



Bug Hunters' Perspectives on the Challenges and Benefits of the Bug Bounty Ecosystem

Omer Akgul, University of Maryland; Taha Eghtesad, Pennsylvania State University; Amit Elazari, University of California, Berkeley; Omprakash Gnawali, University of Houston; Jens Grossklags, Technical University of Munich; Michelle L. Mazurek, University of Maryland; Daniel Votipka, Tufts University; Aron Laszka, Pennsylvania State University

<https://www.usenix.org/conference/usenixsecurity23/presentation/akgul>

This paper is included in the Proceedings of the 32nd USENIX Security Symposium.

August 9–11, 2023 • Anaheim, CA, USA

978-1-939133-37-3

Open access to the Proceedings of the 32nd USENIX Security Symposium is sponsored by USENIX.

Bug Hunters' Perspectives on the Challenges and Benefits of the Bug Bounty Ecosystem

Omer Akgul[◇]
akgul@umd.edu

Taha Eghtesad*
teghtesad@psu.edu

Amit Elazari[§]
aelazari@berkeley.edu

Omprakash Gnawali⁺
gnawali@cs.uh.edu

Jens Grossklags[¶]
jens.grossklags@in.tum.de

Michelle L. Mazurek[◇]
mmazurek@umd.edu

Daniel Votipka[‡]
dvotipka@cs.tufts.edu

Aron Laszka*
laszka@psu.edu

[◇]University of Maryland

*Pennsylvania State University

[¶]Technical University of Munich

[§]University of California, Berkeley

⁺University of Houston

[‡]Tufts University

Abstract

Although researchers have characterized the bug-bounty ecosystem from the point of view of platforms and programs, minimal effort has been made to understand the perspectives of the main workers: bug hunters. To improve bug bounties, it is important to understand hunters' motivating factors, challenges, and overall benefits. We address this research gap with three studies: identifying key factors through a free listing survey ($n=56$), rating each factor's importance with a larger-scale factor-rating survey ($n=159$), and conducting semi-structured interviews to uncover details ($n=24$). Of 54 factors that bug hunters listed, we find that rewards and learning opportunities are the most important benefits. Further, we find scope to be the top differentiator between programs. Surprisingly, we find earning reputation to be one of the least important motivators for hunters. Of the challenges we identify, communication problems, such as unresponsiveness and disputes, are the most substantial. We present recommendations to make the bug-bounty ecosystem accommodating to more bug hunters and ultimately increase participation in an underutilized market.

1 Introduction

Traditionally, organizations relied on internal security experts (e.g., red teams) and outsourced experts (e.g., penetration testing) to discover vulnerabilities in their products. In contrast, bug-bounty programs—also known as vulnerability-reward programs or “crowd-sourced” security—incentivize independent security experts to evaluate the security of an organization's products and report vulnerabilities in exchange for rewards (financial or otherwise, such as the learning opportunity). Bug-bounty programs were initially spearheaded by Netscape in 1995 [26]; now, many companies (e.g., Google, Apple) and governmental agencies (e.g., U.S. Department of Defense) run bug-bounty programs.

However, due to their crowd-sourced nature, bug-bounty programs also suffer from inefficiencies. Bug-bounty programs may receive invalid or duplicate reports, wasting ef-

fort [7, 62]. Further, programs compete to attract productive hunters, and older programs struggle to maintain a hunter pool [48, 55]. On the other hand, bug hunters (hereafter, *hunters*) face uncertainties regarding their findings and rewards, and are often disappointed by program responses [2, 54]. To mitigate some of these issues, bug-bounty platforms such as Bugcrowd [11] and HackerOne [32] have emerged, connecting bug-bounty programs to hunters through a marketplace. However, many issues persist.

Perhaps due to these issues, despite many benefits, bug-bounty adoption by organizations has remained relatively low¹ and well below predictions (as cited in [12, 55]). Further, only a small fraction of hunters find a substantial amount of bugs [25, 46, 59] despite the seemingly populous talent-pool reported [14, 35]. These hunters tend to receive more attention from the ecosystem, while others are ignored [26].

Identifying specific factors that make bug-bounty programs (un)attractive and (un)successful, addressing the most significant challenges, and bolstering commonly enjoyed benefits in the bug-bounty ecosystem could invite more hunters, increase their commitment, enable identification of more bugs, reduce wasted effort in bug hunting and reporting, and streamline the process of fixing reported bugs; all of which could improve the security posture of many companies and improve software security more broadly. Additionally, by understanding hunter motivations, we can also understand the societal impacts of the bug bounty market and guide the efforts of regulators to ensure the market meets society's broader needs.

To improve the bug bounty ecosystem, we must first understand how bug bounties work. Indeed, a number of research efforts have taken steps in this direction [24, 28, 43, 44, 46, 48, 55, 59, 61]. However, a common limitation is that researchers consider data collected only from the perspective of bug-bounty programs (e.g., vulnerability reports and payments). Therefore, they provide only a limited view of bug

¹The two largest bug-bounty platforms host ~3,300 (global) bug-bounty programs [14, 35] compared to more than 10,000 “software publishers” and 60,000 “custom computer programming services” in the U.S. alone [16]. Most (80%) of the Forbes 500 have no vulnerability disclosure program [8].

hunters' work, considering only final outputs but neglecting the hunters' motivations and the challenges that they face.

Recent work has begun to consider decision-making in vulnerability discovery broadly [30, 56, 57]; but none of this work focuses on bug bounties specifically, discussing them only when broached by participants. Additionally, none of this hacker-focused work has empirically evaluated factors that govern hunters' choices of which software to evaluate.

Bug-bounty platforms have themselves issued several reports on hunters' motivations [13–15, 34–36]. However, these brief reports do not focus on challenges faced by hunters, appear to be for marketing, and are not independently verified.

Other research efforts have used interviews with stakeholders throughout vulnerability disclosure, including bug bounty programs, to explore drawbacks of the gig-work model [26] and challenges in the vulnerability discovery ecosystem as a whole [4]. We build upon these broad, qualitative studies with a larger, mixed-methods sample that explicitly and systematically identifies and quantifies the factors that affect hunters' participation in bug-bounty programs. Unlike prior work, we ask hunters to quantify how important individual factors are, allowing stakeholders to know which factors to prioritize. For instance, like other researchers [26, 59], we find reputation to be a motivator; however, we are able to demonstrate that it is in fact one of the least important motivators (§5.1).

Our specific research questions are as follows:

RQ1: What are the factors that hunters consider and challenges they face when participating in bug bounties?

RQ2: How important are these factors to hunters? Why?

We approach these two questions in four contexts: (1) factors considered when choosing between specific programs, (2) challenges faced, (3) benefits of bug bounties in general, and (4) useful features of bug-bounty platforms.

We conduct three studies (§3) to address our research questions: an initial factor-identification survey ($n=56$) to list prominent factors at play (RQ1), a larger survey ($n=159$) to find the importance of the factors (RQ2) and a semi-structured interview study to reveal why factors are important ($n=24$).

As expected, we find the most salient benefits to be monetary, both when choosing between programs and as a general motivator (§5.1). When choosing between programs, hunters consider heuristics (e.g., scope) that increase the probability of finding a bug (§5.3). Aside from monetary benefits, hunters deeply value learning opportunities (§5.2). Contrary to intuition from prior work [25, 55, 59], we find that they value reputation much less than other factors.

We observe that hunters' most prominent challenges are communication issues with bug-bounty program managers who grade reports and decide on payouts. Specific issues include poor responsiveness, bug-grading disputes, and dissatisfaction with mediation and platform triaging (§5.4).

We also find that the gig-work model of bug bounties introduces unique challenges for hunters. While it provides flexibility, it can also create stress and uncertainty (§5.5).

We discuss bug-bounty platform features that hunters consider most useful: public dashboards and easy procedures for reporting bugs and receiving payments (§5.6).

Next, our results show the importance of legal safe harbors to hunters in multiple contexts (§5.7).

The paper concludes with a discussion of how our results expand our understanding of benefits and challenges in bug bounties, as well as recommendations for a bug bounty ecosystem that works better for bug hunters, companies, and software security in general (§6).

2 Related Work

Researchers have tried to understand hunters' motivations through empirical analysis of market behaviors and via direct surveys and interviews with hunters.

Market behaviors To understand how hunters select bug-bounty programs, researchers have studied empirical data produced by bug-bounty programs (e.g., vulnerability reports and payments) [3, 28, 40, 47, 48, 52, 55, 59, 63]. These studies investigate the relationship between hunter activity and various program features, highlighting correlations that might suggest motivations. For example, researchers found that hunter program selections were associated with expected monetary rewards and program age [44, 48]. These results are in line with similar investigations of public reporting from the Google Chrome and Mozilla Firefox bug-bounty programs [28] and public HackerOne data [48]. Though we report some overlapping factors, our work offers a substantially different perspective, as we survey hunters directly, allowing them to tell us their priorities directly rather than inferring them.

Hunters' self-reported motivation Other publications have leveraged surveys to characterize hunter demographics and motivations. The most prominent examples of this work are marketing materials produced annually by HackerOne [35, 36] and Bugcrowd [13–15], the two largest bug-bounty platforms. Each company surveys the hunters participating on their platform, collecting demographics and a high-level view of bug bounty participants' motivations (e.g., money, education). However, these surveys do not provide the same depth of exploration into hunter motivation as our work and do not focus on challenges faced by hunters.

Perhaps most related to our work is a non-profit research organization's interview study of bug bounty ecosystem stakeholders to understand their experiences [26]. The authors suggest but do not systematically define several benefits and challenges for hunters, with a primary focus on criticizing the gig-work model. Our work differentiates itself by employing mixed methods to systematically identify, define, and *quantify* factors relevant to bug bounties under four contexts: choosing between bug-bounty programs, challenges of bug bounties, benefits of bug bounties, and useful features of bug-bounty

platforms. Our quantification of relevant factors enables stakeholders of the bug bounty ecosystem—including bug-bounty platforms but also hunters, companies seeking to improve their security, and potentially regulators or standards bodies concerned with security—to make better informed decisions.

A more general study explored the vulnerability disclosure process as a whole, finding communication to often be an issue [4]. Fulton et al. explored issues marginalized groups face in the vulnerability discovery space (e.g., women, people of color) [30]. While both studies do touch briefly on hunters' perceptions of bug bounties, it is tangential to their research.

3 Method

We designed and conducted three studies to investigate our research questions: an initial *free-listing study* to determine factors at play (RQ1), a *factor-rating study* (RQ2), and finally an *interview study* (RQ2). The first two studies allow us to understand what motivates and challenges hunters, while the interview study contextualizes these results.

Our institutions' ethics review boards approved all three studies. Participants signed consent forms detailing study plans and participant rights before data collection. Identifiable data was only available to authors named on the ethics review.

3.1 Free-listing study (RQ1)

To identify factors that influence participation in bug bounties, we performed an online survey on hunters ($n=56$).

Survey The survey began with open-ended questions asking participants to list factors that affect them in five (later reduced to four, see end of §3.1) contexts (we call these *factor groups*): (1) factors when choosing between bug-bounty programs, (2) reasons for leaving bug-bounty programs, (3) the benefits of participating in bug bounties, (4) challenges faced in general, and (5) useful features of bug-bounty platforms.

For each question, we stressed that initial study participants should list all factors they may consider, even if they do not regard a given factor in every single decision. Additionally, we asked initial study participants to spend time to recall factors if they thought there might be more they could remember. Common in listing exercises, this prompt allows us to elicit less obvious factors [9]. We use an open-ended listing approach, called *free listing*, common in anthropological research when the domain is not well understood [9]. This is useful for eliciting the full breadth of possible factors.

Next, we asked participants to self-report their bug-bounty experience (see Table 1) and skills. Finally, we concluded with standard demographic questions (see Table 1) to understand our sample population. We also asked if participants were willing to be contacted for follow-up studies. The full survey can be found in the extended paper [1].

Pilots Through personal connections, we recruited three security experts who regularly work on bug bounties. We sent them the survey and discussed responses in an online focus-group session. We proceeded with data collection once the questions were clear and provided good face validity [31].

Recruitment We recruited by advertising on social media (through the authors' accounts), mailing lists, and Slack channels that hunters use. In total, we received 61 complete responses to the survey. We removed 5 responses due to poor quality (unintelligible answers, unreasonably fast completion times, or duplicates), leaving 56 responses for analysis. Responses were obtained from May to December 2019.

We concluded data collection after the final 15 responses largely confirmed the factors identified in the first 41 responses, indicating conceptual saturation [17].

Data Analysis We analyzed open-ended survey responses with exploratory open coding [53]. Because we planned to use the identified factors directly in the factor-rating study, we calculated inter-rater reliability [49].

Sometimes factors listed were polysemous. In cases where these responses came from participants who agreed to an interview, we asked for clarification ($n=7$).

We developed the codebook and established reliability on the first 41 responses (73.2% of all responses, 64.0–78.2% of all listed factors per question²). The initial codebook was developed by three researchers using 10 responses (25% of the responses at the time; 15.3–28.0% of factors listed). Two of the three researchers then attempted to establish good reliability by independently coding batches of 10 responses at a time, resolving differences and updating the codebook after each batch. We ran out of new responses without being able to establish our threshold for acceptable reliability (Cohen's $K > 0.8$). However, after 28 days, the researchers revised the codebook and independently re-coded 16 of 41 responses (~40% of the responses at the time; 27.0–38.3% of all factors listed), achieving “almost perfect” [42] reliability (Cohen's $K > 0.8$ for all factor groups; 0.81–0.91). Finally, with reliability established, one researcher re-coded the rest of the responses. The final 15 responses were received after reliability had been established and coded by one researcher. The final codebook contained 78 factors across five factor groups.

Refining the factors Our listing exercise identified many factors, so it was impractical to ask participants to respond to each directly in the factor-rating study. In addition, we noticed significant overlap between two factor groups: reasons for quitting a bug-bounty program, and challenges faced in the bug bounty context more generally. To reduce the number of questions asked, we merged the two, resulting in 54 factors across four factor groups (see Table 2).

²Responses are *unitized* based on number of factors listed in a response (i.e., codes are assigned to individual factors, not entire responses).

3.2 Factor-rating study (RQ2)

The free-listing study identified a comprehensive set of factors considered by hunters, but not their relative importance. We therefore designed a second study asking participants to rate how important each factor is to them, personally.

Survey The second survey started with survey participants rating each of the 54 factors on a seven-point Likert.³ As in the prior survey, the factors were asked in the context of their respective factor groups: factors considered when choosing a program, challenges faced and benefits of bug bounties, and useful bug-bounty-platform features. To better understand hunters' background, we asked an open-ended question on how they started bug bounties. As before, we finished with questions about the participant's bug-hunting experience and demographics. We again asked whether participants would be willing to be contacted for follow-up studies. The full survey can be found in the extended paper [1].

Piloting We piloted the survey on five hunters, focusing on how well participants understood the provided list of revised factors. We made minor revisions to the naming and explanation of the 54 factors based on these comments. The final list of factors to be evaluated appears in Table 2.

Recruitment In addition to the methods used in the free-listing study, we reached out to 586 hunters who had a public bug report in the first half of 2020 on HackerOne or BugCrowd, listed their Twitter accounts, and allowed direct messages from anyone. The vast majority of responses likely came through this method. Further, we advertised on [reddit.com/r/bugbounty](https://www.reddit.com/r/bugbounty). Of 161 completed survey responses, we discarded two with nonsensical responses to multiple open-ended questions, leaving 159 responses for analysis. We estimate there to be 4-6 overlapping participants between the free-listing and factor-rating study.

Data analysis A straightforward (and common) way of analyzing which factors are the most important would be to convert our participants' Likert choices to numeric values and present simple averages. This approach, while seemingly intuitive, has been criticized and discouraged by statisticians [21, 45, 60], because it makes the dubious assumption that the ordinal options that participants select are equidistant (e.g., the distance between "Extremely important" and "Very important" is the same as between "Very important" and "Moderately important"). Thus, we adopt comparison-based techniques, which consider only whether one factor is rated higher than another. Specifically, we employ log-linear Bradley-Terry (LLBT) modeling to synthesize *worth* estimates (π) that represent the relative importance of factors (f) to participants on a preference scale [21, 23]. The probability

³"Extremely challenging/important" to "Not at all challenging/important" or "Extremely useful" to "Extremely useless" as appropriate.

of one factor being preferred over another is given by:

$$p(f_j > f_k | \pi_j, \pi_k) = \frac{\pi_j}{\pi_j + \pi_k}$$

where j, k denote the indices of factors considered [22].⁴

3.3 Interviews

We invited all consenting free-listing study participants with valid responses to take part in remote semi-structured interviews. We asked if and why the identified factors were important to them. The interviews ($n=8$) started by asking how participants got into bug bounties and security in general. Next, we explained each factor group to provide context and then asked the participant to pick the most important factors and explain their choices. Finally, we asked whether and how they would continue participating in bug bounties.

Similarly, we invited factor-rating study participants who consented to interviews, with minor revisions to the interview protocol. Specifically, we used the revised list of factors (54 items), and asked participants ($n=16$) directly about the factors they rated highly in their survey responses. In both interview rounds, interviews started while survey recruitment was still active in order to retain a higher percentage of participants.

It was infeasible for interviews to cover all 54 factors. Thus, we focused on factors the participant cared about the most, asking about other factors if time allowed. Interviewers were careful to avoid asking redundant questions, as discussion on one factor frequently expanded to others.

With 54 factors to cover and many hunters expressing unique considerations, reaching saturation on participants' opinions of each factor was not realistic; instead, we aimed to collect enough data to contextualize the survey results.

We conducted interviews with 24 people in total, each averaging 43 minutes. While survey participants were not compensated,⁵ interviewees were thanked with a \$20 gift card.

Analysis We again used exploratory open coding to analyze our interviews [53]. Two researchers coded three interviews to create an initial codebook. They then independently coded three interviews at a time, meeting to discuss the interviews, resolving differences, and updating the codebook. The final codebook was obtained when all interviews were coded and discussed. Interviews were re-coded with the final codebook, resulting in a total of 1004 coded segments in the 24 interviews. We did not seek inter-rater reliability metric, since we see the interviews primarily as adding context to the outputs of the free-listing study and factor-rating study [49].

⁴We use a version of LLBT that requires converting our data to explicit paired comparisons. Methods without this conversion are computationally infeasible for our high-dimensional data [21, 38]. Researchers who developed these statistical methods confirmed that our approach is appropriate [29].

⁵Our varied recruitment methods, international participants, and relatively short surveys made compensation logistically difficult.

3.4 Limitations

Our methods are primarily based on self-report data, which is subject to well-known limitations. Self-report data often has high levels of noise and therefore does not ensure definitive answers through one measurement. We therefore investigated our overall research goal through three distinct studies, and noted the few inconsistencies that occurred.

Lack of recall can be an issue in free listing [5]. We prompted participants to try to recall all possible factors and only switch to the next question when they could no longer think of any, a best practice for free-listing recall [9].

Hunters might have portrayed themselves as more successful than they are. Similarly, our gift card incentive might have been most attractive to interview participants who make less money in bug-bounty programs, perhaps due to lower skill or experience. We partially addressed this by recruiting the majority of free-listing study participants from non-public hunter messaging channels and factor-rating study participants from hunters from authors of publicly disclosed bugs, implying some level of expertise. Further, the metrics we collected indicate a relatively even distribution across all skill and experience levels (see Table 1).

On the other hand, it is likely we did not capture people who wanted to participate in bug bounties but were unable to at all. This problem of survivor bias is likely unavoidable, as there is no clear way to recruit people interested, but not active in bug bounties. However, we expect some of our participants with less bug bounty participation will have similar experiences.

Further, all three of our studies, like all self-report studies [10], include some sampling and selection biases. Our sample is likely reasonably representative of the current hacker population (§4) and includes diversity in participant location, education, experience, and skill, meaning our results are reasonably likely to generalize to the average hunter. However, the hunter population itself is demographically skewed (e.g., young, white or Southeast Asian, and male [14, 15, 35]), likely introducing inherent biases (e.g., the companies and vulnerabilities that get attention). For a discussion of marginalized populations, we refer readers to the work of Fulton et al. [30].

Finally, to maximize face validity, we rigorously piloted each stage of the study, revising procedures with feedback.

4 Participants

Table 1 summarizes participants’ self-reported demographics and experiences. We had 56 participants in the free-listing study, 159 in the factor-rating study, and 24 interviewees. Participants were mainly from North America, South Asia, and Europe; young in age; and overwhelmingly male.

Exact demographics of hunters are unknown; however, aside lower education levels, our participants are similar to those reported by popular bug-bounty platforms. They report samples consisting of 77-85% < 35 years old, 90-

		FL	FR	I
Gender	Male	51	147	24
	Female	2	2	0
	Self-described	1	2	0
Age	18-29	34	124	16
	30-39	16	27	7
	40-49	4	3	0
	50-59	2	0	0
Residence	North America	21	22	3
	South Asia	14	64	7
	Europe	14	23	9
	Southeast Asia	2	14	0
	Middle East	0	18	0
	Other	5	11	4
Education	≤ Completed H.S.	18	53	11
	Trade/technical/vocational	1	3	0
	College, no degree	13	17	2
	Associate’s degree	3	5	1
	Bachelor’s degree	15	56	3
	Professional/MS/PhD	5	19	4
Weekly Hours Working on Bug Bounties	<5	13	34	3
	5-10	18	53	8
	10-20	12	40	8
	20-30	8	18	2
	30-40	1	13	2
	>40	4	3	0
Total Number of Bugs Found	<10	6	39	2
	10-99	18	65	10
	≥ 100	12	52	8
Years Working on Bug Bounties	<1	2	3	1
	1-2	18	87	12
	3-5	15	46	4
	6-9	6	14	4
	≥ 10	0	5	0
Skill	1 - Fundamental	3	9	1
	2 - Novice	8	23	3
	3 - Intermediate	23	68	9
	4 - Advanced	10	48	6
	5 - Expert	12	13	4
Yearly Income From Bug Bounties	<\$999	10	33	3
	\$1,000 - \$29,999	19	57	9
	\$29,999 - \$74,999	5	13	2
	≥ \$75,000	10	12	3

Table 1: Participant demographics and experience across studies. Questions were similar between studies but not exactly same. FL: free-listing. FR: factor-rating. I: interview study.

96% male, 19-34% with graduate degrees, and 37-67% working < 10 hours a day on bug bounties [14, 15, 35]. HackerOne’s sample—the only marketing survey to report hunter experience—had similar levels of experience (i.e., 47% < three years of hunting) to ours [35].

5 Factors

We identified 54 factors affecting participation in the bug bounty ecosystem under four factor groups. A complete list can be found in Table 2. Some of these factors are reported in prior work; as shown in Table 2, no previous work has sys-

tematically identified all factors, and most focus on benefits, leaving gaps in challenges and in which platform features are most useful. Further, we provide the first in-depth ranking of factor importance.

In our interviews, we identified seven common themes that connect interrelated factors; we use these themes to organize our discussion. For brevity, we primarily discuss the most popular factors; details about more can be found in the extended paper [1]. We report how many interview participants made an argument (I), how many free-listing study participants listed a factor (FL), and the relative importance of factors (worth estimates) generated from the factor-rating study (π).⁶

It is important to note that because each interviewee only had time to comment on a subset of factors, participant counts from interviews are provided for context but cannot be interpreted as prevalence.

5.1 Earning rewards

Though rewards might seem like an obvious motivation [14, 15, 26, 34–36, 61], hunters expressed nuanced details.

Reward considerations Unsurprisingly, *monetary rewards* were commonly listed by free-listing study participants and were highly ranked by factor-rating study participants both as a consideration for choosing a bug-bounty program (FL=36, $\pi=0.120$) and as the top benefit of bug bounties (FL=42, $\pi=0.191$). Hunters similarly prioritized the mapping of bug severity to reward (i.e., the *bounty table*, FL=16, $\pi=0.117$). *Non-monetary rewards* (e.g., swag; FL=4, $\pi=0.013$) were motivators [26], but were ranked least important.

In interviews, hunters described nuanced preferences for how bounties are determined and managed. Many hunters (I=9) argued that bounties should be correlated with the severity of the identified bug; this aligns well with industry practices [6]. Three specifically mentioned valuing bugs based on how much they might cost the company if exploited. Three hunters who specialized in high-severity bugs mentioned the importance of large rewards for these bugs, with less concern about rewards for less critical bugs.

In contrast, three hunters primarily considered the payout amounts for low- and medium-severity bugs when choosing a bug-bounty program, mainly because they considered finding more critical bugs improbable. One hunter said that they would only take high-severity payment levels into account if they planned to commit to a program for a long time. Other hunters (I=7) argued that payments should be proportional to hunter effort, not just the outcome.

Some (I=7) argued the bounty table was not a good predictor of their expected payment for participating, which would be determined largely by factors such as how many bugs they expected to be able to find, possible bonus payments (e.g., for

⁶Worth estimates can only be meaningfully compared within factor groups, which were analyzed separately; Table 2 provides a complete list.

well-written reports), and the potential for multiple payments (e.g., for re-exploiting a previously patched bug). One noted a preference for a program that “doesn’t have really huge payouts, but ... you’re likely to get more out of it because you actually submit more bugs because you know it better.”

Interestingly, eight interviewees indicated there were other, more important motivations than monetary rewards. Generally motivated by finding as many bugs as possible, one hunter noted, “I try sometimes finding such programs, who doesn’t give monetary rewards. ... It’s easy to find vulnerabilities from such programs.” Another said they report bugs regardless of payment, because “I just want to make companies secure.”

In lieu of monetary rewards, as noted in prior work [26], a few interviewees were content to receive only a reputation boost (I=3); more (I=5) found merchandise sufficient compensation. However, factor-rating study participants overwhelmingly rated reputation and non-monetary rewards as the least important motivators.

5.2 Learning opportunities

All three studies highlight the importance hunters place on being able to learn. Hunters report learning in many ways, e.g., from publicly disclosed bugs and community interactions.

Importance of learning *Learning* new hacking techniques was one of the most frequently listed benefits in the free-listing study and was found to be the second most important benefit by the factor-rating study participants (FL=32, $\pi=0.170$). Though not as immediately apparent, hunters’ focus on learning was evident in several other, related factors that were commonly discussed.

Referring to the learning opportunity that bug-bounty programs provide, most interviewees said that learning is an integral part of the process: a hunter will either learn organically or be forced to learn new techniques to stay relevant (I=12). Specifically, three explained the need to learn new technologies and methods when going after certain bugs. One hunter recalled having to learn GraphQL to find a bug: “What I thought was, let’s learn about it first. So I got into learning GraphQL, and the structures, how it’s written, how you get response, what kind of response you get.”

Three hunters noted that bug bounties also provide an opportunity to practice skills, an integral part of learning [58].

In contrast, three hunters noted that the constant pressure to learn can be difficult. “No matter how experienced you are, everyday you will need to learn new things. And I think this is one of the reasons that makes bug hunting a bit difficult, or a bit tricky. That if you stop learning, you will lose the [touch]. In order to find bugs, you will have to stay updated in community to learn something new.”

Learning from public reports Our interviewees confirmed previous speculation [61] about the importance of *public disclosure* to the learning process. With learning as a

	Factor	Definition	Relevant work	FL	Worth(π)
Choosing a program	Scope	Domains or assets included in the program.	[15]	28	0.156
	Reward	Expected monetary or non-monetary rewards.	[15, 28, 35, 44, 61]	36	0.120
	Bounty table	Reward rules and ranges for different bug types.	[28]	16	0.117
	Technology familiarity	Familiarity with the technology of the assets (e.g., familiarity with web or iOS).	[15]	22	0.113
	Legal safe harbor	Includes a commitment to not pursue legal actions after hackers who follow the rules.	[26]	4	0.098
	Program repute	Reputation in the community for being pleasant to work with.	[35]	15	0.086
	Learning opportunity	Lack of familiarity with the technology of the assets and interest in learning.	[35]	1	0.064
	Private or public	Private programs (only by invitation) vs. public programs (accessible by anyone).	[35]	4	0.048
	Company familiarity	Company behind the program is widely known; you or your peers use its products.	[35, 61]	15	0.047
	Saturation	Number of reports received or number of hackers working on the program.		8	0.047
	Career opportunities	Future career opportunities with the company behind the program.	[35]	3	0.027
	Public disclosure	Public vulnerability disclosure is generally allowed following the bug resolution.		6	0.027
	Age	For how long the program has been running.	[48, 55, 59]	6	0.023
	Business domain	Business domain of the company behind the program (e.g., healthcare, retail).	[55, 61]	2	0.021
Country	Where the company behind the program is located.		1	0.006	
Challenges of bug hunting	Poor responsiveness	Lack of responses or slow responses from program managers.	[14, 15, 28, 36]	30	0.130
	Dissatisfaction with responses	Rewards are lower than expected (e.g., downgraded severity, impact).	[26]	26	0.120
	Unclear scope	Program scope is not defined clearly.		3	0.082
	Poor platform support	Dissatisfaction with how platforms handle issues, such as mediations.		1	0.079
	Duplicates	Too many reports marked as duplicates.	[26]	4	0.078
	Assets outside expertise	Assets are outside area of expertise, lacking certain required skills.		11	0.068
	Secure assets	Finding bugs is too difficult.		5	0.064
	Stress and uncertainty	Fear of burning out, social isolation during work, irregular income, etc.	[26]	5	0.062
	Too much labor work	Menial tasks (e.g., CAPTCHA, timeouts, obfuscation, setting up test accounts).		12	0.050
	Boredom	Bored of working on the program or a more interesting program launches.	[26]	8	0.050
	Unrepresentative reputation system	Hackers' reputation points do not reflect real experience.		1	0.047
	Difficulty working with managers	Managers are difficult to work with (e.g., disrespectful, requiring extra work).	[26, 36]	23	0.044
	Not enough time	Not having enough time for participating in bug bounties.		2	0.043
	Limited vulnerability disclosure	Restrictive vuln. disclosure policies and NDAs that may prevent you from publishing.	[26]	2	0.041
Legal threats	Fear of threats of legal implication (civil or criminal).		2	0.028	
Communication or language	Communication difficulties from lack of language skills, anxiety in communication, etc.		2	0.014	
Benefits of bug hunting	Monetary rewards	Monetary compensation.	[14, 15, 26, 34–36, 61]	42	0.191
	Learning	Learning or improving skills.	[14, 15, 26, 34–36]	32	0.170
	Enjoyment	Enjoyment or challenge of white-hat hacking.	[15, 26, 34–36]	20	0.140
	Legal safe harbor	Hacking without the threat of legal actions if they obey the rules.		4	0.118
	Flexibility	Work schedule and place flexibility (compared to traditional employment).	[14, 26]	16	0.095
	Career	Building relations with companies for employment and other opportunities.	[26, 34–36]	11	0.091
	Community	Bug bounty creates a community of hackers.	[26, 35]	3	0.071
	Altruism	Improving cybersecurity to help others, securing the internet.	[14, 15, 26, 34–36]	5	0.062
	Reputation	Earning platform reputation points, building a following, etc.	[26, 34, 35, 59]	14	0.048
	Non-monetary rewards	Non-monetary compensation (e.g., SWAG, hardware, subscriptions).	[26]	4	0.013
Useful platform features	Ease of payment	Receiving payments in a standardized, hassle-free way.		12	0.156
	Ease of reporting	Easy to generate, submit, and track reports and their status.		16	0.142
	Viewing disclosed vulnerabilities	Platform-provided interface for viewing bugs found by others.	[61]	15	0.137
	Private program invitations	Access to private programs on the platform.		5	0.107
	Program directory	Listing many programs in one place, with statistics, details, etc.		17	0.068
	Standardized rules	Platform standardizing how scopes, rewards, criticality, etc. are defined.		3	0.063
	Community	Platform making effort to create a community of hackers.	[26]	11	0.057
	Platform rewards	For example, platform SWAG and funded travel.	[26]	1	0.054
	Mediation	Platform resolving disputes between hackers and programs.		13	0.051
	Platform-managed disclosure	Platform-provided tools/mechanisms to publicly disclose resolved bugs.		6	0.050
	Resources for learning	Platform providing free resources on how to hack (e.g., Bugcrowd University).	[36]	2	0.047
	Reputation system	Platform managed-reputation system for hackers.		6	0.043
Platform triage	Triaging managed by the platform (e.g., HackerOne triages your report instead of Uber).		5	0.023	

Table 2: Factors used in the factor-rating study, organized by factor groups. FL: count of participants who listed the factor in the free-listing study. Worth (π): the estimated relative importance of factors (values only meaningful in comparison, see §3.2 for details). Descriptions shortened for space; full versions are given in Appendix A. Citations show factors identified in prior work.

top benefit, hunters naturally valued public disclosure as a learning opportunity. While factor-rating study participants

did not directly value a program's willingness to disclose as much as other factors (FL=6, π =0.027), they did consider

viewing disclosed vulnerabilities to be the platforms' third most useful feature (FL=15, $\pi=0.137$). This result adds nuance to prior work [25], suggesting that even though hunters do not need a specific bug-bounty program to be disclosure friendly to work on it, they prefer the entire ecosystem to be so. This reinforces the notion that hunters primarily seek public disclosure as a resource for learning, with reputation building less important. Conversely, *resources for learning* provided by bug-bounty platforms were not rated as useful (FL=2, $\pi=0.047$), suggesting room for improvement.

Review of publicly disclosed bugs was hunters' most frequently mentioned learning method (I=18). These reports show hunters how their peers approached the problem (I=9): "It's all about the thought process." Reported bugs can also potentially be directly reproduced in other programs (I=4). A hunter described this as, "everyone will take the exploit or the attack vector from the report, and everyone will be trying to score the same issue on different programs."

Public disclosure is not only about learning new hacking techniques; hunters noted that disclosure also helped with evaluating bug-bounty programs (I=10), particularly the quality of the triage team (I=3) (see §5.4). As one hunter put it, "I read about all the disclosed reports on HackerOne and I see how they communicate with the researcher." Further, two hunters viewed public disclosures as advertising opportunities for programs. The disclosures grabbed their attention as they skimmed platforms' disclosed vulnerabilities lists (e.g., Hacktivity and Crowdstream [20, 37]). These disclosures tell hunters that a company is taking its bug-bounty program seriously, making it a more attractive organization to work with.

Impact of community Free-listing study participants reported the *community* of hunters as a benefit of bug bounties (FL=3). Though not rated as highly as other factors ($\pi=0.071$), we find the community to be an integral part of learning. This community develops naturally through interactions on social media, but many hunters (FL=11, $\pi=0.057$) also noted bug-bounty platforms' contributions (e.g., live hacking events).

Nearly all interviewees noted that the hunter community frequently interacts with each other and shares information, creating a learning environment (I=22). This includes hunters disclosing as a way of "giving back to the community" (I=5), providing specific technical knowledge (I=7), and answering questions ($n=5$). One hunter appreciated the responsiveness of blog-post authors: "If you have any doubt, you just ping them. . . . The people are very helpful. The communities are good." Sharing of resources—commonly free of charge (I=1)—is particularly striking in the competitive world of bug bounties (i.e., only one hunter gets paid for each vulnerability) (I=2).

The bug-bounty community also offers the ability to learn by developing relationships with other hunters. Many participants mentioned gaining professional contacts that were helpful to their career (I=8), and five of those noted these contacts led to bug-finding collaborations. Several also highlighted the social benefits of community engagement (I=7),

which can reduce the isolation of working remotely and individually (I=2). In-person events in particular offered this sense of belonging. One hunter explained, "you get to meet so many people. You tend to have so many opportunities, and people seem to recognize you, and everyone wants to be your friend, you want to be everyone's friend."

Not everyone saw the community as a useful learning tool. Five interviewees noted that community members do not always share useful knowledge, and two said sharing is a relatively recent phenomenon. One noted that "there are a few too many trolls, but usually they're [the average hunter] quite helpful and that does really encourage things."

Contrast to technology familiarity Free-listing study participants noted that *technology familiarity* was important when picking bug-bounty programs (FL=22) and factor-ranking participants ranked it an important factor ($\pi=0.113$).

Most interviewees implied that they preferred programs with technologies they are familiar with because it allows them find bugs more easily ($n=15$). Specifically, some ($n=5$) noted that familiarity with the technology (sometimes referring to specific libraries, "you're familiar for instance with Django, probably with Flask or things like that, so you just go and search if the version is updated.") helps them apply skills they already have. As one put it, ". . . if you want to find any bugs, you will need to be familiar with the technology. . ." Three hunters said that familiarity helps them with finding higher severity bugs, while one said it helped them with identifying "the low hanging fruit."

On the surface this might appear to contradict a focus on learning by emphasizing existing skills. However, interviews suggest otherwise. Participants reported that even existing skills need to be practiced (I=3) [58] and hunters learn new techniques regardless of the target. As such, we argue that hunters picking more familiar technologies does not necessarily limit their learning. Rather, focusing on familiar technologies allows hunters to get started on finding bugs and offers opportunities to learn about new variations or connected modules and to master skills through practice.

5.3 Predicting the likelihood of finding a bug

Many participants noted factors related to the probability of finding a bug as important considerations. These factors include program scope, age and saturation, whether the program is public or private, the likelihood of duplicates, and the impact of the hunter's reputation. There was little consensus as to what aspects of a given factor were beneficial or not.

Extent and clarity of program scope Half of free-listing study participants identified the *scope*—which dictates which assets can be investigated and what bug types will be accepted—to be important for choosing a bug-bounty program (FL=28). Factor-rating study participants ranked scope the most important factor ($\pi=0.156$). Interestingly, this finding has only been

previously mentioned in marketing materials [15].

Many hunters consider scope to be a useful predictor of how many bugs they can expect to find. Most interviewees preferred larger scopes (I=15), reasoning that a larger attack surface should correlate with more potential bugs. For example, one hunter explained, “a good example of a big scope is the U.S. Department [of Defense], . . . you have a lot of systems where you are allowed to identify vulnerabilities, which gives you a lot of possibilities to identify things.”

Larger scopes also allow hunters to evaluate a product as a whole (as a malicious actor could); two said better understanding of the overall architecture of assets ultimately creates a higher chance of finding bugs. As one explained, “I just want to find out if some adversary can get their hands on customer data. And they [adversaries] don’t have scopes, they just hit the target, and I want to do the same.”

Correspondingly, narrow scopes have important drawbacks (I=5). First, they raise the risk of finding out-of-scope bugs (I=3). Further, one argued that bug-bounty programs cannot sufficiently isolate small scopes within highly connected products: “They have this big list of out-of-scope, . . . So I go hit their site and I go to an in-scope target and it starts hitting all of these out-of-scope targets all over the place.” Narrow scopes also increase the chances of bug-bounty programs becoming saturated (defined below), because there are likely more hunters per target (I=3), and the difficulty of finding a bug increases accordingly (I=2).

Conversely, a few did not consider a wide scope to be a major factor. Two interviewees noted scope was more important to beginners compared to the experienced, due to increased skills, “Maybe it’s less true today because I’m more experienced now, but at the beginning at least it was important.”

Participants considered *unclear scopes* to be a major issue with bug bounties (FL=3, $\pi=0.082$). Many interviewees recalled frustrating experiences with imprecise scopes (I=10), including those that were vague (I=5) or outdated (I=2). Two noted that they had to talk to managers for clarification. Unclear scopes are in many cases a type of communication issue; we discuss major challenges related to communication in §5.4.

Age and saturation Participants said program age—time elapsed since launch—affected which bug-bounty program they would select (FL=6, $\pi=0.023$). A closely related factor is saturation: how much attention a program has already received and how many hunters are actively working on it, relative to the scope (FL=8, $\pi=0.047$). Eleven participants use age as a proxy for saturation, generally expecting younger programs to have more remaining bugs. As such, 12 interviewees prefer newer programs. Older programs do roll out new code periodically (I=3), creating new opportunities.

However, there was no consensus on this point. Seemingly contradicting prior work [48, 59], six preferred older programs, due to more experience handling bug reports (I=2) and provide more publicly available information (I=1). One

participant said, “There’s going to be fewer reports about that [younger] program, so you’re not going to learn as much.”

Public vs. private programs A *public* program is openly advertised, and anyone can participate. In contrast, *private* programs allow participation by invitation only. Several interviewees used public/private status to judge the chances of finding a bug when choosing programs; however it was not ranked as a top factor (FL=4, $\pi=0.048$). Interviewees’ preferences were evenly divided (private: I=11; public: I=9).

Private programs were reported to have fewer hunters (I=9), suggesting lower saturation. Conversely, three said that private programs are just as saturated as public: “Most programs tend to have a pretty healthy amount of hackers regardless.” Further, three noted that public programs often have wider scopes. Aside from saturation, some used public/private status as a heuristic for age (I=1), responsiveness (I=1), and possibility of public disclosure (I=2).

Likelihood of duplicates Reporting a *duplicate*—a previously reported bug—frequently goes unrewarded, creating frustration for hunters. This was ranked as the fourth largest issue (FL=4, $\pi=0.078$). Interviewees associated duplicates with bug-bounty program managers not patching bugs soon enough (I=6) and older programs (I=2): “If you’re among the first ones on the program, I think that . . . increases your chances not to hit a duplicate.” Three argued that duplicates are primarily a problem when hunters go after easier to find bugs, and can be reduced by avoiding “the low hanging fruit.”

Some interviewees (I=4) argued that duplicates are inevitable: “Duplicates are always the thing that you just have to get used to.” However, several (I=9) described coping mechanisms: three said it helps knowing the bug they found has been acknowledged, four said being added to the report (even without rewards) provides assurance the bug-bounty program managers are not lying, and two said they would not go after bugs they consider popular among the community.

Impact of reputation One factor in whether a hunter will receive rewards is the hunter’s *reputation*. Reputation can be built through official platform reputation systems, as well as through basic publicity of found bugs with write-ups. Hunters with both high and low reputation (I=9) said they will not be or are not taken seriously when reporting bugs for lack of reputation; however, reputation is typically acquired by reporting valid bugs, making it hard for beginners to get started. Similarly, five noted that hunters’ reputations help them garner private invites, “to make sure that I stay up high enough for them to keep going and giving me private invitations.” Four participants mentioned that reputation helps with being taken seriously by other security researchers as well (I=3).

Though 14 free-listing study participants mentioned reputation as an overall benefit, factor-rating study participants gave it limited importance, rating it the second least important benefit ($\pi=0.048$). Similarly, the *reputation systems* that

bug-bounty platforms provide were rated the second least important feature (FL=6, $\pi=5$). We speculate that although reputation is a very visible aspect of bug bounties, it's ultimately not as beneficial as other factors. This finding contradicts prior work that hypothesizes reputation might be more important than or comparable to monetary compensation [25, 55, 59].

5.4 Communication, disputes, and mediation

The free-listing study revealed multiple factors relating to communication between hunters and bug-bounty program managers, including poor responsiveness, rude or unreasonable bug-bounty program managers, and unexpected responses to reports. Though some of these issues have been discussed in prior work (see Table 2), other important issues, such as the mediation process, were not previously explored.

Poor responsiveness In the free-listing study, a majority of participants (FL=30) mentioned *poor responsiveness*: where bug-bounty program managers do not efficiently communicate with hunters. Responsiveness was also ranked as the top challenge in bug bounties by factor-rating study participants ($\pi=0.130$). Prior work [26] and marketing materials [15] have briefly acknowledged the importance of this issue. Our interviewees (I=17) were able to elaborate on what constitutes a timely response, including time to first response (I=1), time to severity determination (I=1), time to payout (I=5), and/or time to produce a patch (I=1). This diverse list of definitions suggests bug-bounty programs should work to improve all metrics, rather than just picking one.

Some (I=5) recounted frustrating instances where the bug-bounty program managers would not respond to their or others' communication attempts at all. One said, "Sometimes some programs don't fix the issues, which are valid, and close them as not applicable, with no reason. Once I comment on it, they don't respond on it and totally ignore my comment."

Many interviewees suggested ways to make unresponsiveness more tolerable, but with little consensus. Some hunters wanted frequent updates for payout delays (I=4), which are "acceptable, as long as they are giving updates." Two said they could wait longer for higher severity bugs; one, conversely, expected faster resolution for higher-severity bugs.

Difficulty working with managers/triggers Though responsiveness was often the most frustrating communication problem, it was not the only one. Notably, *difficulty working with managers* was mentioned by many free-listing study participants (FL=23). However, this factor ranked among the least challenging ($\pi=0.044$), possibly because we distinguished it from the closely related factor of dissatisfaction with responses (discussed next).

Six interviewees mentioned bug-bounty program managers are not always professional in their communications; they can be rude (I=3) or otherwise dismissive (I=2). As one hunter

said, "It's actually not fun to work with somebody who insults you or is rude or stuff like this."

While most hunters attributed communication difficulties to program managers, nine interviewees blamed hunters too. Six said bug reports are sometimes not clear. In response, one hunter was trying to improve: "Something that we've also been trying to do a lot now is ... try to make the reporting as quality and clear as possible." Two interviewees said hunters are occasionally rude and should behave professionally ($n=2$).

Another source of potential conflict is *platform triagers*, mentioned by five free-listing study participants but ranked as the least useful platform feature ($\pi=0.023$). Some bug-bounty programs hire triagers from bug-bounty platforms, rather than maintaining their own team. A few hunters (I=2) argued that triagers outside the development team increase communication difficulty. Two said for-hire triagers create an unnecessary barrier between the hunter and the team that will fix the bug; another noted that since the company makes the final payout decision, triagers' severity gradings are irrelevant.

Conversely, five interviewees had positive experiences with platform triagers. They noted that platform triagers specialize in dealing with hunters, and therefore are better to work with than bug-bounty program managers. One appreciated that platform triagers help clarify bug reports before they get to bug-bounty program managers, and another said platform triagers helped them trust decisions they disagreed with (e.g., duplicates). Two participants recalled negative experiences with programs that do not have platform triagers.

When asked about dissatisfaction with managers, two hunters explicitly said they had no major issues; one said, "For the most part, I've had a really positive experience."

Unexpected responses Closely related to the previous factor, *dissatisfaction with responses* was a commonly mentioned, highly frustrating problem (FL=26, $\pi=0.120$).

Most interviewees recalled disagreeing with the bug-bounty program managers' evaluation of a bug (I=16), usually about vulnerability applicability and severity levels, duplicate status, and permission for public disclosure.

Eight said bug-bounty program managers make errors related to misunderstanding submitted reports, either because they don't read them thoroughly (I=2) or don't understand technical details (I=4). Conversely, one noted that hunters might not fully read program descriptions, "To be honest, I sometimes forget to read the whole program scope and sometimes I submit some vulnerabilities that are not in scope."

Troublingly, as found before in similar contexts [4], two hunters noted they may not report potential bugs to avoid scope disagreements with bug-bounty program managers.

Five hunters were more pessimistic, arguing that companies might trick hunters in order to avoid paying them. One recalled, "Sometimes, when the program got everything they needed from you, ... they set the status, [to] needs more ... Then, the bug is getting fixed ... severity gets lowered, and you get lower rewards than you expected. And then, they

just suddenly stop responding to your comments, and that's a situation that happens all the time."

Disputes and responsiveness issues can, in rare situations, lead to extreme outcomes. One participant was banned from a major bug-bounty platform. Another participant noted that exploiting the bugs themselves becomes more attractive when they get frustrated with bug-bounty programs.

Attempting to resolve disputes When disputes arise, hunters may enlist the bug-bounty platform as an ostensibly impartial mediator, although they generally cannot override the bug-bounty program managers' final assessment [33]. Although listed by many in the free-listing study as a benefit (FL=13), *mediation* was not ranked a top feature of bug-bounty platforms ($\pi=0.051$). Similarly, *poor platform support* (including for mediation), was seen as one of the biggest challenges faced in bug bounties (FL=1, $\pi=0.079$).

Interviewees were split between negative (I=8) and positive (I=5) reactions to platform mediation. On the positive side, some noted that mediation was useful to clarify technical issues to the bug-bounty program managers (I=5). One explained, "One of the biggest benefits . . . is they do facilitate that conversation. . . . Mostly to help a non-technical program manager understand the impact." A participant found comfort in knowing ". . . the platform has their back. In most cases."

Other participants, however, argued that mediation is inherently biased, because bug-bounty platforms favor the companies that pay them to host (I=4). Two recalled experiences where bug-bounty platforms did not respond to mediation requests; two others had heard of but not experienced such issues. One hunter described requesting mediation, "and nothing happens. You get a reply every two weeks because that's how often you can ping them, and there's no indication that there's anything that happens. It is the biggest joke."

5.5 Gig-work benefits and drawbacks

Multiple factors we identified relate to the gig-work nature of the bug bounty ecosystem. This model has benefits (e.g., flexibility) but also serious drawbacks, including stress and uncertainty. Notably, although they do not formally define and measure the prevalence of each issue, this aspect of bug bounties has been heavily criticized by Ellis and Stevens [26].

Flexibility A commonly referenced and middle-ranked (FL=16, $\pi=0.095$) benefit of the gig-work model is *flexibility*: working from anywhere, at any time and for any duration, as well as full autonomy in what to work on.

Most interviewees mentioned flexible work hours (I=14). Three emphasized that this avoids deadlines and pressure: "Compared to my developer work that was all project based with deadlines . . . here I don't have any deadlines." Others mentioned choosing what to hack (I=2), being able to work remotely (I=2), and the relatively low barrier to entry (not requiring an official position or prior approval, I=1).

A few hunters identified some drawbacks of flexibility. One noted that maintaining your own work hours can be tricky. Another mentioned that choosing when to work is less meaningful for those who hunt bugs essentially full-time: "I can do whenever I want to. . . . This could be harder if your only job is full-time bug bounty hunting."

Further, interviewees argued that the appearance of a low barrier to entry is deceptive: two said they had to build skills over years before they could participate effectively.

Negative aspects of gig-work Bug bounties typically only reward hunters for bugs deemed unique and valid. As such, hunters can spend significant effort for no pay, creating *stress and uncertainty* (FL=5, $\pi=0.062$). This stress is in many cases downstream of key factors we identified in prior subsections, such as unclear scope or likelihood of duplicates.

The most frequent source of uncertainty was payment (I=9). Hunters do not know when they might find a bug, when and if it might be acknowledged, or when managers might decide to pay. Four participants noted the irregular income is especially stressful when bug bounties are the main source of income: "I also have days or even weeks where I don't find any single bug. And this is kind of depressing, I would say, which is the dark side of the whole story."

Payment uncertainty is compounded by the competition to find bugs, "because basically bug bounty is a competition, it's first come, first serve." Three hunters noted that time-restricted events increase this sense of competition.

Even if a hunter finds a bug first, they still must convince a bug-bounty program manager it is valid before they receive payment. As discussed in Section 5.4, this can be challenging for several reasons, creating additional stress and uncertainty.

Interviewees described their approaches for coping with this uncertainty. Confirming prior work [26], four said that for financial stability, they keep a full-time job outside the bug-bounty ecosystem. One said, "I have a full-time job, and that makes my income a lot more stable. But I think if I was doing bug bounties full-time . . . that might have impacted my mental health." Another three said they deal with the stress of bug bounties by taking breaks: "Whenever I feel burnt out, I just do something else. I give training . . . or I go for a walk."

Five interviewees said this uncertainty keeps them from doing bug bounties full time. One specifically said even a potentially high income was not worth the instability: "I have the potential of earning over 150 grand with bug bounty . . . if I work at it, but the staggered amount that you get the money in is a problem. So if you're actually planning to do this full-time, you need to be able to have a bit of savings . . . so you can push back on that in case." Conversely, one participant volunteered that they are a full-time hunter, and another argued that bug bounties can be a valid career by themselves.

5.6 Fundamental platform features

Hunters listed multiple bug-bounty platform features as useful in the free-listing study. Most of these features exist independently of bug-bounty platforms and are discussed in previous sections. Here, we discuss the few that are unique to bug-bounty platforms and important to hunters.

Ease of payment With monetary payouts a significant motivator, ease of payment was also of interest to hunters (FL=12, $\pi=0.156$). The most frequently mentioned perk of bug-bounty platform payments were the many options provided (I=7). For example, six interviewees preferred to be paid in an international currency and three preferred cryptocurrency payments. One explained, “I prefer Bitcoins, because I have had problems with PayPal in the past.” Some hunters liked the simplicity of bug-bounty platform payments (I=6): “I put in my checking account number one time and never think about it again.” Finally, one enjoyed support for splitting bounties between collaborators and another liked that they could track all bounties earned.

Ease of reporting Standardized reporting, submission, and correspondence were appreciated by many free-listing study participants (FL=16). In fact, this feature was rated the second most important a platform provides ($\pi=0.142$).

Interviewees mostly mentioned the usefulness of the reporting interface (I=9) including guidelines for how to write reports, formatting tools (e.g., markdown), and support for visuals (e.g., videos). Hunters also appreciated the interface for tracking bug progress (I=5), including correspondence with and actions taken by the bug-bounty program managers.

A minority saw room for improvement (I=6), including larger size limits for reports and attachments, more consistency in reporting interfaces between different bug-bounty platforms, and a more formal communication process.

Program directories *Program directories* are arguably the main feature of bug-bounty platforms, allowing hunters to discover a variety of programs and compare several aspects of them in one place (I=12). Some of the available metrics reflect what hunters care about the most (e.g., expected bounties, responsiveness). Accordingly, this feature was mentioned by most free-listing study participants (FL=17) and rated relatively important by factor-rating study participants ($\pi=0.068$).

A minority of interviewees said there was room for improvement, such as the publication of more statistics (I=3) and hunter feedback (I=1) for each program.

Viewing disclosed vulnerabilities Perhaps because they are integral to learning (an important motivator), hunters (I=11) appreciate being able to easily *view a program’s disclosed vulnerabilities* through the platforms (e.g., *Hacktivity* on Hackerone) (FL=15, $\pi=0.137$). Interviews suggest platforms are the primary source for accessing disclosed reports. A few (I=5) mentioned finding independent write-ups (e.g.,

blog posts); however, more hunters mentioned disclosures made through bug-bounty platforms (I=11).

5.7 Legal safe harbor

Legal safe harbor is a commitment from companies not to legally pursue hunters who follow the rules, allowing hunters to attack real world targets. This legal aspect of bug bounties was mentioned by free-listing study participants under multiple factor groups: choosing programs, challenges, and benefits. However, it is frequently overlooked in prior work.

Though legal safe harbor was not listed as a benefit by many in the free-listing study (FL=4), it was ranked as relatively important ($\pi=0.118$). Similarly, safe harbor was not listed by many as a factor in choosing new programs (FL=4), but was again ranked as fairly important ($\pi=0.098$).

Most interviewees (I=16) said that bug bounties without safe harbors are risky; six said they would not consider programs without them. This might partly explain why legal safe harbor was not necessarily top of mind during the free-listing study; it may not be very visible, but its absence is noticeable.

Some were less strict, saying they only avoid programs that have previously sued hunters (I=4). “I just need to hack on any program that they actually don’t sue people. It makes my life easier.” Eight recalled instances of legal action taken against hunters, and two warned that companies without safe harbor policies would be likely to sue.

As another subtle benefit (I=2), safe harbor signals that hunters are not malicious, as often portrayed [18]. One interviewee said, “So by this [safe harbor] existing, . . . everybody acknowledges, ‘These hackers are good-guy hackers.’ ”

Not all hunters found legal safe harbor to be essential (I=5), perhaps related to *legal threats* being seen as the second least important challenge (FL=2, $\pi=0.028$). Interviewees explained why: it’s already the law where they live (I=1); all HackerOne programs should already include safe harbors, so double-checking is unnecessary (I=1); and legal threats are rare (I=1). Four said it was the hunter’s responsibility to keep their activities legal. One added, “I know my rights as a researcher. Even if you don’t have a legal safe harbor, they have to prove some sort of damage.”

6 Discussion

When the ecosystem functions well, bug bounties have the potential to improve the security posture of organizations (commercial or governmental) at a low cost [55, 59], while also providing hunters numerous benefits. However, bug bounties have low adoption rates, and only some organizations that run bug bounties consistently attract quality bug reports [48, 55]. On the hunters’ side, while the reported pool of workers is large, only a few receive the full breadth of promised benefits [26]. Essentially, not only are bug bounties underutilized,

but there exist inequalities in how the current ecosystem's benefits are distributed among organizations and hunters.

In this section, we list recommendations for bug-bounty programs, platforms, and policymakers to alleviate the important challenges we have identified, as well as support and expand the benefits hunters appreciate the most. We hope this can bolster bug bounty adoption, helping the security of more organizations, and ensuring more hunters receive the full benefits of this marketplace.

What bug-bounty programs can do The two highest-ranked factors that programs have direct control over are scope and rewards, but both require additional resources to increase. Rewards are directly tied to finances, while an increased scope might increase staffing requirements; we suggest future work explore the nuances of increasing scope. A larger scope, when feasible, is also likely to alleviate some other concerns hunters have, such as duplicates, saturation, and the range of technologies used. Closely related to the extent of the scope, the clarity of the scope was also seen as an important issue. Making sure that the scope is clear and up-to-date should be relatively low-cost and could reduce hunters' frustration with unexpected out-of-scope and invalid responses.

The most significant challenges hunters reported were all related to communication. To address responsiveness, the foremost challenge, increased staffing is likely to be the most effective solution [26]. If additional staffing is not possible, a cheaper option is to provide frequent and transparent updates on the status of a bug report to reduce uncertainty (§5.4).

Unexpected responses could also be reduced by improving communications overall. Bug-bounty programs could start by making scopes and bounty tables as clear as possible, perhaps including examples with payouts per (publicly disclosed) bug and by drawing attention to common out-of-scope submissions and pitfalls. Programs could also train their managers to improve communications with hunters and avoid common pitfalls in interpreting vulnerability reports.

What bug-bounty platforms can do Poor platform support (i.e., mediation) was the most significant platform issue, and dissatisfaction with platform or program responses (leading to disputes and then potentially to mediation) is the second most important challenge overall. Several participants complained that mediators are biased against hunters. Platforms could address this by more clearly communicating their business models—they need hunters as much as bug-bounty programs to function—and increasing transparency by creating periodic reports on the outcomes of mediation.

More generally, platforms could also adopt several of our above suggestions for improving communications issues, including adding more staff where possible to reduce delays, and adding guidance for submitting reports. Platforms could also offer training in how to communicate with hunters and interpret vulnerability reports—based on prior examples on the platform—to participating programs.

We also suggest platforms explore how to reduce the uncertainty hunters face. Platforms could evaluate the utility of implementing insurance policies to ensure hunters are paid even if bug-bounty programs are unwilling to accept the results of mediation; or, more aggressively, attempt to reduce the authority of bug-bounty programs to issue final judgments. For instance, platforms could consider managing more of the triaging process and even deciding on final payments.

Learning was seen as the second most important benefit of bug bounties and perhaps the primary way productive hunters are added to the workforce. Currently, only a few hunters enjoy the full benefits of the bug bounty ecosystem; however, increasing hunters' skill levels could reduce this gap, while also leading to more bugs being found overall, and therefore potentially better software security in general. Unfortunately, learning resources provided by bug-bounty platforms were the third least important feature, indicating room for improvement. We suggest better integrating learning material with previously disclosed bugs (the third most useful platform feature), in order to make this material more useful and relevant.

Beyond improving less popular aspects, bug-bounty platforms should not neglect their most popular features: providing a wide range of payment mechanisms (e.g., for international hunters), standardizing interfaces between programs, and enabling hunters to review (and read others' reviews of) bug-bounty programs.

Bug-bounty platforms could also help distribute hunters' attention among bug-bounty programs by promoting low-attention but well maintained or societally important programs, though this may need to be incentivized by an outside party (as we discuss next).

What legislative bodies and policymakers can do Bug-bounty programs offer important security benefits that could be useful to many companies. However, our results suggest hunters typically focus on the programs with the most resources (e.g., monetary rewards, large scopes). Large companies (e.g., Google), with committed bug-bounty programs, are well positioned to take advantage in ways smaller companies and programs may not be able to, even when these companies are in critical sectors with important security needs. Exacerbating the issue, smaller companies are at higher risk for suffering significant loss to cybercrime [39, 55].

Several government agencies currently recommend companies adopt bug bounties [41, 51]. While this is a good start, without additional support, our results suggest it has the effect of entrenching security inequality. Government agencies could also provide funding to help companies with lower security budgets and staffing afford to run committed bug-bounty programs: increasing scope, better staffing, and higher bounties. Grants (e.g. [19]) could be tied to priorities for improving the bug bounty ecosystem, such as publicizing reports (enabling hunters to assess programs), attracting attention to less popular but security-critical industries (e.g., healthcare, finance [55]), or improving educational resources [58].

Governments have additional roles to play in improving conditions for bug hunters. Legal scholars have proposed legislation to implement legal safe harbors for any good-faith security researcher, including hunters [27], and judicial policy-setters have taken steps in this direction [50]. Our work provides evidence that this is a worthwhile effort, as hunters value a legal safe harbor and use it as a discriminator between programs (§5.7). A generalized safe harbor could therefore incentivize hunter participation for companies that do not currently offer that benefit. Further, labor regulators could consider how best to protect hunters from uncertainty and even abuse associated with gig-work (§5.5, [26]), by setting appropriate standards for communication and even payment, which would help to retain more hunters in the ecosystem.

Acknowledgments

We thank our participants, without whom this study would not be possible; and our reviewers, who helped shape the analysis and framing of the paper. We also thank Jack Cable, Julien Ahrens, and Nathan Reitingner for their assistance and insights.

This material is based upon work sponsored by the National Science Foundation under Grants No. CNS-1850510 and CNS-1801545. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. This research was also supported by a gift from Google.

References

- [1] Omer Akgul, Taha Eghtesad, Amit Elazari, Omprakash Gnawali, Jens Grossklags, Michelle L Mazurek, Daniel Votipka, and Aron Laszka. Bug hunters' perspectives on the challenges and benefits of the bug bounty ecosystem. *arXiv preprint arXiv:2301.04781*, 2023.
- [2] Reed Albergotti. Apple pays hackers six figures to find bugs in its software. Then it sits on their findings. *The Washington Post*, 2021. <https://www.washingtonpost.com/technology/2021/09/09/apple-bug-bounty/>.
- [3] Nikolaos Alexopoulos, Andrew Meneely, Dorian Arnouts, and Max Mühlhäuser. Who are vulnerability reporters? A large-scale empirical study on FLOSS. In *Proceedings of the 15th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, 2021.
- [4] Noura Alomar, Primal Wijesekera, Edward Qiu, and Serge Egelman. "You've got your nice list of bugs, now what?" Vulnerability discovery and management processes in the wild. In *16th Symposium on Usable Privacy and Security (SOUPS)*. USENIX Association, 2020.
- [5] John Annett. Hierarchical task analysis. *Handbook of Cognitive Task Design*, 2, 2003.
- [6] Apple. Example payouts. <https://developer.apple.com/security-bounty/payouts/>.
- [7] Soodeh Atefi, Amutheezan Sivagnanam, Afiya Ayman, Jens Grossklags, and Aron Laszka. The benefits of vulnerability discovery and bug bounty programs: Case studies of Chromium and Firefox. In *2023 ACM Web Conference (WWW)*, May 2023.
- [8] Tod Beardsley, Bob Rudis, Tom Sellers, Curt Barnard, and Kwan Lin. 2021 industry cyber-exposure (ICER): Fortune 500 report, 2020. <https://www.rapid7.com/research/report/2021-industry-cyber-exposure-report/>.
- [9] H. Russell Bernard. *Research methods in anthropology: Qualitative and quantitative approaches*. Rowman & Littlefield, 2017.
- [10] Casey Breen, Cormac Herley, and Elissa M. Redmiles. A large-scale measurement of cybercrime against individuals. In *2022 CHI Conference on Human Factors in Computing Systems*, pages 1–41, 2022.
- [11] Bugcrowd. <https://www.bugcrowd.com>.
- [12] Bugcrowd. Bugcrowd achieves record growth in first half of 2018, 2018. <https://www.bugcrowd.com/press-release/bugcrowd-achieves-record-growth-in-first-half-of-2018/>.
- [13] Bugcrowd. The state of bug bounty, 2018. <https://www.bugcrowd.com/resources/reports/state-of-bug-bounty-2018/>.
- [14] Bugcrowd. Inside the mind of a hacker, 2020. <https://itmoah.bugcrowd.com/>.
- [15] Bugcrowd. Inside the mind of a hacker, 2021. <https://www.bugcrowd.com/resources/report/inside-the-mind-of-a-hacker/>.
- [16] United States Census Bureau. 2019 SUSB annual data tables by establishment industry, 2020. <https://www.census.gov/data/tables/2019/econ/susb/2019-susb-annual.html>.
- [17] Kathy Charmaz. *Constructing grounded theory: A practical guide through qualitative analysis*. Sage Publications, 2006.

- [18] Mitchell Clark. Missouri governor threatens reporter who discovered state site spilling private info, 2021. <https://www.theverge.com/2021/10/14/22726866/missouri-governor-department-elementary-secondary-education-ssn-vulnerability-disclosure>.
- [19] European Commission. Funding opportunities for small businesses. https://commission.europa.eu/funding-tenders/how-apply/eligibility-who-can-get-funding/funding-opportunities-small-businesses_en.
- [20] Crowdstream. <https://bugcrowd.com/crowdstream>.
- [21] Regina Dittrich, Brian Francis, Reinhold Hatzinger, and Walter Katzenbeisser. A paired comparison approach for the analysis of sets of Likert-scale responses. *Statistical Modelling*, 7(1), 2007.
- [22] Regina Dittrich and Reinhold Hatzinger. Fitting loglinear Bradley-Terry models (LLBT) for paired comparisons using the R package pfmmod. *Psychological Test and Assessment Modeling*, 51(2):216, 2009.
- [23] Regina Dittrich, Reinhold Hatzinger, and Walter Katzenbeisser. Modelling the effect of subject-specific covariates in paired comparison studies with an application to university rankings. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 47(4), 1998.
- [24] Amit Elazari. Private ordering shaping cybersecurity policy: The case of bug bounties. *An edited, final version of this paper in Rewired: Cybersecurity Governance*, Ryan Ellis and Vivek Mohan eds. Wiley, 2019.
- [25] Ryan Ellis, Keman Huang, Michael Siegel, Katie Moursouris, and James Houghton. Fixing a hole: The labor market for bugs. In *New Solutions for Cybersecurity*, pages 129–159. MIT Press, 2018.
- [26] Ryan Ellis and Yuan Stevens. Bounty everything: Hackers and the making of the global bug marketplace. *Available at SSRN 4009275*, 2022.
- [27] Daniel Etcovitch and Thyla van der Merwe. Coming in from the cold: A safe harbor from the CFAA and the DMCA § 1201 for security researchers. *Berkman Klein Center Research Publication*, (2018-4), 2018.
- [28] Matthew Finifter, Devdatta Akhawe, and David Wagner. An empirical study of vulnerability rewards programs. In *22nd USENIX Security Symposium (USENIX Security)*. USENIX Association, 2013.
- [29] Brian Francis, Regina Dittrich, and Omer Akgul. Private communication, 2022.
- [30] Kelsey R. Fulton, Samantha Katcher, Kevin Song, Marshini Chetty, Michelle L. Mazurek, Chloé Messdaghi, and Daniel Votipka. Vulnerability discovery for all: Experiences of marginalization in vulnerability discovery. In *To Appear in 32nd USENIX Security Symposium (USENIX Security)*. USENIX Association, 2023.
- [31] Frederick J. Gravetter and Lori-Ann B. Forzano. *Research methods for the behavioral sciences*. Cengage learning, 6 edition, 2018.
- [32] HackerOne. <https://www.hackerone.com>.
- [33] HackerOne. Hacker mediation. <https://docs.hackerone.com/hackers/hacker-mediation.html>.
- [34] HackerOne. Hacker powered security report: 2019, 2019. <https://www.hackerone.com/sites/default/files/2019-08/hacker-powered-security-report-2019.pdf>.
- [35] HackerOne. The 2020 hacker report, 2020. <https://www.hackerone.com/sites/default/files/2020-04/the-2020-hacker-report.pdf>.
- [36] HackerOne. The 2021 hacker report, 2021. <https://www.hackerone.com/resources/reporting/the-2021-hacker-report>.
- [37] Hactivity. <https://hackerone.com/hactivity>.
- [38] Reinhold Hatzinger and Regina Dittrich. Pfmmod: An R package for modeling preferences based on paired comparisons, rankings, or ratings. *Journal of Statistical Software*, 48:1–31, 2012.
- [39] Nicolas Huaman, Bennet von Skarczinski, Christian Stransky, Dominik Wermke, Yasemin Acar, Arne Dreißgacker, and Sascha Fahl. A large-scale interview study on information security in and attacks against small and medium-sized enterprises. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, August 2021.
- [40] Keman Huang, Michael Siegel, Stuart Madnick, Xiaohong Li, and Zhiyong Feng. Poster: Diversity or concentration? Hackers’ strategy for working across multiple bug bounty programs. In *37th IEEE Symposium on Security and Privacy (S&P)*, 2016.
- [41] Christopher C. Krebs. Binding operational directive 20-01 - Develop and publish a vulnerability disclosure policy. 2020. <https://www.cisa.gov/sites/default/files/bod-20-01.pdf>.
- [42] J. Richard Landis and Gary G. Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1), 1977.

- [43] Aron Laszka, Mingyi Zhao, and Jens Grossklags. Banning misaligned incentives for validating reports in bug-bounty platforms. In *21st European Symposium on Research in Computer Security (ESORICS)*, pages 161–178, 2016.
- [44] Aron Laszka, Mingyi Zhao, Akash Malbari, and Jens Grossklags. The rules of engagement for bug bounty programs. In *22nd International Conference on Financial Cryptography and Data Security (FC)*. Springer, 2018.
- [45] Gitta H. Lubke and Bengt O. Muthén. Applying multi-group confirmatory factor models for continuous outcomes to likert scale data complicates meaningful group comparisons. *Structural Equation Modeling*, 11(4):514–534, 2004.
- [46] Donatello Luna, Luca Allodi, and Marco Cremonini. Productivity and patterns of activity in bug bounty programs: Analysis of HackerOne and Google vulnerability research. In *14th International Conference on Availability, Reliability and Security (ARES)*, 2019.
- [47] Ana Magazinius, Niklas Mellegård, and Linda Olsson. What we know about bug bounty programs – An exploratory systematic mapping study. In *International Workshop on Socio-Technical Aspects in Security and Trust (STAST)*. Springer, 2019.
- [48] Thomas Maillart, Mingyi Zhao, Jens Grossklags, and John Chuang. Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs. *Journal of Cybersecurity*, 3(2), 2017.
- [49] Nora McDonald, Sarita Schoenebeck, and Andrea Forte. Reliability and inter-rater reliability in qualitative research: Norms and guidelines for CSCW and HCI practice. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW Issue), 2019.
- [50] U.S. Department of Justice. Department of justice announces new policy for charging cases under the computer fraud and abuse act, 2022. <https://www.justice.gov/opa/pr/departement-justice-announces-new-policy-charging-cases-under-computer-fraud-and-abuse-act>.
- [51] National Institute of Standards and Technology. Security and privacy controls for information systems and organizations. RA-5:245, 2020. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf>.
- [52] Jukka Ruohonen and Luca Allodi. A bug bounty perspective on the disclosure of web vulnerabilities. *arXiv preprint arXiv:1805.09850*, 2018.
- [53] Johnny Saldaña. The coding manual for qualitative researchers. *The coding manual for qualitative researchers*, pages 1–440, 2021.
- [54] Jim Salter. Three iOS 0-days revealed by researcher frustrated with Apple’s bug bounty, 2021. <https://arstechnica.com/information-technology/2021/09/three-ios-0-days-revealed-by-researcher-frustrated-with-apples-bug-bounty/>.
- [55] Kiran Sridhar and Ming Ng. Hacking for good: Leveraging HackerOne data to develop an economic model of bug bounties. *Journal of Cybersecurity*, 7(1), 2021.
- [56] Daniel Votipka, Seth Rabin, Kristopher Micinski, Jeffrey S. Foster, and Michelle L. Mazurek. An observational investigation of reverse engineers’ processes. In *29th USENIX Security Symposium (USENIX Security)*. USENIX Association, 2020.
- [57] Daniel Votipka, Rock Stevens, Elissa Redmiles, Jeremy Hu, and Michelle Mazurek. Hackers vs. testers: A comparison of software vulnerability discovery processes. In *39th IEEE Symposium on Security and Privacy (S&P)*, 2018.
- [58] Daniel Votipka, Eric Zhang, and Michelle L Mazurek. Hacked: A pedagogical analysis of online vulnerability discovery exercises. In *2021 IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2021.
- [59] Thomas Walshe and Andrew Simpson. An empirical study of bug bounty programs. In *2nd IEEE International Workshop on Intelligent Bug Fixing (IBF)*, 2020.
- [60] Chien-Ho Wu. An empirical study on the transformation of likert-scale data to numerical scores. *Applied Mathematical Sciences*, 1(58):2851–2862, 2007.
- [61] Mingyi Zhao, Jens Grossklags, and Peng Liu. An empirical study of web vulnerability discovery ecosystems. In *22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2015.
- [62] Mingyi Zhao, Aron Laszka, and Jens Grossklags. Devising effective policies for bug-bounty platforms and security vulnerability discovery. *Journal of Information Policy*, 7, 2017.
- [63] Mingyi Zhao, Aron Laszka, Thomas Maillart, and Jens Grossklags. Crowdsourced security vulnerability discovery: Modeling and organizing bug-bounty programs. In *4th AAI Workshop on Mathematical Foundations of Human Computation (HCOMP)*, 2016.

A Complete List of Factors

Question	Code Name	Description	FL	μ_r	Worth (π)
Choosing a program	Scope:	Number of domains or assets that are included in the program.	28	6.10	0.156
	Reward:	Expected monetary or non-monetary rewards (e.g., SWAG, hardware, subscription).	36	5.91	0.120
	Bounty table:	Reward rules and ranges set by the managers (e.g., \$50 for low criticality bugs, but \$5000 for high criticality bugs).	16	5.87	0.117
	Technology familiarity:	Familiarity with the technology of the assets (e.g., familiarity with web or iOS).	22	5.86	0.113
	Legal safe harbor:	Language of program includes a commitment to not pursue legal actions after hackers who follow the rules and/or explicitly authorizes testing conducted in accordance with the rules.	4	5.71	0.098
	Program repute:	Program's reputation in the community for being pleasant to work with (i.e., what other hackers say about the program).	15	5.68	0.086
	Learning opportunity:	Lack of familiarity with the technology of the assets (e.g., interest in learning crypto or Android).	1	5.35	0.064
	Private or public:	Private programs (accessible only by invitation) vs. public programs (accessible by anyone).	4	5.16	0.048
	Company familiarity:	Company behind the program is widely known, or you or your peers use its products or services (e.g., working on Uber's program because you like or use their services).	15	5.04	0.047
	Saturation:	Number of reports received or number of hackers working on the program.	8	5.17	0.047
	Career opportunities:	Future career opportunities with the company behind the program.	3	4.49	0.027
	Public disclosure:	Public vulnerability disclosure is generally allowed following the resolution of the issue, permissive NDAs.	6	4.63	0.027
	Age:	For how long the program has been running.	6	4.44	0.023
	Business domain:	Business domain of the company behind the program (e.g., social media, insurance, medical).	2	4.47	0.021
Country:	Where the company behind the program is located.	1	3.34	0.006	
Challenges of bug hunting	Poor responsiveness:	Lack of responses or slow responses from program managers.	30	5.39	0.130
	Dissatisfaction with responses:	Rewards are lower than promised by rules (e.g., downgraded severity, impact, disagreements with duplicates).	26	5.36	0.120
	Unclear scope:	Program scope is not defined clearly.	3	4.99	0.082
	Poor platform support	Dissatisfaction with how platforms handle issues, such as mediating between hackers and programs.	1	5.03	0.079
	Duplicates:	Too many reports marked as duplicates.	4	5.02	0.078
	Assets outside expertise	Assets are outside area of expertise, lacking certain required skills.	11	4.87	0.068
	Secure assets:	Finding bugs is too difficult.	5	4.78	0.064
	Stress and uncertainty:	Fear of burning out, social isolation during work, irregular income, etc.	5	4.8	0.062
	Too much labor work:	Menial tasks (e.g., CAPTCHA, waiting for timeouts, obfuscation, setting up test accounts).	12	4.62	0.050
	Boredom:	Bored of working on the program or a more interesting program launches.	8	4.62	0.050
	Unrepresentative reputation system:	Hackers' reputation points do not reflect real experience and are not transferable between platforms.	1	4.59	0.047
	Difficulty working with managers:	Bug-bounty program managers are difficult to work with (e.g., disrespectful, requiring extra work).	23	4.48	0.044
	Not enough time:	Not having enough time for participating in bug bounties.	2	4.45	0.043
	Limited vulnerability disclosure:	Restrictive vulnerability disclosure policies and NDAs that may prevent you from publishing your work following the resolution/mitigation of the issue.	2	4.49	0.041
Legal threats:	Fear of threats of legal implication (civil or criminal).	2	3.96	0.028	
Lacking communication or language skills:	Communication difficulties because you feel that you lack language skills, experience anxiety in communication, etc.	2	3.45	0.014	
Benefits of bug hunting	Monetary rewards:	Monetary compensation.	42	6.31	0.191
	Learning:	Learning or improving skills.	32	6.18	0.170
	Enjoyment:	Enjoyment or challenge of white-hat hacking.	20	6.08	0.140
	Legal safe harbour:	Hacking without the threat of legal actions if they obey the rules.	4	5.96	0.118
	Flexibility:	Work schedule and place flexibility (compared to traditional employment).	16	5.85	0.095
	Career:	Building relations and reputation with companies for employment and other work opportunities.	11	5.71	0.091
	Community:	Bug bounty creates a community of hackers.	3	5.52	0.071
	Altruism:	Improving cybersecurity for the sake of helping others, hacking to make the internet safer for everyone.	5	5.54	0.062
	Reputation:	Earning platform reputation points, building a following, etc.	14	5.36	0.048
	Non-monetary rewards:	Non-monetary compensation (e.g., SWAG, hardware, subscriptions).	4	4.35	0.013
Useful platform features	Ease of payment:	Receiving payments in a standardized, hassle-free way.	12	6.51	0.156
	Ease of reporting:	Easy to generate, submit, and track reports and their status.	16	6.46	0.142
	Viewing disclosed vulnerabilities:	Platform provided interface for viewing bugs found by others.	15	6.41	0.137
	Private program invitations:	Access to private programs on the platform.	5	6.28	0.107
	Program directory:	Listing many programs in one place, with statistics, details, etc. (being able to view Uber, Paypal, etc. programs on one page with statistics).	17	6.02	0.068
	Standardized rules:	Platform standardizing how scopes, rewards, criticality, etc. are defined.	3	5.99	0.063
	Community:	Platform making effort to create a community of hackers.	11	5.77	0.057
	Platform rewards:	E.g., platform SWAG, funded travel.	1	5.89	0.054
	Mediation:	Platform resolving disputes between hackers and programs.	13	5.77	0.051
	Platform managed disclosure:	Platform provided tools/mechanisms to publicly disclose resolved bugs.	6	5.86	0.050
	Resources for learning:	Platform providing free resources on how to hack (e.g., Bugcrowd University).	2	5.59	0.047
	Reputation system:	Platform managed reputation system for hackers.	6	5.70	0.043
	Platform triage:	Triaging managed by the platform (e.g., HackerOne triages your report instead of Uber).	5	5.06	0.023
None:	There are no useful features that platforms provide.	4	-	-	

Table 3: Factors used in the factor-rating study, organized by factor groups. FL: count of participants who listed the factor in the free-listing study. Worth (π): the estimated relative importance of factors (see §3.2 for details). Likert averages (μ_r) were included to show differences with LLBT-based analysis.