\alpha$) of 0.05, with sufficient power (80%) [95, pg. 296].

| | Feature Areas | Twitter (N=283) | Reddit (N=83) | SE (N=62) | Total (N=428) |
|---|---|---|---|---|---|
| Customize | Scripts | 43 (19%) | 17 (28%) | 21 (41%) | 81 (25%) |
| | Arch. | 17 (8%) | 13 (21%) | 9 (18%) | 39 (12%) |
| | Memory | 7 (3%) | 6 (10%) | 4 (8%) | 17 (5%) |
| | Var/Func Names | 4 (2%) | 1 (2%) | 3 (6%) | 8 (2%) |
| | *Total* | 71 (31%) | 37 (61%) | 37 (73%) | 145 (43%) |
| Current | Tool | 28 (12%) | 8 (13%) | 18 (35%) | 54 (16%) |
| | Setup/Run | 8 (4%) | 10 (16%) | 0 (0%) | 18 (5%) |
| | GUI | 5 (2%) | 4 (7%) | 5 (10%) | 14 (4%) |
| | *Total* | 45 (20%) | 22 (36%) | 23 (45%) | 86 (25%) |
| Other | Overview/ Vulns/ Bonding | 103 (45%) | 9 (15%) | 0 (0%) | 112 (33%) |
| | Learning | 68 (30%) | 10 (16%) | 1 (2%) | 79 (23%) |
| | Info | 4 (2%) | 5 (8%) | 1 (2%) | 10 (3%) |
| | *Total* | 171 (76%) | 24 (39%) | 2 (4%) | 197 (58%) |

TABLE 3: Number of times each feature area is discussed in each forum. Numbers in parentheses indicate the percentage of threads in which the feature area was discussed.

**Customization is most commonly discussed.** The most common group of features discussed were related to the scripting interface provided by Ghidra (N=81). For example, this includes the capabilities and semantics of the scripting language Ghidra provides (N=29), integrating custom scripts into the main analysis pipeline when a new binary is loaded (N=22), extending the main toolset features such as the decompiler or debugger (N=13), and the specific API Ghidra exposes for scripting (N=7). Many threads also considered other topics related to the customization of Ghidra. This included discussions about writing custom architecture (N=39) and memory specifications (N=17) to allow Ghidra to analyze previously unsupported binary formats (e.g., for custom firmware for IoT devices) and programmatically manipulating the data structures Ghidra uses to handle variable and function names (N=8).

Table 4 shows that customization-related features were more commonly discussed when controlling for forum and the initial author's answerer status. This is shown in the table's second row. The log estimate (E) of 1.32 indicates that features were discussed 1.32× more often than current features (the baseline case). The 95% confidence interval (CI), in column four, provides a high-likelihood range for this estimate between 1.01× and 1.73×. Finally, the p-value of 0.042 indicates this result is significant. This supports our prior qualitative findings [57], indicating that reverse engineers are very interested in the ability to customize their tools.

**Reverse engineers commonly discussed the main tool suite.** Discussions about Ghidra's current offerings were predominantly focused on the main tool suite (N=54). This was further dominated by discussions of Ghidra's

| Variable | Value | Log<br>Estimate | CI | *p*-value |
|---|---|---|---|---|
| *Type* | Current | – | – | – |
| | Customize | **1.32** | **[1.01, 1.73]** | **0.042*** |
| *Answerer* | False | – | – | – |
| | True | **0.59** | **[0.44, 0.79]** | **< 0.001*** |
| *Forum* | Reddit | – | – | – |
| | SE | 1.10 | [0.76, 1.58] | 0.598 |
| | Twitter | **1.61** | **[1.17, 2.2]** | **0.003*** |

*Significant effect – Base case (Log Estimate defined as 1)

TABLE 4: Summary of regression over feature discussion counts. Pseudo $R^2$ measures for this model were 0.07 (McFadden) and 0.25 (Nagelkerke). *Type* was the variable of primary interest. The variables included at the bottom of the table were added primarily as covariates to account for variance. Because we consider a random sample of tweets, the *Forum* comparison is an under-approximation.

decompiler (N=22), likely because this is one of the key benefits of Ghidra. It provides decompilation for free at a level comparable only to expensive commercial offerings [19]. However, key differentiating features, such as collaboration support (N=4) and binary version tracking (N=3) were some of the least discussed features. This seems to indicate that the common discussion of Ghidra's decompiler is not spurred only by its novelty, but also the general importance of decompiler support to reverse engineers. In addition to Ghidra's more unique features, reverse engineers discussed core tools such as Ghidra's exporter (N=9), focusing on how Ghidra could integrate with their other tools; its disassembler (N=7); and its debugger (N=3).

**Answerers were less likely to start threads across all feature areas and forums.** Controlling for all variables, non-answerers started 1.69× more feature-specific discussions than answerers. Interestingly, we did not observe any statistically significant relationship between the author's answerer status and feature type. We might have expected to see more active authors discussing more advanced features, while non-answerers focus on the more basic offerings of Ghidra. This was not the case; however, this may be an artifact of time period studied with a defined split occurring as Ghidra moves out of the early adoption phase.

**Features are discussed the most on Twitter, but feature discussions are a minority of Twitter discussions (RQ3).** Finally, we consider variations in feature discussions between forums. Since we are comparing a random sample of tweets to the full set of SE and Reddit comments, it is very likely the observed difference in counts is even more significant in the full sample. While it appears obvious that Twitter dominates the other forums in this metric since we looked at more Twitter threads, authors had the option to post their messages to any forum, so this remains a relevant comparison. For this reason, we directly compare counts of feature types instead of percentages of threads on a given forum.

Looking at the distribution of discussion among forums, we found that features were 1.61× more likely to be discussed on Twitter than on Reddit. Also, while we did not observe a significant difference between feature counts in SE and Twitter—indicated by the overlapping CIs—, we did observe nearly twice the number of feature discussions on Twitter (N=116) as SE (N=60).

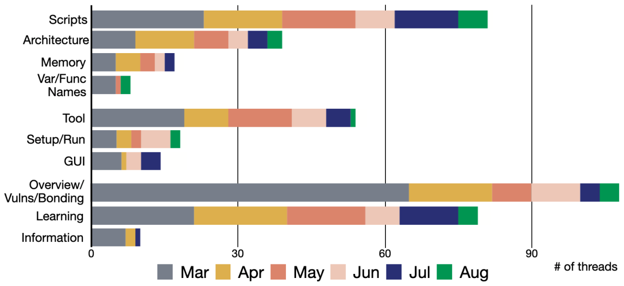

Figure 1: Timeline of feature discussions divided by group.

While Twitter includes a large amount of feature-specific information, only about half of all Ghidra discussions on Twitter are feature-specific (51%). This means reverse engineers reviewing Twitter for specific feature information or reverse engineering tool developers considering which features to prioritize must scan through large numbers of threads (N=171). This likely makes searching for relevant information very difficult for reverse engineers and could potentially outweigh the benefits of a large quantity of threads.

Using a Chi-squared test—appropriate for categorical data [96]—we performed pairwise comparisons of the number of *Other* features discussed between each forum. Because we perform multiple tests, we use a Holm-Bonferroni correction to account for the possibility of introducing false positives [97]. For each comparison, the effect size is calculated by measuring the association of the two variables tested ($\phi$) [98, 282-283]. Based on Cohen's recommendation, we consider a $\phi \geq 0.1$ a small effect, $\geq 0.3$ a medium effect, and $\geq 0.5$ a large effect [94]. After correction, a $p$-value less than 0.05 is considered significant. We found that discussions on Twitter were much less likely to be feature specific than both Reddit ($\phi = 0.81$, $p < 0.001$) and SE ($\phi = 0.99$, $p < 0.001$). Reddit was also less likely to be feature specific than SE ($\phi = 0.33$, $p < 0.001$), though the effect size was less dramatic.

**No clear difference in discussion timeline between features.** After investigating the differences related to feature discussions distribution across the entire time period studied, we sought to understand whether the conversation changed over time. That is, do reverse engineers begin by discussing a particular feature set, then move to other, potentially more complex features later on as they adopt the tool into their practice? Any trend could tell tool developers which features to focus on for initial release and drive the order of work. However, we did not observe any clear differences between the timeline of discussion of the Ghidra feature groups. We did observe that feature-specific discussions began more often at the start of the release period, with about a third of discussions occurring in the first month (38% of current tool discussions and 30% of customization discussions) and half of all discussions happening in the first two months (38% of current tool discussions and 54% of customization discussions). This trend is even more dramatic for topics which are not feature-specific, as 49% of these discussions occurred in the first

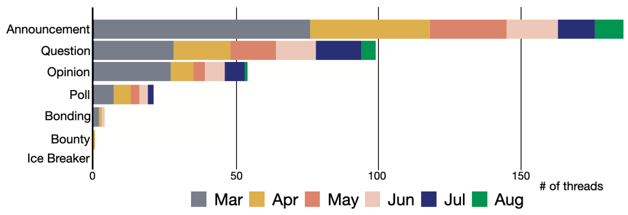

Figure 2: Timeline of threads divided by type.

month and 69% occurred by the second month. This can be seen in Figure 1, which shows the number of discussion threads started in each month. This spike in discussions at the beginning of the collection period is likely due to Ghidra's newness and the publicity it received [41]–[50] and may not generalize to other, more traditional releases.

## 4.2. Thread Types (RQ2)

Knowing the feature areas reverse engineers discuss, we then looked at how reverse engineers were discussing these features across forums. Figure 2 presents the number of discussion threads for each type started in each month of our review.

**Announcements were most common, especially on Twitter (RQ3).** Over half of all threads (55%) were declarative in nature, simply making an announcement sharing a resource or information without looking for any response. For example, many authors shared links to Ghidra tutorials, news articles about Ghidra's release, or open source Ghidra scripts. Interestingly, these announcements were predominantly found on Twitter, making up more than half of all coded Twitter threads (64%). We did find that a large percentage of Reddit threads were announcements (42%), but announcements were tied for most common thread type with question-initiated threads. Conversely, we did not observe any announcements on SE. Finally, announcements were concentrated in the first months after Ghidra's release, with 42% occurring within one month and 63% within two months. This is likely unique to the tool adoption process, showing an effort by reverse engineers to spread the word about the newly available framework.

**Questions were also common, especially on SE (RQ3).** The next most common thread type was questions (29%). A thread was coded as a question if the initial author asked for help with a specific problem. In most cases, authors asked for help troubleshooting an issue with Ghidra or sought suggestions for features that would help achieve a particular goal. In contrast with announcements, questions were most common on SE (94% of threads). Questions were tied for the most common thread type on Reddit (42% of threads), but only made up 8% of all threads on Twitter (third most common thread type). Questions were also more evenly distributed over time, with only 28% occurring within one month, 48% within two months, and 64% within three months, the midpoint in collection.

**Reverse engineers commonly share and seek opinions.** In other common thread types, the initial author either shared their general perspective on a given topic (16% of threads) or polled the community for the collective's opinion (6% of threads). In both thread types, authors discuss their general opinions of Ghidra with statements ranging from "Ghidra is magic!" to "I haven't tried Ghidra nor do I intend to. Fight me." Authors also commented on specific features, for example noting that "Ghidra is pretty much giving me the original source here," referring to Ghidra's decompiler. Another common discussion topic among the community was the value of Ghidra in comparison to the most commonly used, but expensive, reverse engineering framework, IDA Pro [19].

**Very few cases of bonding.** Finally, we observed very few examples of reverse engineers asking others to share experiences of general challenges faced (1% of threads). We also did not observe any examples of ice breaker threads, which have been used in other forums to help members of the community get to know one another [51]. This indicates that there were relatively few direct efforts build a community of users around this new tool. However, it is possible this occurs more often in closed channels, other online organizations, or in direct communication outside the scope of our analysis.

## 4.3. Question Answering (RQ1)

Because questions constituted some of the most prevalent and consistent forms of discussions, we next sought to understand the community's support for answering reverse engineers' questions about Ghidra during tool adoption. Specifically, we consider whether any of the forums are better at providing support and whether the community establishes sufficient knowledge base for particular feature areas to answer questions in this early stage. This can tell reverse engineers which forum to go to in search of answers and would help tool developers predict predict where question answering support would be most beneficial to fill in gaps or lags in community knowledge.

To answer this question, we limited our analysis to threads where the initial author posed a specific question (N=181). We marked each question as answered if we observed a comment on the thread which presented an answer to the question. Note, this is only an approximation, as we do not attempt to assess whether the answer is necessarily correct or sufficiently informative. Prior work has considered whether a question has an "accepted answer"—a common feature in Q&A forums [69]. However, because only one forum we considered included this feature (i.e., SE), we chose not use this method to allow for a fair comparison. Further, while we found only 58% of questions we marked as answered had an "accepted" answer on SE, we found that in all but one case, the "accepted" answer metric was overly conservative. Many questions received answers with multiple upvotes by other authors and responses confirming the correctness of the answer, indicating the question asker may have failed to accept a reasonable answer (N=11). Other answers were given as comments with the asker indicating the answer was correct, but unable to accept the answer because it was not submitted in the right place in the UI (N=4). Finally, other answers appeared not to be accepted because they were related to an open bug in the Ghidra tool itself (N=3) or solving the issue was computationally infeasible (N=2).

**Regression analysis.** To explore trends in question answering behaviors, we performed another regression analysis

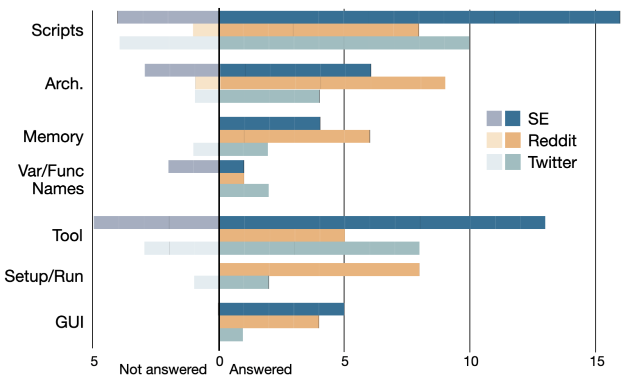

Figure 3: Number of questions answered and unanswered in each feature area in each forum.

| Variable | Value | Odds Ratio | CI | $p$-value |
|---|---|---|---|---|
| *Answerer* | False | – | – | – |
| | True | **5.42** | **[1.54, 19.09]** | **0.009*** |

*Significant effect – Base case (Log Estimate defined as 1)

TABLE 5: Summary of regression over question answering likelihood. Pseudo $R^2$ measures for this model were 0.07 (McFadden) and 0.13 (Nagelkerke).

using the same initial explanatory variables and model selection process described in Section 3.3. However, in this case, because our outcome variable—whether the question is answered—is binary, we performed a logistic regression (appropriate for binary data [95]). The selected regression model is given in Table 5.

**No difference in answering rates between features or forums.** Figure 3 shows the number of questions answered and unanswered for each feature group divided by forum. Overall, we found the majority of Ghidra questions (81%) received an answer. This was true whether reverse engineers asked about current features (84%) or customizations (79%). Additionally, while a higher percentage of questions on Reddit (95%) were answered than both Twitter and SE (74% and 75%, respectively), we did not find this difference to be statistically significant when controlling for feature area and author answerer status (both forum and feature area were not included in our final model).

**Answerers are more likely to get a response.** Question answering activity of the question author was the only variable we observed which had a significant effect on whether the question was answered. Table 5 shows a 5.42× increase in odds of a question being answered when asked by an answerer versus a non-answerer, assuming all other variables are the same. This is somewhat unintuitive as we might expect answerers to ask more complex questions, requiring a higher level of expertise in Ghidra than exists among other members of the community. Instead, it appears answering support is consolidated within a subset of the online reverse engineering community who answer each others' questions, but not those of reverse engineers outside this subset. This interaction between small clusters of reverse engineers is shown in more detail in Section 4.5.

| Variable | Value | Odds Ratio | CI | $p$-value |
|---|---|---|---|---|
| *Answerer* | False | – | – | – |
| | True | **5.32** | **[3.16, 8.99]** | **< 0.001*** |

*Significant effect – Base case (Odds Ratio defined as 1)

TABLE 6: Summary of regression over level of collective sensemaking. Pseudo $R^2$ measures for this model were 0.06 (McFadden) and 0.13 (Nagelkerke).

## 4.4. Sensemaking (RQ2)

After observing that most questions are answered, we considered how relevant knowledge is developed within the community during tool adoption. To determine whether knowledge about Ghidra is developed directly through forum discussions, we consider whether each discussion thread exhibited signs of collective sensemaking—how communities create shared knowledge structures and build meaning together [89]. We might expect this to be particularly prevalent after a tool's initial release as no users have prior experience and build their understanding through iteration of ideas with others.

**Regression analysis.** To explore collective sensemaking trends, we performed another regression analysis continuing to use the same initial explanatory variables and model selection process described in Section 3.3. Our outcome variable for this regression was the level of collective sensemaking observed in each thread (i.e., reaction, lateral engagement, or knowledge transformation), an ordered categorical variable. Therefore, we used an ordinal logistic regression model (appropriate for ordinal data) [99]. The final model is given in Table 6.

**Idea transformation is rare.** Less than half of all threads (42%) exhibited any characteristics of collective sensemaking. In most of these cases, we only observed authors reflecting on points made by others earlier in the thread (19% of threads) or some lateral engagement, where authors went back-and-forth discussing a topic (16% of threads). However, we observed very few cases (7% of threads) of idea transformation, which is demonstrated by authors actually changing their perspective or expressing any new insight derived from the conversation. For example, in one Twitter thread, two authors discuss adding disassembler support for correctly presenting retpolines (a recently developed security mitigation used to prevent branch-target-injection attacks [100]). During their discussion, they both present possible solutions, iteratively building on each others' ideas, and eventually concluding on a new approach based on the combination of both suggestions. This finding draws a clear contrasts between the reverse engineering and software engineering communities as software engineers' discussions typically tend to be more collaborative [30], [53], [54].

**Collective sensemaking is more likely in answerer-initiated threads.** Our regression model-selection process did not include either the forum or feature area in the final model, indicating that no significant difference was observed with respect to those variables. The only variable included in our final regression model was the initiating author's answerer status. Our results indicate answerer initiated threads are 5.32× as likely (compared to non-answerer initiated threads) to increase one level of col-

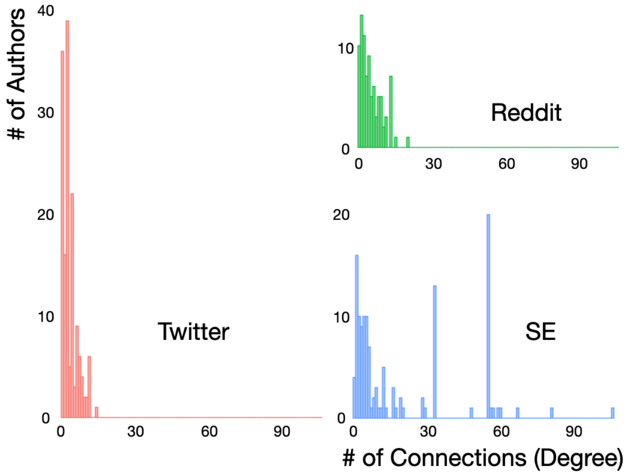

Figure 4: Number of connections for each author divided by forum.

lective sensemaking. This result is intuitive as we might expect reverse engineers who are able to answer several questions to have more experience and therefore discuss more complex topics that require input from others to make sense. It is also in line with our result from Section 4.3 that answerer questions are more likely to receive answers and thus more likely to include transformative discussions.

### 4.5. Knowledge Sharing Dynamics (RQ2)

As we observed limited collective sensemaking occurring directly in discussions, it may instead be the case that knowledge is developed over time through direct information sharing between authors (i.e., author A shares information with author B, who in turn applies their gained knowledge in another discussion with author C). To investigate this theory, we reviewed the network connections between authors, considering relationships at three granularity levels: author, thread, forum. See Section 3.4 for a discussion of how graphs at each granularity level were generated.

**Low density of connections, but high reciprocity.** Looking first at the lowest granularity graph, the author-level, we found low connection density—the number of edges that exist in the graph divided by the possible number of edges that would exist if the graph was fully connected—among authors in each forum (2% for Twitter, 8% for SE, and 4% for Reddit). Note, while there are differences in density between forums, these are expected variations as connection density generally decreases with group size. Additionally, we observed that the degree distribution, the number of other authors each author is connected to in the graph, was right-skewed, indicating most authors had few connection, with a few authors dominating the conversation. Figure 4 presents the degree distribution for each forum, showing a clear right-skew for Twitter and Reddit, while SE includes more high-connectivity nodes. This distribution is common in social networks, as most network members cluster into small communities, with a few highly connected members spanning multiple clusters [101]. While we did not see significant connectivity across all the nodes, we did find high levels of reciprocity between nodes (78% in Twitter, 78% in SE, and 72% in Reddit). Reciprocity is a measure specific to directed graphs that gives the ratio of nodes linked in both directions to the total number of nodes linked in at least one direction [102]. Reciprocity only considers if a bidirectional connection exists between nodes unidirectionally connected, meaning disconnected nodes are not included in the calculation. This indicates that for the authors who are connected, information sharing goes both ways, which is in line with our prior results in Section 4.3. Therefore, while we did not observe much evidence of authors developing knowledge specifically within the forums (i.e., sensemaking), we did observe evidence of information sharing at least within small clusters of authors. Also, the differences we observed in reciprocity between Reddit and the other forums aligns with our expectations. Because Reddit is a more pseudonymous platform, these types of relations are expected to be less common.

**Connections dominated by a central cluster in each forum.** Knowing the network density is low, but reciprocity is high for connected authors, we moved to the thread-level graph to determine how often these connections—hence, information sharing—goes beyond a specific thread. On average, we observed that most authors only participate in one (33%) or two (29%) threads, with two threads being the median. The first three graphs of Figure 5 presents the thread-level graphs for each forum. Each graph shows a central cluster of threads sharing contributing authors, surrounded by threads whose authors are disconnected from those participating in other threads. On Twitter, the majority (51%) of all nodes are isolates (i.e., degree zero nodes), and 77% of all edges are contained in the central cluster. While there are less isolates for both Reddit (31%) and SE (25%), the dominance of a single central cluster is more pronounced, with 87% of all edges contained in a single Reddit cluster and all edges in SE part of the same cluster.

**Reverse engineers comment on multiple feature areas.** Next, we considered whether the clusters we identified covered specific feature areas, indicating possible author specialization. We began by labelling each node in the graphs on the top row of Figure 5 according to the type of feature discussed. We found that a majority of threads in 73% of the non-isolate clusters covered the same type of features. However, we found that most authors who participated in multiple threads (92%) discussed more than one area out of the ten areas given in Table 3, with a mean of 2.48 feature areas discussed by authors. This indicates that during tool adoption, active authors do not focus on a single feature set. This may be specific to the tool adoption discussions with authors specializing in specific areas the community develops.

**Very few reverse engineers participate in multiple forums.** Moving to the highest level graph granularity, we next analyzed the forum-level graph (shown on the far-right of Figure 5). This graphs shows that the central clusters identified in the thread-level graphs, as well as a few of the smaller clusters were sparsely connected. In total we observed seven connections across forums—by six unique authors. On average, authors contributed to 1.03 forums, meaning that authors almost exclusively

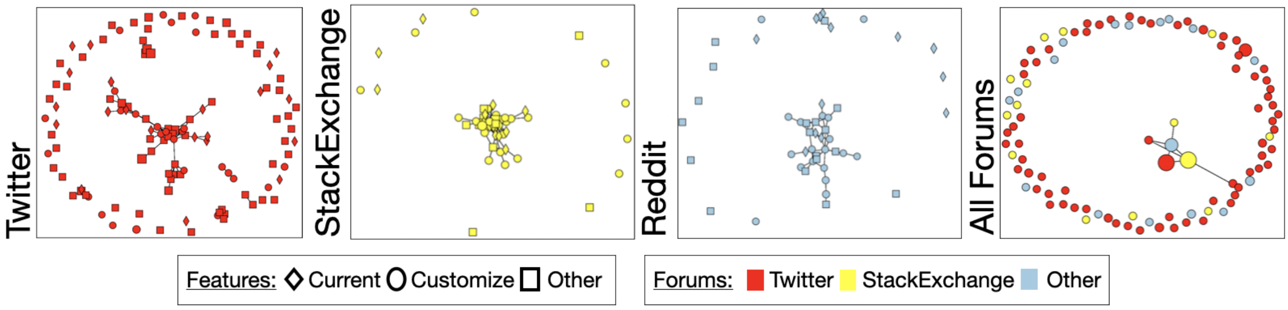
Twitter
StackExchange
Reddit
All Forums
Features: ◇ Current ○ Customize □ Other
Forums: 🟥 Twitter 🟨 StackExchange 🟦 Other

Figure 5: The first three graphs indicate the thread-level relationships across each forum and the furthest-right graph shows forum-level connections. In the thread-level graphs, features areas covered by each thread are indicated by the node shape (as given in the legend). However, no such distinction is made in the forum-level graph and all nodes are shown as circles irrespective of features discussed in the associated clusters.

participate in conversations on a single forum. Note, this is likely somewhat of an underestimate of between-forum connections due to the limitations of our account-linking method (see Section 3.5).

**Most authors ask or answer.** Finally, we sought to understand how information was produced and consumed by authors across forums. Figure 6 plots the number of questions answered (knowledge produced) and asked (knowledge consumed) by each author in a heatmap. The large majority of authors (85%) only participated in one way, with most only answering questions (58%) and a smaller group only asking (27%). This indicates the community is dominated by a group willing to help answer the questions of others, but not asking further questions of others in the community. Additionally, as very few author both produced and consumed information (15% of authors), there is little evidence of knowledge acquisition, development, and sharing directly in the discussions. This contrasts our initial expectation that we would see more questions and knowledge development among authors during Ghidra's adoption phase.

## 4.6. Negativity in the Community (RQ2)

Finally, given that the conversation is dominated by a few individuals and most authors have limited participation in the conversation, we investigated whether the community discourse might be seen as unwelcoming. An unwelcoming atmosphere during the tool adoption period could have a dramatic effect on the community's eventual use of a tool. If all members of the community do not feel comfortable participating, the discussion and consensus around a tool will be shaped by a subset of the population and not benefit from the possible diversity of perspectives available.

Using our coding of all discourse actions derived from Zhang et al.'s codebook [52], we looked at the likelihood of an author receiving a negative response to a comment. We considered two types of negative responses. The first, was sarcasm that poked fun at the comment, but did not clearly attack the commenter. The stronger form of negative comment we observed attacked or mocked the commenter, or expressed disgust, derision, or anger, toward the prior comment. We did not consider a comment negative if it engaged the merits of the prior commenter's points or was trying to offer constructive feedback.

| Variable | Value | Odds Ratio | CI | *p*-value |
|---|---|---|---|---|
| *Forum* | Reddit | – | – | – |
| | SE | 1.28 | [0.40, 4.06] | 0.676 |
| | Twitter | **0.22** | **[0.9, 0.53]** | **< 0.001*** |
| *Answerer* | False | – | – | – |
| | True | **5.61** | **[3.10, 10.14]** | **< 0.001*** |

*Significant effect – Base case (Odds Ratio defined as 1)

TABLE 7: Summary of regression over comment sentiment. Pseudo $R^2$ measures for this model were 0.19 (McFadden) and 0.23 (Nagelkerke).

**Regression analysis.** We utilized a regression analysis to investigate trends in comment sentiment. Because our outcome variable, comment sentiment, is binary (i.e., negative or not), we used a logistic regression (appropriate for binary data [95]). Again, we used the same initial set of explanatory variables and model selection procedure as described in Section 3.3. The final model is given in Table 7. Unlike with prior questions, we included comments from all threads (N=1098) in our analysis instead of limiting our scope to feature-specific discussions. We chose not to limit our analysis in this case because reverse engineer participation in the community is likely affected by the broader climate of discussion.

**Very little negativity overall.** Only a small fraction (7%) of all coded comments were considered negative. Of those, the vast majority were considered sarcastic (6% of comments) with very few demonstrating direct attacks against other authors (1% of comments). While we did find lower percentages of negative comments on feature-specific threads (3% and 4% of comments discussing current tools and customizations, respectively) than other threads (12% of comments), this variable was not included in our final model.

**Negative responses are most common on Twitter.** We did observe a statistically significant effect when comparing Twitter and Reddit, with comments having a 4.55× increase in likelihood of being negative. This was unexpected as Reddit is the more anonymous forum, which commonly leads to more negative behaviors [103], [104]. This may be the result of Reddit's content moderation policies, such as allowing users to flag potentially hateful or harassing messages [105]. Negative comments were also more prevalent on Twitter (14% of comments) than SE (2%

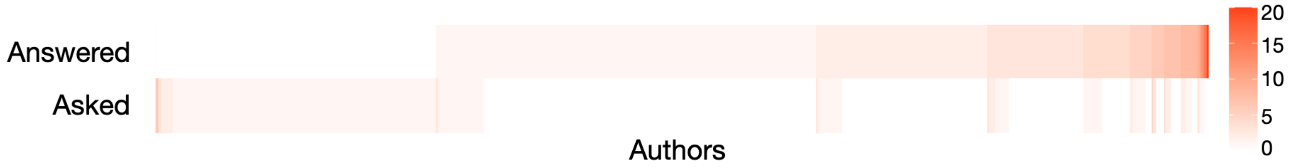

Figure 6: A heatmap indicating the number of questions asked and answered by each reverse engineer. Authors are sorted by the number of questions answered first, then number asked. The darker colors indicated more questions of the given type.

of comments).

**Non-answerers are most likely to make negative comments.** The difference between comments by answerers and non-answerers was the largest gap we found across all variables tested. 16% of comments made by non-answerers were considered negative, compared to only 2% of answerers. Controlling for the forum, comments made by an non-answerer were correlated with a 5.56× increase in likelihood of being negative. This is expected as negative comments, i.e., trolling behaviors, are not expected to come from authors who are also constructive participants in the conversation. It is conceivable that reverse engineers could act differently in different contexts, exhibiting both positive and negative behaviors [106]. However, in this early adoption phase, it appears these activities are mostly bifurcated. Additionally, we did not observe any reverse engineer frequently posting negative comments, as individual reverse engineer posted at most two negative comments and the vast majority (92%) only posted one negative comment.

## 5. Discussion

Our key findings can be summarized as follows:

- Reverse engineers discussed customization features most often. When discussing Ghidra's existing feature set, they most commonly focused on Ghidra's decompiler.
- Most reverse engineers' questions were answered, but answers were less likely when a non-answerer asked the question.
- Collective sensemaking was not common in the forums during tool adoption. Instead, knowledge flowed from a large group of producers to a smaller group of consumers with a single, central group of discussions. This distinguishes reverse engineers from software engineers whose discussions are generally more collaborative [30], [53], [54].
- Twitter includes the most feature-specific threads of any forum. However, these threads make up a lower percentage of the forum's threads than Reddit or SE, they focus more on announcements, are more likely to include negative responses, and often are isolated (i.e., do not share authors with other threads).

With these findings in mind, we suggest recommendations for reverse engineering tool developers to consider when trying to get a tool adopted, for newcomers and experts in the reverse engineering community when approaching a new tool, and for researchers considering directions for future work.

### 5.1. Tool Developers

Likely the most relevant result of our work for reverse engineering tool developer is the finding that most users, at least in the initial adoption period, considered features that allow the user to customize the tool to their specific needs. First, this indicates reverse engineers' interest in and therefore the importance of these features, suggesting tool developers should place particular focus on supporting tool customization. This finding is in line with our prior work investigating reverse engineering processes [57], but shows this is an significant consideration of users during initial adoption, indicating that it should be included from the start. Additionally, the large number of questions about these features may suggest usability difficulties. This may be specific to Ghidra, but it is to be expected for more complex features. Therefore, tool developers should be careful to consider the user when designing customization interfaces by testing their usability and providing additional documentation.

When considering how best to support and interact with the community at this early stage, our results suggest tool developers need to be involved across multiple forums. Because we observed very little overlap in authors across forums, focusing on any particular forum is insufficient for engaging the community. Additionally, authors used each forum for different purposes, further suggesting the need to consider multiple forums.

Within forums, our results suggest tool developers should make efforts to answer a variety of reverse engineers' questions. During the tool adoption phase, engaging many reverse engineers can provide support to individuals we found were more isolated and, in the case of non-answerers, less likely to have their questions answered. Focusing on the subset of most prolific users may waste developer effort, as these users are more likely to be supported by other members of the community.

Finally, tool developers can also benefit from adopting similar analysis methods to the ones we demonstrate in this paper. By tracking feature discussions across forums, developers can produce similar results regarding feature popularity and community questions-answering support to improve early adoption. This information can help developers know which features to prioritize as well as which features require improved documentation. Additionally, tracking this information over time would allow developers to observe the impact of tool changes.

### 5.2. Reverse Engineers

Similar to our recommendations for tool developers, reverse engineers should also consult all forums for tool information and question answering when considering new tool or feature adoption. Additionally, our results suggest reverse engineers will find Twitter most useful when searching for resources to help with tool learning generally (e.g., tutorials, feature overviews), while SE is suggested for information about specific problems, and Reddit provides a mixture of both. While Twitter has more feature-specific threads, it contains even more tangentially related material—as has been shown in other settings [66]—, which may be difficult to sift through. If a reverse engineer is looking to make sense of a new feature or tool through discussion with another reverse engineer, they are most likely to find this in SE (at least during the early stages of tool adoption).

We also call on forum moderators to make changes to the incentive structure of these forums, where possible, to improve knowledge sharing. First, we recommend incentivizing behaviors that bridge isolated pockets of users in the early tool adoption phase. For example, moderators should consider rewarding reverse engineers who start discussions specifically for bonding purposes. This can encourage relationships between authors who have not connected previously and spur future collective sensemaking, accessing a more diverse set of perspectives.

Additionally, moderators should be careful to deter negative comments. Negative comments were rare (although still potentially harmful for beginners [107] especially in early adoption), suggesting a light touch intervention by moderators may be sufficient. For example, moderators could engage in *counterspeeh* as suggested by Mathew et al. [108], by responding to hateful speech directly condemning it and warning that the user may be banned if similar hateful behavior continues. This type of community feedback has been shown to improve user behavior [109]–[111]. Moderators should also closely monitor for negative comments to prevent their number from growing.

## 5.3. Future Work

Because we observed very little sensemaking in any forum, further investigation is required to understand how knowledge around new tools is developed in the reverse engineering community. It may be that knowledge is currently developed individually or among small groups (e.g., coworkers). Alternatively, there may be additional private forums, missed in this analysis, which require further investigation. To provide a thorough understanding of reverse engineering knowledge development and needs, interviews or surveys directly with reverse engineers are necessary.

Our results also suggest the value of efforts aimed at bridging information across forums. Because of the isolated nature of each forum, exposing reverse engineers to relevant information elsewhere could be particularly useful.

Finally, future work should consider methods for simplifying feature extraction to allow broader longitudinal analysis. For example, NLP methods could be leveraged, using our codes and feature tags from GitHub as training data, to automate the discussion feature tagging process.

## Acknowledgment

We thank Kelsey Fulton and the anonymous reviewers for their helpful feedback; and several reverse engineering experts, who wish to remain anonymous, for helping us better understand Ghidra. This research was supported in part by a UMIACS contract under the partnership between the University of Maryland and DoD, and by a Google Research Award.

## References

[1] D. Votipka, R. Stevens, E. M. Redmiles, J. Hu, and M. L. Mazurek, “Hackers vs. testers: A comparison of software vulnerability discovery processes,” *Proc. of the IEEE*, 2018.

[2] M. Ceccato, P. Tonella, C. Basile, B. Coppens, B. De Sutter, P. Falcarin, and M. Torchiano, “How professional hackers understand protected code while performing attack tasks,” in *Proc. of the 25th International Conference on Program Comprehension*, ser. ICPC ’17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 154–164. [Online]. Available: https://doi.org/10.1109/ICPC.2017.2

[3] E. Eilam, *Reversing: secrets of reverse engineering*. John Wiley & Sons, 2011.

[4] D. Fraze, “Computer and Humans Exploring Software Security (CHESS),” DARPA, 2017, (Accessed 05-31-2019). [Online]. Available: https://www.darpa.mil/program/computers-and-humans-exploring-software-security

[5] K. Yakdan, S. Dechand, E. Gerhards-Padilla, and M. Smith, “Helping johnny to analyze malware: A usability-optimized decompiler and malware analysis user study,” in *IEEE S&P ’16*, May 2016, pp. 158–177.

[6] Y. Shoshitaishvili, M. Weissbacher, L. Dresel, C. Salls, R. Wang, C. Kruegel, and G. Vigna, “Rise of the hacrs: Augmenting autonomous cyber reasoning systems with human assistance,” in *Proc. of the 24th ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’17. ACM, 2017.

[7] N. Rutar, C. B. Almazan, and J. S. Foster, “A comparison of bug finding tools for java,” in *Proceedings of the 15th International Symposium on Software Reliability Engineering*, ser. ISSRE ’04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 245–256. [Online]. Available: http://dx.doi.org/10.1109/ISSRE.2004.1

[8] D. Baca, B. Carlsson, K. Petersen, and L. Lundberg, “Improving software security with static automated code analysis in an industry setting.” *Software: Practice and Experience*, vol. 43, no. 3, pp. 259–279, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/spe/spe43.html#BacaCPL13

[9] A. Doupé, M. Cova, and G. Vigna, “Why johnny can’t pentest: An analysis of black-box web vulnerability scanners,” in *Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, ser. DIMVA’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 111–131. [Online]. Available: http://dl.acm.org/citation.cfm?id=1884848.1884858

[10] A. Austin and L. Williams, “One technique is not enough: A comparison of vulnerability discovery techniques,” in *Proceedings of the 2011 International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM ’11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 97–106. [Online]. Available: http://dx.doi.org/10.1109/ESEM.2011.18

[11] N. Antunes and M. Vieira, “Comparing the effectiveness of penetration testing and static code analysis on the detection of sql injection vulnerabilities in web services,” in *Proceedings of the 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing*, ser. PRDC ’09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 301–306. [Online]. Available: http://dx.doi.org/10.1109/PRDC.2009.54

[12] L. Suto, “Analyzing the effectiveness and coverage of web application security scanners,” https://www.beyondtrust.com/resources/white-paper/analyzing-the-effectiveness-and-coverage-of-web-application-security-scanners/, BeyondTrust, Inc, Tech. Rep., 2007.

[13] ——, "Analyzing the accuracy and time costs of web application security scanners," https://www.beyondtrust.com/wp-content/uploads/Analyzing-the-Accuracy-and-Time-Costs-of-Web-Application-Security-Scanners.pdf, BeyondTrust, Inc, Tech. Rep., 2010.

[14] G. McGraw and J. Steven, "Software [in]security: Comparing apples, oranges, and aardvarks (or, all static analysis tools are not created equal," http://www.informit.com/articles/article.aspx?p=1680863, Cigital, 2011, (Accessed 02-26-2017).

[15] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones," in *OSDI '10*. Berkeley, CA, USA: USENIX Association, 2010, pp. 393–407. [Online]. Available: http://dl.acm.org/citation.cfm?id=1924943.1924971

[16] C. Cadar, D. Dunbar, D. R. Engler *et al.*, "Klee: Unassisted and automatic generation of high-coverage tests for complex systems programs." in *OSDI '08*, vol. 8, 2008, pp. 209–224.

[17] S. K. Cha, T. Avgerinos, A. Rebert, and D. Brumley, "Unleashing mayhem on binary code," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, ser. SP '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 380–394. [Online]. Available: http://dx.doi.org/10.1109/SP.2012.31

[18] N. Stephens, J. Grosen, C. Salls, A. Dutcher, R. Wang, J. Corbetta, Y. Shoshitaishvili, C. Kruegel, and G. Vigna, "Driller: Augmenting fuzzing through selective symbolic execution," in *NDSS '16*, no. 2016. Internet Society, 2016, pp. 1–16.

[19] Hex-Rays, "Ida: About," 2019, (Accessed 05-30-2019). [Online]. Available: https://www.hex-rays.com/products/ida/

[20] Vector35, "Binary.ninja: A reverse engineering platform," 2019, (Accessed 05-30-2019). [Online]. Available: https://binary.ninja/

[21] Synopsys, "Coverity scan - static analysis," 2019, (Accessed 05-30-2019). [Online]. Available: https://scan.coverity.com/

[22] ForAllSecure, "Forallsecure," 2019, (Accessed 05-30-2019). [Online]. Available: https://forallsecure.com/

[23] Hex-Rays, "Plug-in contest 2018: Hall of fame," 2019, (Accessed 05-30-2019). [Online]. Available: https://www.hex-rays.com/contests/2018/index.shtml

[24] Vector35, "Vector35/community-plugins," 2019, (Accessed 05-30-2019). [Online]. Available: https://github.com/Vector35/community-plugins/tree/master/plugins

[25] W. Maalej, R. Tiarks, T. Roehm, and R. Koschke, "On the comprehension of program comprehension," *ACM Transactions on Software Engineering Methodology*, vol. 23, no. 4, Sep. 2014. [Online]. Available: https://doi.org/10.1145/2622669

[26] X. Xia, L. Bao, D. Lo, P. S. Kochhar, A. E. Hassan, and Z. Xing, "What do developers search for on the web?" *Empirical Software Engineering*, vol. 22, no. 6, pp. 3149—-3185, 2017. [Online]. Available: https://doi.org/10.1007/s10664-017-9514-4

[27] Y. Wu, S. Wang, C.-P. Bezemer, and K. Inoue, "How do developers utilize source code from stack overflow?" *Empirical Software Engineering*, vol. 24, no. 2, p. 637–673, 2019. [Online]. Available: https://doi.org/10.1007/s10664-018-9634-5

[28] Y. Tian, P. Achananuparp, I. N. Lubis, D. Lo, and E. Lim, "What does software engineering community microblog about?" in *2012 9th IEEE Working Conference on Mining Software Repositories (MSR)*, 2012, pp. 247–250.

[29] Y. Tian and D. Lo, "An exploratory study on software microblogger behaviors," in *2014 IEEE 4th Workshop on Mining Unstructured Data*, 2014, pp. 1–5.

[30] G. Bougie, J. Starke, M.-A. Storey, and D. M. German, "Towards understanding twitter use in software engineering: Preliminary findings, ongoing challenges and future questions," in *Proceedings of the 2nd International Workshop on Web 2.0 for Software Engineering*, ser. Web2SE '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 31–36. [Online]. Available: https://doi.org/10.1145/1984701.1984707

[31] L. MacLeod, M. Storey, and A. Bergen, "Code, camera, action: How software developers document and share program knowledge using youtube," in *2015 IEEE 23rd International Conference on Program Comprehension*, 2015, pp. 104–114.

[32] C. Parnin and C. Treude, "Measuring api documentation on the web," in *Proc. of the 2nd International Workshop on Web 2.0 for Software Engineering*, ser. Web2SE '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 25–30. [Online]. Available: https://doi.org/10.1145/1984701.1984706

[33] C. Parnin, C. Treude, L. Grammel, and M.-A. Storey, "Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow," Rand National Defense Research Institute, Tech. Rep., 2009.

[34] D. Posnett, E. Warburg, P. Devanbu, and V. Filkov, "Mining stack exchange: Expertise is evident from initial contributions," in *2012 International Conference on Social Informatics*, 2012, pp. 199–204.

[35] N. S. Agency, "Ghidra," National Security Agency, 2019, (Accessed 10-23-2020). [Online]. Available: https://www.nsa.gov/resources/everyone/ghidra/

[36] W. McGrew and T. Holland, "Introduction to reverse engineering with ghidra," HORNE Cyber, 2019, (Accessed 10-23-2020). [Online]. Available: https://www.defcon.org/html/defcon-27/dc-27-workshops.html#mcgrew

[37] B. Knighton and C. Delikat, "Ghidra - journey from classified nsa tool to open source," BlackHat, 2019, (Accessed 10-23-2020). [Online]. Available: https://www.blackhat.com/us-19/briefings/schedule/index.html#ghidra---journey-from-classified-nsa-tool-to-open-source-16309

[38] C. Doege, "Intro to reverse engineering with ghidra: Taming the dragon," Bsides SATX, 2019, (Accessed 10-23-2020). [Online]. Available: https://www.bsidessatx.com/presentations-2019.html

[39] C. A, "Ghidra vs the cold war," Bsides Bristol, 2019, (Accessed 10-23-2020). [Online]. Available: https://bsidesbristol.sched.com/event/Qcy7/ghidra-vs-the-cold-war

[40] R. Joyce, "Come get your free nsa reverse engineering tool!" RSA, 2019, (Accessed 10-23-2020). [Online]. Available: https://www.rsaconference.com/industry-topics/presentation/come-get-your-free-nsa-reverse-engineering-tool

[41] C. Cimpanu, "Nsa releases ghidra, a free software reverse engineering toolkit," ZDNet, (Accessed 10-23-2020). [Online]. Available: https://www.zdnet.com/article/nsa-release-ghidra-a-free-software-reverse-engineering-toolkit/

[42] L. H. Newman, "The nsa makes ghidra, a powerful cybersecurity tool, open source," Wired, (Accessed 10-23-2020). [Online]. Available: https://www.wired.com/story/nsa-ghidra-open-source-tool/

[43] K. Sheridan, "Nsa researchers talk development, release of ghidra sre tool," Dark Reading, (Accessed 10-23-2020). [Online]. Available: https://www.darkreading.com/endpoint/nsa-researchers-talk-development-release-of-ghidra-sre-tool/d/d-id/1335536

[44] D. Winder, "Nsa releases security research tool but can you trust it?" Forbes, (Accessed 10-23-2020). [Online]. Available: https://www.forbes.com/sites/daveywinder/2019/03/07/nsa-releases-super-spooks-security-tool-so-would-you-trust-it/#193731362c59

[45] L. Franceschi-Bicchierai, "Releasing the nsa's previously classified tool 'ghidra' for free is a 'game changer'," Vice, (Accessed 10-23-2020). [Online]. Available: https://www.vice.com/en/article/panvm7/nsa-releases-ghidra-for-free-game-changer

[46] S. Vavra, "Nsa's reverse-engineering malware tool, ghidra, to get new features to save time, boost accuracy," Cyberscoop, (Accessed 10-23-2020). [Online]. Available: https://www.cyberscoop.com/ghidra-nsa-new-version-black-hat-2019/

[47] J. Uchill, "Nsa releases cybersecurity tool to the public," Axios, (Accessed 10-23-2020). [Online]. Available: https://www.axios.com/nsa-releases-cybersecurity-tool-open-source-3c94ebe4-8229-428d-876c-47a08e2c08e3.html

[48] S. Khandelwal, "Nsa releases ghidra source code — free reverse engineering tool," Hacker News, (Accessed 10-23-2020). [Online]. Available: https://thehackernews.com/2019/03/ghidra-reverse-engineering-tool.html

[49] M. Team, “Nsa released ghidra, its multi-platform reverse engineering framework,” Cyber Defense Mechanism, (Accessed 10-23-2020). [Online]. Available: https://www.cyberdefensemagazine.com/nsa-released-ghidra-its-multi-platform-reverse-engineering-framework/

[50] P. Paganini, “Nsa releases the source code of the ghidra reverse engineering framework,” Security Affairs, (Accessed 10-23-2020). [Online]. Available: https://securityaffairs.co/wordpress/83341/malware/ghidra-source-code.html

[51] L. Mamykina, D. Nakikj, and N. Elhadad, “Collective sensemaking in online health forums,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI ?15. New York, NY, USA: Association for Computing Machinery, 2015, p. 3217?3226. [Online]. Available: https://doi.org/10.1145/2702123.2702566

[52] A. Zhang, B. Culbertson, and P. Paritosh, “Characterizing online discussion using coarse discourse sequences,” 2017.

[53] M.-A. Storey, L. Singer, B. Cleary, F. Figueira Filho, and A. Zagalsky, “The (r) evolution of social media in software engineering,” in *Future of Software Engineering Proceedings*, ser. FOSE 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 100–116. [Online]. Available: https://doi.org/10.1145/2593882.2593887

[54] T. Lopez, T. Tun, A. Bandara, M. Levine, B. Nuseibeh, and H. Sharp, “An anatomy of security conversations in stack overflow,” ser. ICSE-SEIS ’19. IEEE Press, 2019, p. 31–40. [Online]. Available: https://doi.org/10.1109/ICSE-SEIS.2019.00012

[55] S. Becker, C. Wiesen, N. Albartus, N. Rummel, and C. Paar, “An exploratory study of hardware reverse engineering — technical and cognitive processes,” in *Proc. of the 16th Symposium on Usable Privacy and Security*, ser. SOUPS ’20. USENIX Association, Aug. 2020, pp. 285–300. [Online]. Available: https://www.usenix.org/conference/soups2020/presentation/becker

[56] A. Bryant, “Understanding how reverse engineers make sense of programs from assembly language representations,” Ph.D. dissertation, US Air Force Institute of Technology, 01 2012.

[57] D. Votipka, S. Rabin, K. Micinski, J. S. Foster, and M. L. Mazurek, “An observational investigation of reverse engineers’ processes,” in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 1875–1892. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/votipka-observational

[58] T. Nosco, J. Ziegler, Z. Clark, D. Marrero, T. Finkler, A. Barbarello, and W. M. Petullo, “The industrial age of hacking,” in *Proc. of the 29th USENIX Security Symposium*, ser. USENIX Security ’20. USENIX Association, Aug. 2020, pp. 1129–1146. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/nosco

[59] M. Fang and M. Hafiz, “Discovering buffer overflow vulnerabilities in the wild: An empirical study,” in *Proc. of the 8th International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM ’14. ACM, 2014, pp. 23:1–23:10.

[60] M. Hafiz and M. Fang, “Game of detections: how are security vulnerabilities discovered in the wild?” *Empirical Software Engineering*, vol. 21, no. 5, pp. 1920–1959, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10664-015-9403-7

[61] Y. Acar, M. Backes, S. Fahl, D. Kim, M. L. Mazurek, and C. Stransky, “You get where you’re looking for: The impact of information sources on code security,” in *Proceedings of the 37th IEEE Symposium on Security and Privacy*, ser. IEEE S&P, May 2016, pp. 289–305.

[62] F. Fischer, K. Böttinger, H. Xiao, C. Stransky, Y. Acar, M. Backes, and S. Fahl, “Stack overflow considered harmful? the impact of copy paste on android application security,” in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 121–136.

[63] S. Nadi, S. Krüger, M. Mezini, and E. Bodden, “"jumping through hoops": Why do java developers struggle with cryptography apis?” in *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*, 2016, pp. 935–946.

[64] C. Ragkhitwetsagul, J. Krinke, M. Paixao, G. Bianco, and R. Oliveto, “Toxic code snippets on stack overflow,” *IEEE Transactions on Software Engineering*, pp. 1–1, 2019.

[65] D. Votipka, K. R. Fulton, J. Parker, M. Hou, M. L. Mazurek, and M. Hicks, “Understanding security mistakes developers make: Qualitative analysis from build it, break it, fix it,” in *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, Aug. 2020, pp. 109–126. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/votipka-understanding

[66] L. Singer, F. Figueira Filho, and M.-A. Storey, “Software engineering at the speed of light: How developers stay current using twitter,” in *Proc. of the 36th International Conference on Software Engineering*, ser. ICSE 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 211–221. [Online]. Available: https://doi.org/10.1145/2568225.2568305

[67] F. Calefato, F. Lanubile, and N. Novielli, “How to ask for technical help? evidence-based guidelines for writing questions on stack overflow,” *Information and Software Technology*, vol. 94, pp. 186 – 207, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0950584917301167

[68] L. Ponzanelli, A. Mocci, A. Bacchelli, and M. Lanza, “Understanding and classifying the quality of technical forum questions,” in *Proc. of the 14th International Conference on Quality Software*, 2014, pp. 343–352.

[69] C. Treude, O. Barzilay, and M. Storey, “How do programmers ask and answer questions on the web?: Nier track,” in *Proc. of the 33rd International Conference on Software Engineering (ICSE)*, 2011, pp. 804–807.

[70] M. Allamanis and C. Sutton, “Why, when, and what: Analyzing stack overflow questions by topic, type, and code,” in *2013 10th Working Conference on Mining Software Repositories (MSR)*, 2013, pp. 53–56.

[71] S. Beyer and M. Pinzger, “A manual categorization of android app development issues on stack overflow,” in *2014 IEEE International Conference on Software Maintenance and Evolution*, 2014, pp. 531–535.

[72] A. Barua, S. W. Thomas, and A. Hassan, “What are developers talking about? an analysis of topics and trends in stack overflow,” *Empirical Software Engineering*, vol. 19, pp. 619–654, 2012.

[73] A. Rahman, A. Partho, P. Morrison, and L. Williams, “What questions do programmers ask about configuration as code?” in *Proceedings of the 4th International Workshop on Rapid Continuous Software Engineering*, ser. RCoSE ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 16–22. [Online]. Available: https://doi.org/10.1145/3194760.3194769

[74] C. Rosen and E. Shihab, “What are mobile developers asking about? a large scale study using stack overflow,” *Empirical Softw. Engg.*, vol. 21, no. 3, p. 1192–1223, Jun. 2016. [Online]. Available: https://doi.org/10.1007/s10664-015-9379-3

[75] X.-L. Yang, D. Lo, X. Xia, Z. Wan, and J.-L. Sun, “What security questions do developers ask? a large-scale study of stack overflow posts,” *Journal of Computer Science and Technology*, vol. 31, pp. 910–924, 09 2016.

[76] N. Patnaik, J. Hallett, and A. Rashid, “Usability smells: An analysis of developers’ struggle with crypto libraries,” in *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. Santa Clara, CA: USENIX Association, Aug. 2019. [Online]. Available: https://www.usenix.org/conference/soups2019/presentation/patnaik

[77] M. Tahaei, K. Vaniea, and N. Saphra, “Understanding privacy-related questions on stack overflow,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–14. [Online]. Available: https://doi.org/10.1145/3313831.3376768

[78] Y. Yu, G. Yin, H. Wang, and T. Wang, “Exploring the patterns of social behavior in github,” in *Proc. of the 1st International Workshop on Crowd-Based Software Development Methods and Technologies*, ser. CrowdSoft 2014. New York, NY, USA: Association for Computing Machinery, 2014, p. 31–36. [Online]. Available: https://doi.org/10.1145/2666539.2666571

[79] B. Vasilescu, V. Filkov, and A. Serebrenik, “Stackoverflow and github: Associations between software development and crowd-sourced knowledge,” in *Proc. of the 2013 International Conference on Social Computing*, 2013, pp. 188–195.

[80] M. Squire, ""should we move to stack overflow?" measuring the utility of social media for developer support," in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, vol. 2, 2015, pp. 219–228.

[81] B. Vasilescu, A. Serebrenik, P. Devanbu, and V. Filkov, "How social q&a sites are changing knowledge sharing in open source software communities," in *Proc of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, ser. CSCW '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 342–354. [Online]. Available: https://doi.org/10.1145/2531602.2531659

[82] A. Strauss and J. Corbin, *Basics of qualitative research*. Newbury Park, CA: Sage, 1990, vol. 15.

[83] "King ghidorah - wikipedia," https://en.wikipedia.org/wiki/King_Ghidorah, 2020.

[84] "Nationalsecurityagency/ghidra: Ghidra is a software reverse engineering (sre) framework," https://github.com/NationalSecurityAgency/ghidra, 2020.

[85] D. G. Freelon, "Recal: Intercoder reliability calculation as a web service," *International Journal of Internet Science*, vol. 5, no. 1, pp. 20–33, 2010.

[86] A. F. Hayes and K. Krippendorff, "Answering the call for a standard reliability measure for coding data," *Communication methods and measures*, vol. 1, no. 1, pp. 77–89, 2007. [Online]. Available: http://dx.doi.org/10.1080/19312450709336664

[87] B. Knighton and C. Delikat, "Ghidra - journey from classified nsa tool to open source," 2019, blackHat. [Online]. Available: https://www.blackhat.com/us-19/briefings/schedule/index.html#ghidra---journey-from-classified-nsa-tool-to-open-source-16309

[88] R. Joyce, "Come get your free nsa reverse engineering tool!" 2019, rSA Conference. [Online]. Available: https://www.rsaconference.com/industry-topics/presentation/come-get-your-free-nsa-reverse-engineering-tool

[89] E. Wenger, *Communities of Practice: Learning, Meaning, and Identity*, ser. Learning in Doing: Social, Cognitive and Computational Perspectives. Cambridge University Press, 1999. [Online]. Available: https://books.google.com/books?id=heBZpgYUKdAC

[90] A. C. Cameron and P. K. Trivedi, *Regression analysis of count data*. Cambridge university press, 2013, vol. 53.

[91] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953. [Online]. Available: https://doi.org/10.1007/BF02289263

[92] A. E. Raftery, "Bayesian model selection in social research," *Sociological methodology*, pp. 111–163, 1995.

[93] F. Thung, T. F. Bissyandé, D. Lo, and L. Jiang, "Network structure of social coding in github," in *2013 17th European Conference on Software Maintenance and Reengineering*, 2013, pp. 323–326.

[94] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Lawrence Erlbaum Associates, 1988.

[95] H. J. Seltman, "Experimental design and analysis," *Online at: http://www. stat. cmu. edu/, hseltman/309/Book/Book. pdf*, 2012.

[96] K. P. F.R.S., "X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *Philosophical Magazine*, vol. 50, no. 302, pp. 157–175, 1900.

[97] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979. [Online]. Available: http://www.jstor.org/stable/4615733

[98] H. Cramér, *Mathematical methods of statistics (PMS-9)*. Princeton university press, 2016, vol. 9.

[99] P. McCullagh, "Regression models for ordinal data," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 42, no. 2, pp. 109–142, 1980. [Online]. Available: http://www.jstor.org/stable/2984952

[100] P. Turner, "Retpoline: a software construct for preventing branch-target-injection," Google, (Accessed 10-08-2020). [Online]. Available: https://support.google.com/faqs/answer/7625886

[101] M. E. J. Newman and J. Park, "Why social networks are different from other types of networks," *Phys. Rev. E*, vol. 68, p. 036122, Sep 2003. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.68.036122

[102] M. E. J. Newman, S. Forrest, and J. Balthrop, "Email networks and the spread of computer viruses," *Phys. Rev. E*, vol. 66, p. 035101, Sep 2002. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.66.035101

[103] W. P. Wu and C. C. Lien, "Cyberbullying: An empirical analysis of factors related to anonymity and reduced social cue," in *Information, Communication and Engineering*, ser. Applied Mechanics and Materials, vol. 311. Trans Tech Publications Ltd, 5 2013, pp. 533–538.

[104] P. G. Zimbardo, "The human choice: Individuation, reason, and order versus deindividuation, impulse, and chaos." in *Nebraska symposium on motivation*. University of Nebraska press, 1969.

[105] E. Chandrasekharan, M. Samory, S. Jhaver, H. Charvat, A. Bruckman, C. Lampe, J. Eisenstein, and E. Gilbert, "The internet's hidden rules: An empirical study of reddit norm violations at micro, meso, and macro scales," *Proc. ACM Hum.-Comput. Interact.*, vol. 2, no. CSCW, Nov. 2018. [Online]. Available: https://doi.org/10.1145/3274301

[106] J. Cheng, M. Bernstein, C. Danescu-Niculescu-Mizil, and J. Leskovec, "Anyone can become a troll: Causes of trolling behavior in online discussions," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, ser. CSCW '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1217–1230. [Online]. Available: https://doi.org/10.1145/2998181.2998213

[107] E. Seymour and N. M. Hewitt, *Talking About Leaving: Why Undergraduates Leave the Sciences*. Westview Press, 2000.

[108] B. Mathew, H. Tharad, S. Rajgaria, P. Singhania, S. K. Maity, P. Goyal, and A. Mukherjee, "Thou shalt not hate: Countering online hate speech," in *AAAI International Conference On Web and Social Media*, ser. ICWSM '19, 2019.

[109] L. Blackwell, T. Chen, S. Schoenebeck, and C. Lampe, "When online harassment is perceived as justified," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, no. 1, Jun. 2018. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/15036

[110] J. Cheng, C. Danescu-Niculescu-Mizil, and J. Leskovec, "How community feedback shapes user behavior," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 8, no. 1, May 2014. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14518

[111] T. O. Cunha, I. Weber, H. Haddadi, and G. L. Pappa, "The effect of social feedback in a reddit weight loss community," in *Proceedings of the 6th International Conference on Digital Health Conference*, ser. DH '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 99–103. [Online]. Available: https://doi.org/10.1145/2896338.2897732

# Appendix

In this appendix, we give full descriptions of each codebook used to analyze discussion threads.

## 1. Ghidra Features

For each discussion thread, we coded the set Ghidra features discussed. The individual codes are taken from the issue tags on the offical Ghidra github page [84]. Features are also grouped into related feature areas.

- Scripts
  - Extensions - Threads about updates to the framework itself (e.g., script manager, exporter, decompiler). For example, changing the script manager to use scripts in unsupported languages (like Ruby).

  Extensions can mess with the UI (add buttons or menu items). Extensions can also produce and consume events and can share information with other extensions through a service architecture.
  - FID - Threads about any analyses that identify functions that match common functions.
  - API - Threads about the API provided to extension and scripts that allow them to use features of the main Ghidra framework.
  - Script Manager - Threads about the function that runs the scripts and allows reverse engineers to pick which things you want to run.
  - Analysis - Threads about things that get run whenever you import a binary and decompile (which could include scripts). For example, threads about getting some analysis running in Ghidra (either included or one they created or downloaded).
  - Scripting - Threads about code users can write to extend the analysis provided by Ghidra. Scripts do not update the UI (e.g., add buttons or menu items), that's extensions. An example of a script is code that creates a new analysis.
- Architecture
  - Processor - Threads about supporting a new architecture or fixing support for an architecture.
  - Sleigh - Threads about the code used for converting a given architecture into decompiled C representation.
  - Loader - Threads related to different binary file formats. Loaders tell Ghidra how new files are formatted, where different information is stored, how it's stored, and how it can be used to reconstruct the decompiled code. For example, when threads discuss how Ghidra represents specific addresses internally.
- Memory
  - Data Types - Threads about adding custom data types or structs to represent groups of data points and then applied to the binary.
  - BitFields - Threads related to the feature that allows the struct to define a variable with a given set of bits (instead of using a whole byte). For example, the reverse engineer only needs one bit for a boolean variable, so they only need a bitfield with one bit instead of setting aside 8 bits for that variables.
  - Memory - Threads about the actual bytes of a program split into sections (e.g., data section, code section, etc). For example, how Ghidra determines which segment of code belong to which part. This is different from *Loader* in that it is talking about sections broadly, not specific addresses.
  - C-Parser - Threads discussing the feature that allows reverse engineers to create a C header with all the data types they want instead of manually adding each by hand. Reverse engineers can write a header in C, then Ghidra parses it and updates the decompiled code to use them where appropriate.
- Variable/Function Names
  - Symbol Table - Threads about the database that stores the list of symbols (variable and function names) of a program.
  - Symbol Tree - This features is similar to the symbol table, but a different version of the visualization that shows symbols in a hierarchical view. This means the reverse engineer see the symbols in a struct and maybe some other metatdata if they expand the symbol. This looks something like a file system tree.
  - Program Tree - Threads about the Ghidra-specific abstract syntax tree (called p-code) representation of the program.
  - PDB - Threads related to importing debug symbols into the program to provide additional information (or debugging the import of this information).
  - DWARF - Threads about the specific DWARF format of debugging information file.
- Tool
  - Tool - Threads discussing one-off features the provide support for specific needs such as an entropy window (e.g., how similar are bytes across the program, helping you figure out if a program is compressed), search bar, and bookmarks.
  - Byte Viewer - Threads about the hex editor that lets reverse engineers look at the raw hex of the binary.
  - Decompiler - Threads related to the tool that converts the binary to a high-level C-like language.
  - Demangler - Some compilers change the names of functions if they are overloaded (e.g., same name, different arguments) so that the differences is referenced in the function name (this is important for C linkers). Whenever an reverse engineer wants to decompile, they need to reverse this process. Discussions of this process are coded as *Demangler*.
  - Disassembler - Discussion about the tool that shows the assembly view of the program.
  - Filesystem - Ghidra allows the reverse engineer to unpack and traverse firmwares that include internal filesystems. This code covers any of the related features.
  - Version Tracking - Threads about Ghidra's tool that will look at a new version of a binary the reverse engineer worked with previously, that tries to identify similarities and port over changes and annotations into the new binary that the reverse engineer made to the previous version to limit duplication of effort.
  - Version Control - Threads about multi-user support that allows multiple people to edit the binary together (similar to git).
  - Server - Threads regarding the collaborative use of Ghidra. People can set up a Ghidra server to share projects between multiple users.
  - Emulation - Threads discussing Ghidra's API that will let you emulate different architectures and run the program.
  - Exporter - Threads related to exporting modified (patched) versions of binaries or other data generated when using Ghidra (e.g., variable renaming).
- Setup/Run
  - Build - Threads discussing compiling Ghidra from source.
  - Launch - Threads related to getting Ghidra running.

- Eclipse - Threads related to building scripts with the GhidraDev Eclipse plugin.
  - Platform - Threads about getting Ghidra working on a particular operating system (e.g., Windows, Linux, Mac).
- GUI
  - Graphing - Threads about Ghidra's Control flow graph view within the UI.
  - GUI - Threads about the buttons, menus, tables, and views shown to the user.
  - Headless - Threads related to running Ghidra in headless mode (e.g., just running scripts and analysis without a UI).
- Overview/Vulnerabilities/Bonding
  - Overview - Threads that provides a resource giving a walkthrough of Ghidra.
  - Vulnerability - Threads related to a vulnerability in Ghidra.
  - Community - Threads about building a community of Ghidra users.
  - Support - Threads about the Ghidra team making bug fixes.
  - Humor - Threads that do not really provide any information, but just make a joke about Ghidra.
- Learning
  - Learning - Threads related to teaching people how to use Ghidra.
  - RE Example - Threads that demonstrate the use of Ghidra.
  - Comparison - Threads comparing Ghidra to another reverse engineering tool.
- Information about Ghidra
  - Documentation - Threads about getting documentation on Ghidra.s
  - Website - Threads about the Ghidra website (https://ghidra-sre.org/) itself where information is hosted (Not the program).

## 2. Conversational Actions

This codebook is divided into two parts. First, we coded the type of conversation initiated by the initial post in each thread. This codebook is based on the codebook developed by Mamykina et al. [51]. Next, we coded each comment of each discussion thread to determine the type of discourse each comment represented. This codebook is based on the codebook developed by Zhang et al. [52].

### 2.1. Thread type.

- Announcements - Posts of a declarative nature that simply make a statement about an event, product, or feature of interest without looking for any particular response from others.
- Questions - Threads initiated by individuals seeking advice in regards to a specific personal issue.
- Statement of strong opinion threads (Opinion) - Posts that state a strong opinion in regards to a certain issue.
- Opinion/experience polls (Poll) - Posts that seek opinions or experiences of others on a topic of interest.
- Bonding threads - Threads that asked individuals to share their experiences related to the topic in general, rather than to any aspect of it in particular.
- Bounty - Threads where the initial author wants to pay someone else to do something with Ghidra (e.g., write a new script).
- Ice-breakers - Simple games usually not related to the topic.

### 2.2. Discourse Acts.

- Question - A comment with a question or a request seeking some form of feedback, help, or other kinds of responses. While the comment may contain a question mark, it is not required. For instance, it might be posed in the form of a statement but still soliciting a response. Also, not everything that has a question mark is automatically a QUESTION. For instance, rhetorical questions are not seeking a response.
- Answer: A comment that is responding to a QUESTION by answering the question or fulfilling the request.
- Statement of a perspective: Related information, but not solving the question.
- Reframing the problem - Elaboration or restatement of the original question
- Sharing a resource: A comment that is presenting some new information to the community, such as a piece of news, a link to something, a story, an opinion, a review, or insight.
- Agreement: A comment that is expressing agreement with some information presented in a prior comment. It can be agreeing with a point made, providing supporting evidence, providing a positive example or experience, or confirming or acknowledging a point made.
- Appreciation: A comment that is expressing thanks, appreciation, excitement, or praise in response to another comment. In contrast to AGREEMENT, it is not evaluating the merits of the points brought up. Comments of this category are more interpersonal as opposed to informational.
- Disagreement: A comment that is correcting, criticizing, contradicting, or objecting to a point made in a prior comment. It can also be providing evidence to support its disagreement, such as an example or contrary anecdote.
- Negative Reaction: A comment that is expressing a negative reaction to a previous comment, such as attacking or mocking the commenter, or expressing emotions like disgust, derision, or anger, to the contents of the prior comment. This comment is not discussing the merits of the points made in a prior comment or trying to correct them.
- Elaboration on answer: A comment that is adding additional information on to another answer. Oftentimes, one can imagine it simply appended to the end of the comment it elaborates on. Note: This must be about another commenter's statement.
- Synthesis of previously stated perspectives: A comment summarizes previously provided information. This can summarize information from other communications threads.
- Personal reconciliation: For example, "I'm sorry if I offended you."
- Humor: This comment is primarily a joke, a piece of sarcasm, or a pun intended to get a laugh or be silly

but not trying to add information. If a comment is sarcastic but using sarcasm to make a point or provide feedback, then it may belong in a different category.

## 3. Sensemaking

Our final codebook considered the level of collective sensemaking displayed in each thread. This codebook is based on the codebook developed by Mamykina et al. [51].

- Reaction to previous perspectives - Many posts in threads with a high degree of collective sensemaking begin with reflection on previously stated perspectives. These simple references place new posts in the context of other contributions and help to move the discussion forward. A thread is considered to contain a reaction if at least one commenter reflects on the comments of others.
- Lateral engagement between participants - Thread participants not only express their own individual perspectives but also engage in the back and forth negotiation of meaning. For example, authors might interrogate each other's perspectives, weighing evidence or sharing relevant experiences. Requires back and forth discussion and reflection between commenters.
- Transformation of ideas - Threads not only reflect on the previously proposed perspectives, but also developed those ideas beyond their original form.